Mobile Crowdsensing from a Selfish Routing Perspective

Julia Buwaya and José D. P. Rolim
Computer Science Department, University of Geneva, Switzerland
E-mail: {julia.buwaya | jose.rolim}@unige.ch

***Abstract*—We present a selfish routing model to optimize the allocation of tasks in a mobile crowdsensing (MCS) system. The players of our game are sensing service requesters that wish to route their demand along paths that are made up of resources belonging to the crowd participants. Resource usage involves load-dependent costs and one resource may serve several requests at the same time. Due to human involvement and mobility there exists uncertainty, which we address by introducing *certainty* parameters. For the Nash equilibria of our game, we can transfer efficiency guarantees, i.e., the worst-case ratio between the welfare of an equilibrium and the welfare of a social optimum is provably bounded by a small constant when cost functions are polynomials. An ϵ-approximation of a Nash equilibrium solution can be computed in polynomial time for affine cost functions. Based on our model, we develop a mechanism for the automation of efficient task allocations in MCS systems and we present a proof for the truthfulness of this mechanism.**

I. Introduction

The use of the collective intelligence and performance of *crowds* - online communities providing resources to solve tasks - has become more and more popular. These activities are grouped under the umbrella term *crowdsourcing*, a term marked in particular by Jeff Howe and Mark Robinson and their discussions on how work can be outsourced to individuals by using the Internet [1]. Examples of crowdsourcing include Amazon Mechanical Turk [2], LEGO Ideas [3], and the IoT Lab [4]. Unlike old-known markets where goods are offered by companies, here the goods are the cumulative product of a large number of people. The participants in crowdsourcing usually operate no specialized business but qualify due to circumstances that they own desired resources. They obtain benefits for participation in form of idealistic, monetary, or other personal rewards. The sub-category of *mobile crowdsensing (MCS)* is a recent and emerging paradigm which leverages the sensing data from the mobile devices of a crowd to serve various goals. These include business goals as for example designing and evaluating a health care product by using the cumulative data gathered by a crowd through the powerful sensors integrated in smartphones. Moreover, MCS systems are used to improve public and individual services. Concrete examples include weather monitoring in rural areas of East Africa [5] as well as participatory citizen sensing systems for sharing information on water conditions and flooding in Vicenza, Italy and Doncaster, UK [6]. The term MCS was coined in [7] presenting a categorization of various MCS applications and pointing out research challenges including the scheduling of sensing and communication tasks. In MCS systems, the allocation of tasks (also called *load balancing*) among a huge number of heterogeneous mobile personal devices is a fundamental and non-trivial issue. Tasks should be allocated such that short- and long-term system performance are optimized with respect to costs, quality of results, user satisfaction and further application-specific metrics.

In this paper, we study this issue of load balancing in MCS systems for a specific setting: A requester wants to collect sensing data about an area, but implementing commercial sensors is not feasible or too expensive. Instead, the requester engages a number of people present in this area that are equipped with personal mobile devices that have sensing capabilities. Mobile devices usually have a lower computing performance and limited power supply compared to standard personal computers. Data generation and data transfer via the Internet involves load-dependent costs. The question, we would like to answer is "How should tasks be assigned to participants for the efficient completion of the tasks?" The overall sensing task (covering the area) should be successfully completed with high probability and the involved costs should be minimized. We point out that such a setting may for example arise in area-related MCS applications on weather monitoring [5].

We present an atomic routing game to depict the problem. Each provider (crowd participant) owns a node (resource) in a multi-commodity network. On the other side, requesters balance their demand over feasible paths (resource bundles that are capable of successfully completing desired sensing tasks) in a cost-minimizing manner. A resource may serve several requests at the same time, which can be modeled efficiently by the network model. Resource usage involves load-dependent costs. From the computed costs, we may infer minimal rewards to engage participants. A special aspect of our problem comes with the mobility and autonomy of crowd participants producing uncertainty about location and, hence uncertainty about the participants' suitability to accomplish tasks ahead of time. We address this aspect by introducing *certainty* parameters into the model. The exact values of these parameters in practice could for example be

This paper will appear in the Proceedings of the IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPS Workshops) 2017.

derived from the results of mobility prediction algorithms (see e.g., [8] and [9]). The model can furthermore be extended to include participation constraints of the crowd (when a personal costs limit is exceeded).

