# Online Incentive Mechanism Design for Smartphone Crowd-sourcing

Ashwin Subramanian, G Sai Kanth, Sharayu Moharir and Rahul Vaze

***Abstract*—In this paper, we consider the problem of online incentive mechanism design for smart-phone crowd-sourcing. We consider the online setting where users arrive in a sequence and each user participating in crowd-sourcing submits a set of tasks it can accomplish and its corresponding bid. The platform then selects the users and their payments to maximize its utility while ensuring truthfulness, individual rationality, profitability, and polynomial algorithm complexity. The decision whether to accept or reject each user is made instantaneously, with no revocation. We propose an algorithm and show that it satisfies all the four desired properties of an *efficient* auction. Through extensive simulations, we evaluate the performance of our online algorithm.**

## I. Introduction

Crowd-sourcing using smart-phones is a new idea that has gained widespread interest [1]–[8]. Smart phones are equipped with multiple sensors that can be used to monitor key features of the surrounding environment, which help in improving the user experience or reducing human effort. Collectively using data derived from multiple smart phones (called crowd-sourcing) helps in improving the social welfare, e.g. helps public utility companies to track potholes locations, electricity failure, emergency relief operations, traffic congestion etc.

Several commercial applications like such as Sensorly [2], Nericell [1], Google voice recognition, and Apple's Siri use smart phone crowd-sourcing from users spread across multiple locations to improve their services. In some of these applications, users volunteer to share their data since as it helps in improving their own utility. This is, however, not true in general, and necessitates an incentive mechanism design, where users are externally incentivized in the form of payments for the data/tasks they are willing to share/perform.

In this paper, we consider the incentive mechanism design problem for smart phone crowd-sourcing by modeling it as a reverse auction. The platform announces a set of tasks that it wants to accomplish, and each user submits the list of tasks it is ready to perform, and the corresponding bid. The platform has a utility function associated with the set of tasks, and the problem is to find the set of users and their corresponding payments to maximize the system utility.

There are two basic paradigms for smart-phone crowd-sourcing, the offline setting and the online setting. In the offline setting, all users are present/active simultaneously, and send their profiles to the platform at the same time. In the online setting, users arrive sequentially and submit their profiles, and the platform must decide immediately whether to accept or reject the user and how much to pay the user. A decision once made, is irrevocable. For example, the offline scenario is applicable for current traffic congestion monitoring, while the online case is more suited for potholes tracking type of applications that are localized, where users pass over potholes in a given area sequentially. The online scenario is more general than the offline case, as all potentially participating users may not be active at the same time. Therefore, in this work, we focus on the online setting.

The platforms objective is to select the set of users and their payments to maximize its utility, subject to the following four requirements [13], [14]: *(i)* computational efficiency – the algorithm implemented by the platform has polynomial run time complexity, *(ii)* individual rationality – the selfish utilities of all users involved are non-negative, *(iii)* profitability – the platform utility is non-negative after the auction concludes, and *(iv)* truthfulness – no user has any incentive to bid differently from its true valuation.

The first three properties ensure that the proposed algorithm is feasible. Truthfulness makes the reverse auction free from market manipulation. It establishes that there is no incentive for users to manipulate their bids in the hope of higher individual profits.

### A. Related Work

An algorithm called M-Sensing, proposed in [8], is an incentive design mechanism smart-phone crowd-sourcing in the offline setting, and it works in a iterative manner. In each iteration, M-Sensing adds the user with maximum incremental utility in a greedy manner. It pays each user in the selected set, the maximum value which that user can bid and still be selected at some possible position in the greedy selection phase. More recently, in [15], both offline and online algorithms for sensing time schedules have been proposed for the crowd-sourcing problem. The proposed algorithms are shown to be truthful, and more importantly, analytical performance guarantees have been found on the performance of both offline and online algorithms in [15]. However, unlike our setting, [15] only considers a linear utility function. The problem of online incentive mechanism design for crowd-sourcing with strict budget constraints has been studied in [20]. The algorithms proposed in [20] are shown to be truthful, and performance guarantees have been provided for non-negative sub-modular utility functions under some additional assumptions on the user arrival process.

One way to ensure that the auction remains truthful is the VCG mechanism proposed in [9]. In a forward auction, the VCG mechanism charges each individual the harm it causes to other bidders in terms of the social welfare utility,

978-3-9018-8273-9/15/ ©2015 IFIP

while in a reverse auction, it pays each user an amount equal to the value contributed by the user to the auction [9]. However, the task for finding the winning set of users in a VCG mechanism is a combinatorial problem, and has exponential complexity as discussed in [10]. Therefore, for a computationally feasible operation like smart-phone crowd-sourcing, the VCG mechanism cannot be employed. Variations of VCG mechanism have been proposed [11], [12].

For the online smart-phone crowd-sourcing problem, we take motivation from the online $k$-secretary problem [16], [17]. In the online $k$-secretary problem, $N$ secretaries with arbitrary ranks arrive in a uniformly random order and the problem is to select the $k$ best ranked secretaries in an online manner. The best known algorithms for solving the online $k$-secretary problem reject the first $m = N/e$ secretaries, and generate a threshold set from the first $m$ secretaries, which is then used to select the $k$ best secretaries among the remaining $N - m$ secretaries.

### B. Contributions

1) We propose an online algorithm called SMART which improves on M-Sensing, and uses techniques from the VCG mechanism and the optimal solution to $k$-secretary problem, and prove that it satisfies the four desired properties, i.e., computational efficiency, individual rationality, profitability, and truthfulness.
2) The performance of an online algorithm is characterized by its competitive ratio [18], which is defined to be the ratio of the utility obtained by the online algorithm to that obtained by the optimal offline algorithm. We present extensive simulations results to characterize the competitive ratio of SMART as a function of various system parameters.

## II. System Model

We model the problem of smartphone crowd-sourcing as a reverse auction. The platform declares a set of tasks $\Gamma = \{t_1, t_2, t_3, \ldots, t_m\}$, where the value of a task $t_k$ to the platform is denoted by $\chi(t_k)$ (or $\chi_k$ for the sake of brevity). The cumulative value of the set of tasks $\tau \subseteq \Gamma$ is $\chi(\tau)$, where $\chi$ is any arbitrary combinatorial function.

The set of users is denoted by $U$, such that $|U| = n$. Users arrive one at a time and submit their profiles to the platform, where its profile contains a list of task the user can perform ($\tau_i$), and the user's bid ($b_i$). The platform must decide immediately whether to accept or reject the user and how much to pay the user. If chosen, the payment of user $i$ (denoted by $p_i$) should not be less than its bid.

