

RESEARCH ARTICLE

CAPR: context-aware participant recruitment mechanism in mobile crowdsourcing

Hongli Zhang¹, Zhikai Xu¹, Xiaojiang Du^{2*}, Zhigang Zhou¹ and Jiantao Shi¹¹ School of Computer Science Engineering, Harbin Institute of Technology, Harbin, 150001, China² Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, U.S.A.

ABSTRACT

With the advances of sensing, wireless communication, and mobile computing, mobile crowdsourcing has become a new paradigm for data collection and retrieval that has attracted considerable attention. This paper addresses the fundamental research issue in mobile crowdsourcing: Which participants should be selected as winners in each time slot with the aim of maximizing the total utility of the service provider in the long term? First, a double-sided combinatorial auction model is introduced to describe the relationships between the mobile users and requesters from the perspective of supply and demand at a given time. Then, the coupling between the utility values of the system in different time slots is investigated. Based on the aforementioned analyses, this paper proposes a context-aware participant recruitment mechanism, in which the mobile crowdsourcing system dynamically adjusts the participant recruitment mechanism depending on the ratio between the numbers of mobile users and requesters. Context-aware participant recruitment consists of two main components: (1) a heuristic algorithm based on the greedy strategy to determine the winning participants and (2) a critical payment scheme, which guarantees the rationality of the proposed mechanism. Finally, extensive simulations demonstrate that the proposed mechanism achieves high system utility in the long term. Copyright © 2016 John Wiley & Sons, Ltd.

KEYWORDS

mobile crowdsourcing; incentive mechanism; auction model; mobile computing

*Correspondence

Xiaojiang Du, Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, U.S.A.

E-mail: dxj@ieee.org

Received 6 August 2015; Revised 3 November 2015; Accepted 15 February 2016

1. INTRODUCTION AND RELATED WORK

The ubiquity of sensor-equipped mobile phones and the development of wireless communication technology have enabled mobile users to collect and share information about their surroundings, for example, individuals, human society, and environments. Empowered by these capabilities, mobile users have shifted from being data consumers to being data providers, offering a new service model for mobile cloud computing, that is, mobile crowdsourcing (MC) [1,2]. By integrating mobile computing and crowdsourcing, MC provides requesters with real-time data regarding points of interest (POIs). It offers a number of advantages (e.g., low deployment cost, high spatiotemporal coverage and easy to update) over traditional sensor networks [3–6], which entails deploying a large number of static wireless sensor devices.

An MC system typically involves three main actors: the mobile users that contribute the sensor data (the workers),

the service provider (SP) that processes the collected data, and the end users that subscribe to the service (the requesters). The requesters and the mobile workers are collectively called the participants. An example is illustrated in Figure 1. A requester who desires information regarding antigovernment demonstrations in various locations of a city issues a query to the SP. Subsequently, the SP outsources the data collection task to the well-suited mobile users, who can perform data sensing using their sensor-equipped mobile devices. Then, the selected mobile users physically travel to the designated locations and upload sensing data (e.g., pictures, video, or audio) to the SP through wireless connection. Finally, the SP processes the collected data to generate the query result for the requester. A common challenge for the system is to identify the participants (both the requesters and mobile workers) who can contribute more value to the system and motivate their participation in the system.

Although many MC systems have been proposed, most of them (e.g., [1,2,7,8]) have simply assumed that the

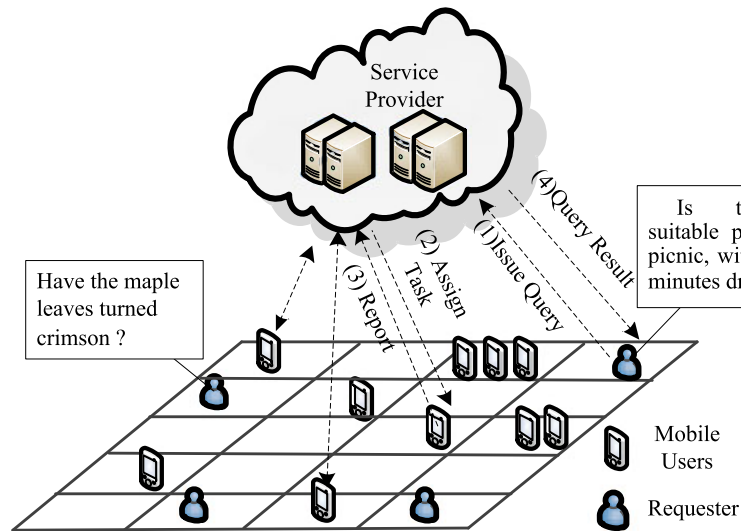


Figure 1. Example of a mobile crowdsourcing system: (1) a requester issues a query, (2) the service provider (SP) assigns the task to mobile users, (3) the selected mobile users (workers) upload their sensing data to the SP, and (4) the SP returns the query result to the requester.

workers voluntarily participate in the system to perform sensing tasks. However, mobile users may not be willing to join an MC system unless they will receive adequate reward. As is well known, some costs (e.g., time, battery, and wireless communication expenses) are incurred by the mobile users who perform these MC tasks. Moreover, the sensing data may include personal information, such as the users' locations, and thus, sharing these data poses potential privacy threats to the participating users. All of these bring the reward-based MC system to the fore.

Many approaches (e.g., [9–22]) have been proposed in recent years to address the participant recruitment issue in reward-based MC. According to the different purpose of employing these strategies, the existing work can be roughly divided into the following categories: (1) user-centric approach: This approach argues that the cost of the participants is rationally reflected in the fees that the users claim and attempts to minimize the side effects to the participants during the data collection, such as travel costs (e.g., [9]), energy consumption (e.g., [10,11]), and privacy leaks (e.g., [12–16]); (2) platform-centric approach: This approach focuses on how to evaluate the participants' contributions and improve the information gain achieved by the platform. Most platform-centric approaches (e.g., [17–22]) used an auction or reverse auction mechanism to reduce the overall cost of the platform. However, these studies have some limitations, which motivated our work. First, they always focused on “atomic query” and assumed that all queries are independent. However, in reality, there could be cases in which the SP may need to recruit several mobile users in different locations to collectively fulfill a complex query, which will be accomplished if and only if all of the spatial sub-tasks are accomplished. Second, they focused on “single-shot”

scenarios and assumed that the SP always has a sufficient number of workers to handle all queries. However, an MC system consists of dynamic participants who can join or leave at any time. Therefore, such an assumption is not viable. To the best of the authors' knowledge, [17] and [22] are the only studies that explicitly addressed the long-term participation incentive mechanism in MC. In [17], Lee *et al.* introduced a virtual credit to low the bids of users who lost in the previous auction round and are participating in the current auction round, hence increasing their probabilities of winning. In [22], Gao *et al.* introduced a virtual queue to store candidate users that will exit the system and studied the queue stability based on the Lyapunov drift [23]. However, these authors did not consider the dynamic nature of MC, namely, the ratio between the mobile users and workers is dynamically changing. To address the issues identified previously, a new participant recruitment strategy for MC must be developed.

This paper addressed the fundamental research issue in participant recruitment: *Which participants (both requesters and workers) should be selected as winners in each time slot with the aim of maximizing the total utility of the SP in the long term?* This problem is challenging for several reasons. First, the competition among the participants makes them partially in conflict with one another. Different requesters may compete for the same worker, and different workers may compete for the same query. In addition, a requester's query will be satisfied if and only if all of the necessary spatial sub-tasks are accomplished. Second, the system dynamics couples the utility values of the SP in different time slots. The workers participating in the MC system may suffer certain indirect costs even when they are not performing sensing tasks (such as periodically reporting their

locations to the SP). Thus, if a worker is rarely selected as a participant, that worker may become frustrated with the SP and decide to exit the system. A decline in the number of workers participating in the system may affect the future utility of the SP.

To address this problem, this paper first studies the relationship between the requesters and workers and then introduces a double-sided combinatorial auction model to describe the interactions between the numbers of the requesters and workers from the perspective of supply and demand. Then, the paper introduces and investigates the context-aware participant recruitment (CAPR) problem, in which the ratio between the requesters and workers is not constant and isolated along the temporal dimension. Next, the problem is transformed into how to achieve a good balance between the workload and the utility at each point, with the aim of maximizing the long-term system utility. On the basis of the aforementioned analyses, this paper presents the design of the CAPR mechanism that consists of two main components. In the first component, a novel metric called “bid density” is introduced to evaluate the value of the requesters’ query, and a heuristic algorithm based on the greedy strategy is then proposed to determine the winning participants within a suitable time frame. The second component is a critical payment scheme, which guarantees the rationality of the proposed mechanism.