In our selfish routing model, we are interested in the computation and efficiency properties of Nash equilibria corresponding to stable outcomes. An ϵ-approximation of a Nash equilibrium can be computed in polynomial time for all $\epsilon > 0$, given that the cost functions on the resources are affine [10]. In addition, in our atomic routing model, well-known results regarding the price of anarchy (the welfare ratio between a worst-case equilibrium and a social optimum of a game) apply, i.e., the solution quality of a Nash equilibrium is provably bounded, given that the cost functions are polynomials. Based on the model, we develop a mechanism for the automation of efficient task allocations in MCS systems. This paper extends [11], which also presented an atomic routing model for MCS systems. [11] focused on comparing several different solution concepts and an experimental study. In this work, we extend theoretical parts of [11] by providing a proof for the *truthfulness* of a mechanism generating equilibrium outcomes in a MCS system. Truthfulness is a game-theoretic property of mechanisms describing that players (here the sensing service requesters) will not lie about their private values in order to manipulate the mechanism (see e.g., [12]). A formal definition will be given in the course of this paper.

In summary, this paper makes the following main contributions: (1) An analysis of MCS systems using a game theoretical selfish routing model that explicitly incorporates strategic behavior and uncertainty. In particular, the chosen model allows the transfer of well-known theoretical results on the existence and quality of some solutions. (2) A mechanism for the automation of efficient distributed task allocations in MCS systems including a full proof for its truthfulness.

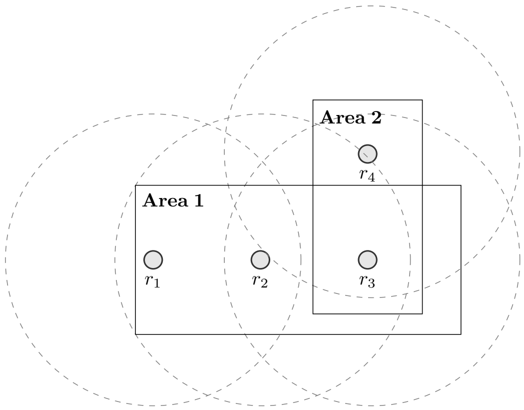
A. Related Work

Load balancing has different forms depending on the considered problems and issues introduced by the MCS system designer. It is important whenever data is collected from a large number of people and has frequently been studied in the field of game theory in relation to congestion control in communication networks (see e.g., [13], [14], [15]) and also in crowdsourcing systems (see e.g., [16], [17], [18]). In MCS, notably [19], [20] and [21] deal with the issue of load balancing. In [19], an auction formulation for the distributed generation of task allocations in MCS systems is presented. Participants can offer bids for undertaking sensing tasks. A centralized operator selects winners based on the bids and pays them after the completion of the tasks. In contrast to [19], the quality of our results is actually provably bounded. In [20], cost-minimal mechanisms that provide a certain quality level are proposed. In contrast to our work, in [20], participants may only be involved in one sensing task at a time. [21] also offers an auction mechanism rather than an efficient network formulation as presented in our work. In [18], the authors provide a resource allocation problem formulation for more general crowdsourcing systems. Nevertheless, they do not exploit the efficient structure of network routing games. The same holds for [16] and [17].

Related to our work are furthermore crowdsourcing contests. The authors of [22] study such crowdsourcing contests and introduce a mechanism that calls for competition in the crowd to produce a product within a limited time frame with a hard deadline. Only the product with maximum quality will be of value for the principal. Nevertheless, every participant that spends effort receives a payment. The authors of [23] model crowdsourcing contests as variation of all-pay auctions and compare crowdsourcing contests to more conventional means of procurements. In contrast to our work, in the latter articles, each participant produces one complete end-product. Instead, we focus on crowdsensing of complementary resources. Both of the latter articles extend ideas of [24] which established the connection between crowdsourcing and all-pay auctions.

In addition, [25] studying innovation management in crowds is related to our work. In [25], the problem of assigning tasks of unknown difficulty to crowd participants of unknown skill is modeled as an extended resource allocation problem. They present a decentralized mechanism which produces a hierarchy such that participants' skill levels and task difficulty levels are matched properly. Due to the special properties of MCS (complementary goods and data collection limited by the abilities of sensors and mobile devices), we present a simple efficient network flow model depicting the crowdsourcing environment instead.

