Atomic Routing Mechanisms for Balance of Costs and Quality in Mobile Crowdsensing Systems

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

***Abstract*—We study the problem of distributing loads in mobile crowdsensing systems (MCS). In this context, we present a multi-commodity network game, more explicitly, an atomic routing game, to depict the linking of several crowd participants into bundles that are capable of successfully completing desired sensing tasks. The nodes of the network correspond to the resources of the crowd participants and the players of our game are sensing service requesters that wish to route their demand along paths trough the network. One resource may serve several requests at the same time, which can be modeled efficiently using the network structure. Resource usage involves load-dependent costs. Our model caters for the uncertainty inherent from crowd involvement and mobility by incorporating *certainty* parameters in the model. These certainty parameters describe the quality of the partial result a participant can produce. Requesters may set a minimum certainty level for the successful completion of their overall sensing tasks that has to be met.**

In our model, we analyze four different solution concepts for balancing loads with respect to costs and quality of results: (1) a distributed brute force approach (engaging all suitable crowd participants), (2) a random selection of suitable crowd participants, (3) a Nash equilibrium (as result of decentralized selfish cost-minimizing game play) and (4) a (centralized) social optimum. All considered distributed solutions or an ϵ-approximation of a solution can be computed efficiently (for affine cost functions). Furthermore, well-known results for the *price of anarchy* of atomic routing games can be transfered to our model, i.e., the relative solution quality of a Nash equilibrium compared to a social optimum is provably bounded. In addition, we provide an extensive experimental study that supports theoretical results and gives further suggestions on the impact of uncertainty. We merge the findings of our analysis into a truthful distributed mechanism such that requesters have no incentive to deviate from an efficient solution.

I. Introduction

Mobile crowdsensing systems (MCS) are a recent and emerging paradigm, which leverages the sensing data from the mobile devices of people (the 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 are used to improve public and individual services. Concrete examples include weather monitoring in rural areas of East Africa [1] as well as participatory citizen sensing systems for sharing information on water conditions and flooding in Vicenza, Italy and Doncaster, UK [2]. In MCS, a fundamental and non-trivial issue is the distribution of *loads* in terms of time and energy consumption as well as computational effort among heterogeneous mobile personal devices. These loads should be *balanced* to optimize short- and long-term system performance with respect to costs, quality of results, user satisfaction and further application-specific metrics. In this paper, we study the issue of balancing loads in MCS more specifically: A requester wants to collect sensing data about an area, but implementing commercial sensors is not feasible or 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. A concrete real-world MCS application that would benefit from the efficient balancing of loads is for example the area-related MCS application on weather monitoring presented in [1]. Load balancing in MCS helps allocating tasks in a cost and quality balancing manner.

In this paper, we present an atomic routing game to depict and analyze the problem of balancing loads in MCS. Each provider (crowd participant) owns a node (resource) in a multi-commodity network. On the other side, requesters route their demand over feasible paths (resource bundles that are capable of successfully completing desired sensing tasks). A resource may serve several requests at the same time, which can be modeled efficiently using the network model. Resource usage involves load-dependent costs. From the computed costs, we may infer minimal rewards to engage participants. Special aspects of our problem are the mobility and the autonomy of crowd participants. Theses aspects produce uncertainty about the suitability of a participant to complete a sensing task. We address these aspects by incorporating *certainty* parameters in the model. The exact values of these parameters in practice could for example be derived from the results of mobility prediction algorithms (see e.g., [3] and [4]). We furthermore include participation constraints for the crowd (when a personal costs limit is exceeded).

This paper will appear in the Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS) 2017.

In our model, we analyze four different solution concepts with respect to costs and quality of results: (1) a distributed brute force approach (engaging all suitable crowd participants), (2) a random selection of suitable crowd participants, (3) a Nash equilibrium (as result of decentralized selfish cost-minimizing game play) and (4) a (centralized) social optimum. A brute force solution and a random solution can be computed efficiently in polynomial time and in an environment with uncertainty, these methods cannot be dismissed as disadvantageous approaches without testing. Brute force may lead to an expensive but certain solution and a random solution is very easily generated and may in an environment with little knowledge about the participants be the best choice. An ϵ-approximation of a Nash equilibrium can also be computed in polynomial time for all $\epsilon > 0$, given that the cost functions on the resources are affine [5]. Furthermore, in the 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 the Nash equilibrium in relation to the social optimum is provably bounded. We will describe how to compute solutions and provide an extensive experimental study comparing solutions and giving further suggestions on the impact of uncertainty. We will merge the results of our evaluation into a decentralized mechanism for MCS that produces efficient distributed Nash equilibrium solutions. Our model and mechanisms are in particular suitable for MCS that deal with gathering sensing data about an area.

