Mobile Crowdsensing Incentives under Participation Uncertainty

Panagiota Micholia
Department of Informatics
Athens University of
Economics and Business
104 34 Athens, Greece
panamixo@aueb.gr

Merkourios Karaliopoulos
Department of Informatics
Athens University of
Economics and Business
104 34 Athens, Greece
mkaralio@aueb.gr

Iordanis Koutsopoulos
Department of Informatics
Athens University of
Economics and Business
104 34 Athens, Greece
jordan@aueb.gr

ABSTRACT

Mobile crowdsensing applications rely on crowds of contributors who are willing to carry out certain tasks of interest. This willingness varies widely across users and tasks and (monetary) incentives are often engaged to strengthen it or make up for its total absence. These incentives need to carefully account for the highly non-homogeneous response of users to external motivation.

We propose a framework that explicitly accounts for the implicit uncertainty about the eventual participation and contributions of users. In the framework, the impact of incentives on the user choice to contribute or not is modeled probabilistically. First, we formulate a convex optimization problem for incentive allocation with the goal of achieving maximum expected quality while taking into account task budget limitations and constraints related to the physical locations of users and tasks. We then propose an iterative algorithm to alleviate the complexity of the original problem with two basic steps: an allocation step that applies incentive allocation to a set of available contributors; and a refinement step that revokes portions of the allocated incentives. We study the performance characteristics of our algorithm by comparing it to a default solver for convex optimization problems.

CCS Concepts

•**Human-centered computing → Ubiquitous and mobile computing;** *Ubiquitous and mobile computing theory, concepts and paradigms;*

MSCC'16: 3rd ACM Workshop for Mobile Sensing, Computing and Communication '16 Paderborn, Germany.

DOI: http://dx.doi.org/10.1145/2940353.2940357

Keywords

Mobile crowdsensing; participation uncertainty; incentive allocation

1. INTRODUCTION

Photo shoots of city points of interest, regional noise or pollution maps, monitoring of automatic vendor machines across an area are only few application examples powered by *mobile crowdsensing.* In all cases, the eventually delivered service is made up of small individual contributions of mobile user crowds through their smart devices. The announcement of crowdsensing campaigns, the recruitment of contributors out of the mobile crowd and, often, the transformation of the raw crowd data into a meaningful service are under the responsibility of an intermediate entity, called the *service provider.*

Yet, the participation of mobile users in crowdsensing campaigns cannot be taken for granted. Personal interests, time availability, and intrinsic motivation vary widely across individual users and more often than not (monetary) incentives are indispensable for encouraging their contributions. More importantly, the user incentivization process needs to acknowledge the uncertainty about the users' response to the offered incentives. Although, stronger incentives, i.e. higher payments, increase the chances of user participation, they do not guarantee it. End users are the ones that make the final decisions. Hence, it is crucial to find ways for incentivizing their contributions while explicitly accounting for their participation uncertainty.

1.1 Related work

One major research thread in mobile crowdsensing incentives relates to reverse auction-based incentive mechanisms. Users are recruited for tasks, and payments are made to them depending on the bids they submit, stating how much they charge for their contributions. The main concern and criterion for selecting users varies across studies. Non-exhaustively speaking, in the participatory system in [5], the payments made to the users are based on their participation level; in [3] the recruitment of sensors takes into account coverage concerns; whereas the quality of contributed data is the decisive factor for the reimbursement of users in [4] and in [11].

As far as user's participation is concerned, Lee et al. in [6] designed a reverse auction dynamic pricing mechanism assuming that user participation is ensured if the benefit they attain is above a certain threshold. Lin et al. [1] study a location and time aware sensor selection problem and propose a Lyapunov, based Vickrey-Clarke-Groves (VCG) auction policy for compensating the sensing cost of selected users. They connect users' participation to the probability of being selected and wish to keep it over a threshold. In [12] we find approaches with implicit user involvement in the determination of the incentives (platform-centric) and explicit user involvement (user-centric). Interested users decide to participate and submit sensing plans knowing that they will get reimbursed. Common to all incentive mechanisms relying on reverse auctions is the assumption that the determined payments commit the users' participation to the crowd-sensing campaigns, even when those payments fall short of the strategically submitted bids.