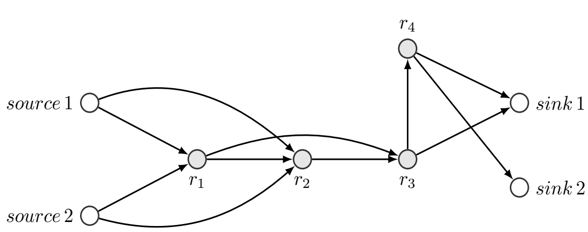
Finally, we would like to emphasize that one of the roots of MCS systems lies in wireless sensor networks (WSNs). WSNs can be envisioned as large collections of autonomous smart sensor nodes, which can distributively form an ad hoc wireless communication network. The nodes can be deployed in a region of interest in order to monitor crucial events and propagate sensed data to a base station, but classical WSNs do not integrate personal mobile devices such as smartphones. Many fundamentals have been established for WSNs regarding representative models and balancing algorithms which maximize network lifetime (see e.g., [26], [27]). In MCS systems, load balancing plays a different role, i.e., we usually do not have multi-hop communication with the high potential of causing early fall-outs of bottleneck nodes. Instead, we have a higher degree of unpredictability and unreliability due to (a different level of) mobility and human involvement. The design of incentive mechanisms to engage crowd participants while taking into consideration preferences and individual behavior is necessary. The purpose of load balancing in MCS systems is to optimize shortand long-term system performance with respect to costs, quality of results, user satisfaction and further application-specific metrics.



(a) Area Coverage Representation

Bundle	Area 1	Area 2
$r_1 r_2 r_3 r_4$	yes	yes
$r_1 r_2 r_3$	yes	no
$r_1 r_2 r_4$	no	no
$r_1 r_3 r_4$	yes	yes
$r_2 r_3 r_4$	yes	yes
$r_1 r_2$	no	no
$r_1 r_3$	yes	no
$r_1 r_4$	no	no
$r_2 r_3$	yes	no
$r_2 r_4$	no	no
$r_3 r_4$	no	yes
r_1	no	no
r_2	no	no
r_3	no	no
r_4	no	no

(b) Table indicating feasible bundles



(c) Network Representation

Figure 1. Graphical representations of a MCS problem instance with two requesters and four crowd participants. Every participant owns one resource indicated by r_1, r_2, r_3, r_4. In (a) the areas that the requesters wish to examine are represented as boxes. Requester 1 wants to cover Area 1 and requester 2 wants to cover Area 2. The radius of the subarea covered by a resource is indicated by the dotted circle surrounding it. In (b) a table is given that indicates whether resource bundles cover an area or not. In (c) a network representation of the problem instance is given. For requester 1 (Area 1), choosing a feasible resource bundle corresponds to choosing a path in the network from source 1 to sink 1. Accordingly, requester 2 (Area 2) chooses a path from source 2 to sink 2.

II. Model

To motivate our model, we present a simple example instance of our problem in Fig. 1 with four crowd participants and two sensing service requesters. Fig. 1a shows an area coverage representation of the considered problem instance. Each requester wants to cover an area of interest, i.e., requester 1 wants to cover Area 1 and requester 2 wants to cover Area 2. In these areas several participants are present (with certain probability). Several combinations of crowd participants, or more explicitly, the resources they own (indicated by r_1, r_2, r_3, r_4), yield feasible coverage of the areas of interest. A single participant covers the area within a certain radius around her. It is rather unlikely that she will be able to set her sensors to only cover a smaller subarea within this radius. Hence, when engaging several participants that have overlapping radii, it is not possible to decrease costs by assigning them smaller subareas. When contributing to the application, a participant will have to spend the full effort to cover the area within her radius. The table in Fig. 1b indicates the feasible resource bundles covering Area 1 and Area 2, respectively. A network representation as given in Fig. 1c can be used to indicate feasible resource bundles as network paths. For requester 1 (Area 1), choosing a feasible resource bundle corresponds to choosing a path in the network from source 1 to sink 1. Accordingly, requester 2 (Area 2) chooses a path from source 2 to sink 2.