Without loss of generality, we assume that every user in $U$, is such that the total value of all tasks it can perform ($v_i$) is greater than its bid.[1] We also assume that the user does not know its time of arrival with reference to the time of arrival of the other users. For a set of users $S \subseteq U$, the set of tasks performed by them are denoted by $\tau(S) = \bigcup_{i \in S} \tau_i$, and the value of $S \subseteq U$ is $v(S) = \chi(\tau(S))$. The marginal value of a user $i$ with respect to a set $S \subseteq U$ is denoted by $v_i(S)$, where, $v_i(S) = v(S \cup \{i\}) - v(S)$. The marginal utility of user $i$ is equal to the difference between the marginal value of user $i$ and its payment, i.e., $u_i(S) = v_i(S) - p_i$. The net marginal utility of user $i$, is denoted by $\sigma_i(S) = v_i(S \backslash \{i\}) - b_i$. The personal utility of a user $i$ is denoted by $\omega_i$. If selected, $\omega_i = p_i - c_i$, where $c_i$ is the cost incurred by user $i$, and zero otherwise. The utility of a set of users $S$ to the platform is denoted by $u(S)$, where $u(s) = v(S) - \sum_{i \in S} p_i$.

We define the problem of smartphone crowd-sourcing as:

*Problem 1:* Given a set $U$ of users,

$$\max_{T \subseteq U} u(T),$$

subject to the four properties of an efficient auction, namely, computational efficiency, individual rationality, profitability and truthfulness.

[1] If an incoming user has $v_i < b_i$, then it is rejected immediately and does not count as a user in $U$.

## III. Our Algorithm: SMART

We propose an algorithm called Search for Marginal Appropriate Replacement Tasks users (SMART). Motivated by the $k$-secretary problem, SMART consists of two phases, the *observation phase* (Phase I) and the *winner selection phase* (Phase II).

In Phase I, i.e, the observation phase, the algorithm rejects the first $k = \lfloor \frac{n}{c} \rfloor$ (where $c$ is a constant chosen by the platform) users on their arrival. It runs SMART-Phase I on the bidding profiles of $U[1 : k]$ and stores the output set of winning users as a reference set $R$. The algorithm uses this set $R$ as a reference for selection of the $n - k$ users remaining users, i.e., $U[n - k : n]$ in Phase II or the winner selection phase.

### A. SMART-Phase I: Algorithm Description

SMART-Phase I is inspired by the VCG mechanism, which is known to be truthful, but computationally expensive. SMART first shortlists the potential winners using an iterative greedy procedure by picking the user with the best marginal utility in each iteration. After the shortlisting process, each user in the shortlist is retained or dropped or replaced depending on the marginal utility that user brings to the platform. At a high level, the payment strategy of SMART is similar to VCG mechanism, where each user is paid for the marginal utility it brings to the platform.

SMART-Phase I consists of three sub-phases, the screening sub-phase followed by the winner selection sub-phase and finally the bad user removal sub-phase. In the winner selection sub-phase, the payment made to each winning user is also determined. We first describe the algorithm in words, followed by its formal description.

*1) **Screening Sub-Phase**:* In the screening sub-phase, the algorithm follows a greedy approach to select users. Starting from an empty set $S$, the algorithm iterates through $U \setminus S$ and selects the user with the maximum difference in marginal value and bid with respect to current set $S$, i.e., $\arg\max_{i \in U \setminus S}(v_i(S) - b_i)$. The selected user is added to $S$, and the process is repeated as long as there are users with a positive difference in marginal value and bid with respect to $S$.

*2) **Winner-Selection Sub-phase**:* This sub-phase takes the output of the screening sub-phase ($S$) as the input. Initially, the set of winning users $T$ is set as $S$, and the algorithm iterates through $T$ in the order in which users entered $S$ in the screening phase.

Recall that $v_i(T \setminus \{i\})$ is the marginal value of user $i$ given $T \setminus \{i\}$, and $b_i$ is the bid of user $i$, i.e., the minimum payment to be given to user $i$ if user $i$ is selected. By definition, the maximum possible marginal contribution of user $i$ to the overall utility is at most $\sigma_i(T) = v_i(T \setminus \{i\}) - b_i$.

**Case I: Negative marginal utility ($\sigma_i(T) < 0$).**

- **Remove User $i$.**
  For user $i \in S$, if $\sigma_i(T) < 0$, the overall utility increases by removing $i$ from $T$. Therefore, user $i$ is removed from the set of winning users, i.e., $T$ is updated to $T \setminus \{i\}$.
- **Replace.**
  To find a replacement for user $i$, the algorithm finds the user with the maximum marginal net utility, currently not in $T$, i.e., $j = \arg\max_{k \in U \setminus T} v_k(T) - b_k$. If $\sigma_j(T) > 0$, i.e., the overall utility can be increased by adding $j$ to $T$, $j$ is added to the set of winning users, i.e., $T$ is updated to $T \cup \{j\}$. The payment of user $j$ ($p_j$) is fixed to $v_j(T)$.

**Case II: Positive marginal utility ($\sigma_i(T) \geq 0$).**

If $\sigma_i(T) \geq 0$, the algorithm checks if there is a user $j \neq i$, currently not in $T$, such that replacing user $i$ with user $j$ increases overall utility. The candidate for replacing user $i$ is the user with the maximum marginal net utility, currently not in $T$, i.e., user $j$ such that $j = \arg\max_{k \in U \setminus T} v_k(T \setminus \{i\}) - b_k$.

- **Case II.a: Replace user $i$ with user $j$.**
  User $i$ is replaced with user $j$ if the following two conditions are satisfied.
  - C1: $\sigma_j(T) > 0$: Overall utility increases by adding user $j$ to $T \setminus \{i\}$.
  - C2: $\sigma_j(T) < \sigma_i(T)$: This condition implies that $b_i > v_i(T \setminus \{i\}) - \sigma_j(T)$, i.e., the bid of user $i$ is high enough to ensure that replacing user $i$ with user $j$ will lead to an increase in overall utility.

  If user $j$ satisfies both C1 and C2, replacing user $i$ with $j$ will increase overall utility. In this case, user $i$ is removed from the set of winning users and replaced with user $j$, i.e., $T$ is updated to $(T \setminus \{i\}) \cup \{j\}$. To determine the payment for user $j$, the algorithm finds user $j^*$ such that $j^* = \arg\max_{k \in U \setminus T} v_k(T \setminus \{j\}) - b_k$. If $v_{j^*}(T \setminus \{j\}) - b_{j^*} > 0$, the payment of user $j$, $p_j = v_j(T \setminus \{j\}) - v_{j^*}(T \setminus \{j\}) + b_{j^*}$; else $p_j = v_j(T \setminus \{j\})$.
- **Case II.b: Retain user $i$.**
  If C1 is not satisfied, i.e., if $\sigma_j(T) < 0$ for all $j \notin T$, the overall utility cannot be increased by adding any user currently not in $T$ to $T \setminus \{i\}$.
  If C2 is not satisfied, $b_i < v_i(T \setminus \{i\}) - \sigma_j(T)$, i.e., the bid of user $i$ is low enough to ensure that replacing user $i$ with user $j$ will not lead to an increase in overall utility. Therefore, if either C1 or C2 is not satisfied, user $i$ is retained in $T$. The payment of user $i$ ($p_i$) is fixed to $\min\{v_i(T), \beta_i\}$, where $\beta_i$ is the minimum value of the bid of user $i$ such that user $i$ enters the screening set $S$.