The contributions of this paper are summarized as follows:

- (1) A double-sided combinatorial model is introduced that, for the first time, describes the interactions between the workers and requesters from the perspective of supply and demand. The model is suitable for the spatially complex queries.
- (2) A CAPR mechanism is introduced and investigated, in which the ratio between the requesters and workers is not constant and isolated along the temporal dimension.
- (3) A novel metric called “bid density” is introduced to evaluate the value of the requesters’ queries, and a heuristic algorithm based on the greedy strategy is then designed to determine the winning participants. Moreover, a critical payment is presented that guarantees the rationality of the proposed mechanism.
- (4) The proposed algorithms are evaluated by means of extensive simulations to shed light on how the CAPR mechanism improves the utility of the SP in the long term.

The remainder of this paper is organized as follows. Section 2 introduces the system framework and formulates the problem. The process of deriving the workload constraint is presented in Section 3. Section 4 describes the details of the proposed scheme. Section 5 presents performance evaluation results. Finally, conclusions are drawn in Section 6.

2. SYSTEM MODEL AND PROBLEM FORMULATION

This section first presents the motivation and the basic idea of the proposed solution, and then a formal description of the problem is provided.

2.1. Motivation and basic idea

Most existing approaches utilize the Vickrey–Clarke–Groves (VCG) mechanism and its variants to recruit participants with the aim of minimizing the total payment of the SP. These studies have some limitations, which motivated our work.

First, they always focused on “atomic query” and assumed that all queries are independent. However, this assumption is inadequate for real-world applications. An example is illustrated in Figure 1: A requester issues the query “Is there a suitable park for a picnic within a 10-minute drive?” to the SP. To answer this question, the SP must determine the following: (1) the air quality of nearby parks and (2) the degree of traffic congestion. The requester’s query will be satisfied if and only if all of the sub-tasks are successfully completed. The traditional auction mechanism, in which the requester bids on the two queried resources in two sequential, independent auctions, is inefficient in the situation.

Second, they focused on a “single-shot” scenario and assumed that the SP always has a sufficient number of workers to handle all queries. However, an MC system consists of dynamic participants who can log in or out at any time; thus, that assumption may not be valid. In addition, the workers participating in the MC system may suffer certain indirect costs (such as periodically reporting their locations to the SP) even when they are not performing sensing tasks. Thus, if a worker is rarely selected as a winner, that worker may become frustrated with the SP and decide to exit the system. Therefore, the participant recruitment mechanism determines the remaining workers in the system, which is closely related to the utility of the SP in the future. Thus, the utility of the SP in different time slots is coupled. The system utility must therefore be considered from a long-term perspective.

For the first problem, the approach taken in this paper is to study the interactions between the requesters and workers from the perspective of supply and demand in the economy. To this end, a double-sided combinatorial auction model is introduced in which the requesters are the buyers (buying the queried resources) and the workers are the sellers (selling the queried resources). An example is illustrated in Figure 2. There are three requesters and five workers in the MC system. User 1 seeks to buy queried resources r_1 and r_2 with a bidding of \$5, user 2 bids \$8 for queried resources r_1 , r_2 , and r_3 , and user 3 bids \$5 for queried resource r_2 . All of these users submit their bids to the online auction platform. Five workers are participants in the auction, and all of these workers also submit their bids to the platform. At the end of the bidding period, the winner determination module computes the winners and prices for the auction. Then, the results are announced to the requesters and workers.

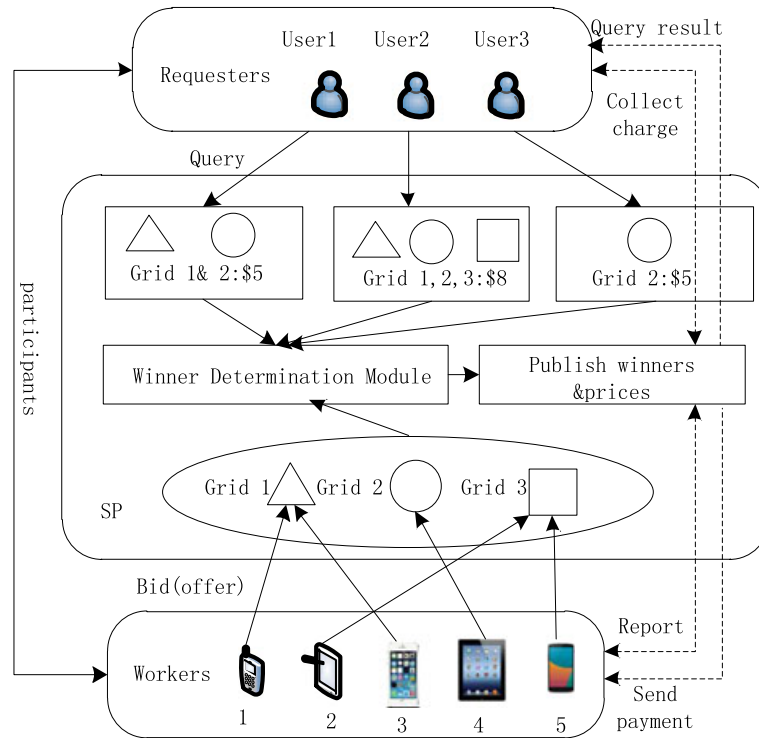


Figure 2. A participant recruitment scenario consisting of three requesters and five mobile workers distributed across three different grids. Each query from a requester may require several grid resources.

For a mobile user who required two queried resources, bidding for two items in one combinatorial auction is more efficient than in two sequential auctions separately. Moreover, double auction allows requesters and workers to bid simultaneously in one auction, which also improves system efficiency. However, in the combinatorial auction, one requester may require several resources, and different requesters may compete for the same resource, while different workers may possess the same resource and may compete for the same query. Thus, determining how to select the winners is a challenge.

For the second problem, the CAPR problem is introduced and investigated. In this scenario, the ratio between the requesters and workers is not constant and isolated along the temporal dimension. The basic idea of CAPR is that the SP should offer the capability to dynamically adjust the participant recruitment mechanism depending on the ratio between the requesters and workers. This paper argues that the SP should place a greater emphasis on load balance (increasing the amount of the workers remaining in the system) when there is a high ratio of requesters to workers. Otherwise, the SP should place more value on the system utility (maximizing the system utility at a given moment). Then, the problem is formalized as a problem striking a good balance between workload and utility. However, it is not easy to determine the appropriate balance between workload and utility at the present moment with the aim of maximizing the long-term system utility.

2.2. Problem formulation

The symbols used throughout the paper are defined in Table I. This paper considers an MC system to include an SP and many participants. As shown in Figure 1, it is assumed that the SP decomposes the spatial space into a set $R = \{r_1, r_2, \dots, r_l\}$ of l grids. The MC system provides users with real-time data. Thus, the service coverage of a worker is associated with that worker's location. Therefore, it is assumed that each worker in the same grid cell

Table I. Symbols used in this paper

Symbols	Meaning
q_i	The query from requester i
b_i	The bid price of requester i
s_j	The bid request of worker j
$r(j)$	Worker j possesses the grid resource $r(j)$.
c_j	The claimed cost of worker j
Q_i^k	The set of k grid resources required to complete query q_i
$S(r_i)$	The set of workers that possess the grid resource r_i
$Q(r_i)$	The set of queries that require grid resource r_i
Q	The set of all queries
$\tau(j)$	The drop threshold of worker j
ε	The drop rate
γ_i	The bid density of requester i
$p(s_j)$	The critical payment of worker j
$p(q_i)$	The critical payment of requester i
τ	The competitive ratio

provides the SP with the same sensing capability. This paper considers the system utility in a long period to consist of a set $T = \{1, 2, \dots, T_1\}$ of T_1 time slots. There are m requesters and n workers in the system at time t . The set of requesters is denoted by $M(t) = \{1, 2, \dots, m\}$, and the set of workers is denoted by $N(t) = \{1, 2, \dots, n\}$. At the beginning of each time slot, the interactions between the SP and the participants are described as follows; an example scenario is also illustrated in Figure 2.