We turn to the formal definition of our model. The problem of allocating resources of crowd participants to requesters which desire bundles of complementary goods can be modeled as *atomic splittable routing game* $G = \{N, R, Q = \bigcup_{i \in N} Q^i, (c_r)_{r \in R}, (d_i)_{i \in N}, (\pi_i)_{i \in N}\}$. (See e.g., [28] for an introduction to atomic routing games.) The finite set of requesters is denoted by $N = \{1, ..., n\}$ and the finite set of crowd participants each owning one resource is denoted by $R = \{1, ..., m\}$. Since each participant is uniquely tied to a resource and this resource to her, we will denote a participant simply as a resource when convenient. In our game, a requester $i \in N$ has a total demand of $d_i \in \mathbb{Z}_+$ units of feasible resource bundles. Let set Q^i denote the set of all feasible resource bundles of requester i. If a player decides to use resources, she must distribute (split) demand in integral parts over feasible resource bundles $Q^i_j \in Q^i$. Such a combination of feasible resource bundles corresponds to a strategy of requester i. (Note that $Q^i_j \subseteq R$.) For resource usage, there are load-dependent costs, i.e., costs are not fixed per unit of demand but depend on the total amount of demand the resource is serving. Resource costs are given by functions $c_r : \mathbb{R}_+ \to \mathbb{R}_+$ for all $r \in R$. We consider non-negative, non-decreasing, and normalized (in the sense that for all $r \in R$, we have $c_r(0) = 0$) polynomial cost functions. Every requester i has a *willingness to pay* $\pi \in \mathbb{R}_+$. This value indicates the maximum amount a requester is willing to spend.

A strategy may also be interpreted as a combination of paths in a multi-commodity network. In that network, the resources in R correspond to nodes. In addition, for each requester $i \in N$, there exists a distinct source node s_i and a distinct sink node t_i. A feasible bundle Q^i_j corresponds to a path going from s_i to t_i passing exactly through the resources of set Q^i_j. Let $D^G = (V, E)$ be the multi-commodity network associated with game G. The set of network nodes is given by $V = R \cup \{s_i, t_i\}_{i \in N}$ and the set of network edges is given by E holding the edges of all paths. A feasible outcome of G is a *feasible flow* $\mathbf{x} = (x_{Q^i_j})_{Q^i_j \in Q}$ in multi-commodity network D^G with

$$\sum_{Q^i_j \in Q^i} x_{Q^i_j} \leq d_i \qquad \forall i \in N \, , \tag{1a}$$

$$x_{Q^i_j} \in \mathbb{Z}_+ \qquad \forall Q^i_j \in Q \, . \tag{1b}$$

Variable $x_{Q^i_j}$ denotes the amount of flow on path Q^i_j. Equations (1a) and (1b) indicate that a feasible flow is non-negative, integral and does not exceed the total demand of a requester. (Recall that d_i is also a non-negative integer.) Given a feasible flow $\mathbf{x}$, the load $x_r = \sum_{Q^i_j \in Q:\, r \in Q^i_j} x_{Q^i_j}$ of resource $r \in R$ corresponds to the total flow running through the corresponding node r in network D^G. The implied costs for the crowd participant owning resource r are $c_r(x_r) \cdot x_r$. The effective costs for a requester i, given flow $\mathbf{x}$, equal the sum of load-dependent costs of resources contained in the chosen paths multiplied by the amount of flow belonging to the requester, i.e., $\sum_{Q^i_j \in Q^i} \sum_{r \in Q^i_j} c_r(x_r) x_{Q^i_j}$. If all feasible combinations of paths have costs higher than the requesters willingness to pay, she will reduce the amount of demand to be routed, so in addition we have

$$\sum_{Q^i_j \in Q^i} x_{Q^i_j} \sum_{r \in Q^i_j} c_r(x_r) \leq \pi_i \quad \forall i \in N \, . \tag{2}$$

Note that we defined a feasible flow explicitly over the feasible paths and not over the edges it contains. Therefore, edges connecting resources, source, and sink of a requester such that a path is created that does not correspond to a feasible bundle, do not pose a problem. The other way around, a feasible bundle always corresponds to a feasible path in the network.

A. Nash Equilibrium

We will assume that a requester will always route the highest possible demand such that costs do not exceed her willingness to pay. Among these options of highest possible demand, a requester chooses a feasible combination of paths that will minimize her effective costs. A requester may split her demand into integral parts over feasible paths to minimize costs. A stable outcome in this scenario is a Nash equilibrium of our defined atomic splittable routing game G defined as follows.

