EGAIM: Enhanced Genetic Algorithm based Incentive Mechanism for Mobile Crowdsensing

Samad Saadatmand
samad.saadatmand@unsw.edu.au

Salil S. Kanhere
salil.kanhere@unsw.edu.au

School of Computer Science and Engineering, UNSW Sydney, Australia

Abstract

Mobile Crowdsensing (MCS) systems take advantage of the ubiquity and sensing power of smartphones in data gathering. Designing an incentive mechanism for motivating the individuals to participate in such systems is vital. Reverse auction is a popular incentive framework in which the users bid their expected returns for their contributions, and the mechanism then selects a number of them as the participants based on their value for the system. In this paper, we consider the goal of participant selection as maximising the total contribution within a budget constraint where the user contributions may be disparate and coverage overlap is possible. We propose a genetic algorithm approximation solution for this optimisation problem. We call the mechanism as Genetic Algorithm based Incentive Mechanism (GAIM). We also propose an enhanced version of this approach (EGAIM) in which an improved parent selection strategy is utilised to overcome two limitations of GAIM which arise in situations where the budget is limited. We compare EGAIM with GAIM and a greedy algorithm under two real-world scenarios, and show that using EGAIM can save up to 55% of budget for achieving at the same level of contribution.*

1. Introduction

Crowdsourcing [1] is the practice of engaging a crowd towards a common goal such as problem solving, idea generation or even improving efficiency. The Internet has been a strong enabler of crowdsourcing and may success stories abound, e.g. Wikipedia [2], Kickstarer [3], Amazon Mechanical Turk [4], etc. More recently, the proliferation of mobile devices that are equipped with a wide variety of embedded sensors including camera, microphone, GPS has enabled a new sensing paradigm known as Mobile Crowdsensing (MCS). MCS leverages people to act as sensors and collect large amount of sensory data from a widespread geographic region. For example, Earphone [5] is a noise monitoring application that converts the phone microphone into a sound level meter. Waze [6] collects real time traffic information from participants' phones as they drive around. Mapillary [7] collects geotagged street-level images from phone cameras.

The success of MCS hinges on sufficient user participation. While some individuals altruistically volunteer to MCS tasks, they are usually few and far between. The vast majority of users that participate in MCS will expect some incentives (monetary or virtual) in exchange for their contributions given the fact that they will be expending resources such as time, battery life, mobile bandwidth and privacy [8]. Thus, a well-formed incentive mechanism is essential for motivating individuals to join MCS campaigns, and more importantly to ensure that they continue to participate over a long period of time after the initial excitement has waned [9].

Reverse auction [10] is a popular incentive mechanism that has been shown to be well suited for MCS. In this setting, the users who are willing to participate in the task bid their expected reward. The mechanism will select a number of users based on their value, purchase their sensing data and pay them their bid price as a reward. The task publisher often has a limited budget and would like to maximise her return in the form of the sensed data received from the participants. RADP [11], is a popular reverse auction based incentive mechanism in which all users are assumed to have the same amount and quality of contributions. In order to achieve a specific amount of cumulative contribution, this scheme selects a predetermined number of users based on the ascending order of their bids. This mechanism however does not minimise the required budget in case users have different amount of contributions. This mechanism also does not maximise the total contribution in case coverage overlaps is possible, i.e., when a region is within the sensing range of multiple users, which is often the case in practical settings.

Thus, assuming the participants have different coverages and the coverage overlap is possible, the problem is selecting a subset of the participants in a reverse auction with the aim of maximising the total contribution within a budget constraint. Selecting the optimal subset of users (from a user population of U) would require checking all possible $2^{|U|}$ subsets of U. However, there is no polynomial time algorithm that can find this optimal solution.

In this paper, we formulate this problem as an optimization problem. We propose a GA as an approximation solution for the participant selection problem. In this algorithm, referred as Genetic Algorithm based Incentive Mechanism (GAIM), every potential answer (i.e., the subset of selected users) is encoded as a vector (i.e. chromosome in GA terminology) of 0 and 1s, representing the absence or presence of a user in the subset.

GAIM starts with a set of P randomly generated chromosomes and iterates through a number of generations in which the

MobiQuitous 2017, November 7–10, 2017, Melbourne, VIC, Australia

ACM ISBN 978-1-4503-5368-7/17/11…$15.00
https://doi.org/10.1145/3144457.3144504

chromosomes evolve through a sequence of bio-inspired operations, namely, parent selection, crossover and mutation. We found that the standard parent selection does not select valuable chromosomes as parents for low budgets. This issue significantly affects the answers found by this algorithm. In fact, we show in the simulations that GAIM is unable to find a suitable subset of users in these circumstances. As a modification, we propose an improved parent selection which offers suitable parents for all budgets. We refer to the algorithm which uses the improved parent selection as Enhanced GAIM (EGAIM).

The contributions of this paper can be summarised as follows:

- To the best of our knowledge, this is the first work that proposes the use of GA for solving the participation selection problem in the context of a reverse auction based incentive mechanism for MCS.
- We outline two major limitations with the parent selection strategy used in the standard GA strategy (referred to as GAIM) and show that they can significantly impact the ability of the algorithm to find a suitable solution.
- We propose an improved parent selection strategy referred to as Enhanced GAIM (EGAIM) to overcome these limitations.
- We conduct extensive experiments under two real-word scenarios and first show that EGAIM can achieve performance that is very close to the optimal solution. We also show EGAIM can achieve up to 55% and 45% savings in budget while achieving similar level of contribution when compared with a greedy algorithm, and GAIM.

In Section 2 we outline related work. The reverse auction framework is outlined in Section 3. In Section 4 the GAIM is proposed. Section 5 first outlines the limitation of GAIM, and then proposes the improved parent selection strategy. In Section 6 we present the experiments and results, and the paper finally is concluded in Section 7.

2. Related Work

Incentive design in MCS has been an active area of research in recent years. There has been ongoing research that has addressed various aspects of incentive design including: quality of information, trustworthiness, fairness, privacy, types of incentives, user selection and coverage. We first provide a short summary of related work that addresses the aforementioned issues.

The notion of Quality of Information (QoI) in MCS was first introduced by Liu et al in [12]. The authors also proposed a distributed QoI-aware energy-management scheme for improving the QoI of the total sensed data. Tham et al. [13] proposed a new metric called Quality of Contributed Service (QCS) for characterising the quality of data. The authors in [14] developed a novel quality-aware task assignment mechanism which is based on the multi-armed bandit problem. Their strategy assigns users the tasks which they are good at.

Restuccia et al. [15] proposed a trust-based mechanism, called FIDES, which eliminates unreliable data and thus achieves considerable savings in the total expended reward. Wu et al. proposed a reputation framework called EndorTrust [16] for predicting and assessing the trustworthiness of users using machine learning methods.