A. Our Contribution

In this work, we make the following major contributions: (1) We present an analysis of MCS involving complementary sensing resources using a game theoretical model that explicitly incorporates strategic behavior and uncertainty. (2) We present an efficient model in which participants may take part in several sensing tasks at the same time. (3) Using our model, distributed allocations of sensing tasks can be generated efficiently and the chosen atomic routing model allows furthermore the transfer of well-known theoretical results on the existence and quality of some solutions. In addition, (4) we provide an extensive experimental study that supports theoretical results and gives further suggestions on the impact of uncertainty. Furthermore, (5) we derive a decentralized load balancing mechanism such that requesters have no incentive to deviate from an efficient distributed Nash equilibrium solution. Our model and mechanism are easy to employ and suitable for various MCS.

B. Outline

In the following Sec. II, we give a brief overview on related work. In Sec. III, we present a formal definition of our model and considered solution concepts. In Sec. IV, we give a theoretical and experimental evaluation. In Sec. V, we propose a decentralized mechanism and in Sec. VI, we give a conclusion and an outlook to our work.

II. Related Work

There exists an increased interest in load balancing algorithms for MCS in recent years. Load balancing comes in different facets depending on the considered problems and issues introduced by the system designer. It is important whenever data is collected from multiple people and has frequently been studied in the field of game theory in relation to congestion control in communication networks (see e.g., [6], [7], [8]) and in crowdsourcing systems (see e.g., [9], [10], [11]). In MCS, notably [12], [13] and [14] deal with the issue of load balancing. In [12], an auction formulation for the distributed generation of task allocations in MCS 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 [12], not only our experimental tests suggest close-to-optimal solutions, the quality of these solutions is actually provably bounded. [13] presents cost-minimal mechanisms that provide a certain quality level. In contrast to our work, in [13], participants may only be involved in one sensing task at a time. Also [14] proposes an auction mechanism rather than an efficient network formulation as presented in our work. In [11], the authors provide a resource allocation problem formulation for more general crowdsourcing systems. Nevertheless, they, as well as [9] and [10], do not exploit the efficient structure of network routing games.

Related to our work are furthermore models of crowdsourcing contests. The authors of [15] study such crowdsourcing contests and present 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 [16] 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 works, each participant produces one complete end-product. We focus on crowdsensing of complementary resources, instead. Both of the latter works extend ideas of [17], which established the connection between crowdsourcing and all-pay auctions.

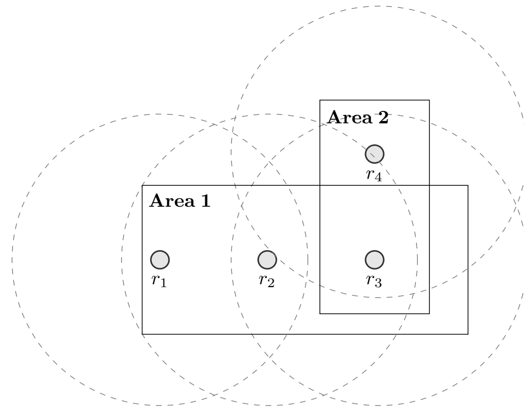
Also related to our work is [18], which analyzes crowdsourcing with focus on innovation. In [18], the problem of assigning tasks of unknown difficulty to crowd participants of unknown skill is modeled as an extended resource allocation problem. The authors present a decentralized mechanism, which produces a hierarchy such that participant 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 emphasize that one of the roots of MCS 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 load balancing algorithms, which maximize network lifetime (see e.g., [19] and [20]). In MCS, 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 the human factor. 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 is to optimize short- and long-term system performance with respect to costs, quality of results, user satisfaction and further application-specific metrics.

III. Model and Solution Concepts

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. In Fig. 1a, we give 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.