Definition 1 (Nash Equilibrium): A feasible flow $\mathbf{x}^{NE}$ is a *Nash equilibrium* (NE), if for all $i \in N$ and for all $Q^i_j, Q^i_k \in Q^i$ with $x^{NE}_{Q^i_j} > 0$, we have

$$\sum_{r \in Q^i_j \setminus Q^i_k} c_r(x^{NE}_r) \leq \sum_{r \in Q^i_k \setminus Q^i_j} c_r(x^{NE}_r) \, . \tag{3}$$

Equation (3) indicates that the costs would increase if flow was shifted from path Q^i_j (with positive flow in the NE) to path Q^i_k. As cost functions are non-decreasing, existence of an NE is guaranteed (see e.g., [28]). For tighter conditions on the cost functions, we can even guarantee uniqueness, i.e., when all cost functions c_r are polynomials of degree at most three (see [29]). A Nash equilibrium of our game can be computed via a Best-Response Dynamic (see e.g., [12]). An ϵ-approximation of a Nash equilibrium can be computed in polynomial time for all $\epsilon > 0$, given that the cost functions on the resources are affine [10]. Under certain symmetry conditions on the network structure, [30] furthermore present an exact polynomial time algorithm to compute a Nash equilibrium in atomic splittable routing games with affine cost functions.

B. Social Welfare and Social Optimum

The social optimum of our game corresponds to a solution of a central planner optimizing social welfare. In our model social welfare corresponds to the difference between the willingness to pay of the requesters and the total load-dependent costs.

Definition 2 (Social Welfare and Social Optimum): Given feasible flow $\mathbf{x}$, the value of function

$$S(\mathbf{x}) = \sum_{i \in N} \left(\pi_i - \sum_{Q^i_j \in Q^i} x_{Q^i_j} \sum_{r \in Q^i_j} c_r(x_r) \right) \tag{4}$$

is called the *social welfare* of $\mathbf{x}$. A *social optimum* corresponds to a vector $\mathbf{x}^S$ maximizing the social welfare function S over all feasible flow vectors.

C. Aspects of the Crowd

We now present several model extensions to depict relevant aspects in the interaction of the requesters with the crowd.

1) Engagement Constraints (Incentivization): Participants may only engage when a certain wage (monetary, idealistic or other form of a reward) level is reached. In monetary terms, the lowest wage level for engagement should be one that covers the costs for resource usage. From the costs of the solutions (that can be generated using the approaches described in Sec. II-A and II-B), we may infer these minimum rewards to engage crowd participants. The costs, respectively, rewards, may vary depending on the chosen solution approach but in any case correspond to the burden laid on the participants for using their resources during the operation of the MCS system.

As an extension of our model, additional rewards that are not related to load-dependent costs could be integrated by adding terms to the existing resource cost functions. More complex pricing schemes in our model will be the subject of a planned follow-up work.

2) Withdrawal Constraints: Crowd participants may withdraw from the MCS application, respectively, network, if certain costs are exceeded. To avoid this, for each resource $r \in R$, we introduce a threshold $b_r \in \mathbb{R}_+$. If the costs of a flow unit passing through r would exceed b_r, the participant would withdraw from the application. Solutions for this extension of our model can be computed by adding the following constraints to the flow formulation in Equations (1a), (1b), and (2):

$$c_r(x_r) \leq b_r \qquad \forall r \in R\,. \tag{5}$$

3) Quality under Uncertainty: When engaging participants in a MCS application, there may exist uncertainty about the quality of the results. We include uncertainty in our model by introducing a *certainty* γ_r for every resource $r \in R$. This certainty corresponds to the probability that a resource delivers a full quality result for its subtask. From the certainty of the resources, we can derive the *certainty of a path* $Q^i_j \in Q$ as

$$\gamma_{Q^i_j} = \prod_{r \in Q^i_j} \gamma_r\,. \tag{6}$$

Note that alternative measures for certainty are possible (see e.g., [19]). Every requester $i \in N$ will have a level of certainty δ_i for the average quality of her results that needs to be met. Solutions for this extension of our model can be computed by adding the following constraints to the flow formulation in Equations (1a), (1b), and (2):

$$\frac{\sum_{Q^i_j \in Q^i} x_{Q^i_j} \gamma_{Q^i_j}}{d_i} \geq \delta_i \qquad \forall i \in N\,, \tag{7}$$

i.e., the average certainty of one unit of demand routed must be greater or equal than the desired level of certainty for every requester.