**SMART-Phase I**

```
// Screening Sub-Phase
S ← ∅, P ← {p_1, p_2, p_3, ..., p_n}
i ← arg max_{j∈U} (v_j(S) − b_j)
while v_i(S) > b_i and S ≠ U do
  S ← S ∪ {i}
  i ← arg max_{j∈U} (v_j(S) − b_j)
endwhile
for each i ∈ U do
  p_i ← 0
endfor
// Winner-Selection Sub-Phase
T ← S
for i = 1, 2, ..., |T| do
  (j, γ_i, β_j) ← Next Best User (i, U, T)
  σ_i(T) ← v_i(T\{i}) − b_i
  β_i ← User Entry Payment(U, i)
  // Case I - Positive marginal utility
  if σ_i(T) > 0 then
    // Pay next best user's bid
    if γ_i − b_i ≥ 0 and γ_i ≤ β_i and γ_i ≠ ∞
      p_i ← γ_i
    // Replace with next best user
    else if γ_i < b_i and γ_i ≤ β_i and γ_i ≠ ∞
      (T, p_j) ← Replace User (U, T, i, j)
    // Pay marginal value in S
    else if γ_i > β_i or γ_i = ∞
      p_i ← min(σ_i(T) + b_i, β_i)
    endif
  // Case 2 - Negative marginal utility
  else if σ_i(T) ≤ 0 and γ_i ≠ ∞
    (T, p_j) ← Replace User (U, T, i, j)
  else if σ_i(T) ≤ 0 and γ_i = ∞
    T ← T\{i}
  endif
endfor
// Bad-User Removal Sub-Phase
for each i in T do
  if u_i(T\{i}) ≤ 0 then
    T ← T\{i}
    p_i ← 0
  endif
endfor
Return (T, P)
```

**Next Best User** $(i, U, T)$

```
j ← arg max_{k∈U\T} v_k(T\{i}) − b_k, β_j ← ∞
if v_j(T\{i}) − b_j > 0 then
  γ_i ← v_i(T\{i}) − v_j(T\{i}) + b_j
else
  j ← −1, γ_i ← ∞
endif
Return (j, γ_i, β_j)
```

*3) **Bad User Removal Sub-phase**:* After the Winner-Selection sub-phase, the algorithm iterates through all elements of $T$, and users $\in T$ with non-positive marginal utility with respect to $T$ are removed from $T$. This ensures that only users with positive marginal utility are retained.

This marks the end of SMART-Phase I. At this point, the reference set $R$, which is used as a guideline in the next phase, i.e., Phase II, is initialized to $T$.

*Remark 3.1:* As mentioned before, SMART-Phase I, rejects the first $k$ users and determines the set of winning users among

**Replace User** $(U, T, i, j)$

$T \leftarrow (T \backslash \{i\}) \cup \{j\}$
$(k, \gamma_j, \beta_k) \leftarrow$ Next Best User $(j, U, T)$
**if** $\gamma_j < v_j(T \backslash \{j\})$ then
  $p_j = \gamma_j$
**else if** $\gamma_j \geq v_j(T \backslash \{j\})$
  $p_j = v_j(T \backslash \{j\})$
**endif**
Return $(T, p_j)$

**User Entry Payment** $(U, i)$

$S_i \leftarrow \emptyset$, $\beta_i \leftarrow 0$, $j \leftarrow \arg\max_{k \in U \backslash \{i\}} (v_k(S) - b_k)$
**while** $v_i(S_i) > b_i$ and $S_i \neq U \backslash \{i\}$ do
  $\beta_i \leftarrow \max(\beta_i, v_i(S_i) - v_j(S_i) + b_j)$
  $S_i \leftarrow S_i \cup \{j\}$;
  $j \leftarrow \arg\max_{k \in U \backslash \{i\}} (v_k(S) - b_k)$
**endwhile**
Return $\beta_i$

the first $k$ users in an offline manner. This set of winning users is used as a reference set for future decisions. The allocation (set of winning users and their payments) obtained on the first $k$ users by SMART-Phase I satisfies the following properties.

P1: SMART-Phase I is computationally efficient.
P2: SMART-Phase I is individually rational.
P3: SMART-Phase I is profitable.
P4: SMART-Phase I is truthful.

The proofs of the first two properties are straightforward and are omitted due to space constraints. The proofs of P3 and P4 are discussed in the Appendix.

### *B. SMART-Phase II: Algorithm Description*

The algorithm uses the output of Phase I, i.e., the set of winning users ($R$) as a reference for selection of users among the remaining $n-k$ users, denoted by $U[n-k:n]$. In the beginning of Phase II, i.e., the set of final winners ($W$) is empty, and users are processed as they arrive. We first describe the algorithm in words followed by a formal description. For each user $i \in U[n-k:n]$:

**Case I:** $|R| < m$, **where** $m = |\Gamma|$

– **Case I.a: Positive marginal utility** $(v_i(R) - b_i > 0)$
The overall utility can be increased by selecting user $i$. The algorithm adds user $i$ to both $R$ and $W$ and makes a payment $p_i = v_i(R)$.

– **Case I.b: Negative marginal utility** $(v_i(R) - b_i \leq 0)$
The algorithm determines if it is profitable to replace a user $j \in R$ with user $i$.
If there exists a user $j \in R$, such that, $b_i < v((R \backslash \{j\}) \cup \{i\}) - v(R) + b_j$, replacing user $j$ with user $i$ in $R$ increases the overall utility of the set $R$. Therefore, user $j$ is replaced with user $i$ in $R$, i.e., $R$ is updated to $R \backslash \{j\} \cup \{i\}$. User $i$ is added to the set of winners $W$, and the payment of user $i$, $p_i = v_i((R \backslash \{j\}) \cup \{i\}) - v(R) + b_j$. If no user $j \in R$ can be replaced profitably, user $i$ is rejected.

**Case II:** $|R| = m = |\Gamma|$
The presence of $m$ selected users implies that at least $m$, i.e. all tasks can be completed by the current set of winning users.