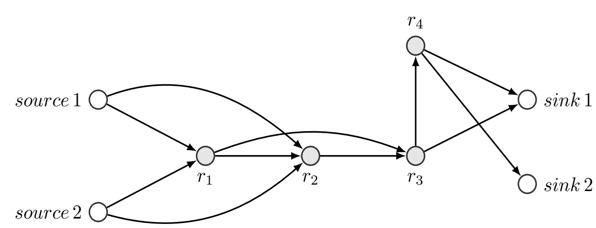
We turn to the formal definition of our model. The problem of allocating resources of crowd participants to requesters that desire bundles of complementary goods can be modeled as *atomic splittable routing game* $G = \{N, R, Q = \bigcup_{i \in N} Q^i, (c_r)_{r \in R}, (d_i)_{i \in N}, (\pi_i)_{i \in N}\}$. (See e.g., [21] 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



(a) Area Coverage Representation

Bundle	Area 1	Area 2
$r_1r_2r_3r_4$	yes	yes
$r_1r_2r_3$	yes	no
$r_1r_2r_4$	no	no
$r_1r_3r_4$	yes	yes
$r_2r_3r_4$	yes	yes
r_1r_2	no	no
r_1r_3	yes	no
r_1r_4	no	no
r_2r_3	yes	no
r_2r_4	no	no
r_3r_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

Fig. 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.

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 $x_r = \sum_{Q^i_j \in Q \colon r \in Q^i_j} x_{Q^i_j}$ 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

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

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 corresponds to a feasible path in the network.

In the following subsections, we present several routing options for the requesters (solution concepts) and how to model aspects relevant in the interaction with the crowd. We will assume that a requester will always route the highest possible demand such that the corresponding solution adheres to her willingness to pay and to the chosen solution concept. Later in our evaluation in Sec. IV, we will then analyze induced costs and the impact of introducing uncertainty and quality guarantees.

A. Brute Force

Every requester chooses to engage all available participants within her area of interest and, hence, to route demand along her longest feasible path (going through all resources within her area of interest).

B. Random Choice

Every requester randomly chooses a set of participants covering her area of interest and, hence, to route demand along some randomly chosen feasible path.

C. Nash Equilibrium

Every requester i chooses a feasible combination of paths that will minimize her effective costs (while routing the maximum amount of demand through the network). 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.

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., [21]). 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 [22]). A Nash equilibrium of our game can be computed via a Best-Response Dynamic (see e.g., [23]). 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 [5]. Under certain symmetry conditions on the network structure, [24] furthermore presents an exact polynomial time algorithm to compute a Nash equilibrium in atomic splittable routing games with affine cost functions.

D. 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.

E. Aspects of the Crowd

1) Engagement Constraints (Incentivization): In cases where participation is not voluntary, participants may only engage when a certain wage level (monetary, idealistic or other form of a reward) 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. III-A to III-D), 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 MCS.

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., [12]). 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.

IV. Evaluation

In this section, we present a theoretical and an experimental evaluation our model and the various solution concepts.

A. Theoretical Evaluation

We begin with a theoretical analysis. 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 in Sec. V. As efficiency measure, we choose the *price of anarchy* (PoA), introduced by Koutsoupias and Papadimitriou [25] and ever since frequently studied in selfish routing games (see e.g., [21] 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:

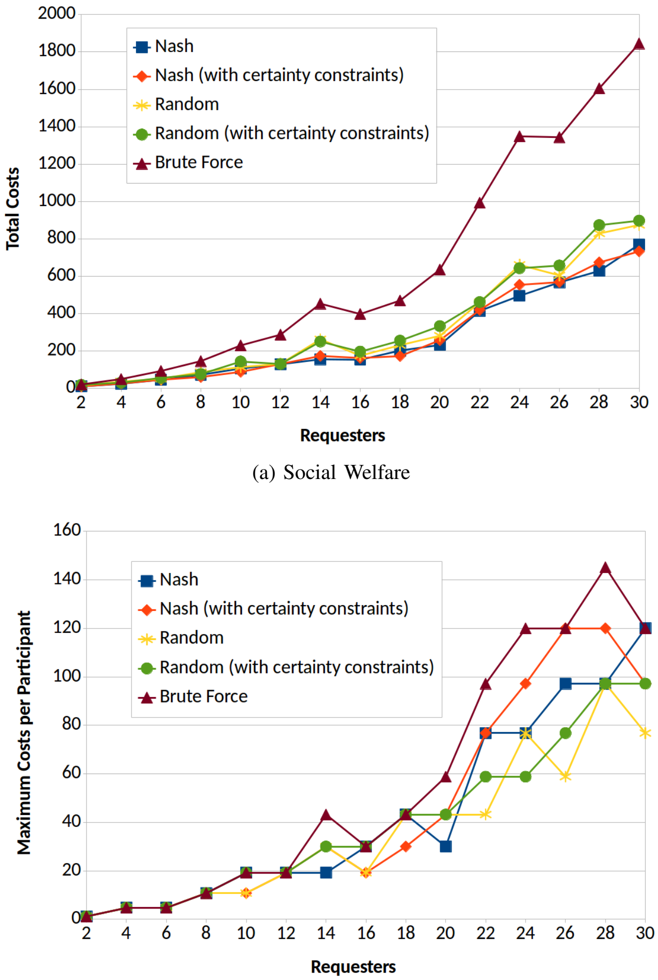
$$PoA = \max_{I \in G} \frac{S(\mathbf{x}^S)}{S(\mathbf{x}^{NE})}\,.$$

Recently, in [26], 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. III-C.

B. Experimental Evaluation

In this subsection we present a comprehensive simulation study. We use experimental tests to draw conclusions on the induced costs of the distributed solutions and the impact of introducing quality guarantees under uncertainty. Simulations are performed using Python 2.7.12. Linear and quadratic programs to compute solutions are solved using Gurobi 7.0.1.

1) Simulation Settings: In our simulations, we are looking at square fields with width $A = 100$ (e.g., in meters), and thus fields of size $A^2 = 10,000$ (e.g., in square meters). The width w of a subsquare covered by a single crowd participant is set to 10 (for simplicity reasons, we assume that subareas covered by participants equal squares instead of circles). We set the demand of every requester equal to $d = 1$ in order to simplify the evaluation of other aspects



(a) Social Welfare

(b) Maximum Costs per Participant

Fig. 2. Social welfare (positive total costs) and maximum costs imposed on a participant in the distributed solution concepts of the Nash equilibrium (without and with certainty constraints), the Random Choice approach (without and with certainty constraints) and Brute Force versus the number of requesters in the MCS.

such as quality under uncertainty and the difference between the various solution concepts. We choose homogeneous linear resource cost functions $c(x_r) = 1.2 \cdot x_r$ (with x_r being the load on resource $r \in R$ in a solution). When uncertainty is considered, we set the desired certainty parameter of every requester to $\delta = 0.8$. In our experimental tests, we will leave out the withdrawal constraints of the participants (presented as options in Sec. III-E2) as well as the willingness to pay of the requesters (i.e., we set the willingness to pay to infinity for every requester). In our experiments, these artificial thresholds would disturb our results and analysis, which here rather focuses on the aspects of costs and quality under uncertainty. We set the average number k of crowd participants within a subsquare of the overall area under consideration to 5. For each number n of requesters between 2 and 30 (in steps of 2), we randomly create 50 test instances and compute resulting costs for the different distributed solution concepts (with and without certainty constraints). Per instance, we conduct 60 iterations of the simulation to compute the average performance of a solution concept. We depict results for the mean values for each metric as results demonstrate strong concentration around the mean. An instance of our game is generated as follows:

Given width A, we assume that the overall area under consideration for potential sensing tasks is a square of size A^2. To determine the minimum number of participants needed to make any request within the overall area feasible, we divide the area into squares of width w, i.e., we tessellate the overall area into $\hat{m} = \lceil \frac{A^2}{w^2} \rceil$ squares of the same size and put a participant in the middle of each square. We then randomly generate $(k-1) \cdot \hat{m}$ other points within the overall area and place a participant there. Now, for each participant, we randomly draw a certainty between 0.1 and 1.0 from a normal distribution. For each requester, we randomly draw a central point within the overall area from a uniform distribution as well as the width of her subarea between $0.2 \cdot A$ and $0.6 \cdot A$. (Note that with this construction it is possible that a requester only needs data from a single participant if her central point lies "close" to the borders of the overall area). The following steps describe how to determine the feasible resource bundles for the requesters from here.