Luo and Tham proposed two incentive mechanisms called Incentive with Demand Fairness (IDF) [17] and Iterative Tank Filling (ITF) [18]. In these schemes, the users are considered as data contributors as well as service consumers at the same time. Using Jain's fairness index the authors show that the ADF is max-min fair. They also show that ITF maximise social welfare using Nash equilibrium.

Zhang et al. [19] highlighted potential privacy leakage issues with existing incentive design mechanisms. They proposed a participant coordination mechanism that allows the task publisher to achieve optimal QoI without tracking the location of users. Singla et al. [20] designed a budget feasible and truthful mechanism in which the users only submit their bids, but not their locations. A user will be asked to reveal her location only if she is selected.

Since the users normally move in and out of the sensing environment, real-time decision-making is required. The incentive mechanisms which make decisions instantaneously are referred to as online. Zhang et al. [21] proposed an online mechanism in which the tasks are assigned to the users only after they enter the target sensing region. Zhao et al. [22] proposed an online scheme for selecting a subset of participants before a deadline with the aim of maximising the value within a budget limit. Kang et al. [23] proposed a probabilistic online quality-aware model for the task assignment problem in which the goal is optimising the tasks' total quality through assigning appropriate tasks to the users.

Ueyama et al. [24] proposed a hybrid mechanism that uses both virtual (e.g., badges, leader board, etc.) and monetary incentives and showed that this achieves significant savings in the total rewards being paid to the participants. In a different work, Song et al. [25] proposed a light-weight negotiation scheme wherein a bid price is offered to the users based on their spatiotemporal properties. This approach approximates the distribution of data based on the user availabilities, and then offers the users bid prices within the budget.

In the rest of this section, we focus on related research in the realm of the participant selection problem in a reverse auction setting, which is closest to our work. In [26], the authors propose a simple solution which selects users randomly and pays them fixed and equal rewards. However, due to different expectations, a fixed reward may not satisfy all users. The authors in [11] proposed an alternative approach where the rewards are not fixed but rather defined dynamically according to user expectations. They proposed a reverse auction based dynamic pricing incentive mechanism, called RAPD, wherein users declare their expected rewards as bids and their mechanism selects the users who are willing to participate in the tasks declare their expected rewards, as bids, and the system selects a number of users based on their bids and pays them their expected rewards. However, in this greedy selection strategy, users with higher expected rewards are less likely to be selected and could thus drop out with passage of time. As a modification, the authors proposed RADP-VPC [27] in which a virtual credit is deducted from the bids of the users who were not selected in the previous auctions. This increases the chance of selection of these users in subsequent auctions and thus overcomes the dropout issue. RADP-VPC-RC [27] is an extension of RADP-VPC in which the highest bid of the winners in the last auction is revealed to users who drop out as a motivation for them to join the system and adjust their bids accordingly. It should be noted that all versions of RADP assume that all user contributions are equal. These schemes thus cannot be used in scenarios where user contributions are different. Our work focuses on such scenarios.

In [28], the authors propose an extensive of RADP-VPC-RC called GIA which incorporates the greedy set cover algorithm as an approximation solution to select a user set (or population) that maximizes the coverage of the target region while also satisfying the budget limit. It should be noted that the contribution of each user is assumed to be proportional to the region from which the user can collect sensing data. The same authors propose an extension of GIA called SPREAD [29] which combines GIA with weighted variance maximization algorithms. As these algorithms are approximation, they do not necessarily maximise the total contributions, specifically when there are coverage overlaps. In this paper, we also focus on finding a suitable set of participants such that the total contribution is maximised within the budget. We assume that both the contributions and rewards of each user can be different and that coverage overlap may be possible.

3. System Model

We consider a mobile crowdsensing system wherein, one or more task providers have created a series of T sensing tasks. Each task $t \in \{1, \ldots, T\}$ has an associated budget β^t which serves as the upper bound on the total rewards paid to the users who are selected to contribute their sensed data to the task. We assume that the system operates under the reverse auction framework for selecting and rewarding participants. Let U^t denote the set of users that are willing to participate in task t. In the (reverse) auction for task t, each user $u_i \in U^t$ bids her expected reward, denoted by $r(u_i)$. A participant selection algorithm chooses a subset $U_s^t \subseteq U^t$ of the bidders. The system buys the sensed data from these users and pays them their expected. The users are selected based on the value of their contributions as well as their expected rewards. We will outline the specific details of our proposed algorithm in Sections 4 and 5.

Our framework is generic and allows for different ways to associate value to the contributions. In this paper, we assume that the value of data is commensurate with the area of the physical region sensed by a user. As shown in Figure 1, we represent the target region as a 2D grid of $N \times M$ equally sized cells, in which each cell $a_{n,m}$ has an associated weight $w(a_{n,m})$ which indicates the importance of the data sensed from that cell for the task provider. The weights are defined by the provider at the time of task creation. Each user u_i has a sensing radius $s(u_i)$, which implies that the user can sense data from all cells within distance $s(u_i)$ from the current cell in which u_i is located. Each user u_i can covers at most $\left((s(u_i) \times 2) + 1\right)^2$ cells, including the cell she is in. In Figure 1, $s(u_1) = 2$ and $s(u_2) = 3$, and the grey cells show the covered cells by them. Each user includes her current location and sensing radius in her bid. The contribution $c(u_i)$ of user u_i is the total weight of the cells covered by u_i, which is represented as follows,

$$c(u_i) = \sum_{n=1}^{N} \sum_{m=1}^{M} w(a_{n,m}) * k_i(a_{n,m}), \qquad (1)$$

where $k_i(a_{n,m})$ is 1 if cell $a_{n,m}$ is in the coverage range of user u_i, otherwise is 0. It is possible that the coverage area of multiple users may overlap. For example, the dark shaded cells in Figure 1 are within the coverage area of both users u_1 and u_2.

The goal of the incentive mechanism is to select a subset of the bidders (U_s^t) such that the cumulative contribution of the selected users is maximized while also ensuring that the total reward paid out to them is not greater than the budget, β^t. We refer to the above as the *participant selection* problem. The following equation formalises this problem for task t in which $C(U_s^t)$ and $R(U_s^t)$ respectively denote the total contribution and the total expected reward of the selected users in U_s^t.

$$\begin{cases} \textit{Select } U_s^t \subseteq U^t \textit{ subject to} \\ \textit{Maximise } C(U_s^t) \textit{ while } R(U_s^t) \leq \beta^t \end{cases}. \qquad (2)$$

In the above, $R(U_s^t) \leq \beta^t$ represents the budget constraint. The total reward being paid to the selected users is the same as sum of their expected rewards, $R(U_s^t) = \sum_{u_i \in U_s^t} r(u_i)$. Due to the possibility of coverage overlap (as noted above), $C(U_s^t)$ may be potentially lower than the sum of the contibutions of each selected user, i.e.

$$C(U_s^t) \leq \sum_{u_i \in U_s^t} c(u_i). \qquad (3)$$

The optimal solution for the participant selection problem cannot be readily found using a greedy algorithm since there is the potential of coverage overlap among the users. Finding the optimal answer requires checking all possible subsets of U^t, i.e. $2^{|U^t|}$, which cannot be achieved by a polynomial time algorithm. In this paper, we propose an approximation approach that uses genetic algorithms as a solution. Table 1 shows a list of the frequently used notations in the rest of this paper.