**SMART-Phase II**

**// Initialization**
$W \leftarrow \emptyset$, $R \leftarrow \emptyset$
$P \leftrightarrow \{p_1, p_2, \ldots p_n\}$
$n = |U|$, $k = \lfloor n/c \rfloor$
**// Initialize Reference Set**
$R \leftarrow$ set of winning users from Phase I
$p_1, p_2, \ldots p_n \leftarrow 0$
**// Online Winner Selection**
**for** each $i = k+1, k+2, \ldots, n$ and $|W| \leq m$ do
  **if** $|R| < m$ then
    **if** $v_i(R) - b_i > 0$ then
      $(R, W, P) \leftarrow$ Add User $(R, i)$
    **else**
      $(R, W, P) \leftarrow$ Try To Replace $(R, i)$
    **end if**
  **else if** $|R| = m$
    $(R, W, P) \leftarrow$ Try To Replace $(R, i)$
  **end if**
  $R \leftarrow$ Remove Bad Users $(R, W)$
**endfor**
Return $(W, P)$

**Add User** $(R, i)$

$p_i \leftarrow v_i(R)$
$R \leftarrow R \cup \{i\}$, $W \leftarrow W \cup \{i\}$
Return $(R, W, P)$

As a result, adding any more users to $R$ can only decrease the marginal utility of $R$.
Therefore, as in Case 1.b, the algorithm determines if any user $j \in R$ can be replaced by the current user. If no user can be replaced profitably, user $i$ is rejected.

Between the arrival of two users in Phase II, the algorithm iterates through $R \backslash W$, and removes any user $i$ with $v_i(R \backslash \{i\}) - b_i \leq 0$. This ensures that all users in $R$ have a positive difference in marginal value and bid, and consequently form a "good" reference for the incoming users. The algorithm terminates when either all users in $U$ have arrived or when $|W| = m$.

*Remark 3.2:* SMART satisfies the following properties.

P1: SMART is computationally efficient.
P2: SMART is individually rational.
P3: SMART is profitable.
P4: SMART is truthful.

The proofs are presented in the Appendix.

**Try To Replace** $(R, i)$

$j = \arg\max_{k \in R \backslash W} \Big( v(R \backslash \{k\}) \cup \{i\}) - v(R) + b_k - b_i \Big)$
**if** $v((R \backslash \{j\}) \cup \{i\}) - v(R) + b_j > b_i$ then
  $p_i \leftarrow v((R \backslash \{j\}) \cup \{i\}) - v(R) + b_j$
  $R \leftarrow (R \backslash \{j\}) \cup \{i\}$ and $W \leftarrow W \cup \{i\}$
**end if**
Return $(R, W, P)$

**Remove Bad Reference Users** $(R, W)$

**for** each $j \in R \backslash W$ do
  **if** $v(R) < v(R \backslash \{j\}) + b_j$ then
    $R \leftarrow R \backslash \{j\}$
  **end if**
**endfor**
Return $R$

### C. Discussion

SMART works by initially rejecting a few users, whose profiles are used by the SMART-Phase I algorithm to generate a reference set that is used to select/reject future users and to decide their respective payments. The reference set allows the platform to select users with positive marginal utility. Also, since in the online scenario we assumed that each user does not know when it arrives in relation to other users, SMART ensures truthfulness.

In general, the "goodness" of any online algorithm is measured by its competitive ratio, i.e. the ratio of the utility of the online algorithm with the utility of the offline algorithm. For the $k$-secretary problem, the competitive ratio is known to be $1-1/e$, where the first $1/e$ users are rejected, assuming that users arrive uniformly randomly. Analytically characterizing the competitive ratio of the SMART is very challenging, since one can compare the users that are selected with SMART and any offline algorithm, but not their payments, as SMART have to ensure user truthfulness. This does not allow any tractable analytical solution for finding the competitive ratio of SMART, and to understand its behavior with respect to any offline algorithm, we turn to extensive numerical simulations. Not surprisingly, it turns out it is optimal to approximately reject the first $k = 1/3$ users (similar to $k$-secretary problem), to get the best competitive ratio.

## IV. Simulation Results

In this section, we compare the performance of SMART to the performance obtained by running SMART-Phase I on the entire arrival sequence in an offline manner in order to characterize the competitive ratio of SMART.

In Figure 1, we plot the competitive ratio of the SMART as a function of the fraction of users observed in Phase I, i.e., $k$. We fix the number of users $n = 100$, and each user bids for 25% of the tasks randomly. Each bid is a random variable, uniformly distributed between 5 and 50. The number of tasks $m = 30$ and the value of each task is a random variable, uniformly distributed between 0 and 40. Figure 1 indicates that for this set of system parameters, observing 32% (around 1/e) of the total number of users, and using their profiles to form a reference set maximizes the competitive ratio of SMART.

Next, we study the variation in the competitive ratio of SMART as a function of the fraction of tasks each user can complete on average. The larger the fraction, the larger is the overlap in the tasks completed by different users. Consequently, users arriving in the first phase, i.e., the observation phase, have a lot of common tasks with the users arriving in selection phase, thereby allowing SMART to make "good" user selections. It is reasonable to expect that the competitive ratio would increase as the average number of tasks that each user completes increases. This hypothesis is confirmed via simulations, as shown in Figure 2. We plot the competitive ratio of SMART as a function of the fraction of tasks completed by each user. For each data-point, we empirically optimize the value of $k$, and plot the maximum competitive ratio.

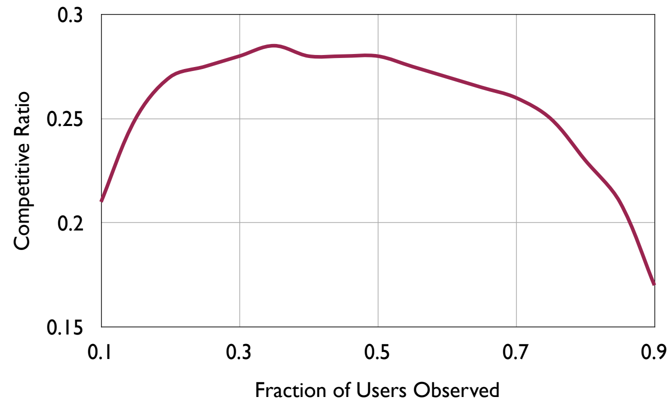


Fig. 1. Variation of the competitve ratio with the fraction of users observed.