Departing from standard reverse auctions, authors in [7] use all-pay auctions, where users exert some effort and submit it to the platform. However, this kind of contest could risk the participation of users that do not get reimbursed. Tullock contests are proposed in a follow-up work in [8] to offer all users a probability to win. Authors in [2] present a pricing mechanism where users have explicit involvement in the determination of their payments as they individually bargain their prices with the platform in order to reach an arrangement. Uncertainty is considered in [9], which is probably the most relevant work to ours. Thereby users aim to capture the responsiveness of users to service consumption, but the study is restricted to a single task scenario.

1.2 Our contribution

In this paper, we deviate from the main body of the existing literature and explicitly consider the uncertainty about the users' participation in mobile crowdsensing campaigns. Contrary to [6] and [1], we model user willingness to participate as a continuous function of the incentives rather than a step function with a predefined threshold, whose definition in real world applications is unclear. Compared to [9], we relate the probability of contributors' participation on quantitative incentives offered in a *multiple task* scenario. We formulate a convex optimization problem (section 2) for the incentive allocation process, aiming at maximizing the total expected quality of contributions under task- and contributor related constraints. We propose an iterative approximate algorithm (section 3) that decomposes the convex optimization problem into two basic iterative steps: a payment allocation step where the user-to-task physical proximity constraints are relaxed, and a payment refinement step, where payments are corrected to account for the former constraints. This decomposition greatly reduces the computational complexity of the algorithm and renders it amenable to parallelization. The algorithm's efficiency and properties are demonstrated in section 4.

2. SYSTEM MODEL

2.1 Problem description

We consider a platform that manages a set of crowdsensing tasks $\mathcal{K}$, with $|\mathcal{K}| = K$. Each task is indexed by a number $j \in \{1, 2, \ldots, K\}$ and is directly related to a physical location i.e. a photo shoot of a particular point of interest. Tasks are requested by customers i.e. an owner of a local coffee shop that issues a request for a crowd photo shoot of their shop, and they demand contributions from a set of available potential contributors $\mathcal{N}$, with $|\mathcal{N}| = N$. Each member i of this set, $i \in \{1, 2, \ldots, N\}$ possesses different skills and interests, which render them more suitable for certain tasks; quality indices $q_{ij} \in [0, 1]$ capture the quality of contribution that user i can make to task j. Likewise, users differentiate with respect to their sensitivity γ_i to extrinsic incentives (payments) and the overall distance D_i they are willing to traverse in order to make their contributions to various tasks.

Each customer of the platform offers a certain budget B_j for incentivizing contributions to its requested task j. This budget is managed by the service provider who manages the platform of crowdsensing tasks. Upon particular points in time, the service provider considers the distance of each user i from task j denoted d_{ij} and shares the task budgets allocating payments p_{ij}. A crucial assumption in our model is that the propensity of users to contribute to a task varies with the amount of extrinsic incentives presented to them, albeit to different extent per user. Some of them are more responsive than others to even small payments, whereas others may only be mobilized by higher offers. Technically, we capture this participation uncertainty in willingness functions $w_{ij}(p_{ij})$ with

$$w_{ij}(p_{ij}) = 1 - e^{-\gamma_i p_{ij}} \tag{1}$$

Hence, the willingness functions are increasing concave probabilistic functions within $[0, 1]$ exhibiting diminishing returns to the offered amounts of payments, i.e. they strictly monotonically increase with payments but at lower rate as the latter get higher. The parameter γ_i captures the user-specific sensitivity to payments; it makes sense to allocate payments to contributors that will be more responsive to them.

2.2 Problem formulation

The goal, then, of the service provider is to maximize the expected aggregate quality of attained contributions over all tasks and contributors by determining proper payment offers to the mobile crowd. The users receive the offers for ongoing tasks via their smartphone devices and decide whether to participate and to how many tasks, accounting for their physical distance of the proposed tasks and the total walking effort to reach the respective locations. Formally, the task of the service provider is to allocate payments among users so as to maximize the total expected quality of tasks (P1). The problem is a convex optimization problem with task-oriented linear equality constraints referred to as *budget constraints* and user-oriented convex inequality constraints referred as *distance constraints.* The budget constraints reflect the fact that the to-be-allocated payments are ultimately constrained by limited task budgets designated from the customers to the platform.

$$\begin{aligned} \underset{p_{ij}}{\text{maximize}} \quad & \sum_{j\in\mathcal{K}}\sum_{i\in\mathcal{N}} q_{ij} w_{ij}(p_{ij}) \\ \text{subject to} \quad & \sum_{i\in\mathcal{N}} p_{ij} = B_j, \ \ \forall j \in \mathcal{K} \qquad (P1) \\ & \sum_{j\in\mathcal{K}} d_{ij} w_{ij}(p_{ij}) \le D_i, \ \ \forall i \in \mathcal{N} \\ & p_{ij} \ge 0, \ \ \forall i \in \mathcal{N}, \forall j \in \mathcal{K} \end{aligned}$$

The distance constraints capture the fact that the expected total distance contributors are willing to traverse in response to the offers made to them, is limited by their individual "walking budgets" D_i.