- (1) Each requester i issues a query $q_i = \{\text{query}(i), b_i\}$ to the SP, where $\text{query}(i)$ denotes the content of the requester's query and b_i is the bid price, which represents the value of the query to the requester. Note that a requester's query may consist of several sub-tasks corresponding to different locations.
- (2) Each worker j issues a bid $s_j = \{\text{loc}(j), c_j\}$ to the SP, where $\text{loc}(j)$ denotes the worker's location and c_j is the worker's claimed cost at time t .
- (3) When the SP receives a query from a requester, the SP establishes a mapping relationship between query (i) and the grids. Thus, the query is expressed as $q_i = \{Q_i^k, b_i\}$, where Q_i^k denotes the set of k grid resources required to complete query q_i . As shown in Figure 2, $q_1 = \{\{r_1, r_2\}, b_1\}$.
- (4) When the SP receives a bid from the worker, the SP establishes a mapping relationship between $\text{loc}(j)$ and the grid. Thus, the bid is expressed as $s_j = \{r(j), c_j\}$, where $r(j)$ denotes that worker j is located in grid $r(j)$, namely, j possesses grid resource $r(j)$ at time t . As shown in Figure 2, $s_1 = \{r_1, c_j\}$.
- (5) The SP determines whether each participant is winning and then informs the participants of the auction results. Then, the winning workers perform their tasks and upload the collected data reports to the SP through wireless connections.
- (6) The SP processes the collected data and returns the query results to the requesters. Then, each worker j is paid an amount of money $p(s_j)$ corresponding to its winning bid s_j , and each winning requester i pays a certain amount of money $p(q_i)$ corresponding to its winning query q_i .

During the interaction, the challenge faced by the SP is as follows: *Which participants (both requesters and workers) should be selected as winners in each time slot with the aim of maximizing the long-term total utility of the SP?* The system utility is defined as follows:

Definition 1 (system utility). The system utility is introduced as a performance index to reflect the social welfare that is generated by the SP. The system utility can be regarded as the difference between the total value of all requesters and the total social cost of all workers. Thus, at time t , the utility of the SP $U(t)$ is computed as follows:

$$U(t) = \sum_{q_i \in Q} y_i b_i - \sum_{s_j \in S} x_j c_j \quad (1)$$

where y_i denotes whether the SP selected query q_i (from requester i) as a winning participant (y_i is equal to 1, when query q_i is selected as a winner) and x_j denotes whether the SP selected bid s_j (of worker j) as a winning participant (x_j is equal to 1 when bid s_j is selected as a winner). The set $Q = \{q_i | 0 < i \leq m\}$ denotes the set of all queries, and the set $S = \{s_j | 0 < j \leq n\}$ denotes the set of all bids from the workers. The first summation in (1) is the value of the winning requesters, and the second summation denotes the total social cost of the winning workers.

The goal of the SP is to maximize its total utility in the long term, which is calculated as follows:

$$\max \sum_{t \in T} U(t) \quad (2)$$

As mentioned previously, the utility values of MC in different time slots are coupled. However, the system is dynamic. Workers may log in or out at any time. The information concerning the future state is incomplete, and the SP cannot explicitly determine the winning participants in each time slot in advance and therefore must determine the winning participants using only the currently available information.

Thus, the CAPR mechanism is introduced and investigated in which the ratio between the requesters and workers is not constant and isolated along the temporal dimension. It is argued that the SP should offer the capability to dynamically adjust its participant recruitment mechanism, depending on the ratio between the numbers of the requesters and workers. It is proposed that the SP should place a greater emphasis on load balance (increasing the number of the workers remaining in the system and decreasing the worker drop rate) when there is a high ratio of requesters to workers, whereas otherwise, the SP should place more value on system utility (maximizing the current system utility). The mathematical formulation of the CAPR is defined later.

Definition 2 (CAPR problem). For each subset $M(t)$ of the requesters and each subset $N(t)$ of the workers, the problem of determining winning participant is defined as follows:

$$\max \left(\sum_{q_i \in Q} y_i b_i - \sum_{s_j \in S} x_j c_j \right) \quad (3)$$

$$\text{s.t. } x_i, x_{ij} \in \{0, 1\} \quad (4)$$

$$y_i = \begin{cases} 1 & \forall r_j \in Q_i^k, x_{ij} = 1 \\ 0 & \text{else} \end{cases} \quad (5)$$

$$\sum_{q_i \in Q} x_{ij} \leq |S(r_j)| \quad \forall r_j \quad (6)$$

$$\varepsilon < f(M(t), N(t)) \quad (7)$$

Remarks. The definition of the CAPR problem indicates that the objective of the SP in selecting the winning participants is to maximize the system utility, as defined in (3). The set $S(r_j) = \{s_i | r(i) = r_j, 0 < i \leq n\}$ denotes the set of

workers that possess the grid resource r_j . The variable x_{ij} is equal to 1 when the SP allocates grid resource r_j to q_i ; otherwise, the variable x_{ij} is equal to 0. The constraint defined in (5) indicates that the requester's query is satisfied if and only if all of the corresponding spatial sub-tasks are completed. The constraint defined in (6) indicates that the number of times that grid resource r_j that is allocated cannot exceed the maximum number $|S(r_j)|$. In this paper, it is assumed that each worker can complete only one sub-task at a time. However, the problem formulation can be easily extended to address other situations. The constant defined in (7) is the load balance constraint, and it indicates whether the drop rate ε of the participants is sufficient to satisfy the basic requirements of the SP for load balancing, which is determined by the set of requesters $M(t)$ and the set of workers $N(t)$. The process of deriving the formula $f(M(t), N(t))$ will be discussed in the next section.

3. LOAD BALANCE VERSUS UTILITY MAXIMIZATION

This section first further investigates the CAPR problem and then describes the derivation process of the tradeoff between system utility and load balance.

3.1. Context-aware participant recruitment problem

We consider a scenario in which each worker j would exit the system if his or her winning probability (the probability of being selected as a winner) is smaller than a certain threshold $\tau(j)$, which is called the drop threshold of worker j . The drop rate of the workers is used as the metric for the load balance. The drop rate is defined as follows.

Definition 3 (drop rate). Let $W(t)$ denote the set of winning participants at time t . Let $s_j[t] \in \{0, 1\}$ denote whether the worker j is selected as a winning participant. If $j \in W(t)$, $x_j[t] = 1$; otherwise, $x_j[t] = 0$. Then, the worker drop rate ε_t at time t is calculated as follows:

$$\varepsilon_t = \frac{|\{j | p_j < \tau(j), j \in N(t)\}|}{|N(t)|} \quad (8)$$

$$p_j = \frac{\sum_{t \in T} x_j(t)}{|T|} \quad (9)$$

where p_j denotes the winning probability of worker j up through time t ; p_j can be calculated using (9). Then, the number of workers at time $t+1$ is calculated as follows:

$$N(t+1) = N(t) - \varepsilon_t N(t) + \lambda(t+1) \quad (10)$$

where $\lambda(t+1)$ denotes the stochastic future information at time $(t+1)$, which combines the effects of a variety of

factors, such as a worker logging out of the system to attend some urgent matter or a new worker joining. In this paper, it is argued that the SP can summarize the general trend of the queries based on the queries log and use this trend to predict further queries in the short-term time. Thus, the number of requesters at time $t+1$ is calculated as follows:

$$M(t+1) = \alpha M(t) + (1 - \alpha)F_t \quad (11)$$

where $F(t)$ denotes the predicted numbers of the requesters and $0 < \alpha \leq 1$ is a control parameter that is chosen to achieve a desirable tradeoff between the predicted value and the observation.

As mentioned previously, the selection of winning participants at the current time determines the number of workers available in the next time slot to a certain extent. In addition, the system utility is closely related to the number of workers. The utility values of MC in different time slots are coupled. The goal of the SP is to maximize the long-term utility of the system. Thus, the participant recruitment mechanism must consider not only the utility at the present time but also the load balance (drop rate), which is closely related to the system utility as follows.