III. Evaluation

In this section, we present a theoretical evaluation of the Nash equilibria of our model. Based on the model, we furthermore present a distributed mechanism for the automated generation of equilibria and, hence, for the efficient allocation of tasks in MCS. We conclude this section with a proof for the truthfulness of the mechanism.

A. Efficiency

We are interested in the *efficiency* of solutions, i.e., the relation between the social welfare of a solution and the social optimum. Of particular interest for us is the relation between the social optimum and the Nash equilibrium which will be included in our mechanism to balance loads in MCS systems in Sec. III-B. As indicator for the level of efficiency, we choose the *price of anarchy* (PoA), introduced by Koutsoupias and Papadimitriou [31] and ever since frequently studied in selfish routing games (see e.g., [28] for an introduction). The PoA corresponds to the worst possible ratio of social welfare of a social optimum and social welfare of an equilibrium.

Definition 3 (Efficiency and Price of Anarchy): For an instance I of game G, let $\mathbf{x}^{NE}$ denote the requesters flow vector at the instance's NE. Given furthermore a social optimum $\mathbf{x}^S$ of I, the quotient

$$PoA(I) = \frac{S(\mathbf{x}^S)}{S(\mathbf{x}^{NE})}$$

denotes the *efficiency* of instance I. The *price of anarchy* (PoA) is defined to be the maximum possible ratio over considered instances:

$$PoA = \max_{I \in G} \frac{S(\mathbf{x}^S)}{S(\mathbf{x}^{NE})}\,.$$

Recently, in [32], a set of exact bounds for atomic splittable routing games in the special case of a bounded-degree polynomial cost function with non-negative coefficients were presented. For a polynomial cost function of degree three this bound is 5.063, for degree two it is 2.549, and for degree one it is even as low as 1.5. In practical terms this means that we can provide distributed solutions of a proven level of quality that (or rather an ϵ-approximation of it) can be computed efficiently as described in Sec. II-A. As discussed in Sec. II-C1, the computed costs may also be interpreted as the minimum rewards needed to incentivize participants.

B. Truthful Equilibrium Generation

In this subsection, we present a simple mechanism to allocate tasks in a MCS system serving several requesters. A mechanism allocating tasks in MCS systems will in the further course of this paper be called *MCS mechanism*.

The Nash equilibrium provides efficient solutions for the task allocation problem in MCS systems. It is a decentralized outcome that gives no (selfish) requester an incentive to deviate from the strategy chosen in the equilibrium. It is a stable outcome under the assumption that requesters learn or have full knowledge about their own and the other requesters available strategy sets. It is an outcome of strategic, i.e., rational, selfish, and profit maximizing, requesters.

We will make use of these facts to create a MCS mechanism in which requesters will chose the equilibrium strategies in a decentralized way. We add additional aspects to the mechanism to make sure that requesters are truthful about available strategies and their willingness to pay. A pseudo code of the MCS mechanism is given in Algorithm 1. As a result of our mechanism, requesters individually choose strategies that converge to an equilibrium under the assumption that requesters are rational. Such a Nash

Algorithm 1 Truthful Equilibrium Generation

Initialize *Willingness to Pay*: Set and publish a fixed price per unit of demand that every requester using the MCS system must agree on paying in the worst case.

for all requester $i \in N$ **do**

Generate Feasible Sets: Requester i must announce her area of interest and will only receive data for this area.

end for

for all requester $i \in N$ **do**

Broadcast Knowledge: Release full (anonymized) knowledge about the strategies of all requesters and cost functions of all resources in the system.

end for

equilibrium could be also computed by a central operator of the MCS system and be given to the requesters in order to support a fast convergence towards an equilibrium. Our mechanism adheres to individual rationality, i.e., the profit of a requester will always be non-negative if she reveals her true strategy set.

Before we give a proof for the truthfulness of our MCS mechanism, we provide a formal definition of the term truthfulness (also see [12] for a definition of truthful mechanism in auctions). Without truthfulness the mechanism would be vulnerable to market manipulation and may produce very poor outcomes (see e.g., [33]).