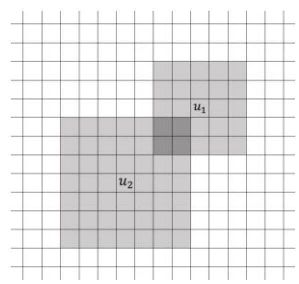

Figure 1. *Representation of the sensing area and coverage of individual users*

Notation	Meaning
β^t	Budget of task t
U^t	Set of users who bid in the auction related to task t
u_i	User i
$c(u_i)$	Contribution of user u_i
$r(u_i)$	Expected reward of user u_i
$s(u_i)$	Sensing radius of user u_i
U_s^t	The subset of selected users for task t; $U_s^t \subseteq U^t$
$C(U_s^t)$	Total contribution of all users in subset U_s^t
$R(U_s^t)$	Total rewards of all users in subset U_s^t
G	The number of generations
P	The population size in each generation
g_{Pr}, g_{of}, g_{Ch}	The parents, offspring and children populations
$\varphi^{p,g_{st}}$	Chromosome p at step st of generation g
$\varphi_i^{p,g_{st}}$	Gene i of chromosome p at step st of generation g
$C(\varphi^{p,g_{st}})$	Total contribution of chromosome p at step st of generation g
$R(\varphi^{p,g_{st}})$	Total reward of chromosome p at step st of generation g

$a_{n,m}$	Cell n, m
$w(a_{n,m})$	The weight of the cell $a_{n,m}$

Table 1. *Frequently used notations*

4. Genetic Algorithm based Incentive Mechanism (GAIM)

In this section, we propose a genetic algorithm (GA) as a heuristic for solving the participant selection problem which is the central to the incentive mechanism in crowdsensing as outlined in Section 3. We first provide a brief overview of GA in Section 4.1. Next, in Section 4.2, we outline the way in which the participation selection problem is mapped to a genetic representation. Finally, the three main steps of the GA are outlined in Sections 4.3-4.5. In this paper, we refer to the plain vanilla GA outlined in this section as Genetic Algorithm based Incentive Mechanism (GAIM).

4.1. Overview and basic definitions

Genetic algorithms [30] draw inspiration from the principles of natural selection and have been widely used to solve search and optimisation problems in various domains such as Bioinformatics [31], Game Theory [32] and Design Automation [33]. Each potential answer is represented as a set of parameters, referred to as *genes*, which are concatenated together to form a string of values (usually binary), known as a *chromosome* [35]. Genetic algorithms start with a population of candidate solutions (i.e. chromosomes) and rely on bio-inspired operators such as *selection*, *crossover* and *mutation* to iteratively evolve toward better solutions. Each iteration is referred to as a *generation*. In each generation, a portion of the existing population is selected to breed a new generation. The selection is dictated by a *fitness function*, which is typically the value of the optimisation problem that is being solved. The crossover operation takes two selected parent chromosomes and combines them to create a pair of offsprings. Finally, mutation is applied to each offspring chromosome to alter each gene with a small probability. The resulting set of chromosomes are referred to as the children of this generation. Figure 2 illustrates the operations involved in each generation. Let g_{Pr}, g_{of} and g_{Ch} represent the three core steps (i.e., selection, crossover and mutation) for a generation g. The output chromosomes of these steps are parents, offspring and children chromosomes, respectively. Let $\varphi_i^{p,g_{st}}$ represent gene i of chromosome p after step st of generation g. Thus, chromosome p after step st of generation g can be represented as,

$$\varphi^{p,g_{st}} = \bigcup_{i=1}^{n} \varphi_i^{p,g_{st}}, \tag{3}$$

where $p \in \{1, \dots, P\}$, $g \in \{1, \dots, G\}$, $st \in \{pr, of, ch\}$, and $i \in \{1, \dots, n\}$ assuming that each chromosome consists of n genes. Fig. 2 shows the relation between the three operations and the chromosomes generated within generation g.

4.2. Mapping

Before GA can be executed, a suitable representation (or coding) for the problem must be devised. In the context of the participant selection problem outlined in Section 3, the GA should select a subset of users U_s^t from the set U^t of all participating users. We represent a solution as a binary vector of length $|U^t|$. The i-th element of the vector corresponds to the presence or absence of the i-th user in U_s^t. A value of 1 implies presence while a 0 stands for absence. In GA terminology, this vector representation is referred to as a chromosome where each element is a gene. The following equation formalises this mapping.

$$\begin{gathered} encoding: \varphi_i^{p,g_{st}} = \begin{cases} 1, & if\ u_i \in U_s^t \\ 0, & if\ u_i \notin U_s^t \end{cases} \\ decoding: U_s^t = \left\{ \bigcup u_i \in U^t \right\} if \varphi_i^{p,g_{st}} = 1 \end{gathered} \tag{4}$$

Note that, not all of the $2^{|U^t|}$ possible chromosomes will satisfy the budget constraint outlined in Equation 2. This is because the expected reward for the corresponding subset of selected users may exceed the task budget, β^t. We thus introduce the notion of *validity* for a chromosome. A chromosome is considered *valid* if the corresponding subset of selected users meet the budget constraint and *invalid* otherwise.

The fitness score (referred to as *value*) of a chromosome is defined as the total contribution of the selected users in the corresponding subset U_s^t if the chromsome is valid. For invalid chromosomes, the value is assumed to be zero as the budget constraint is violated. The following equation formalises the value of chromosome $\varphi^{p,g_{st}}$, shown by $\delta(\varphi^{p,g_{st}})$ as

$$\delta(\varphi^{p,g_{st}}) = \begin{cases} C(U_s^t), & if\ R(U_s^t) \le \beta^t \\ 0, & Otherwise \end{cases}. \tag{5}$$

For clarity, we represent each operation as $f(x) \to (y)$ wherein f is a symbol for the operation, and x and y are the input and output, respectively.