With complete future information, it is easy for the SP to determine the optimal tradeoff between system utility and load balance at each time point to maximize the long-term system utility. As shown in (10) and (11), the number of workers and requesters changes over time, and the ratio between the requesters and workers cannot be predicted in an exact manner. This leads to a dilemma regarding how to achieve a suitable tradeoff between load balance and system utility with incomplete information. To address the problem, a CAPR mechanism is proposed in which the SP dynamically adjusts the participant recruitment mechanism, depending on the ratio between the numbers of the workers and requesters at the present time.

3.2. The derivation of the workload constraint

This subsection presents a logistic model to describe the interactions between the requesters and workers. For simplicity, it is initially assumed that the number of requesters remains fixed at y to investigate how changes in the number of workers affect participant recruitment.

The basic premise is that the SP should place a greater emphasis on the drop rate (load balance) when there is a high ratio of requesters to workers, whereas otherwise, the SP should place more value on the current system utility.

Let x denote the number of workers at time t , and let the function $h(x)$, which takes the number of workers x as an input, denote the upper limit on the drop rate. The drop rate ε_t should be less than or equal to $h(x)$ when the number of workers is x . As the number of the workers increases, the SP should pay greater attention to the system utility. Thus, $h(x)$ is a monotonically increasing function of

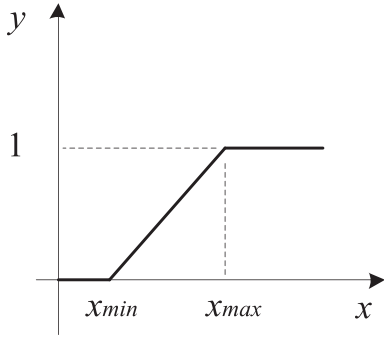


Figure 3. Function $h(x)$: upper limit on the drop rate.

x , and $0 \leq h(x) \leq 1$. One can simply assume that $h(x)$ is a linear function. Then, as shown in Figure 3, the function $h(x)$ can be specified as follows:

$$h(x) = \begin{cases} 1 & x > x_{\max} \\ a + bx & x_{\min} \leq x \leq x_{\max} \\ 0 & x < x_{\min} \end{cases} \quad (12)$$

where x_{\max} is the critical point at which there is a sufficient number of workers such that the SP only focuses on the system utility, x_{\min} is the critical point at which the SP only focuses on the drop rate, and a and b are the normalization parameters, which are calculated as follows:

$$\begin{cases} a + bx_{\max} = 1 \\ a + bx_{\min} = 0 \end{cases} \quad (13)$$

From earlier, the values of these parameters can be computed.

$$\begin{cases} a = \frac{-x_{\min}}{x_{\max} - x_{\min}} \\ b = \frac{1}{x_{\max} - x_{\min}} \end{cases} \quad (14)$$

Then, the relationships between the number of requesters y and the critical points (x_{\max} and x_{\min}) can now be quantified. It is assumed that there are y requesters in the system, that their queries will require an average of z workers to complete, and that the drop threshold of each worker is $1/k$.

Definition 4 (competitive ratio). Given the number of queries y and the number of workers x , the competitive ratio τ is calculated as follows:

$$\tau = \frac{x}{y * z} \quad (15)$$

where y denotes the size of $M(t)$ and x denotes the size of $N(t)$.

Theorem 1. The maximum number of workers that will remain in the system once the system has converged to a stable state (the number of workers is fixed, and no worker

will log out because of an excessively low winning probability) is $k * y * z$.

Proof. Assume that there are $k * y * z + 1$ workers in the system at time t . Because the system is stable, there will also be $k * y * z + 1$ workers in the system at time $t + k$. However, the system assigns only $k * y * z$ total tasks between time t and time $t + k$. Therefore, there must be at least one user who does not receive any tasks during this period. In this scenario, the worker's winning probability will become smaller than $1/k$, and the worker will exit the system. This conflicts with the assumption that the system has converged to a stable state. And in this case, the competitive ratio τ is equal to k .

Theorem 2. For all queries to be completed, at least $y * z$ workers must be present in the system.

Proof. The queries will require an average of z workers to complete, and each worker can accept only one task at a time. Thus, there must be at least $y * z$ workers in the system. And in this case, the competitive ratio τ is equal to 1.

Theorem 1 indicates that the SP will focus only on the system utility once the competitive ratio τ reaches the required maximum competitive ratio τ_{\max} , which is lower than or equal to k . Theorem 2 indicates that the SP will focus only on the drop rate when the competitive ratio τ drops below the required minimal competitive ratio τ_{\min} , which is greater than or equal to 1. Thus, the workload constraint in (7) can be calculated as follows:

$$f(M(t), N(t)) = \begin{cases} 1 & \tau > \tau_{\max} \\ a + bN(t) & \tau_{\min} \leq \tau \leq \tau_{\max} \\ 0 & \tau < \tau_{\min} \end{cases} \quad (16)$$

where a and b are calculated using (14) and the competitive ratio τ is calculated using (15).

4. PROPOSED SCHEME

The proposed schemes consist of two components: The first component solves the CAPR problem to determine the winning participants, and the second component is a payment scheme. Before the algorithm for the first component is described, it is first proven that the CAPR problem is non-deterministic polynomial (NP)-hard and the bid density of the requesters is analyzed. Then, a heuristic algorithm based on the greedy strategy to solve the CAPR, which can effectively determine the winners within a suitable time frame, is proposed to solve the CAPR problem. Finally, a payment scheme is proposed to motivate the requesters and workers to continue to participate in the system.

Theorem 3. The CAPR problem is NP-hard.

Proof. It can be proven that the CAPR problem is NP-hard by proving that its decision version is NP-complete. For

the decision problem, it should be demonstrated that it belongs to NP and then another known NP-complete problem should be identified that can be reduced to the decision version of the CAPR in polynomial time. The decision version of the CAPR is a modified mixed 0–1 integer programming problem in which certain sub-tasks belong to the same query, that is, they should win simultaneously. The decision problem belongs to NP because checking whether a solution is correct requires polynomial time. Therefore, the CAPR is NP-hard. Thus, it is critical to solve the CAPR with a time-efficient algorithm.

4.1. Bid density of the requesters

To determine the winning participants, a metric called “bid density” is introduced to evaluate the value of the queries (of the requesters).

Definition 5 (bid density γ). The bid density is introduced as a performance index to reflect the value of queries. A requester with a higher bid density will have a higher probability of winning. For a query $q_i = \{Q_i^k, b_i\}$ from requester i , the query’s bid density γ_i is calculated as follows:

$$\gamma_i = b_i / Q_i^k \quad (17)$$

However, given any two queries q_i and q_j within the requester’s set $M(t)$, the grid resources that are required to complete these queries may be different. In other words, set Q_i^k may not be equal to set Q_j^k . Thus, in most cases, the bid densities of different queries cannot be directly compared with one another via (17). One straightforward method is to compare the size of the set Q_i^k . Thus, the query’s bid density γ_i is calculated as follows:

$$\gamma_i = b_i / |Q_i^k| \quad (18)$$

However, as shown in Figure 4, the distributions of queries and grid resources vary significantly across the grids. The relationships between supply and demand for

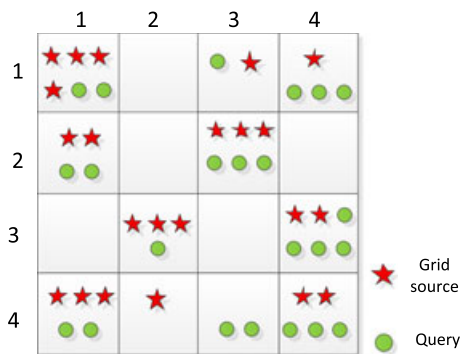


Figure 4. The bid density of a requester’s query, r_{ij} corresponding to the grid in line i and column j .

the grid sources are different in different regions. As a simple example, consider the following scenario. Given two queries, q_i and q_j , assume that the bidding price b_i is equal to b_j and that $Q_i^2 = \{r_{11}, r_{32}\}$ and $Q_j^2 = \{r_{14}, r_{34}\}$. Then, Equation (18) indicates that γ_i is equal to γ_j . However, as shown in Figure 4, the winning probability of query q_i is higher than the winning probability of query q_j . Based on the aforementioned analysis, the bid density of the requester can be analyzed from the perspective of supply and demand in the economy, yielding the following expression for the bid density γ_i of a query:

$$\gamma_i = \frac{b_i}{|Q_i^k|} * \frac{\sum_{r_i \in R} |S(r_i)|}{\sum_{r_i \in R} |S(r_i)|} * \frac{\sum_{r_i \in R} |Q(r_i)|}{\sum_{r_i \in Q_i^k} |Q(r_i)|} \quad (19)$$

where $Q(r_i) = \{q_j | r_i \in Q_j^k, 0 < j \leq m\}$ denotes the set of queries that require the grid resources r_i and $S(r_i) = \{s_j | r(j) = r_i, 0 < j \leq n\}$ denotes the set of workers that possess the grid resource r_i .