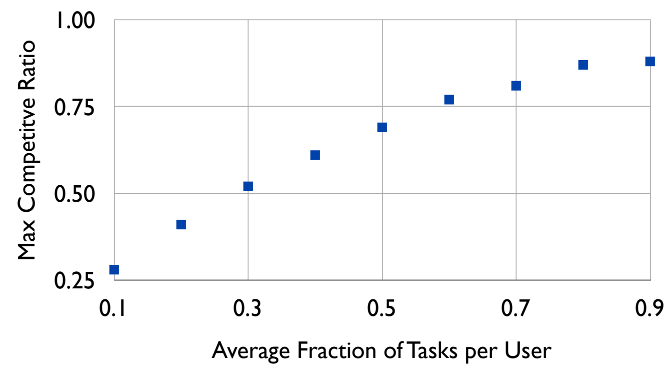


Fig. 2. Variation of the competitive ratio with the fraction of total tasks completed by the average user

Next, the question we try to answer is: given that we know that each user completes a certain fraction of tasks on average, what fraction of users should SMART observe, i.e., when should the observation phase end in order to maximize the competitive ratio. In Figure 3, for each value of the fraction of tasks each user can complete on average, we plot the fraction of users observed by SMART which leads to the maximum competitive ratio.

In Figure 3, we observe that the fraction of users that need to be observed by SMART has its maximum when each user can complete 30% of the total tasks on average. This behavior is quite intuitive since, while choosing a certain fraction of users to observe, there is a tradeoff between the quality of the reference set and the number of users left for selection. If the fraction of users observed is low, the reference set formed in observation phase will be of low quality and hence, the selection of winning users based on the reference set by SMART will be poor. If the fraction of users observed is high, the reference set formed in the observation phase will be of high quality, however, there will be very few users left for selection.

## References

[1] "Nericell," in *http://research.microsoft.com/en-us/projects/nericell/*.
[2] "Sensorly," in *www.sensorly.com*.

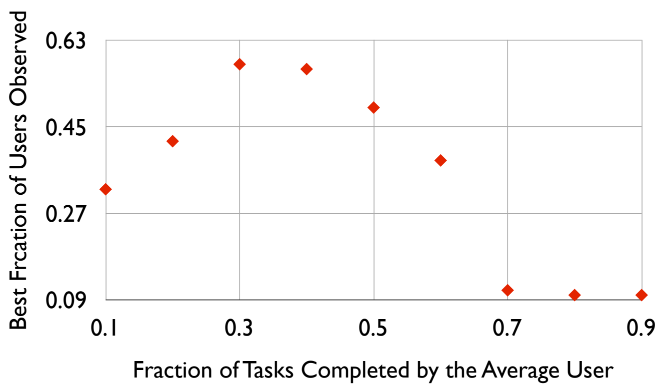


Fig. 3. Variation of the fraction of users in the observation phase with the fraction of total tasks completed by an average user

[3] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: using a mobile sensor network for road surface monitoring," in *Proceedings of the 6th international conference on Mobile systems, applications, and services*. 2008, pp. 29–39.

[4] 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*. 2010, pp. 105–116.

[5] T. Yan, M. Marzilli, R. Holmes, D. Ganesan, and M. Corner, "mcrowd: a platform for mobile crowdsourcing," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. 2009, pp. 347–348.

[6] X. Sheng, J. Tang, and W. Zhang, "Energy-efficient collaborative sensing with mobile phones," in *Proceedings of IEEE INFOCOM, 2012*. 2012, pp. 1916–1924.

[7] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Walrand, "Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing," in *Proceedings of IEEE INFOCOM, 2012*. 2012, pp. 1701–1709.

[8] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *Proceedings of the 18th ACM Annual International Conference on Mobile Computing and Networking*. 2012, pp. 173–184.

[9] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders," *The Journal of Finance*, vol. 16, no. 1, pp. 8–37, 1961. [Online]. Available: http://dx.doi.org/10.1111/j.1540-6261.1961.tb02789.x

[10] V. Conitzer and T. Sandholm, "Failures of the vcg mechanism in combinatorial auctions and exchanges," in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. 2006, pp. 521–528.

[11] D. Parkes and S. Singh, "An mdp-based approach to online mechanism design," in *Advances in neural information processing systems*, 2003, p. None.

[12] S. De Vries and R. V. Vohra, "Combinatorial auctions: A survey," *INFORMS Journal on computing*, vol. 15, no. 3, pp. 284–309, 2003.

[13] D. Garg, Y. Narahari, and S. Gujar, "Foundations of mechanism design: A tutorial part 1-key concepts and classical results," in *Sadhana (Academy Proceedings in Engineering Sciences)*, vol. 33, no. 2. Indian Academy of Sciences, 2008, pp. 83–130.

[14] ——, "Foundations of mechanism design: A tutorial part 2-advanced concepts and results," *Sadhana*, vol. 33, no. 2, pp. 131–174, 2008.

[15] K. Han, C. Zhang, and J. Luo, "Truthful Scheduling Mechanisms for Powering Mobile Crowdsensing," *ArXiv e-prints*, Aug. 2013.

[16] R. Kleinberg, "A multiple-choice secretary algorithm with applications to online auctions," in *Proceedings of the sixteenth annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2005, pp. 630–631.

[17] M. Babaioff, N. Immorlica, D. Kempe, and R. Kleinberg, "A knapsack secretary problem with applications," *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pp. 16–28, 2007.

[18] A. Borodin and R. El-Yaniv, *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[19] R. B. Myerson, "Optimal auction design," *Mathematics of operations research*, vol. 6, no. 1, pp. 58–73, 1981.

[20] D. Zhao and X. Li and H Ma. "How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint," in *Proceedings of IEEE INFOCOM, 2012*

## APPENDIX

### A. Properties of SMART-Phase I

To show that SMART-Phase I is profitable, we first show that the utility of SMART-Phase I is at least as much as that of M-Sensing, and the result follows since M-Sensing [8] is profitable.

*Lemma 1:* The utility of SMART-Phase I is greater than or equal to the utility of M-Sensing [8].

*Proof:* The output set $S$ of the screening phase of SMART-Phase I is equal to the final selection set of M-Sensing. SMART-Phase 1 then refines $S$ further through the Winner Selection sub-phase and the Bad User Removal sub-phase. In the winner selection sub-phase, SMART-Phase I initially sets the set of winning users $T$ to be equal to $S$ and then iteratively updates $T$. Moreover, note that with M-Sensing, and SMART-Phase 1, the payment for each user $i \in S$, is $\beta_i$, and $p_i = \min\{\gamma_i, \sigma_i(T) + b_i, v_i\beta_i\}$, respectively. Thus, $p_i \leq \beta_i$.

So initially, $v(S) - \sum_{l\in S}\beta_l = v(T) - \sum_{l\in T}\beta_l$, since $T = S$. In each iteration of the winner selection sub-phase of SMART-Phase 1 the current user, say user $i$ is either retained, removed or replaced.