4.3. Parent Selection

For the very first generation, a randomly generated set of P chromosomes are selected as parents. For all subsequent generations, g, the parents are selected from the children of the previous generation, $g-1$. We employ the widely used *roulette wheel* selection strategy [30], wherein, the probability with which a child from generation $g-1$ is chosen as a parent for generation

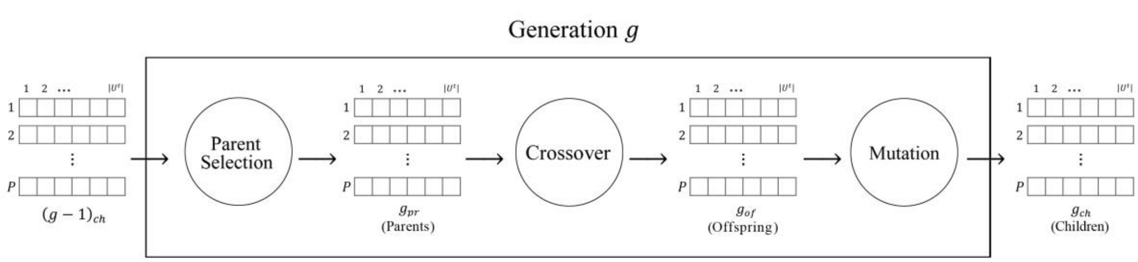

Figure 2. *The various steps involved in a generation g*

g is based on the ratio of its value to the sum of the values of all children of generation $g-1$. More precisely, each child chromosome $\varphi^{p,(g-1)_{ch}}$ is selected with probability $\rho\left(\varphi^{p,(g-1)_{ch}}\right)$ which is calculated as

$$\rho\left(\varphi^{p,(g-1)_{ch}}\right)=\frac{\delta\left(\varphi^{p,(g-1)_{ch}}\right)}{\sum_{i=1}^{P}\delta\left(\varphi^{i,(g-1)_{ch}}\right)}. \qquad (6)$$

Given $c\rho\left(\varphi^{p,(g-1)_{ch}}\right)=\sum_{i=1}^{p}\rho\left(\varphi^{i,(g-1)_{ch}}\right)$ denotes the cumulative probability of the children chromosome p. Algorithm 1 outlines the roulette wheel selection strategy. In simple words, this strategy ensures that from the pool of children of generation $g-1$, those with higher values are more likely to be selected as parents for generation g. However, it is still possible (albeit with a lower chance) for low valued children to be selected.

Algorithm 1: Roulette Wheel Parent selection

Input: set of children chromosomes $\left\{\bigcup_{i=1}^{P}\varphi^{i,(g-1)_{ch}}\right\}$
Output: a parent chromosome $\varphi^{p,g_{pr}}$

```
Begin
    r ← random[0, 1]
    j ← P
    while j > 1 do
        if cρ(φ^{j,(g-1)_ch}) ≤ r then
            φ^{i,g_pr} ← φ^{j,(g-1)_ch}
            return φ^{p,g_pr}
        end if
        j ← j − 1
    end while
End
```

Due to the randomness associated with this process, a particular child may never be selected as a parent while another child may be selected multiple times. The parent selection operation ζ^{g} for generation g is represented as follows,

$$\zeta^{g}\left(\left\{\bigcup_{p=1}^{P}\varphi^{p,(g-1)_{ch}}\right\}\right)\rightarrow\left(\left\{\bigcup_{p=1}^{P}\varphi^{p,g_{pr}}\right\}\right) \qquad (7)$$

4.4. Crossover

After selection, the parents are grouped into pairs, $\langle\varphi^{2k-1,g_{pr}},\varphi^{2k,g_{pr}}\rangle$ where $k=\{1,\ldots,\frac{P}{2}\}$. In the crossover operation, two offsprings are born by combining each pair of the parents. We employ the standard *single point* crossover technique [30], in which a randomly selected crossover point, by θ, $1\leq\theta<|U^{t}|$, is chosen. The two parent chromosomes are cut at the crossover point, producing two head and two tail segments. The tail segments are swapped to produce two offsprings. The two offsprings thus inherit some genes from each of the parent chromosomes. The crossover operation for pair $\langle\varphi^{2k-1,g_{pr}},\varphi^{2k,g_{pr}}\rangle$ at the crossover point θ in generation g is represented as

$$\mathbb{C}_{\theta}^{g}\left(\langle\varphi^{2k-1,g_{pr}},\varphi^{2k,g_{pr}}\rangle\right)\rightarrow\left(\langle\varphi^{2k-1,g_{of}},\varphi^{2k,g_{of}}\rangle\right) \qquad (8)$$

in which $\varphi^{2k-1,g_{of}}$ and $\varphi^{2k,g_{of}}$ are the offsprings and the values of their genes are determined as follows,

$$\varphi_{i}^{2k-1,g_{of}}=\begin{cases}\varphi_{i}^{2k-1,g_{pr}}, & if\ i\in\{1,\ldots,\theta\}\\ \varphi_{i}^{2k,g_{pr}}, & if\ i\in\{\theta+1,\ldots,|U^{t}|\}\end{cases},$$
$$\varphi_{i}^{2k,g_{of}}=\begin{cases}\varphi_{i}^{2k,g_{pr}}, & if\ i\in\{1,\ldots,\theta\}\\ \varphi_{i}^{2k-1,g_{pr}}, & if\ i\in\{\theta+1,\ldots,|U^{t}|\}\end{cases}. \qquad (9)$$

The above operation is applied to every $P/2$ pairs of the parents to create P offsprings.

4.5. Mutation

The goal of the mutation operation is to increase diversity amongst the offspring chromosomes. Each offspring can be selected for manipulation with a low probability (0.1 in our experiments in Section 6. If an offspring is chosen, then one randomly selected gene is chosen and its value is flipped (i.e., a 1 is changed to 0 and vice versa). Let ϑ denote the index of the selected gene, the signature of the mutation operation on chromosome p is represented as,

$$\mathcal{M}_{\vartheta}^{g}(\varphi^{p,g_{of}})\rightarrow(\varphi^{p,g_{ch}}), \qquad (10)$$

where $\varphi^{p,g_{ch}}$ is the mutated child and its genes are as

$$\varphi_{i}^{p,g_{ch}}=\begin{cases}\neg(\varphi_{i}^{p,g_{of}}), & if\ i=\theta\\ \varphi_{i}^{p,g_{of}}, & otherwise\end{cases} \qquad (11)$$

in which $\neg(\varphi_{i}^{p,g_{of}})$ is 0 if $\varphi_{i}^{p,g_{of}}=1$, and is 1 if $\varphi_{i}^{p,g_{of}}=0$. Since the mutation probability is low, typically only a handful of offsprings are modified while the rest remain unaltered. The set of mutated and unchanged offsprings together are the children of the current generation.

5. Enhanced GAIM (EGAIM)

In this section, we highlight two drawbacks that exist with the parent selection strategy outlined in Section 4.3 which ultimately impacts the ability of GAIM to find a solution that is close to optimal (Sections 5.1 and 5.2). In Section 5.3, we present an improved parent selection strategy that overcome these issues. The crossover and mutation operations remain the same as in Sections 4.4 and 4.5. We refer to this variant of the GA strategy as Enhanced GAIM (EGAIM).