3. AN APPROXIMATE ALGORITHM

At first sight, we could solve the convex problem (P1) for its global optimum via brute-force application of a standard solver, i.e. the $MATLAB^{®}$ built-in solver for convex optimization problems. However, it turns out (Section 4) that the running times of such solvers scale badly with the number of tasks and users, limiting their actual applicability to practical instances of the problem. Therefore, we propose a low-complexity approximation algorithm that decomposes the original problem (P1) into simpler subproblems that can be solved optimally and fast through a number of iterations.

Each iteration involves two basic steps, as shown in Algorithm 3. First, in the *payment allocation step* of Algorithm 1, tentative payment offers are computed for the users without taking into account their distance constraints. Then, in the *payment refinement* of Algorithm 2, the payments are cut to make them compatible with the distance constraints. The budget that is accumulated out of the excess payment cuts is anew allocated to contributors, who are not yet constrained by distance constraints. The steps are repeated until the computed offers are constraint-compatible for all contributors.

The problems encountered in each step are simpler and can be solved optimally. Moreover, in each step we end up solving these problems *separately* for each task (payment allocation step) and each user (payment refinement step). This lends to trivial parallelization of the involved computation effort and results in even more savings in actual implementations. We further demonstrate these advantages of our algorithm in Section 4.

3.1 Initialization step

We relax the original problem (P1) by omitting the distance inequality constraints and derive problem (P2i):

$$\begin{aligned} \text{maximize} \quad & \sum_{j\in\mathcal{K}}\sum_{i\in\mathcal{N}} q_{ij} w_{ij}(p_{ij}) \\ \text{subject to} \quad & \sum_{i\in\mathcal{N}} p_{ij} = B_j, \ \ \forall j \in \mathcal{K} \qquad (P2i) \\ & p_{ij} \ge 0, \ \ \forall i \in \mathcal{N}, \forall j \in \mathcal{K} \end{aligned}$$

Problem (P2i) is a convex optimization problem with equality constraints. It is solved exactly by the water-filling algorithm [10], determining the optimal payments to contributors in the absence of individual distance constraints. Intuitively, the optimal value of (P2i) is greater or equal than the one of (P1) since the feasible solution space of (P2i) is a superset of that of (P1).

3.2 Payment refinement step

The solution of (P2i) yields two candidate contributor sets. A set of contributors that violate the distance constraint, $\mathcal{N}_v^0$, and a set of contributors that do not, $\mathcal{N}_a^0$. More generally, by the time this step initiates iteration t, $t \in \{1, 2, 3, \dots\}$, the candidate contributors are partitioned in sets $\mathcal{N}_v^t$, with $|\mathcal{N}_v^t| = N_v^t$ and $\mathcal{N}_a^t$, with $|\mathcal{N}_a^t| = N_a^t$. For contributors in $\mathcal{N}_v^t$, there exist different options to remove part of the non-zero tentative payments computed for them in the payment allocation step of the previous iteration so that they do comply with the distance constraints. Apparently, these payment cuts, δ_{ij}, are upper bounded by the current payment allocations $\delta_{ij} \le p_{ij}$. The goal in this step, is to find the optimal combination of such values δ_{ij} such that the resulting loss in the expected quality $L_Q(\delta_{ij})$ is minimized:

$$L_Q(\delta_{ij}) = \sum_{j\in\mathcal{K}}\sum_{i\in\mathcal{N}_v^t} q_{ij} \cdot \big(w_{ij}(p_{ij}) - w_{ij}(p_{ij} - \delta_{ij})\big) \quad (2)$$