For every requester i:

1) Create the largest feasible resource bundle equivalent to all the participants (resources) present in her subarea of interest.
2) For every participant in the bundle, remove her and verify if the result is still a feasible bundle, i.e., covers the desired subarea.
3) If true, add the newly constructed bundle to the set of feasible bundles and go to step 2), else dismiss the bundle.

This finishes the description of the generation of a game instance. For the computation of solutions, we use the according equations presented in Sec. III.

2) Performance Measures and Evaluation: We evaluate our solution concepts with respect to social welfare of a solution and the maximum costs imposed on a participant. In Fig. 2, the results of our experimental tests for different numbers of requesters are displayed. Please note the differing scales of the graphics in the figure. We show result for the Nash equilibrium, the Random Choice approach (see Sec. III-B) and the Brute Force approach (see Sec. III-A).

a) Social Welfare (Total Costs): In Sec. III-D, we gave a formal definition of the social welfare of a solution. If a willingness to pay for the requesters is given, it corresponds to the difference between the willingness to pay of the requesters and the total load-dependent costs. In our evaluation, we focus on quality and costs of a solution, hence the willingness to pay was not set for the requesters. Social welfare simply corresponds to the (negative) total costs imposed on the participants for the completion of sensing tasks. In Fig. 2a, the social welfare (positive total costs) for the different distributed solution concepts with respect to the number of requesters is presented. Following a basic intuition, Brute Force performs much worth than the other distributed approaches. With growing number of requesters in the application, total costs increase to more than double of the costs of the other solutions. The Nash equilibrium performs best and the distance between its costs and the costs of the Random Choice approach increases with growing number of requesters. Nevertheless, Random Choice might not be a bad choice when information necessary to generate an equilibrium is not given.

The certainty requirements introduced in the system, slightly diminish the performances with respect to costs for all solution concepts. We conclude that with a "balanced" distribution of certainties in the instances (as done in the settings of our experiments), quality under uncertainty can be achieved without major cutbacks.

b) Maximum Costs per Participant: The maximum costs per participant give us additional insides on the impact of a solution on an individual of the crowd. In Fig. 2b, we present the simulation results for the maximum costs imposed on a crowd participant in a solution. Interestingly, in contrast to the total costs imposed on the system, when looking at local costs at the participants side, the choice of the solution approach has a lower (worst-case) impact. Maximum individual costs are clearly rising with the number of requesters in the system, while even in the best social welfare solution among our approaches (i.e., for the Nash equilibrium), a single participant may have higher costs than in the Random Choice approach. Still, also here the Brute Force approach is the one that demands the most of an individual.

In summary, we conclude that the Nash equilibrium outperforms the other two distributed approaches regarding social welfare. When uncertainty is present and there are restrictions to assure quality of the solutions, our experimental tests suggest that the effect on costs is only minor as long as certainty is "well-balanced" over all participants. Maximum costs for a participant are very similar for all approaches. For a participant, the choice of the solution approach may not make a big difference, while for the requesters, the Nash equilibrium is the best choice. In particular, for the requesters this is also the case because, in addition, theoretical results guarantee that costs are bounded in comparison to the social optimum. As discussed in Sec. III-E1, the computed costs may also be interpreted as the minimum rewards needed to incentivize participants.

V. Mechanism

In this section, we derive a simple mechanism to balance quality and costs in MCS serving several requesters.

Our evaluation in Sec. IV yielded that the Nash equilibrium provides favorable solutions for the load balancing problem in MCS. The Nash equilibrium 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 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.

Our mechanism works as follows:

1) *Demand and Willingness to Pay*: Set a fixed price per unit of demand (e.g., in time units) that every requester using the MCS must agree on paying in the worst case.
2) *Feasible Sets*: The requesters must announce their areas of interest and will only receive data for these areas.
3) *Broadcast Knowledge*: Provide full (anonymized) knowledge about the strategies and costs of all requesters to all requesters in the system.

The logical result of our mechanism is that requesters individually choose strategies that converge into an equilibrium under the assumption that requesters are rational. Such a Nash equilibrium could also be computed by a central operator of the MCS and be given to the requesters as suggestion in order to support a fast convergence towards an equilibrium. Note that lying about her strategies would hurt a requester as she would then either have to pay for coverage of the whole area indicated and/or get data that does not satisfy her needs.