5.1. First Generation

Recall that, in the parent selection strategy used in GAIM, the parent chromosomes for the first generation are generated randomly. Since each gene of a randomly generated chromosome can be either 0 or 1 with equal probability, on average $|U^{t}|/2$ genes are equal to 1. Let $\mu_{r}(U^{t})$ denote the average of the expected rewards of the users in U^{t}. Thus, the total expected reward of a randomly generated chromosome is $\frac{|U^{t}|\times\mu_{r}(U^{t})}{2}$. If this term is greater than the budget for the task, β^{t}, then this randomly generated chromosome will be invalid. To demonstrate how likely the aforementioned issue is, we collated relevant data from the simulation experiments for GAIM, the details of which are outlined in Section 6.1. In the simulations 50 parent chromosomes are used in each generation. We consider the scenarios consisting of 250 users. GAIM is executed for different budgets ranging from

100 to 9000. Table 2 collates the total fraction of invalid chromosomes generated in the first generation. Observe that for a budget less than or equal to 3600, all randomly generated chromosomes are invalid. Even with a budget of 4200, about 64% of the chromosomes are invalid.

As will be shown in Section 6.2, this significantly impacts the ability of GAIM to find a suitable subset of users. With higher budget, the likelihood of the aforementioned issue diminishes. Instead of randomly generating chromosomes in the first generation, EGAIM uses the strategy outlined Section 5.3 for generating P parent chromosomes.

Budget	The likelihood of being invalid
≤3600	100%
3800	96.5%
4000	85.6%%
4200	63.8%
4400	39%
4600	15.9%
4800	4.7%
5000	1.1%
≥5200	0%

Table 2: *Invalidity percentage of random chromosomes for different budgets*

5.2. All Subsequent Generations

For all subsequent generations in GAIM, the parents for a generation g are selected in a probabilistic manner from the children of the previous generation, $g-1$, as per the roulette selection strategy outlined in Section 4.3. Recall that, the probability with which a chromosome is selected is based on its value. Since, invalid chromosomes have zero value, they are never selected as a parent. This implies that if any of the children of a generation are invalid, then one or more the valid children are picked more than once as parents in the next generation. Furthermore, the greater the number of invalid children, the higher the likelihood of multiple parents being similar in the next generation. It should be noted that even if all parents are valid, the roulette wheel strategy can still select the same children (particularly those with higher values) as parents more than once. All in all, this reduces the diversity of the parent population.

To demonstrate the likelihood of the aforementioned issue, we collated relevant data from the simulation experiments for GAIM, the details of which are outlined in Section 6.1. In these experiments, GAIM is run for 200 generations with 50 chromosomes in each generation for different budgets (ranging from 100 to 9000) with 250 users. Our analysis revealed that on average only 64% of the parents (i.e., 32 out of 50) selected in each generation are unique, which suggests that over a third of the parents (36%) are duplicates.

Recall that in the crossover operation, two offsprings are created by combining a pair of parents. If it so happens that the two parents are similar, then their offsprings will be identical to their parents and thus each other. Thus, the presence of similar parents can reduce the overall diversity in the chromosome pool which is known to adversely impact the speed of evolution of GA [34].

In EGAIM, if X out of P children are invalid, then the valid $P-X$ children are automatically selected as parents. The remaining X chromosomes are generated using the strategy outlined in Section 5.3

5.3. An improved approach for producing valid and valuable chromosomes

We propose an improved strategy for creating chromosomes which are valid and also possessing high value.

Each time a chromosome needs to be generated, our strategy starts with a zero chromosome, i.e., a chromosome where all genes are zero, which is denoted by $\varphi^{p,g_{pr}}$. This chromosome is iteratively modified until the total expected rewards of resultant chromosome can no longer meet the budget constraint in Equation 2. Each iteration consists of the following steps: (1) A random index j from 1 to $|U^t|$ (which has not been chosen before) is selected. (2) The gene at that index, i.e. $\varphi_j^{p,g_{pr}}$, is changed to 1, provided the expected reward after the modification is within the budget. If not, then the gene is left unaltered and the process is terminated. Thus, we can guarantee that the resulting chromosome will always be valid. The detailed algorithm is illustrated below.

Algorithm 2: Improved Chromosome Generation

Input: zero chromosome $\varphi^{p,g_{pr}}$
Output: chromosome $\varphi^{p,g_{pr}}$

Begin
  $IndexPool \leftarrow \{1, \dots, |U^t|\}$
  while $True$ **do**
    $j \leftarrow PickFrom(IndexPool)$
    $IndexPool \leftarrow IndexPool - \{j\}$
    if $R(\varphi^{p,g_{pr}}) + r(u_j) \le \beta^t$ **then**
      $\varphi_j^{p,g_{pr}} \leftarrow 1$
    else
      Return $\varphi^{p,g_{pr}}$
    end if
  end while
End

The above algorithm is used to replace the random generation of chromosomes in the first generation of GAIM. Moreover, as discussed in Section 5.2, the above algorithm is used to generate the remaining X chromosomes to replace the corresponding invalid children from the previous generation.

Since there is still some randomness associated with the above process, it may be possible that when it is used to generate multiple chromosomes, some of them are likely to be identical. However, we argue that the likelihood of this is very low. As in Section 5.2, we analysed the relevant data from the execution of EGAIM in our simulations (Section 6). We could not find a single instance where this algorithm generated two identical chromosomes. It should be noted that the proposed chromosome generation strategy not only produces chromosomes that are valid but possessing high value. This will be evident in the simulation results explained in Section 6.2.

When used for producing the chromosomes in the first generation, our strategy has similar complexity as incurred while generating random chromosomes, which is $O(|U^t|)$. Admittedly, the proposed strategy when used in subsequent generations has higher complexity, $O(|U^t|)$, as compared to $O(P)$ for the roulette wheel strategy. However, the parent selection operation inherently requires calculating the values (and validity) of all the children