If the current user $i$ is *retained* there is no change in $T$, and still $T = S$, however, since $p_i \leq \beta_i$,

$$v(S) - \sum_{l\in S}\beta_l \leq v(T) - p_i - \sum_{l\in S\backslash\{i\}}\beta_l. \tag{1}$$

The current user $i$ is *removed* if and only if $\sigma_i(T) - b_i \leq 0$. Therefore, $v_i(T\backslash\{i\}) - b_i \leq 0$, which implies $v(T) - v(T\backslash\{i\}) - b_i \leq 0$, and finally, $v(T\backslash\{i\}) \geq v(T) - b_i$.

Since M-Sensing is rational, $\beta_i \geq b_i$, we have $v(T\backslash\{i\}) \geq v(T) - \beta_i$.

Therefore, after removing user $i$, with $T \leftarrow T\backslash\{i\}$,

$$v(S) - \sum_{l\in S}\beta_l \leq v(T) - \sum_{l\in T}\beta_l. \tag{2}$$

The *replacement* of the current user $i$ can occur at two points in the SMART-Phase I algorithm, in lines 23, 24 and in lines 30, 31. In both cases, the Replace User function is called to replace user $i$ with another user say user $j \in U\backslash T$. Let the set of winning users be $T_{\text{old}}$ before replacement and $T_{\text{new}}$ after replacement. $T_{\text{new}} = (T_{\text{old}}\backslash\{i\}) \cup \{j\}$.

Consider the replacement done in lines 23 and 24. In this situation $\gamma_i \leq b_i$, $\gamma_i \leq \beta_i$ and $\gamma_i \neq \infty$. The Replace User function first performs replacement and then calls the Next Best User function with inputs as $\{j, U, T_{\text{new}}\}$. The Next Best User function finds $k = \arg\max_{m\in U\backslash T_{\text{new}}} v_m(T_{\text{new}}\backslash\{j\}) - b_m$ and returns $\gamma_j = v_j(T_{\text{new}}\backslash\{j\}) - v_k(T_{\text{new}}\backslash\{j\}) + b_k$. Since $i \in U\backslash T_{\text{new}}$, therefore, $v_k(T_{\text{new}}\backslash\{j\}) - b_k \geq v_i(T_{\text{new}}\backslash\{j\}) - b_i$.

This implies that in the worst case, the replacement user $k$ is same as user $i$. Therefore, in the worst case $\gamma_j = v_j(T_{\text{new}}\backslash\{j\}) - v_i(T_{\text{new}}\backslash\{j\}) + b_i$. Since the payment made in the Replace User function is $p_j = \min(\gamma_j, v_j(T_{\text{new}}\backslash\{j\}))$, $p_j \leq \gamma_j$. Therefore,

$p_j \leq v_j(T_{\text{new}}\backslash\{j\}) - v_i(T_{\text{new}}\backslash\{j\}) + b_i$, which on rearranging, $v_i(T_{\text{new}}\backslash\{j\}) - b_i \leq v_j(T_{\text{new}}\backslash\{j\}) - p_j$. Since $T_{\text{new}}\backslash\{j\} = T_{\text{old}}\backslash\{i\}$, it follows that $v_i(T_{\text{old}}\backslash\{i\}) - b_i \leq v_j(T_{\text{new}}\backslash\{j\}) - p_j$.

As M-Sensing is individually rational, $\beta_i \geq b_i$, therefore, $v_i(T_{\text{old}}\backslash\{i\}) - \beta_i \leq v_j(T_{\text{new}}\backslash\{j\}) - p_j$.

Consider the replacement done in lines 30 and 31. In this situation $\sigma_i(T_{\text{old}}) \leq 0$ and $\gamma_i \neq \infty$. Since, $\gamma_i \neq \infty$, $v_j(T_{\text{old}}\backslash\{i\}) - b_j \geq 0$. Since, $\sigma_i(T_{\text{old}}) \leq 0$, $v_i(T_{\text{old}}\backslash\{i\}) - b_i \leq 0$. As M-Sensing is individually rational, $\beta_i \geq b_i$ therefore, $v_i(T_{\text{old}}\backslash\{i\}) - \beta_i \leq 0$.

Also, since the payment made in the replace user function $p_j = \min(\gamma_j, v_j(T_{\text{new}}\backslash\{j\}))$, $p_j \leq v_j(T_{\text{new}}\backslash\{j\})$. Hence, $v_j(T_{\text{new}}\backslash\{j\}) - p_j \geq 0$.

Therefore, $v_i(T_{\text{old}}\backslash\{i\}) - \beta_i \leq v_j(T_{\text{new}}\backslash\{j\}) - p_j$.

Consider the first iteration of SMART-Phase 1 in the winner selection phase for replacement. Here, $S = T_{\text{old}}$ and $T_{\text{new}}\backslash\{j\} = S\backslash\{i\}$. Therefore, $v_i(S\backslash\{i\}) - \beta_i \leq v_j(T_{\text{new}}\backslash\{j\}) - p_j$, which implies $v(S) - v(S\backslash\{i\}) - \beta_i \leq v(T_{\text{new}}) - v(T_{\text{new}}\backslash\{j\}) - p_j$. Since $v(T_{\text{new}}\backslash\{j\}) = v(S\backslash\{i\})$, $v(S) - \beta_i \leq v(T_{\text{new}}) - p_j$. Subtracting $\sum_{l\in S\backslash\{i\}} \beta_l$ to both sides, we get

$$v(S) - \sum_{l\in S} \beta_l \leq v(T_{\text{new}}) - p_j - \sum_{l\in S\backslash\{i\}} \beta_l. \quad (3)$$

Now consider the situation when $n$th replacement occurs. Let the set of winning users before replacement be $T_{\text{old}_n}$ and let the set of winning users after replacement be $T_{\text{new}_n}$. Therefore,

$$T_{\text{old}_n} = (S\backslash\{i_1, \ldots, i_{n-1}\}) \cup \{j_1, \ldots, j_{n-1}\}), \quad (4)$$
$$T_{\text{new}_n} = (S\backslash\{i_1, \ldots, i_n\}) \cup \{j_1, \ldots, j_n\}). \quad (5)$$

Generalizing (3) at the $n$th iteration, we get, $v(S) - \sum_{l\in S} \beta_l$

$$\leq v(T_{\text{new}_n}) - \sum_{m\in\{j_1,\ldots,j_n\}} p_j - \sum_{l\in S\backslash\{i_1,\ldots,i_n\}} \beta_l. \quad (6)$$

Let (6) hold true. Lets analyze the $n+1^{th}$ iteration.

$$T_{\text{old}_{n+1}} = (S\backslash\{i_1, \ldots, i_n\}) \cup \{j_1, \ldots, j_n\}), \quad (7)$$
$$T_{\text{new}_{n+1}} = (S\backslash\{i_1, \ldots, i_{n+1}\}) \cup \{j_1, \ldots, j_{n+1}\}) \quad (8)$$

At the $n+1^{\text{th}}$ replacement,

$$v_{i_{n+1}}(T_{\text{old}_{n+1}}\backslash\{i_{n+1}\}) - \beta_i \leq v_{j_{n+1}}(T_{\text{new}_{n+1}}\backslash\{j_{n+1}\}) - p_j.$$

Therefore, $v(T_{\text{old}_{n+1}}) - v(T_{\text{old}_{n+1}}\backslash\{i_{n+1}\}) - \beta_i \leq v(T_{\text{new}_{n+1}}) - v(T_{\text{new}_{n+1}}\backslash\{j_{n+1}\}) - p_j$. Since, $T_{\text{old}_{n+1}}\backslash\{i_{n+1}\} = T_{\text{new}_{n+1}}\backslash\{j_{n+1}\}$, $v(T_{\text{old}_{n+1}}) - \beta_i \leq v(T_{\text{new}_{n+1}}) - p_j$.