VI. Conclusion

To the best of our knowledge, our analysis represents the first initiative to explicitly investigate load balancing in MCS using a game-theoretic network flow model with load-dependent costs. After the presentation of a theoretical and experimental analysis that provided favorable results for the Nash equilibrium as distributed solution concept in MCS, we derived a distributed mechanism for the operation of these systems. The costs of the distributed Nash equilibrium solutions are provably bounded. Our experiments suggested that even when uncertainty is present in the system and there are restrictions to assure the quality of the solutions, the costs do not drastically increase as long as the certainties of the crowd participants follow a normal distribution. Extending our mechanism to dynamic changes of the system (regarding number of participants and requesters) is planned for future work. Furthermore, in future work, we plan to investigate how pricing schemes as presented in [16] and [12] improve solution performance. Intuitively, setting prices should reduce uncertainty in the model. To what extent this holds, would also be an interesting topic for a real-world study. In addition, in an extension of our model, we plan to integrate further aspects related to the mobility and the autonomy of the participants. E.g., participants could be empowered to choose a position anticipating the requesters choice of resource usage. More explicitly, in this scenario, the areas of interest of the requesters would be announced to the participants such that the participants could choose favorable positions maximizing their profit. Changing positions may involve additional costs that have to be included in the model and participants could choose multiple positions over time. We also consider to explicitly include mobility prediction algorithms for ad hoc scenarios of MCS.

Acknowledgment

This work was partially supported by the Swiss National Science Foundation (SNSF) - CRSII2-154458 *SwissSenseSynergy* project and an University of Geneva Digital Fellows Grant. In addition, the authors would like to thank the anonymous reviewers of this paper for their valuable comments.

References

[1] "The Mahali Space Weather Monitoring Project," http://mahali.mit.edu/, retrieved Jan. 6, 2017.
[2] "WeSenseIt," http://wesenseit.eu, retrieved Jan. 6, 2017.
[3] 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.
[4] 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.
[5] 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.
[6] J. Anselmi and N. S. Walton, "Decentralized proportional load balancing," *SIAM Journal on Applied Mathematics*, vol. 76, no. 1, pp. 391–410, 2016.
[7] 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.
[8] 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.
[9] 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.
[10] I. Abraham, O. Alonso, V. Kandylas, and A. Slivkins, "Adaptive crowdsourcing algorithms for the bandit survey problem," in *Conference on Learning Theory (COLT)*, 2013.
[11] 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.
[12] 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.
[13] I. Koutsopoulos, "Optimal incentive-driven design of participatory sensing systems," in *IEEE Conference on Computer Communications (INFOCOM)*, 2013.
[14] T. Luo, H.-P. Tan, and L. Xia, "Profit-maximizing incentive for participatory sensing," in *IEEE Conference on Computer Communications (INFOCOM)*, 2014.
[15] R. Cavallo and S. Jain, "Efficient crowdsourcing contests," in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2012.
[16] S. Chawla, J. D. Hartline, and B. Sivan, "Optimal crowdsourcing contests," *Games and Economic Behavior*, 2015.
[17] D. DiPalantino and M. Vojnovic, "Crowdsourcing and all-pay auctions," in *ACM Conference on Electronic Commerce (EC)*, 2009.
[18] D. Acemoglu, M. Mostagir, and A. Ozdaglar, "Managing innovation in a crowd," National Bureau of Economic Research, Tech. Rep., 2014.
[19] 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.
[20] 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.
[21] T. Roughgarden, *Selfish routing and the price of anarchy*. MIT press Cambridge, 2005.
[22] 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.
[23] N. Nisan, T. Roughgarden, É. Tardos, and Vijay V. Vazirani (Eds.), *Algorithmic Game Theory*. Cambridge University Press, 2007.
[24] C.-C. Huang, "Collusion in atomic splittable routing games," *Theory of Computing Systems*, vol. 52, no. 4, pp. 763–801, 2013.
[25] E. Koutsoupias and C. Papadimitriou, "Worst-case equilibria," in *Symposium on Theoretical Aspects of Computer Science (STACS)*, 1999.
[26] 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.