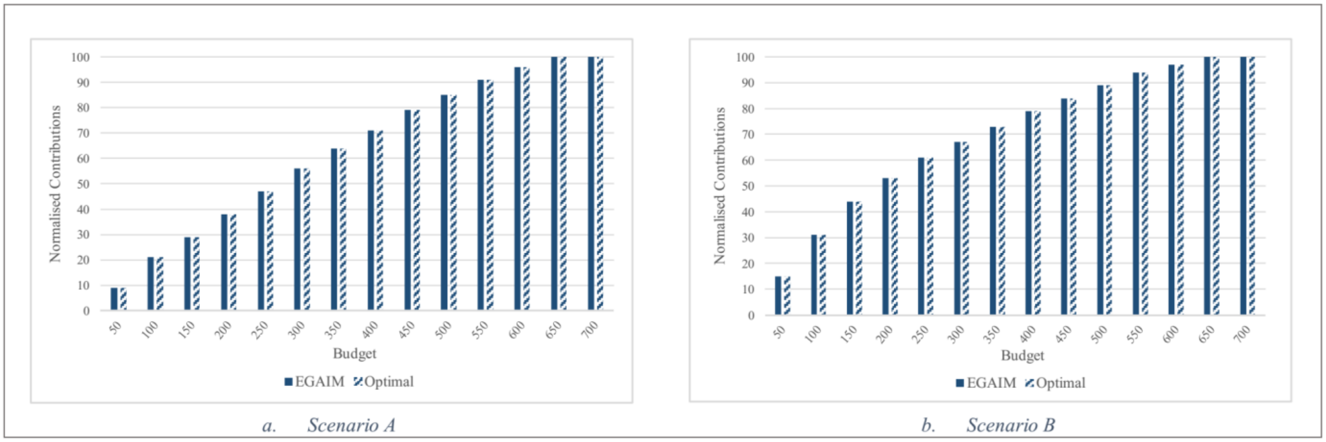

Figure 3. *Comparison of EGAIM with Optimal for 20 users*

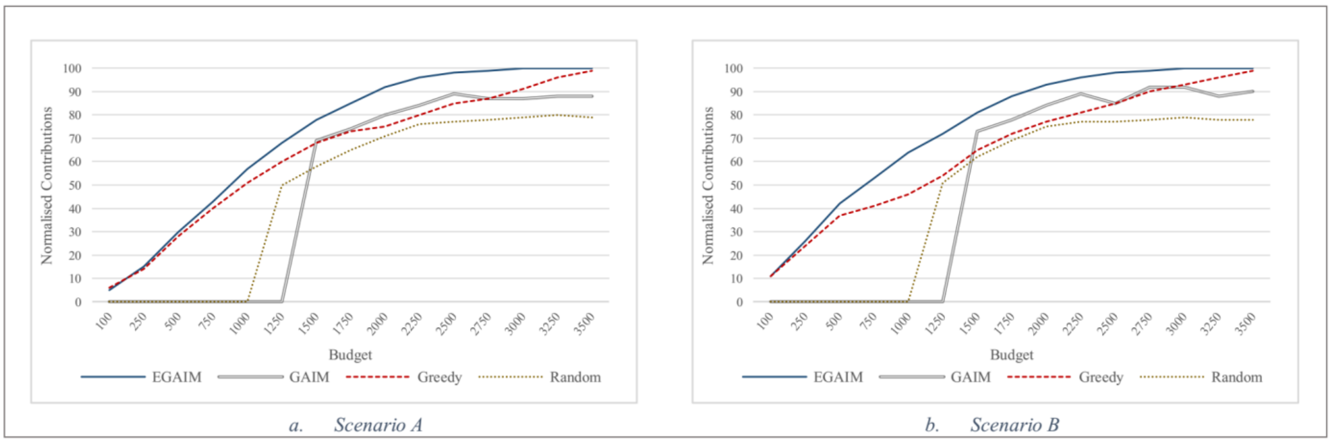

Figure 4. *Comparison of EGAIM, GAIM, Greedy and Random for 100 users*

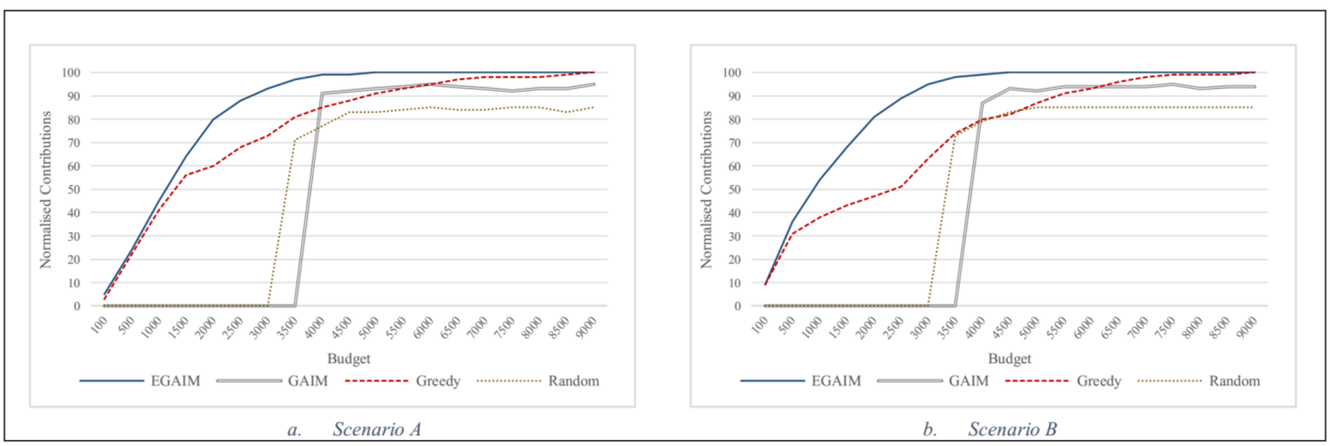

Figure 5. *Comparison of EGAIM, GAIM, Greedy and Random for 250 users*

chromosomes in the previous generation, $g-1$, by the fitness function as Equation 5 with complexity $O(|U^t|)$. Since the P is constant, the complexity of the parent selection in GAIM is $O(P \times |U^t|) = O(|U^t|)$. Thus, complexity of the improved parent selection with $O(2 \times |U^t|) = O(|U^t|)$ is the same as that of the parent selection strategy used in GAIM. Since the time complexity of the crossover and mutation operations are also $O(|U^t|)$, the total time complexity of the GAIM and EGAIM are $O(|U^t|)$.

6. Experiments and Results

In this section, we evaluate the effectiveness of our proposed GA-influenced participant selection strategy under two real-world scenarios. Section 6.1 outlines the two scenarios and experimental settings. The results are presented in Section 6.2. In the first instance, we compare the performance of EGAIM to the optimal solution. Next, we compare the performance of EGAIM with three other strategies, namely, a greedy algorithm, a random algorithm and the GAIM strategy outlined in Section 4.

6.1. Experiment Settings

The following settings are the common to all the experiments:

We consider the sensed area to be a square grid of 50×50 cells. We assume that the user positions are randomly selected in the 50×50 grid. The sensing radius $s(u_i)$ of each user u_i is randomly selected to be either 2 or 3 cells. Each user can sense data from at most $\left((s(u_i) \times 2) + 1\right)^2$ cells (as discussed in Section 3). The region sensed by user u_i is denoted by $Cov(u_i)$. The expected reward $r(u_i)$ of each user u_i is commensurate with her coverage, and is assumed to be randomly distributed between $0.8 \times Cov(u_i)$ and $1.2 \times Cov(u_i)$. The contribution of a user u_i is equal to the sum of the weight of the cells that are contained within $Cov(u_i)$ and is defined in Equation (1) (see Section 3). The total contribution of a subset U of users is equal to the some of the weights of the cells that are within the coverage range of all users in U, and is defined as

$$C(U) = \sum_{n=1}^{N} \sum_{m=1}^{M} w(a_{n,m}) * K_U(a_{n,m}), \qquad (13)$$

where $K_U(a_{n,m})$ is 1 if cell $a_{n,m}$ is covered by at least one user in U, and is 0 otherwise.