From (5) and (7), we note that $T_{\text{old}_{n+1}} = T_{\text{new}_n}$. Substituting for $T_{\text{new}_n}$ from (6), we get, $v(S) - \sum_{l\in S} \beta_l - \beta_{i_{n+1}} \leq v(T_{\text{new}_{n+1}}) - p_{j_{n+1}} - \sum_{m\in\{j_1,\ldots,j_n\}} p_j - \sum_{l\in S\backslash\{i_1,\ldots,i_n\}} \beta_l$. Hence, $v(S) - \sum_{l\in S} \beta_l$

$$\leq v(T_{\text{new}_{n+1}}) - \sum_{m\in\{j_1,\ldots,j_{n+1}\}} p_j - \sum_{l\in S\backslash\{i_1,\ldots,i_{n+1}\}} \beta_l. \quad (9)$$

Therefore, from (3), (6), and (9), using induction we get that, $v(S) - \sum_{l\in S} \beta_l$

$$\leq v(T_{\text{new}_n}) - \sum_{m\in\{j_1,\ldots,j_n\}} p_j - \sum_{l\in S\backslash\{i_1,\ldots,i_n\}} \beta_l, \quad (10)$$

is true for any arbitrary $n$.

In the winner selection sub-phase, SMART-Phase 1 iterates through all the elements of $S$ in a sequential manner. Using (1), (2), and, (10), we can conclude that at the end of the winner selection phase, $v(S) - \sum_{l\in S} \beta_l \leq v(T) - \sum_{l\in T} p_l$, where the LHS represents the utility of M-Sensing and the RHS represents the utility of SMART-Phase 1, since the Bad User Removal function does not decrease the utility. ■

Next, we show the most important property of SMART-Phase I, its truthfulness. Towards that end we will use the Myerson's Theorem [19].

*Theorem 1:* [19] A reverse auction is considered truthful if and only if

- The selection rule is monotone. If a user $i$ wins the auction by bidding $b_i$, it would also win the auction by bidding an amount $b'_i$, where $b'_i < b_i$.
- Each winner is paid a critical amount. If a winning user submits a bid greater than this critical value, it will not get selected.

*Lemma 2:* SMART-Phase I is truthful.

*Proof:* We prove this lemma by showing that SMART-Phase I satisfies the two properties of Theorem 1.

*Monotonicity of SMART-Phase I:* Consider a user $i$ that is selected by SMART-Phase I with bid $b_i$, i.e., $i \in T$. Let user $i$ change its bid to $b'_i$, where $b'_i < b_i$. Let $\sigma'_i(T) = v_i(T\backslash\{i\}) - b'_i$. Now we look at two cases where the user $i$ with bid $b_i$ could have entered $T$.

a) Assume that user $i \in S$, in the user screening phase, with bid $b_i$, and was then retained in $T$. By the definition of the User Screening phase, if $i \in S$, then $i = \arg\max_{k\in U\backslash S} v_k(S) - b_k$ at some iteration $r_i$, where user $i$ entered $S$. If user $i$ instead bid $b'_i < b_i$, then again user $i$ enters $S$ at iteration $r_i$ or earlier. Since user $i$ is retained in $T$ with bid $b_i$, by definition of SMART-Phase 1, we have $\sigma_i(T) > 0$ and $\gamma_i > b_i$. Consequently, $\sigma'_i(T) > 0$ and $\gamma_i > b'_i$. Thus, even if user $i$ bid $b'_i$, both Cond. 1 and Cond. 1.1 in the Winner Selection phase are satisfied, and therefore user $i$ is retained in $T$.

b) If user $i$ entered into $T$ by replacing some user $j \in T$, then user $i$ has to satisfy $i = \arg\max_{k\in U\backslash T} v_k(T\backslash\{j\}) - b_k$ in the Next Best User function when called from Line 14 of the SMART-Phase 1. Therefore, should user $i$ decrease its bid to $b'_i$ it would still replace user $j \in T$ and enter $T$.

*Existence of a Critical Bid Amount with SMART-Phase 1:* We claim that the payment $p_i$ made by SMART-Phase 1 is critical, i.e., if any user $i$ bids in excess of its critical amount

$p_i$, then SMART-Phase 1 will not select it. Two possible conditions exist with SMART-Phase 1,

*Case I*: A winning user $i \in S, i \in T$ receives $p_i = \min(v_i(T\backslash\{i\}), \beta_i, \gamma_i)$, where $S$ is the set of screened users. Let's assume that $p_i = v_i(T\backslash\{i\})$, and if user $i$ changes its bid to $b'_i$, $b'_i > v_i(T\backslash\{i\})$, then $\sigma'_i(T) = v_i(T\backslash\{i\}) - b'_i < 0$. From Cond 2 of SMART-Phase 1, we conclude that user $i$ would be replaced by another user $j$ if possible in $T$ or removed from $T$. If $p_i = \beta_i$ and $b'_i > \beta_i$, then by the definition of the User Entry Payment function, user $i$ will not enter the screening set $S$ in the User Screening phase. Therefore, user $i$ will not enter $T$. Finally, if $p_i = \gamma_i$ and $b'_i > \gamma_i$, then from Cond 1.2 user $i$ will be replaced by some user $j$ in $U\backslash T$.

*Case II*: A winning user $i \in T$ and $i \notin S$ receives $p_i = \min(v_i(T\backslash\{i\}), \gamma_i)$. Such a user does not belong to $T$ at the beginning of the Winner Selection Phase and is a replacement user. Note that, replacement users are paid inside the Replace User Function. In the function a replacement user say $j \notin S$ replaces a user $i \in T$. Let $T$ before the replacement be $T_{\text{old}}$, and after the replacement be $T_{\text{new}} = T \cup \{j\}\backslash\{i\}$. The Replace User function calls the Next Best User function to compute $\gamma_j$ for user $j$. This function returns both the second best user to user $j$, user $k$ and the value of $\gamma_j$.