4.2. Heuristic greedy algorithm for the context-aware participant recruitment problem

To achieve the desired computational efficiency, a heuristic greedy algorithm is proposed to solve the CAPR problem. The SP first sorts the workers who possess the same grid resources in ascending order by their claimed costs. Then, the SP adopts a greedy strategy to solve the problem. The SP selects the query ($q_i \in Q, Q = \{q_i | 0 < i \leq m\}$) that has the highest bid density and adds it into W_1 (the set of winning requesters) until all of the requesters (that belong to Q) have been assigned. In each iteration, the bid density γ_i is updated, and all queries that could not be selected together with the existing winning queries are deleted (from Q). Once the winning requesters have been determined, the grid resources required to complete the queries are identified. Then, the SP selects the worker ($L[i][j]$) with the lowest claimed cost from the set $L[i]$ of workers that possess the grid resource r_i and places that worker into W_2 (the set of winning workers). The detail of this algorithm is shown in Algorithm 1.

Algorithm 1: Heuristic greedy algorithm for solving the CAPR problem
Input: $Q = \{q_i | 0 < i \leq m\}, S(x_i), Q(x_i)$
Output: $W_1 // W_1$: the set of winning requesters
 $W_2 // W_2$: the set of winning workers
1. Initialize $W_1 \leftarrow \emptyset; W_2 \leftarrow \emptyset; \text{int } a[|R|]; \text{int } j;$
// Sort the workers who possess the same grid resource in ascending order by the claimed cost.
2. for i from 1 to $|R|$ // grid resource r_i
3. Sort c_j for all $s_j \in S(r_i)$ in ascending order and the resulting list is denoted by $L[i]$.
4. $L[i][j]$ denotes the j th element of $L[i]$.
5. end for
6. while $Q \neq \emptyset$
7. Sort γ_i for all $q_i \in Q$ in descending order, and the resulting list is denoted by queue L
8. q_a denotes the first element in L
9. $W_1 \leftarrow W_1 \cup \{q_a\}$ // Determine the winning requesters
10. Delete q_a from Q


```

11. for each  $x_i$  in  $Q_s^k$ 
12.    $j = a[i]++$ ;
13.    $W_2 \leftarrow W_2 \cup \{L[i][j]\}$ ; // Determine the winning workers.
14.   Delete  $L[i][j]$  from  $S(x_i)$  //Delete the requesters that
conflict with the winning requesters
15.   if  $|S(x_i)| <= 0$ 
16.     for each  $q_i$  in  $Q(x_i)$ 
17.       delete  $q_i$  from  $Q$ ;
18.     end for
19.   end if
20. end for
21. end while
22. return  $W_1, W_2$ ;

```

Algorithm 1 focuses on system utility; however, as mentioned previously, the set W_2 of winning workers should satisfy the basic requirements of the SP for load balancing, which are defined in (7). Thus, the SP first counts the set of the workers $D(t) = \{j | p_j < \tau(j), j \in N(t)\}$ to determine who will drop out using (8). Then, the SP determines whether the winning worker set W_2 satisfies the requirements of the system, which can be calculated using (16). If this condition is not met, then the SP will adopt a reassignment strategy to solve the problem. The SP will select the worker with the lowest claimed cost and uses that worker to replace the worker with the highest claimed cost in set W_2 . The SP will repeat this process until the drop rate meets the necessary requirement.

4.3. Payment scheme for the context-aware participant recruitment

To motivate participants to join MC and continue to participate in the MC system, a reasonable payment scheme should be designed that satisfies the following critical properties: (1) individual rationality: Each participant can expect a non-negative utility upon participating in the system; and (2) budget balance: The system can run auctions without deficits. Thus, the rule of critical payment introduced in [24] is used to determine the amount that the requesters should pay and the amounts that the workers should receive.

Definition 6 (critical value). Given a query $q_i = \{Q_i^k, b_i\}$ from requester i , if the bid value b_i is greater than or equal to $p(q_i)$, then requester i will be selected as a winning participant; otherwise, the requester will lost the bid. The bid value $p(q_i)$ is called the critical value for requester i . The concept of a critical value is also applied to the bid request from the workers.

First, the critical values for the workers are calculated. In the proposed model, each worker can complete only one sub-query at a time (although the model can easily be extended to other situations). Additionally, workers that possess the same grid resources compete for the same queries. In each time slot, they issue their respective bid requests to the SP. Thus, the entire set of sales offer can be regarded as a Vickrey auction, which is also known as a sealed-bid second-price auction [18].

Assume that there are n workers ($c_1 < c_2 < \dots, c_k < c_k + 1, \dots, c_n$) that possess the same grid resource r_i and that the worker with a claim cost of c_k was selected as a

winning participant in the final round. Then, the critical value for the worker j , who also possesses the grid resource r_i ($r(j) = r_i$), can be calculated as follows:

$$p(s_j) = \begin{cases} c_{k+1} & 0 < j \leq k \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

Then, the critical values for the requesters are calculated. As shown in Algorithm 1, a requester fails to win in the auction when the requester's query conflicts with the queries that have won ($\exists r_i \in Q_i^k, S(r_i) = \emptyset$). In this situation, there are an insufficient number of workers to satisfy the requester's query. Thus, determining the critical payment (of q_i) involves deleting q_i and greedily selecting other queries as shown in Algorithm 1 until the remaining workers cannot satisfy query q_i . It is assumed that query q_i fails to win in the auction when query q_s is selected as a winner. Thus, the critical value of the requester (the payment that the requester should pay) is calculated as follows:

$$p(q_i) = \begin{cases} \max \left(\left(\frac{\sum_{r_i \in R} |S(r_i)| \sum_{r_i \in Q_i^k} |Q(r_i)|}{\sum_{r_i \in Q_i^k} |S(r_i)| \sum_{r_i \in R} |Q(r_i)|}, \sum_{r_i \in Q_i^k} p(r_i) \right) & q_i \in W_1 \\ 0 & \text{else} \end{cases} \quad (21)$$

where $p(r_i)$ denotes the payment that a worker possessing grid resources r_i will receive and γ_s denotes the bid density of q_s . To guarantee of the rationality of the auction, the payment $p(q_i)$ should be lower than or equal to the bid value b_i . Otherwise, the SP will delete q_i from the winning set W_1 . The detail of this algorithm is shown in Algorithm 2.

Algorithm 2: Critical value algorithm for the requester
Input: query q_m , other submitted $Q_{-(m)} = \{q_i | i \neq m, 0 < i \leq n\}$
Output: $p(q_m)$ //critical value
1. $W_1 \leftarrow \emptyset$; int $flag \leftarrow 0$; int $a[] = \{|S(x_1)|, |S(x_2)|, \dots, |S(x_n)|\}$
2. while $Q_{-(m)} \neq \emptyset$ && $flag = 0$ do
3. Sort γ_i for all $q_i \in Q_{-(m)}$ in decreasing order, and the resulting list is denoted by queue L
4. q_s denotes the first element of L
5. $W_1 \leftarrow W_1 \cup \{q_s\}$
6. delete q_s from Q ;
7. for each x_i in Q_s
8. if $-a[i] < 1$
9. if x_i in Q_m^k // Determine whether q_m conflicts with the winner
10. $flag = 1$;
11. break;
12. end if
13. for each q_i in $Q(x_i)$
14. delete q_i from $Q_{-(k)}$; //Delete the queries that conflict with the queries that have won
15. end for
16. end if
17. end for
18. end while
19. if $p(q_m) > b_m$
20. $p(q_m) = 0$
21. end if
22. return $p(q_m)$;

Algorithm 2 is *individually rational* because the participants are never charged more than their valuations as a result of the allocation. Moreover, Algorithm 2 is *budget*

balanced because the payment obtained from the requesters is always greater than or equal to the amount paid to the workers.