Since the first term does not involve δ_{ij}, the problem of minimizing $L_Q(\delta_{ij})$ is equal to maximizing the second term. Moreover, the problem can be tackled independently for each user so that we are faced with N_v^t different convex optimization problems (P2r), one for each user i in $\mathcal{N}_v^t$, with a concave equality constraint and a bound on the values of δ_{ij}. These problems are of the following form:

$$\begin{aligned} \underset{\delta_{ij}}{\text{minimize}} \quad & -\sum_{j\in\mathcal{K}} q_{ij} w_{ij}(p_{ij} - \delta_{ij}) \\ \text{subject to} \quad & \sum_{j\in\mathcal{K}} d_{ij} w_{ij}(p_{ij} - \delta_{ij}) = D_i \qquad (P2r) \\ & 0 \le \delta_{ij} \le p_{ij}, \forall j \in \mathcal{K} \end{aligned}$$

Noticing that the term $w_{ij}(p_{ij} - \delta_{ij})$ appears in the objective function and the equality constraint, we can transform the problem (P2r) to a linear optimization counterpart problem (P3r) setting $x_{ij} = w_{ij}(p_{ij} - \delta_{ij})$.

$$\begin{aligned} \underset{x_{ij}}{\text{minimize}} \quad & -\sum_{j\in\mathcal{K}} q_{ij}x_{ij} \\ \text{subject to} \quad & \sum_{j\in\mathcal{K}} d_{ij}x_{ij} = D_i, \qquad (P3r) \\ & 0 \le x_{ij} \le 1 - e^{p_{ij}}, \ \ \forall j \in \mathcal{K} \end{aligned}$$

where $\delta_{ij} = \ln(1-x_{ij})+p_{ij}$. By the end of this step, the contributors in $\mathcal{N}_v^t$ are allocated payments $p_{ij}-\delta_{ij}$ such that the distance constraints are satisfied (and active). If $\mathcal{N}_s^t$ is the set of these "saturated" contributors by iteration t, with $|\mathcal{N}_s^t| = N_s^t$, it holds $\mathcal{N}_s^t = \mathcal{N}_s^{t-1} \cup \mathcal{N}_v^{t-1}$, i.e. contributors who end up violating the distance constraints in the previous payment allocation step, are turned into saturated contributors in the subsequent payment refinement step and are no further considered in future payment allocation steps.

3.3 Payment allocation step

Summing across tasks the payment cuts δ_{ij}, determined in the payment refinement step, we get residual budget amounts Θ_j that can be re-distributed across contributors in $\mathcal{N}_a^t$.

$$\Theta_j = \sum_{i\in\mathcal{N}_v^t} \delta_{ij} \tag{3}$$

The aim with these allocations is to maximize their impact on the expected aggregate quality of contributions. The complementary payments θ_{ij} should be such that the gain $G_Q(\theta_{ij})$ is maximized.

$$G_Q(\theta_{ij}) = \sum_{i\in\mathcal{N}_a^t}\sum_{j\in\mathcal{K}} q_{ij}\cdot\big(w_{ij}(p_{ij}+\theta_{ij}) - w_{ij}(p_{ij})\big)$$

The second term does not change with θ_{ij}, so the optimization is applied to the first term, which can be split into $k \le K$ different summation components, one for each task j with $\Theta_j > 0$. Therefore, at this step, we solve k different problems of the following form

$$\begin{aligned} \underset{\theta_{ij}}{\text{minimize}} \quad & -\sum_{i\in\mathcal{N}_a^t} q_{ij}w_{ij}(p_{ij}+\theta_{ij}) \\ \text{subject to} \quad & \sum_{i\in\mathcal{N}_a^t} \theta_{ij} = \Theta_j \qquad (P2a) \\ & \theta_{ij} \ge 0 \ \ \forall i \in \mathcal{N}_a^t \end{aligned}$$

(P2a) is a convex optimization problem with linear constraints and can be solved optimally applying the same waterfilling algorithm that we use for the solution of (P2i). By the end of this step, the computed offers for contributors in $\mathcal{N}_a^t$ equal $p_{ij}+\theta_{ij}$.

3.4 Algorithm correctness

Intuitively, there are two possibilities for the algorithm termination. In the first case, upon execution of the payment allocation step (P2a, P2i) during iteration $t \in \{0,1,2,\dots\}$, the distance constraints are not violated by any contributor, $\mathcal{N}_v^t = \varnothing$. This implies that $N_a^t > 0$ and $N_s^t < N$, i.e. the budgets of tasks are fully allocated and there is at least one contributor for whom the distance constraint is not active. The second possibility is that after a payment refinement step $N_s^t = N$, i.e. the set of contributors with active distance constraints grows all the way to include the full set of users and $N_a^t = 0$. Hence, there is budget unspent since no contributor's payment can grow more without violating their corresponding distance constraint.