Definition 4 (Truthful MCS Mechanism): A MCS mechanism is truthful if announcing true values for the willingness to pay and the areas of interest in the MCS system is a dominant strategy for every requester, i.e., if it is always better to give true values than lying about values.

Theorem 1: The mechanism presented in Algorithm 1 is truthful under the condition that the whole area under consideration in the MCS system is covered by crowd participants and requesters costs are bounded by the willingness to pay.

Proof: We prove the theorem by contradiction. Assume that it can be more lucrative for a requester to announce an untruthful value of her willingness to pay or her area of interest or a combination of both.

Case 1: Announce an untruthful value of the willingness to pay and a truthful value about the area of interest: There is a value for the willingness to pay that is fixed by the mechanism operator. Any announcement of a lower or higher willingness to pay will not change the utility (true willingness to pay minus effective costs) of a player.

Case 2: Announce a truthful value of the willingness to pay and an untruthful value about the area of interest: It could be that a player announces (1) an area that includes the true area of interest but is greater or (2) an area that only includes parts of the true area, or (3) an area that does not include the area of interest. The willingness to pay is fixed and we assume that all players have positive values for every subarea of their true area of interest. Obviously, the choice of (3) is irrational as the whole area under consideration in the MCS system is covered by crowd participants and the requester will have to pay for areas that are not of interest for her while receiving no useful data. If the area is greater than the actual area of interest as described in (1), the requester will pay for areas that are not of interest for her which is less lucrative than announcing her true value for the area of interest. If, as described in (2), the area announced only includes parts of the area of interest, the requester will not receive positive value for the rest of her area of interest and in addition pays for areas that are not of interest.

Case 3: Announce untruthful values about the willingness to pay and the area of interest: If a player actually has a higher willingness to pay than the value fixed by the mechanism operator, she could consider announcing an untruthful value about her area of interest. This cannot be lucrative, because all players are selfish and because of the assumption that costs are limited such that no (other) player will drop out (i.e., requesters costs are bounded by the willingness to pay). If (1) the area announced is greater, the effective costs may be below her willingness to pay but in any case higher than if she would announce her true area. In our game, feasible sets are tied to areas. This means, even if certain players place some demand in some subareas that are not of interest to them, this will never increase the costs in a fashion that another player drops out. So, costs for a player announcing a false area of interest will only be higher than for the true value. If (2) the area announced only holds parts of the area of interest or (3) the area announced does not include the area of interest, the player can increase her utility by shifting to the true area of interest. Announcing a lower willingness to pay cannot influence the system in any way has there exists a fixed value set by the mechanism operator.

■

IV. Conclusion

To the best of our knowledge, together with [11] our analysis is among the first initiatives to explicitly investigate task allocation in MCS systems using a game-theoretic selfish routing model. After the presentation of theoretical results for the Nash equilibrium, we outlined a distributed mechanism for the operation of MCS systems and proved its truthfulness. The costs of the generated distributed Nash equilibrium solutions are provably bounded. We conclude that selfish routing can be useful to model MCS systems.

Acknowledgment

This work was partially supported by the *Swiss National Science Foundation* (SNSF) - CRSII2-154458 *SwissSenseSynergy* project. The authors would also like to thank the reviewers of this paper for their valuable comments.

References

[1] W. Safire, "On Language," *New York Times*, February 5, 2009.

[2] "Amazon Mechanical Turk," http://www.mturk.com, retrieved Dec. 28, 2016.

[3] "LEGO Ideas," http://ideas.lego.com, retrieved Dec. 28, 2016.

[4] "IoT Lab," http://www.iotlab.eu, retrieved Jan. 6, 2017.

[5] "The Mahali Space Weather Monitoring Project," http://mahali.mit.edu/, retrieved Jan. 6, 2017.

[6] "WeSenseIt," http://wesenseit.eu, retrieved Jan. 6, 2017.

[7] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges." *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[8] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.

[9] M. Papandrea and S. Giordano, "Location prediction and mobility modelling for enhanced localization solution," *Journal of Ambient Intelligence and Humanized Computing*, vol. 5, no. 3, pp. 279–295, 2014.

[10] R. Cominetti, J. R. Correa, and N. E. Stier-Moses, "The impact of oligopolistic competition in networks," *Operations Research*, vol. 57, no. 6, pp. 1421–1437, 2009.