Lets assume that user $j$ is paid an amount equal to $\gamma_j$. From the Next Best User Function $\gamma_j = v_j(T_{\text{new}}\backslash\{j\}) - v_k(T_{\text{new}}\backslash\{j\}) + b_k$. If user $j$ bids an amount $b'_j > \gamma_j$, then the following happens. Since $T_{\text{new}}\backslash\{j\} = T_{\text{old}}\backslash\{i\}$ from line 1 of the Next Best User function we can conclude that user $k$ would take user $j$'s place as a replacement user at line 14 of SMART-Phase 1.

Consider the alternate case when user $j$ is paid an amount equal to $v_j(T_{\text{new}}\backslash\{j\})$. Note that user $j$ was selected as a replacement for user $i$, through the Next Best User function call made in line 14 of SMART-Phase 1 for the computation of $\gamma_i$ for user $i$. At this point the set of winning users is $T_{\text{old}}$. If user $j$ bids $b'_j > v_j(T_{\text{new}}\backslash\{j\})$, then since $T_{\text{new}}\backslash\{j\} = T_{\text{old}}\backslash\{i\}$, $v_j(T_{\text{old}}\backslash\{i\}) = v_j(T_{\text{new}}\backslash\{j\})$, then from line 2 of the Next Best User function called at line 14 of SMART-Phase 1, since $b'_j > v_j(T_{\text{new}}\backslash\{j\})$, the Next Best User function either returns some other user $k$ as a replacement for user $i$ or if the user $j$ is the maximizer in line 1 of the Next Best User function, then $j = -1$ is returned. In conclusion, user $j$ does not replace user $i$ if $b'_j > v_j(T_{\text{new}}\backslash\{j\})$. ■

### B. Properties of SMART

*Lemma 3:* SMART is computationally efficient.

*Proof:* It can be shown that the computational complexity associated with SMART is $O(nm^3)$. Hence, SMART is computationally efficient. ■

*Lemma 4:* SMART is individually rational.

*Proof:* To prove that SMART is individually rational, it is sufficient to prove that $\forall_{i \in W}\ p_i \geq b_i$. In Phase II, a user enters the final winner's set $W$ through either the Add User function or the Try To Replace function. The Add User function is called when the incoming user $i$ has $v_i(R) > b_i$. Since the Add User function pays user $i$, $p_i = v_i(R)$, $p_i > b_i$. An incoming user $i$ replaces an existing user $j \in R\backslash T$ through the Try To Replace function if and only if $v((R\backslash\{j\}) \cup \{i\}) - v(R) + b_j - b_i > 0$. Since the payment made in the Try To Replace function to such a selected user $i$ is $p_i = v((R\backslash\{j\}) \cup \{i\}) - v(R) + b_j$, therefore $p_i > b_i$. ■

*Lemma 5:* SMART is profitable.

*Proof:* An incoming user $i \in U[k+1:n]$ enters $W$ through either the Add User function or the Try To Replace function.

Let user $i$ enter $W$ through the Add User function. At any point in the algorithm the set of winning users ($W$) is always a subset of the reference set ($R$), that is $W \subseteq R$. This implies that for a user $i \in U\backslash R$, the marginal tasks with respect to $R$ are always a subset of the marginal tasks with respect to $W$, i.e. $\tau_i(R) \subseteq \tau_i(W)$. Therefore, $\chi(\tau_i(R)) \leq \chi(\tau_i(W))$. The payment made to user $i$ is $p_i = v_i(R) = \chi(\tau_i(R))$. Hence, $v_i(W) - p_i = \chi(\tau_i(W)) - \chi(\tau_i(R)) \geq 0$. The platform utility of $W \cup \{i\}$ is $u(W \cup \{i\}) = v(W) + v_i(W) - p_i - \sum_{j \in W} p_i = u(W) + v_i(W) - p_i$. Therefore $u_i(W) = u(W \cup \{i\}) - u(W) \geq 0$, i.e. the incremental utility change is always non-negative.

Let user $i$ enter $W$ through the Try To Replace function. The payment made to user $i$ is $p_i = v((R\backslash\{j\}) \cup \{i\}) - v(R) + b_j = \chi(\tau_i(R\backslash\{j\})) + v(R\backslash\{j\}) - v(R) + b_j$. The increase in the utility of the platform is $u(W \cup \{i\}) - u(W) = u_i(W)$, where $u_i(W) = v_i(W) - p_i = \chi(\tau_i(W)) - \chi(\tau_i(R\backslash\{j\})) + v(R) - v(R\backslash\{j\}) - b_j$. Since user $j \in R\backslash W$, $W \subseteq R\backslash\{j\}$. Therefore, $\chi(\tau_i(W)) - \chi(\tau_i(R\backslash\{j\})) \geq 0$. Further the Remove Bad Reference Users function ensures that for any $j \in R$, $v(R) - v(R\backslash\{j\}) - b_j \geq 0$. Therefore $u_i(W) \geq 0$.

Since at the start of the Selection phase $u(W) = 0$ and the addition of a user $i$ to $W$ through the Add User function or the Replace User function yields $u_i(W) \geq 0$, ONLINE-SMART is profitable. ■

*Lemma 6:* SMART is truthful.

*Proof:* We use Theorem 1, to prove this Lemma. Let us assume that user $i$ enters the final winners set $W$ with bid $b_i$.

Let user $i$ change its bid to $b'_i$ with $b'_i < b_i$. If user $i$ entered $W$ through the Add User function, then $v_i(R) - b_i > 0$. Since $v_i(R) - b'_i > 0$ user $i$ enters $W$. If user $i$ entered $W$ through the Try To Replace Function then for some $j \in R\backslash W$, $v((R\backslash\{j\}) \cup \{i\}) - v(R) + b_j - b_i > 0$. Further user $i$ replaces user $j$ in $R$. Since $v((R\backslash\{j\}) \cup \{i\}) - v(R) + b_j - b_i > 0$, user $i$ still replaces user $j$ in $R$ and enters $W$. Hence the selection rule of SMART-Phase II is monotone.

Let user $i$ change its bid to $b'_i$ with $b'_i > p_i$. If user $i$ entered $W$ through the Add User function then $p_i = v_i(R)$. Since $b'_i > v_i(R)$, $v_i(R) - b'_i < 0$ and user $i$ no longer enters $W$. If user $i$ entered $W$ through the Try To Replace Function, then it replaces some $j \in R\backslash W$ and is paid $p_i = v((R\backslash\{j\}) \cup \{i\}) - v(R) + b_j$. Since $v((R\backslash\{j\}) \cup \{i\}) - v(R) + b_j - b'_i < 0$, user $i$ no longer replaces user $j$ in $R$ and consequently it does not enter $W$. Hence the payment made by SMART-Phase II is critical. ■