Algorithm 1: Payment allocation step

```
Data: $B_j, N_a^t, K, q_{ij}$
Result: optimal payments: $p_{ij}$
Allocate budget $B_j$ to $N$ contributors in $N_a^t$ to
  maximize expected quality of tasks $K$.
```

Algorithm 2: Payment refinement step

```
Data: $D_i, N_v^t, K, q_{ij}, d_{ij}, p_{ij}$
Result: optimal payment cuts: $\delta_{ij}$
Remove $\delta_{ij}$ from contributors in $N_v^t$ to minimize
  the loss in expected quality of tasks $K$.
```

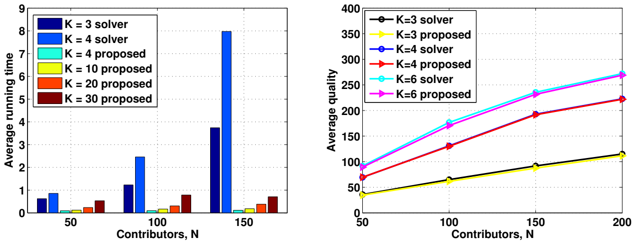
More formally, we can show that $\mathcal{N}_a^t$ decreases over time till it either becomes the null set or remains constant across two successive iterations, through an induction argument. *Base case*: For t = 0, $N_a^0 = N$. At the end of the first iteration $t = 1$, we have that $N_a^1 = N_a^0 - N_v^1$. If $N_v^1 > 0$, we get $N_a^1 < N_a^0$; if $N_v^1 = 0$, then the algorithm terminates. *Assumption step*: Let the statement about the algorithm's termination conditions hold for $t = n$, i.e. $N_a^n \le N_a^{n-1}$. *Induction step*: We will show that it holds for $t = n+1$, i.e. $N_a^{n+1} \le N_a^n$. If the inequality in the assumption step holds as equality, then $N_a^{n+1} = N_a^n = N_a^{n-1}$ and the algorithm already terminates by the n^{th} iteration. Otherwise, it either holds $N_a^{n+1} < N_a^n$, when $N_v^{n+1} > 0$, or $N_a^{n+1} = N_a^n$ when $N_v^{n+1} = 0$. Hence, the original statement about the two termination conditions of the algorithm holds generally.

3.5 Complexity

The complexity of the iterative algorithm depends on the complexity of the algorithms used in the individual steps and the required number of iterations till it terminates. Since problems (P2i), (P3r), and (P2a) are separable, the allocation (refinement) step can be independently solved for each task (respectively user). Hence, the computation procedure can be highly parallelized. The convex optimization problem with a linear constraint in (P2i) is solved using the waterfilling algorithm from [10]. The algorithm execution for each task involves sorting the N initial contributors in $\mathcal{O}(N\log N)$ steps and iterating $\mathcal{O}(N)$ times resulting to complexity

Algorithm 3: Proposed algorithm for (P1).

```
Remove distance constraint from (P1).
Perform **allocation step** solving (P2i).
$t = 0$, $flag = 0$.
Categorize contributors into sets $N_v^0$ and $N_a^0$.
**if** $N_v^0 > 0$ **then**
    $t = 1$
    **while** $N_v^t > 0$ *and* $flag = 0$ **do**
        Perform **refinement step** solving (P3r).
        Compute excess budget $\Theta_j$.
        Perform **allocation step** solving (P2a).
        Update contributor sets $\mathcal{N}_v^t$ and $\mathcal{N}_a^t$.
        **if** $N_v^t = 0$ **then**
            Feasible solution reached.
            $flag = 1$
        **if** $N_v^t > 0$ *and* $N_a^t = 0$ **then**
            Solution infeasible.
            $flag = 1$
        $t = t + 1$
**else**
    Feasible solution reached.
```