5. PERFORMANCE EVALUATION

5.1. Simulation setup

Foursquare is one of the most popular location-based social networks in the world; it has 15 million members as of June 2011 and keeps growing every month. The distribution of the check-in locations of Foursquare users represents the distribution of potential MC participants. Thus, the performance of the proposed mechanisms was evaluated through extensive simulations based on a real-world Foursquare dataset that was made available by Gao [25]. This dataset contains the check-in histories of 18 107 users from March 2010 to January 2011. The previous check-in locations and corresponding check-in times for each user are available. For a user i , the corresponding check-in sequence is a set of POIs ordered by check-in time $C(i) = \langle (p_1, t_1), \dots, (p_i, t_i), \dots, (p_m, t_m) \rangle$, where (p_i, t_i) indicate that user i checked-in POI p_i at time t_i .

In the simulation, the POIs were treated as the grid cells. Without loss of generality, 20 POIs were selected as the set of the grid resources R . Workers in the same grid cell provide the SP with the same sensing capability. Thus, a user that checks in POI p_i at time t_i is assumed to possess grid resource r_i . The real costs of the users were generated according to a normal distribution, where the mean μ of the real costs was set to 5. For query q_i in the system, k ($1 \leq k \leq 5$) POIs were randomly selected from set R as a set of the grid resources (Q_i^k) required to complete query q_i . Similarly, the real value of the queries was generated according to a normal distribution where the mean μ of the real value was set to 20. The default settings of the parameters are summarized in Table II.

5.2. Evaluation results at a single point

The performance of the proposed mechanism was first evaluated at a single point. In this evaluation, the focus was placed on examining the allocation performance of the proposed mechanism in different scenarios. The performance of the system utility predominantly depends on the value of the winning queries and the social cost. Thus, the following metrics was used: the completion ratio of the queries, the unit social cost, and the system utility.

Because no prior study has implemented online combinatorial double auction in MC, in the simulations, the proposed winning participants recruitment mechanism

(denoted by “CAA”) was compared with the following mechanisms: (1) the optimal offline allocation mechanism (denoted by “OPT”); (2) the combinatorial double auction mechanism based on a random policy (denoted by “CAR”), in which the SP randomly selects the winning queries; and (3) the single-sided auction mechanism (denoted by “AUC”) proposed in [11], in which the requesters bid for the queried resources in several sequential, independent auctions.

- (1) Completion ratio of all requesters’ queries

The simulations were performed under different background conditions. Thus, the total value of the completed queries was different. To compare the results obtained in the different experiments, a metric called the completion ratio η was introduced, which is calculated as follows:

$$\eta = \frac{\sum_{q_i \in W_1} b_i}{\sum_{q_i \in Q} b_i} \quad (22)$$

where b_i denotes the value of query q_i , W_1 denotes the winning queries, and Q denotes all queries that were received by the SP. Figure 5 plots the values of completion ratio η when the number of queries is fixed at 100 and the number of workers varies from 500 to 100. In Figure 6, the completion ratio is evaluated when the number of workers is fixed at 100 and the number of queries varies from 100 to 300. Figure 5 illustrates that the completion ratio η increases with an increasing number of workers, whereas in Figure 6, the completion ratio η decreases with an increasing number of queries.

As shown in Figure 5, when n (the number of workers) is equal to 500, all of the considered mechanisms result in a high completion ratio. However, as the number of workers decreases, the completion ratio of the AUC mechanism reduces more rapidly than those of the other mechanisms. Its completion ratio ($n = 300, \eta = 0.27$) is even lower than that of the CAR mechanism ($n = 300, \eta = 0.41$) when n is less than or equal to 300. This lower completion ratio occurs because the SP satisfies a requester’s query if and only if the requesters won all the auctions for grid resources required to

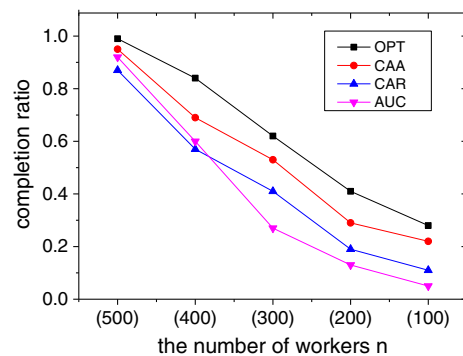


Figure 5. Completion ratio versus number of workers n .

Table II. Summary of the default settings

Parameter name	Default value
Number of grid resources	20
Range of cost c_j	[0,10]
Range of real value b_i	[10,30]
Number of sub-tasks in a query	[1,5]

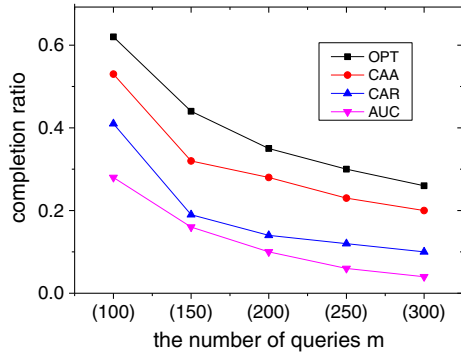


Figure 6. Completion ratio versus number of queries m .

complete the query. The simulation results demonstrate that the CAR mechanism is inefficient and that the CAA model is more suitable for participant recruitment in MC.

As shown in Figures 5 and 6, the completion ratio of the CAA mechanism ($n = 100, \mu = 0.22$) is always greater than that of the CAR mechanism ($n = 100, \mu = 0.11$) and is closer to that of the OPT mechanism ($n = 100, \mu = 0.28$). The results show that the proposed evaluation metric “bid density”, which is calculated using (19), can adequately reflect the true values of the queries.

(2) Unit social cost of the workers

The social cost has an important influence on the system utility. However, the completion ratios of the different schemes are different. Therefore, the total social costs of the queries completed under the different mechanisms are not directly comparable. For this reason, a metric called the unit social cost β , which denotes the average social cost of all winning queries, was introduced to evaluate the social cost of the different schemes. Formally, this metric is calculated as follows:

$$\beta = C_{total} / |W_1| \tag{23}$$

where W_1 denotes the set of winning queries and C_{total} denotes the total social cost of all winning queries. C_{total} is calculated using (24).

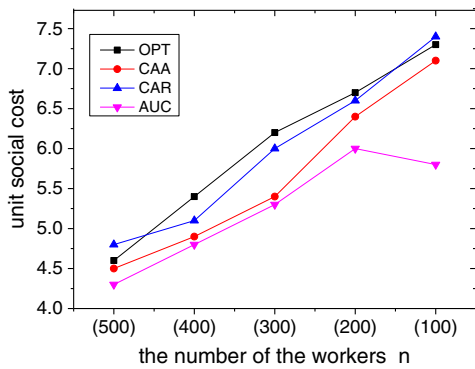


Figure 7. Unit social cost versus number of workers n .

$$C_{total} = \sum_{j \in W_2} c_j \tag{24}$$

where W_2 denotes the set of winning workers and c_j denotes the social cost of worker j .

Figure 7 plots the social costs when the number of queries remains fixed at 100 and the total number of workers varies from 500 to 100. In general, the unit social cost decreases with an increasing number of workers. For example, as shown in Figure 7, when n is equal to 100, the unit social cost of the CAA mechanism is 7.1, whereas the unit social cost is 4.5 when n is equal to 500. This is because as the number of workers decreases, the SP becomes less likely to find low-cost workers and thus incurs higher unit social costs. The results demonstrate that the SP needs a rational participant mechanism designed to incentivize mobile users to participate in the MC.

(3) System utility at a single point

According to Definition 1, the system utility can be regarded as the difference between the total value of all requesters and the total social cost of all workers. The simulations were performed under different background conditions. Thus, the values of the system utility were different. To compare the results obtained in the different experiments, the system utility of the OPT scheme was defined as 1 in each simulation. Figure 8 depicts the system utility values when the number of queries remains fixed at 100 and the total number of workers varies from 500 to 100. In general, the system utility increases with an increasing number of workers. For example, when n is equal to 100, the system utility of the CAA mechanism is 0.95, whereas the system utility is 0.79 when n is equal to 500. Among all investigated schemes, CAA has the highest system utility, primarily because it could achieve a high completion ratio.