[11] J. Buwaya and J. D. P. Rolim, "Atomic routing mechanisms for balance of costs and quality in mobile crowdsensing systems," in *International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2017.

[12] N. Nisan, T. Roughgarden, É. Tardos, and Vijay V. Vazirani (Eds.), *Algorithmic Game Theory*. Cambridge University Press, 2007.

[13] J. Anselmi and N. S. Walton, "Decentralized proportional load balancing," *SIAM Journal on Applied Mathematics*, vol. 76, no. 1, pp. 391–410, 2016.

[14] D. Mukherjee, S. C. Borst, J. S. H. van Leeuwaarden, and P. A. Whiting, "Efficient load balancing in large-scale systems," in *Annual Conference on Information Science and Systems (CISS)*, 2016.

[15] Y. Zhao, S. Wang, S. Xu, X. Wang, X. Gao, and C. Qiao, "Load balance vs. energy efficiency in traffic engineering: A game theoretical perspective," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2013.

[16] V. Burger, M. Hirth, C. Schwartz, T. Hoßfeld, and P. Tran-Gia, "Increasing the coverage of vantage points in distributed active network measurements by crowdsourcing," in *Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance*. Springer, 2014, pp. 151–161.

[17] I. Abraham, O. Alonso, V. Kandylas, and A. Slivkins, "Adaptive crowdsourcing algorithms for the bandit survey problem," in *Conference on Learning Theory (COLT)*, 2013.

[18] A. Chatterjee, M. Borokhovich, L. Varshney, and S. Vishwanath, "Efficient and flexible crowdsourcing of specialized tasks with precedence constraints," in *IEEE Conference on Computer Communications (INFOCOM)*, 2016.

[19] J. Wang, J. Tang, D. Yang, E. Wang, and G. Xue, "Quality-aware and fine-grained incentive mechanisms for mobile crowdsensing," in *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2016.

[20] I. Koutsopoulos, "Optimal incentive-driven design of participatory sensing systems," in *IEEE Conference on Computer Communications (INFOCOM)*, 2013.

[21] T. Luo, H.-P. Tan, and L. Xia, "Profit-maximizing incentive for participatory sensing," in *IEEE Conference on Computer Communications (INFOCOM)*, 2014.

[22] R. Cavallo and S. Jain, "Efficient crowdsourcing contests," in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2012.

[23] S. Chawla, J. D. Hartline, and B. Sivan, "Optimal crowdsourcing contests," *Games and Economic Behavior*, 2015.

[24] D. DiPalantino and M. Vojnovic, "Crowdsourcing and all-pay auctions," in *ACM Conference on Electronic Commerce (EC)*, 2009.

[25] D. Acemoglu, M. Mostagir, and A. Ozdaglar, "Managing innovation in a crowd," National Bureau of Economic Research, Tech. Rep., 2014.

[26] A. Jarry, P. Leone, S. Nikoletseas, and J. D. P. Rolim, "Optimal data gathering paths and energy-balance mechanisms in wireless networks," *Ad Hoc Networks*, vol. 9, no. 6, pp. 1036–1048, 2011.

[27] J. Buwaya and J. D. P. Rolim, "Bounding distributed energy balancing schemes for WSNs via modular subgames," in *International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2016.

[28] T. Roughgarden, *Selfish routing and the price of anarchy*. MIT press Cambridge, 2005.

[29] E. Altman, T. Basar, T. Jimenez, and N. Shimkin, "Competitive routing in networks with polynomial costs," *IEEE Transactions on Automatic Control*, vol. 47, no. 1, pp. 92–96, 2002.

[30] C.-C. Huang, "Collusion in atomic splittable routing games," *Theory of Computing Systems*, vol. 52, no. 4, pp. 763–801, 2013.

[31] E. Koutsoupias and C. Papadimitriou, "Worst-case equilibria," in *Symposium on Theoretical Aspects of Computer Science (STACS)*, 1999.

[32] T. Roughgarden and F. Schoppmann, "Local smoothness and the price of anarchy in splittable congestion games," *Journal of Economic Theory*, vol. 156, pp. 317–342, 2015.

[33] P. Klemperer, "What really matters in auction design," *The Journal of Economic Perspectives*, vol. 16, no. 1, pp. 169–189, 2002.