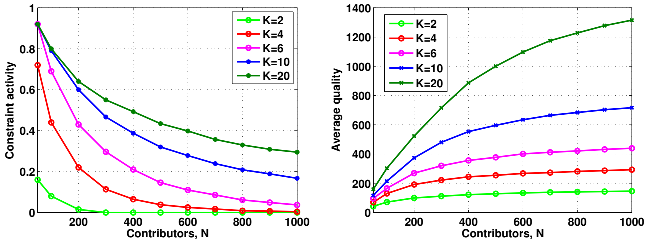
(a) (b)

Figure 1: Average running time in seconds (a) and (b) average quality for the proposed algorithm and solver.

of $\mathcal{O}(N \log N)$ if parallelization is supported. (P3r) is a linear optimization problem that can be solved in polynomial time $\mathcal{O}(K^p L)$, where $p > 0$ and L is the number of input bits. For example, the Karmarkar algorithm achieves $p = 3.5$. The algorithm needs to be carried out once for each contributor who violates the distance constraint. Finally, the convex optimization problem (P2a) is also solved by the waterfilling algorithm. In the allocation step, the algorithm is executed as many times as the tasks that have budget to reallocate and each execution demands $\mathcal{O}(N \log N)$ steps. In the worst case that one contributor violates the distance constraint at each iteration, the payment allocation and refinement steps are executed N times. Hence, the complexity is $\mathcal{O}(max\{NK^pL, N^2 \log N\})$.

4. RESULTS

In the simulations that follow, we consider a rectangular area (60x60) and randomly place users and tasks all over it. The quality indices and the payment sensitivity parameters are uniformly distributed in [0,1], while the physical distance thresholds are sampled from the interval [0,100]. Fifty simulation runs are carried out for each fixed combination of problem parameters.



(a) (b)

Figure 2: (a)Normalized number of contributors violating distance constraint (b) average quality achieved,varying contributors and tasks, for the proposed algorithm.

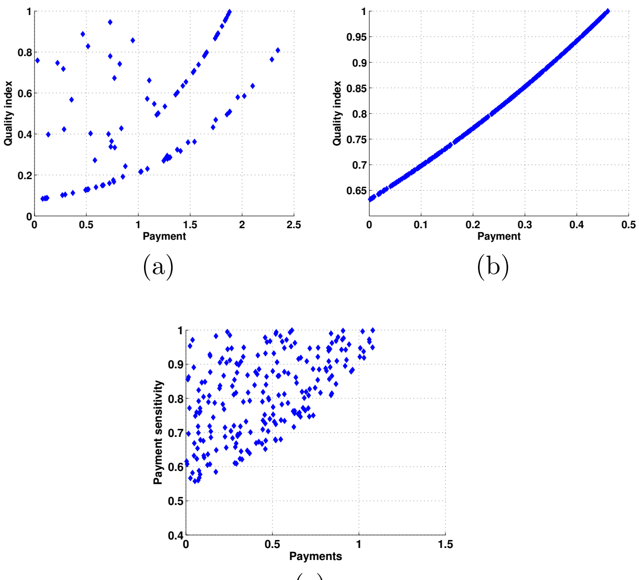
4.1 Algorithm efficiency

We compare our proposed algorithm with a typical convex optimization problem solver built in MATLAB (for our problem the solver implements the active-set algorithm). Fig. 1a shows that the average running time grows exponentially with the number of candidate contributors for the Matlab solver, whereas its increase is linear with our proposed algorithm. Increasing the number of tasks from $K = 3$ to $K = 4$ doubles the solver's running time for $N = 150$, while the respective increase is subtle even when we let the number of tasks scale to $K = 30$. Notably, this advantage in terms of running time comes at almost no expense in terms of the solution quality: our algorithm consistently yields expected quality values very close to the optimal ones, as seen in Fig. 1b.

4.2 Properties of our algorithm

Fig. 2a presents the portion of contributors violating the distance constraint during an iteration of our algorithm. For $K = 2$, the problem (P2) results in (P2i) with no users violating the constraint. The impact in the average expected quality for each added contributor becomes smaller as N grows in Fig. 2b (87% of the contributors are paid for $N = 100$ and 38% when $N = 1000$). Since the same amount of budget is distributed amongst a larger variety of contributors, payments eventually get smaller and their effect to individual willingness of participation eventually drops along with their expected traveling distance and expected quality. On the contrary, the growth of K augments the expected traveling distance per contributor (Fig. 2a), and works in favor of the aggregate expected quality (Fig. 2b).