We assume that the two GA strategies are run for 200 generations (G) and that the user population (P) is 50. Each experiment is run for 100 times and average results are presented. We observed that the results for each individual run do not deviate significantly from the average. Thus we do not show the standard deviation (or variance) in our results. The position of each user, their expected reward and contribution are the same for each run of all algorithms under consideration. Different values are used in each subsequent run.

Recall that the output of each algorithm will be a subset of users U_s^t that are selected to contribute data for task t. The evaluation metric that best measures the quality of the selected subset U_s^t is the total contribution of the chosen subset $C(U_s^t)$ normalized by the maximum cumulative contribution of all users, i.e. $C(U^t)$ assuming an unconstrained budget, expressed as percentage, i.e. $C(U_s^t)/C(U^t) \times 100$. The evaluation metric used is the average of the normalised contributions.

We consider the following two scenarios: (i) *Scenario A*, wherein all cells have equal weight (of 1), which implies that data from all cells in the grid are equally important. (ii) *Scenario B*, wherein cells located in a central 20×20 cell region have higher weights (of 3) as compared to the other cells (which have a weight of 1). This reflects an application scenario where data from a certain region of interest (e.g., the city centre) may have higher importance. We simulate both scenarios with 100 and 250 users ($|U^t|$).

In our experiments, we compare the performance of the basic GA strategy (Section 4) and the enhanced GA strategy incorporating our improved parent selection algorithm (Section) with the following algorithms: (i) Greedy: In this strategy, users are ordered in descending order of the ratio of their contribution to reward, i.e., $c(u_i)/r(u_i)$. Starting with the highest ratio, users are progressively added to the set U_s^t if the addition of a user does not violate the budget constraint outlined in Section 3. (ii) Random: In this strategy, a number of random subsets (i.e., chromosomes) are generated from the possible $2^{|U^t|}$ subsets. The subset with the highest cumulative contribution while also meeting the budget constraint is selected as the solution. For a fair comparison with the GA variants, we generate $P \times G$ random subsets.

6.2. Results

Comparison with Optimal Strategy

In the first instance, we compare EGAIM with the optimal solution. To determine the optimal solution, we generate all possible subsets (i.e. $2^{|U^t|}$) and select the subset that achieves the highest contribution while meeting the budget constraint. As outlined in Section 3, this is an NP-Complete problem and thus we can only test this for a small user population. Thus, for this comparison we only consider a population of 20 users. We also use a reduced budget (the budget is varied from 50 to 700) as compared to the rest of simulations. The rest of the experimental settings are as outlined in Section 6.1. It is evident from Figure 3 the results for EGAIM are very close (and often similar) to the optimal strategy regardless of the budget for both scenarios. When comparing the two scenarios, the same budget achieves a greater normalized contribution in Scenario B compared to Scenario A. Recall that in Scenario B, the central region (20×20) has higher weight associated with it as compared to the other cells. Both schemes (i.e. EGAIM and optimal) select users that tend to include the cells with higher weights within their coverage range, which in turn results in increased contribution with the same budget (as compared to Scenario A).

Comparing EGAIM with Greedy, Random AND GAIM

Figure 4 and 5 provides a comparison of the four strategies under consideration for 100 users and 250 users, respectively. It is evident from both figures that EGAIM consistently outperforms the other three strategies. For example, consider Scenario B with 250 users (i.e. Figure 5b). EGAIM can achieve 90% of the normalised contribution with a budget of around 2500 units. However, to achieve the same level of contribution, Greedy and GAIM require 5500 and 4500 units of budget, respectively. Notice that the random strategy does not achieve this level of contribution even when the budget_is 9000 units (for which achieves 85%). This implies that a task creator could potentially save 55% and 45% of her budget for achieving the same level of contribution with EGAIM as compared to Greedy and Random, respectively.

For low budgets, both Random and GAIM are unable to find a suitable solution. For example, in Scenario A with 100 users (i.e., Figure 4.a), consider the random strategy with a budget ≤ 1000 and GAIM with a budget ≤ 1250. This can be attributed to the issue outlined in Section 5.1. Recall that, in GAIM, the parent chromosomes in the first generation are randomly generated. When the budget is low, the likelihood that the expected reward corresponding to each randomly generated chromosome is greater than the budget is very high, thus rendering all chromosomes to be invalid. The random strategy suffers from this same issue.

Amongst the four strategies, Random tends to perform the worst under all circumstances. Since the subsets (i.e. chromosomes) are generated randomly from a large sample size (of $2^{|U^t|}$) the likelihood of selecting subsets that achieve good cumulative contributions while also meeting the budget constraint are low. Increasing the budget beyond a certain point (e.g., in Figure 5.a, increasing the budget beyond 5000) does not achieve a better result due to the aforementioned issue.

The greedy strategy achieves a similar result as EGAIM at the two extremes, i.e. lowest and highest budget. However, in all other instances EGAIM outperforms the greedy strategy. Observe that the greedy strategy tends to perform worse compared to EGAIM in Scenario B than Scenario A. Recall that, the greedy strategy prioritizes users based on the ratio of their contribution to reward. Also in Scenario B, the central 20×20 region has higher weights compared to the other cells. Thus the greedy strategy tends to pick users that are highly concentrated in this central region. However, these users are likely to have significant coverage overlap, which results in reduced normalised contribution as compared with EGAIM which does not suffer from this drawback. Comparing Figures 4 and 5, it is also evident that when the participant pool is large (i.e. 250 vs 100) greedy performs worse than EGAIM. This can also be attributed to a similar reason, i.e. the fact that greedy strategy completely neglects coverage overlap. In contrast, EGAIM seeks to find a subset of users that will achieve the highest cumulative contribution while also meeting the budget constraints. So this algorithm tends to find users who can cumulatively cover greater portions of the target region and thus avoids selecting users with coverage overlap.

Apart from the issue outlined earlier that arises for lower budgets, GAIM is for the most part closest to EGAIM in performance. The lower performance of GAIM can be attributed to the reduced diversity in the chromosome population as was outlined in Section 5.2.

We implemented all the algorithms in C#. For both user populations (100 and 250), GAIM and EGAIM require only a few seconds to execute on a standard laptop (Intel Core i7, 16 GB RAM).