5.3. Evaluation results of the entire time period

Finally, the performance of the proposed mechanism was evaluated from a long-term perspective. The performance of the proposed CAPR mechanism (denoted by “CAPR”)

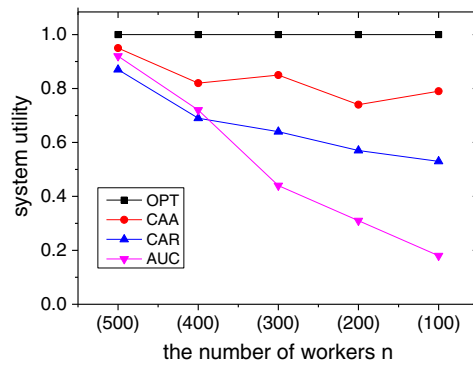


Figure 8. System utility versus number of workers n .

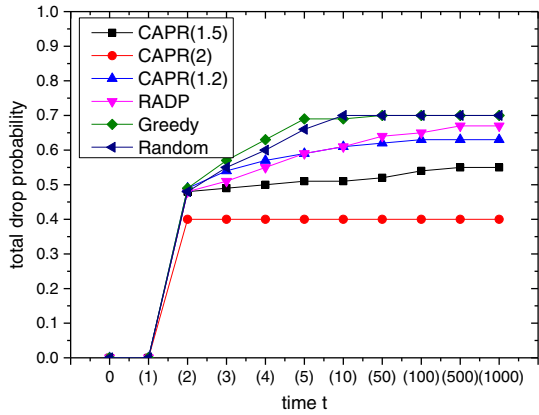


Figure 9. Total drop probability in scenario (a). CAPR, context-aware participant recruitment.

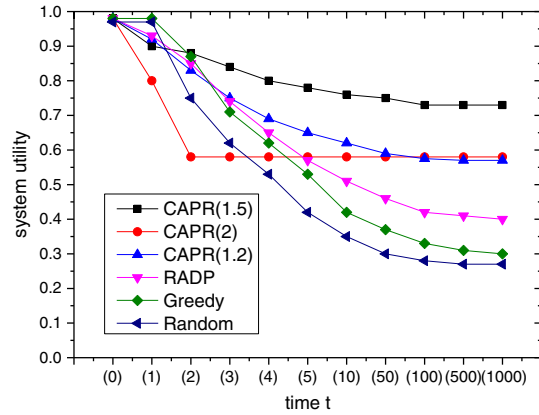


Figure 11. System utility in scenario (a). CAPR, context-aware participant recruitment.

was compared with the performances of the following mechanisms: (1) the CAA mechanism without long-term incentives, in which the objective of the SP is solely to maximize the system utility in each round (denoted by “Greedy”); (2) the combinatorial double auction mechanism based on a random policy (marked as “Random”), in which the SP randomly selects the winning queries; and (3) the CAA mechanism combined with the RADP-VPC policy proposed in [11] (denoted by “RADP”), in which a virtual credit is introduced to lower the bids of users who lost in the previous auction round and are participating in the current auction round, hence increasing their probabilities of winning.

In this simulation, the drop threshold was set to 0.5 ($k=2$) for all workers. Therefore, a worker would exit the system if his or her allocation probability was less than 0.5. Two different simulation scenarios, (a) and (b), were considered. In scenario (a), the number of queries was fixed at 120, and the number of workers was initialized at 200. No new users were added to the system during the simulation. Then, we ran the system over a period of 1000 time slots, which was sufficiently

long to achieve stable outcomes under the adopted policies. In scenario (b), the number of queries was fixed at 120, and the number of workers was initialized at 0. Workers were added to the system in the order that they appeared in the check-in sequence. The notation CAPR (x) indicates that the required minimal complete ratio τ_{min} was x . The simulation was repeated 20 times on different days.

(1) Drop probability

As shown in Figure 9, in scenario (a), more than 70% of the workers dropped out of the system under the greedy and random selection mechanisms, and approximately 65% of the workers dropped out under the RADP policy. However, by virtue of the context-aware participant mechanism, the proposed method retained more users in the system (CAPR (1.5), $\tau_{min} = 1.5$); the total drop probability was 56%. Moreover, the number of remaining workers increased with an increasing competitive ratio τ . When $\tau_{min}=2$ (CAPR (2)), the total drop probability closes to the theoretical lower bound of 40%.

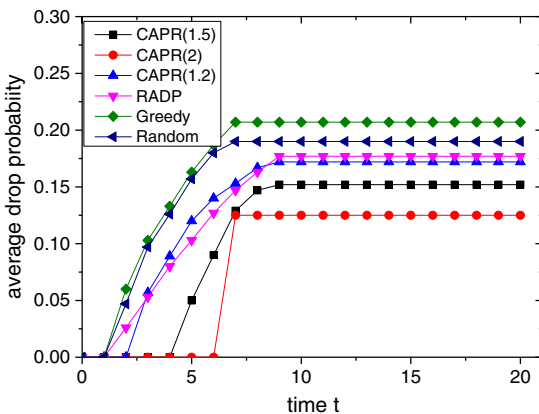


Figure 10. Average drop probability in scenario (b). CAPR, context-aware participant recruitment.

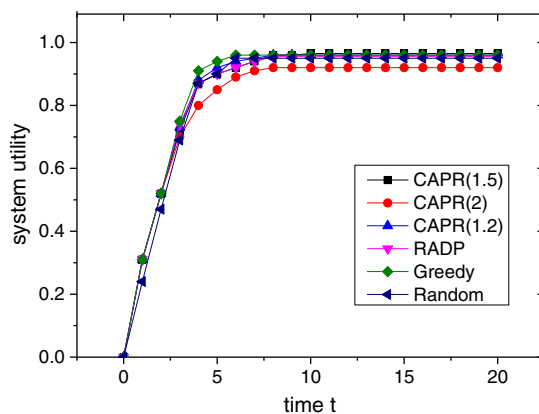


Figure 12. System utility in scenario (b). CAPR, context-aware participant recruitment.

Because the total number of workers changes in scenario (b) is dynamically changing, to enable direct comparison, the average drop probability in each time period was used, instead of the total drop probability, to evaluate the aforementioned investigated mechanisms. Figure 10 plots the change in the average drop probability over time. The average drop probability of the proposed mechanism was the lowest throughout the simulations because our participant recruitment mechanism is adaptive. When the ratio between the workers and queries is low, the SP places a greater emphasis on the load balance (increasing the number of workers that remain in the system). Thus, in the initial phase, the average drop probability was relatively low compared with the other mechanisms. When the system reached the balance point ($\tau_{min}=2$, approximately 950 workers), the proposed mechanism could retain more users in the system, while the same number of new users was being added to the system. Thus, in this scenario, the average drop probability of the proposed mechanism remained relatively low.

(2) System utility

As shown in Figure 11, in scenario (a), the system utility of all mechanisms decreased until the system reached the balance point. The system utility under the proposed mechanism ($\tau_{min}=1.5$, $\tau_{min}=1.2$) was higher than those under the other mechanisms. When τ_{min} was equal to 1.5, the system utility was always greater than 0.7, whereas the system utility was always greater than 0.55 when τ_{min} was equal to 1.2. This occurred because the system utility is closely related to the number of workers, as demonstrated in previous sections. Thus, the proposed mechanism offers a good performance gain because of its low drop rate. However, the performance of the proposed mechanism was poor when the SP only focused on the load balance ($\tau_{min}=2$). Thus, the performance of our mechanism depends on the choice of the parameter τ_{min} . A suitable value of this parameter can be determined for a given platform through historical data analysis.

As shown in Figure 12, in scenario (b), the system utility values of all mechanisms increased until the system reached the balance point. The process can be divided into two stages. In the first stage (from time t_0 to time t_8), the number of workers was not sufficient to satisfy all of the requesters' queries. Thus, during that stage, the system utility increased with near-linear efficiency as the number of workers increased. In the second stage, the platform attempted to select the workers with low social costs to increase the system utility. Thus, the growth gradually decreased. As the workers became abundant, most of the mechanisms achieved high efficiency.