To get more insights into how our algorithm allocates payments, we study the correlation between contributors' payments and their quality indices in Fig. 3a and



(a) (b)

(c)

Figure 3: Correlation of contributor payments to quality for (a) N = 100 and (b) N = 1000 and (c) correlation of contributor payments to sensitivity for N=1000.

3b. The reimbursement of contributors according to their quality indices in the allocation step induces a positive correlation between the two quantities. For N = 100, we find two types of correlation, which correspond to the number of iterations performed by the algorithm (Fig. 3a). The contributors deviating from the emerged groups, correspond to those, whose allocated payments were refined due to their distance constraints. In Fig. 3b, the algorithm terminates in one iteration and all contributors are categorized in a single group.

We also plot the user sensitivity to payments γ_i, which ultimately reflects their participation uncertainty. In Fig. 3c, contributors with payments over 0.5 exhibit $\gamma_i > 0.6$ and payments over 1 are offered to quite responsive contributors with $\gamma_i > 0.9$. The impact of payments is, thus, greater when offered to contributors with high sensitivity values, in agreement with intuition.

5. CONCLUSION

We proposed a framework for incentive allocation in mobile crowdsensing and considered contributor participation uncertainty as well as contributor inherent constraints in executing tasks, which are abstracted here as bounded traversed distance. Desiring to achieve highest expected quality of contributions, we formulated a convex optimization problem with respect to task and contributor limitations. We designed a two-step iterative algorithm that reaches effective near-optimal and low-complexity solutions. Our crowdsensing framework is flexible enough to be applied to various types of networks. Its specific properties can even adapt to the characteristics and limitations of a community network. This setting involves the scenario that nodes of the community network can crowdsource tasks to a pool of users that are connected to the network.

6. ACKNOWLEDGMENTS

The authors acknowledge the support of the European Commission through the Horizon 2020 project netCommons (Contract number 688768, duration 2016-2018).

7. REFERENCES

[1] L. Gao, F. Hou, and J. Huang. Providing long-term participation incentive in participatory sensing. In *Proc. IEEE INFOCOM*, pages 2803–2811, April 2015.

[2] S. He, D. H. Shin, J. Zhang, and J. Chen. Toward optimal allocation of location dependent tasks in crowdsensing. In *Proc. IEEE INFOCOM*, pages 745–753, April 2014.

[3] L. Jaimes, I. Vergara-Laurens, and M. Labrador. A location-based incentive mechanism for participatory sensing systems with budget constraints. In *IEEE PerCom*, pages 103–108, March 2012.

[4] H. Jin, L. Su, D. Chen, K. Nahrstedt, and J. Xu. Quality of information aware incentive mechanisms for mobile crowd sensing systems. In *16th Proc. ACM MobiHoc*, pages 167–176, June 2015.

[5] I. Koutsopoulos. Optimal incentive-driven design of participatory sensing systems. In *Proc. IEEE INFOCOM*, pages 1402–1410, April 2013.

[6] J.-S. Lee and B. Hoh. Dynamic Pricing Incentive for Participatory Sensing. *Pervasive Mobile Computing*, Vol.6:pp.693–708, December 2010.

[7] T. Luo, S. Kanhere, S. Das, and H.-P. Tan. Optimal prizes for all-pay contests in heterogeneous crowdsourcing. In *11th IEEE Conf. on MASS*, pages 136–144, October 2014.

[8] T. Luo, S. Kanhere, H.-P. Tan, F. Wu, and H. Wu. Crowdsourcing with tullock contests: A new perspective. In *IEEE INFOCOM*, pages 2515–2523, April 2015.

[9] T. Luo and C.-K. Tham. Fairness and social welfare in incentivizing participatory sensing. In *9th IEEE SECON*, pages 425–433, June 2012.

[10] D. P. Palomar and J. R. Fonollosa. Practical algorithms for a family of waterfilling solutions. *IEEE Trans. on Signal Processing*, Vol. 53:pp.686–695, 2005.

[11] D. Peng, F. Wu, and G. Chen. Pay as how well you do: A quality based incentive mechanism for crowdsensing. In *16th Proc. ACM MobiHoc*, pages 177–186, June 2015.

[12] D. Yang, G. Xue, X. Fang, and J. Tang. Crowdsourcing to Smartphones: Incentive Mechanism Design for Mobile Phone Sensing. In *18th Proc. ACM MobiCom*, pages 173–184, August 2012.