7. Conclusion

In this paper, we proposed GAIM as an approximation solution for the participant selection problem in a reverse auction based incentive mechanism. We found that there are two major issues with the standard parent selection strategy used in the GAIM for some low budgets. Then, we proposed EGAIM as an enhanced version of GAIM in which the parent selection is improved. We compared the EGAIM with a greedy algorithm, a random algorithm and the GAIM by considering two real-world scenarios and different budgets, and we showed that using EGAIM can save up to 50% of the budget for achieving the same cumulative contribution.

References

[1] S. S. Kanhere, "Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces," 12th IEEE International Conference on Mobile Data Management (MDM), Vol. 2, Pages 3-6, 2011

[2] Wikipedia, [Online]. Available: https://en.wikipedia.org

[3] KickStarter, [Online]. Available: www.kickstarter.com

[4] Amazon Mechanical Turk, [Online]. Available: www.mturk.com

[5] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-Phone: An End-to-End Participatory Urban Noise Mapping System," in Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks - IPSN 10, 2010

[6] Waze, [Online]. Available: www.wase.com

[7] Mapillary, [Online]. Available: www.mapillary.com

[8] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, and X. Mao, "Incentives for Mobile Crowd Sensing: A Survey," IEEE Communication Surveys & Tutorials, Vol. 18, No, 1, 2016

[9] L. G. Jaimes, I. J. Vergara-Laurens, and A. Raij, "A Survey of Incentive Techniques for Mobile Crowd Sensing," IEEE INTERNET OF THINGS JOURNAL, VOL. 2, NO. 5, 2015

[10] H. Gao, C. H. Liu, W. Wang, J. Zhao, Z. Song, X. Su, J. Crowcroft, and K. K. Leung, "A Survey of Incentive Mechanisms for Participatory Sensing," IEEE COMMUNICATION SURVEYS & TUTORIALS, VOL. 17, NO. 2, 2015

[11] J.-S. Lee and B. Hoh, "Dynamic pricing incentive for participatory sensing," Pervasive and Mobile Computing, vol. 6, 2010

[12] C. H. Liu, P. Hui, J. W. Branch, C. Bisdikian, and B. Yang, "Efficient network management for context-aware participatory sensing," in IEEE SECON'11, pp. 116–124, 2011

[13] C-K. Tham and T. Luo, "Quality of contributed service and market equilibrium for participatory sensing," in IEEE DCOSS'13, pp. 133–140, 2013

[14] N. An, R. Wang, Z. Luan, D. Qian, J. Cai, and H Zhang, "Adaptive Assignment for Quality-Aware Mobile Sensing Network with Strategic Users", IEEE 17th International Conference on High Performance Computing and Communications (HPCC), 2015

[15] F. Restuccia and S. Das, "Fides: A trust-based framework for secure user incentivization in participatory sensing," A World of Wireless, Mobile and Multimedia Networks (WoWMoM), IEEE 15th International Symposium on, 2014

[16] C. Wu, T. Luo, F. Wu, and G. Chen, "EndorTrust: An Endorsement-based Reputation System for Trustworthy and Heterogeneous Crowdsourcing," Global Communications Conference (GLOBECOM), IEEE 2015

[17] C. K. Tham and T. Luo, "Fairness and social welfare in service allocation schemes for participatory sensing," Computer Networks 73 (2014) 58–71, 2014

[18] T. Luo and C. K. Tham, "Fairness and Social Welfare in Incentivizing Participatory Sensing," 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012

[19] B. Zhang, C. Harold Liu, J. Lu, Z. Song, Z. Ren, J. Ma, and W. Wang, "Privacy-preserving QoI-aware participant coordination for mobile crowdsourcing", Computer Networks 101, 29–41, 2016

[20] A. Singla and A. Krause, "Incentives for Privacy Tradeoff in Community Sensing", Proceedings of the First AAAI Conference on Human Computation and Crowdsourcing, 2013

[21] X. Zhang, Z. Yang, Z. Zhou, H. Cai, L. Chen, and X. Li, "Free market of crowdsourcing: Incentive mechanism design for mobile sensing," IEEE Transactions on Parallel and Distributed Systems, 2014.

[22] D. Zhao, X.-Y. Li, and H. Ma, "How to Crowdsource Tasks Truthfully without Sacrificing Utility: Online Incentive Mechanisms with Budget Constraint", IEEE INFOCOM 2014 - IEEE Conference on Computer Communications

[23] Y. Kang, X. Miao, K. Liu, L. Chen, and Y. Liu, "Quality-Aware Online Task Assignment in Mobile Crowdsourcing", 2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems

[24] Y. Ueyama, M. Tamai, Y. Arakawa, and K. Yasumoto, "Gamification-based incentive mechanism for participatory sensing," in Pervasive Computing and Communications Workshops PERCOM Workshops), 2014 IEEE International Conference on, March 2014.

[25] Z. Song, E. Ngai, J. Ma, and W. Wang, "A novel incentive negotiation mechanism for participatory sensing under budget constraints," in Quality of Service (IWQoS), 2014 IEEE 22nd International Symposium of, May 2014.

[26] S. Reddy, D. Estrin, M. Hansen, and M. Srivastava, "Examining micro-payments for participatory sensing data collections," in

Proceedings of the 12th ACM International Conference on Ubiquitous Computing. ACM, 2010

[27] J-S. Lee and B. Hoh, “Sell your experiences: a market mechanism based incentive for participatory sensing,” Pervasive Mob. Comput., 2010

[28] L. G. Jaimes, I. Vergara-Laurens, and M. A. Labrador, “A Location-Based Incentive Mechanism for Participatory Sensing Systems with Budget Constraints”, IEEE International Conference on Pervasive Computing and Communications, Lugano, 2012

[29] L. Jaimes, I. Vergara-Laurens, and A. Chakeri, “Spread, a crowdsensing incentive mechanism to acquire better representative samples,” The First International Workshop on Crowdsensing Methods, Techniques, and Applications, 2014

[30] M. Mitchell, “An Introduction to Genetic Algorithms. Cambridge,” MA: MIT Press, 1996

[31] C. Gondro and B.P. Kinghorn, “A simple genetic algorithm for multiple sequence alignment,” Genetics and Molecular Research. 6 (4): 964–982. PMID 18058716, 2007

[32] J. Periaux, H.Q. Chen, B. Mantel, M. Sefrioui, and H.T. Sui, “Combining game theory and genetic algorithms with application to DDM-nozzle optimization problems”, Finite Elements in Analysis and Design 37, 417-429, 2001

[33] Y. Li, K. H. Ang, G. C-Y. Chong, W. Feng, and K. C. Tan, “CAutoCSD-Evolutionary search and optimisation enabled computer automated control system design”. International Journal of Automation and Computing, 2004

[34] D. Gupta and S. Ghafir, “An Overview of methods maintaining Diversity in Genetic Algorithms,” International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, Volume 2, Issue 5, 2012

[35] D. Whitley, “A genetic algorithm tutorial,” Statistics and computing, Springer, 1994