6. CONCLUSION

This paper investigated the important problem of designing participant recruitment mechanism for maximizing the system utility of the SP in the long term. First, a double-sided combinatorial auction model was introduced to

address the participant recruitment at a given time. The problem of maximizing system utility in the long term was then investigated. The problem was formalized as how to achieve a tradeoff between load balance and system utility at each time point. Based on the aforementioned analysis, we designed a CAPR mechanism. First, a novel metric called bid density was proposed to evaluate the values of the users' queries, and a greedy heuristic algorithm was presented to determine the winning participants in polynomial time. Then, a critical payment scheme was proposed to guarantee the rationality of the mechanism. Finally, extensive simulations demonstrated that the proposed mechanism achieves high system utility in the long term.

Based on the work presented here, future research will address the following issues. First, several workers may collude to gain a better payoff. An extension of the CAPR mechanism to resist collusion attacks will be proposed. Second, mobile participants must submit sensitive information, such as their locations, to the SP; this may introduce a risk of privacy breach. The issue of privacy protection in MC should therefore be studied. Finally, because simulation-based evaluations generally cannot capture all factors that contribute to the problem in reality, there are also plans to conduct real experiments to further evaluate the effectiveness of the proposed mechanism.

ACKNOWLEDGEMENTS

This research was partially supported by the National Basic Research Program of China under grant nos 2011CB302605 and 2013CB329602, and the National Natural Science Foundation of China (NSFC) under grant nos 61202457, 61173144, 61402137, and 61402149, as well as the US National Science Foundation (NSF) under grant CNS-1065444, and the US Army Research Office under grant WF911NF-14-1-0518.

REFERENCES

1. Alt F, Shirazi AS, Schmidt A. Location-based crowdsourcing: extending crowdsourcing to the real world. In *Proceedings of the 6th IEEE Nordic Conference on Human-Computer Interaction (NordCHI 10)*, NY, USA, 2010.
2. Kazemi L, Shahabi C. Geocrowd: enabling query answering with spatial crowdsourcing. In *Proceedings of the 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL'12)*. NY, USA, 2012.
3. Du X, Guizani M, Xiao Y, *et al.* Secure and efficient time synchronization in heterogeneous sensor networks. *IEEE Transactions on Vehicular Technology* 2008; **57**(4): 2387–2394.
4. Abusalah L, Khokhar A, Guizani M. A survey of secure mobile ad hoc routing protocols[J]. *Communications Surveys & Tutorials, IEEE* 2008; **10**(4): 78–93.

5. Du X, Guizani M, Xiao Y, *et al.* A routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks. *IEEE Transactions on Wireless Communications* 2009; **8**(3): 1223–1229.
6. Du X, Xiao Y, Guizani M, *et al.* An effective key management scheme for heterogeneous sensor networks[J]. *Ad Hoc Networks* 2007; **5**(1): 24–34.
7. He Z, Cao J, Liu X. High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility. In *Proceedings of the 34th IEEE International Conference on Computer Communications (INFOCOM'15)*, Hong Kong, China, 2015.
8. Hachem S, Pathak A, Issarny V. Probabilistic registration for large-scale mobile participatory sensing. In *Proceedings of the 11th IEEE International Conference on Pervasive Computing and Communications (PERCOM'13)*, San Diego, California, 2013.
9. Deng D, Shahabi C, Demiryurek U. Maximizing the number of worker's self-selected tasks in spatial crowdsourcing. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL'13)*. Florida, USA, 2013.
10. Song Z, Zhang B, Liu C.H, *et al.* QoI-aware energy-efficient participant selection. In *Proceedings of the Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON'14)*, Singapore, 2014.
11. Zhang D, Xiong H, and Wang L, *et al.* CrowdRecruiter: selecting participants for piggyback crowdsensing under probabilistic coverage constraint. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'14)*, Washington, US, 2014.
12. Li Q, Cao G. Providing privacy-aware incentives for mobile sensing. *Pervasive computing and communications*. In *Proceedings of the 11th IEEE International Conference on Pervasive Computing and Communications (PERCOM'13)*, San Diego, California, 2013.
13. Niu B, Li Q, Zhu X, *et al.* Enhancing privacy through caching in location-based services. In *Proceedings of the 34th IEEE International Conference on Computer Communications (INFOCOM'15)*, Hong Kong, China, 2015.
14. To H, Ghinita G, Shahabi C. A framework for protecting worker location privacy in spatial crowdsourcing. *Proceedings of VLDB Endow* 2014; **7**(10): 919–930.
15. He D, Chan S, Guizani M. Privacy and incentive mechanisms in people-centric sensing networks[J]. *Communications Magazine, IEEE* 2015; **53**(10): 200–206.
16. He D, Chan S, Guizani M. User privacy and data trustworthiness in mobile crowd sensing[J]. *Wireless Communications, IEEE* 2015; **22**(1): 28–34.
17. Lee J.S, and Hoh B. Sell your experiences: a market mechanism based incentive for participatory sensing. In *Proceedings of the 8th IEEE International Conference on Pervasive Computing and Communications (PERCOM'10)*, Mannheim, Germany, 2010.
18. Yang D, Xue G, Fang X, and Tang J. Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing. In *Proceedings of the 18th Annual ACM/IEEE International Conference on Mobile Computing and Networks (MOBICOM'12)*, Istanbul, Turkey, 2012.
19. Liu T, Zhu Y. Social welfare maximization in participatory smartphone sensing. In *Proceedings of the 8th International Conference on Wireless Algorithms, Systems, and Applications (WASA'13)*. Zhangjiajie, China, 2013.
20. Luo T, Tan H.P, and Xia L. Profit-maximizing incentive for participatory sensing. In *Proceedings of the 33rd IEEE International Conference on Computer Communications (INFOCOM'14)*, Toronto, Canada, 2014.
21. Feng Z, Zhu Y, Zhang Q. TRAC: truthful auction for location-aware collaborative sensing in mobile crowdsourcing. In *Proceedings of the 33rd IEEE International Conference on Computer Communications (INFOCOM'14)*, Toronto, Canada, 2015.
22. Gao L, Hou F, Huang J. Providing long-term participation incentive in participatory sensing. In *Proceedings of the 34th IEEE International Conference on Computer Communications (INFOCOM'15)*, Hong Kong, China, 2015.
23. Neely MJ. Stochastic network optimization with application communication and queuing systems. *Synthesis Lectures on Communication Networks* 2010; **3**(1): 1–211.
24. Nisan N, Roughgarden T, Tardos E. *Algorithmic Game Theory*. Cambridge University Press: Cambridge, 2007.
25. Gao H, Tang J, and Liu H. Exploring social-historical ties on location-based social networks. In *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media (ICWSM'12)*, Dublin, Ireland, 2012.

AUTHORS' BIOGRAPHIES



Hongli Zhang received her BS degree in Computer Science from Sichuan University, Chengdu, China in 1994, and her PhD degree in Computer Science from Harbin Institute of Technology (HIT), Harbin, China, in 1999. She is a professor in Computer Science and Technology Department of HIT and the vice director of National Computer Information Content Security Key Laboratory. Her research interests include network and information security, network measurement and modeling, and parallel processing.



Zhikai Xu received the BS degree in Computer science from Dalian University of Technology (DUT), Dalian, China in 2009, and his MS degree in Computer Science from Harbin Institute of Technology, Harbin, China, in 2011. From 2011 to now, he has been working as a PhD candidate in Department of

Computer Science and Technology at Harbin Institute of Technology, Harbin, China. His research interests include Mobile Cloud computing and security.



Xiaojiang (James) Du is currently an associate professor in the Department of Computer and Information Sciences at Temple University. Dr. Du received his BE degree from Tsinghua University, China, in 1996, and his MS and PhD degrees from the University of Maryland, College Park in 2002 and 2003, respectively, all in Electrical Engineering.

His research interests are security, systems, wireless networks, and computer networks. Dr. Du has published over 140 journals and conference papers in these areas and has been awarded more than \$5M research grants from the US National Science Foundation and Army Research Office. He serves on the editorial boards of four international journals.



Zhigang Zhou received the BS and MS degree in Computer Science from Dalian Jiaotong University, Dalian, China, from 2004 to 2011. From 2011 to now, he has been working as a PhD candidate in Department of Computer Science and Technology at Harbin Institute of Technology (HIT), Harbin, China.

His research interests include Cloud computing and security.



Jiantao Shi received the BS, MS, and PhD degrees in Computer Science from Harbin Institute of Technology (HIT), Harbin, China, from 2000 to 2011. He is currently a lecturer in School of Computer Science and Technology in HIT. His research interests include Peer-to-Peer system, cloud computing, and information.