Truthful User Recruitment for Cooperative Crowdsensing Task: A Combinatorial Multi-Armed Bandit Approach

Hengzhi Wang, Yongjian Yang, En Wang, Wenbin Liu, Yuanbo Xu, and Jie Wu, *Fellow, IEEE*

Abstract—Mobile Crowdsensing (MCS) is a promising paradigm that recruits users to cooperatively perform a sensing task. When recruiting users, existing works mainly focus on selecting a group of users with the best objective ability, e.g., the user's probability or frequency of covering the task locations. However, we argue that the task completion effect depends not only on the user's objective ability, but also on their subjective collaboration likelihood with each other. Furthermore, even though we can find a well-behaved group of users in the single-round scenario, while in the multi-round scenario without enough prior knowledge, we still face the problem of recruiting previously well-behaved user groups (exploitation) or recruiting uncertain user groups (exploration). Additionally, we consider that each user has a different cost, and the platform recruits users under a cost budget; thus, the problem becomes more challenging: users may report fake costs to gain more profits. To address these problems, assuming that the user's information is known, we first convert the single-round user recruitment problem into the min-cut problem and propose a graph theory based algorithm to find the approximate solution. Then, in the multi-round scenario where the user's information is estimated from the previous rounds, to balance the trade-off between exploration and exploitation, we propose the multi-round User Recruitment strategy under the budget constraint based on the combinatorial Multi-armed Bandit model (URMB), which is proven to achieve a tight regret bound. Next, we propose a graph-based payment strategy to achieve truthfulness and individual rationality of users. Finally, extensive experiments on three real-world datasets show that URMB always outperforms the state-of-th-art strategies.

Index Terms—Mobile crowdsensing, collaboration likelihood, combinatorial multi-armed bandit, min-cut, truthful

1 Introduction

With the proliferation of smart devices equipped with powerful sensors, Mobile CrowdSensing (MCS) has attracted much attention [2]. It recruits distributed mobile users to cooperatively perform various sensing tasks [3], [4] such as air quality monitoring [5], traffic mapping construction [6], and target tracking [7]. However, due to the budget constraint, we try our best to select a group of effective users, which raises the fundamental user recruitment problem in MCS.

Existing user recruitment works mainly focus on selecting a group of users with the best objective ability, e.g., the user's probability or frequency of covering the task locations [3], [8]. However, in this paper, we argue that, for the MCS task, the completion effect depends not only on the user's objective ability, but also on its subjective collaboration likelihood with each other. For example, a real-time target tracking task [9] requires a group of users to track a target such as a moving vehicle by taking photographs. Due to the real-time mobility of the target, when a user finds the target and takes photographs, the user has to immediately send the target information to other cooperative users to help them effectively track the target in the next location, which inevitably exposes the user's sensitive spatiotemporal information. Thus, due to the privacy concerns [10], [11], social relationship [12], and task attribute [13], a user may be unwilling to cooperate with certain users, e.g., they are not familiar or familiar but have a bad relationship. In this case, a user may hesitate to upload the target location when finding the target, which results in the bad task completion effect. Therefore, when recruiting users for these cooperative tasks, we consider not only their objective abilities but also the subjective collaboration likelihood with each user.

An example of the user recruitment in this paper is shown in Fig. 1, where the whole process is divided into K rounds. Three users (u_1, u_2 and u_3) are moving around the locations of the sensing task. At the beginning of each round, users are selected to perform the task, and their objective abilities as shown in Fig. 1 (the left part) are defined as the frequency that they pass through the locations of the sensing task. Additionally, the subjective collaboration likelihood is shown in Fig. 1 (the central part),

- Hengzhi Wang, Yongjian Yang, En Wang, Wenbin Liu, and Yuanbo Xu are with the Department of Computer Science and Technology and Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, Jilin 130012, China. E-mail: {wanghz17, liuwb16}@mails.jlu.edu.cn, {yyj, wangen, yuanbox}@jlu.edu.cn.
- Jie Wu is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA. E-mail: jiewu@temple.edu.

Manuscript received 8 Nov. 2020; revised 30 Jan. 2022; accepted 8 Feb. 2022. Date of publication 23 Feb. 2022; date of current version 5 June 2023.
This work was supported in part by the National Key R&D Program of China under Grants 2021ZD0112501 and 2021ZD0112502, in part by the National Natural Science Foundations of China under Grants 61772230, 61972450, 62072209, and 62102161, in part by the CCF-Baidu Open Fund under Grant 2021PP15002000, and in part by NSF under Grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, CNS 1651947, and CNS 1564128.
(Corresponding author: En Wang.)
Recommended for acceptance by R. Zheng.
Digital Object Identifier no. 10.1109/TMC.2022.3153451

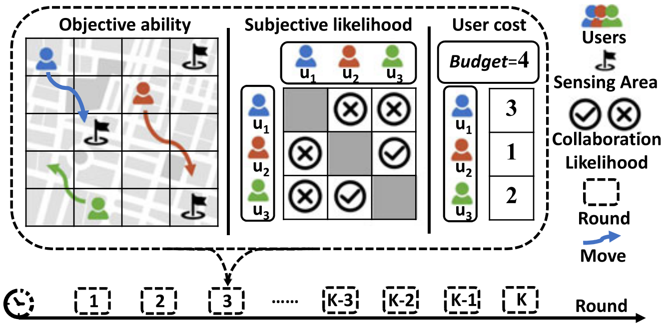


Fig. 1. An example to illustrate the multi-round user recruitment problem taking both objective ability and subjective collaboration likelihood into consideration.

which illustrates that u_2 and u_3 are willing to cooperate with each other, while they do not want to cooperate with u_1. Moreover, Fig. 1 (the right part) shows that the costs of users u_1, u_2 and u_3 are 3,1 and 2, respectively, and the platform has to recruit users under the budget constraint 4. In this single-round scenario, we face the problem of selecting a group of users who have strong individual objective ability as well as good collaboration likelihood under the budget constraint, i.e., the single-round user recruitment problem.

We further consider a multi-round scenario, where users' information is unknown. Several works [14], [15] study this scenario and have proposed several recruitment strategies, which unfortunately ignore users' collaboration likelihood. Motivated by this, we focus on the combination of them, in which users' actual objective abilities and subjective collaboration likelihood need to be estimated according to the feedback (the task completion effect of selected users) of previous rounds. Even so, we still face the problem of recruiting previously well-behaved user groups or exploring other unknown user groups, i.e., the multi-round user recruitment problem. Additionally, the cost of a user seriously affects whether it is selected; however, user costs are usually also unknown to the platform. In other words, the platform can estimate user abilities based on the task completion effect but cannot estimate their costs because the task completion effect has no relationship to users' costs. Thus, we are also confronted with the problem of obtaining the truthful user costs when selecting users under the budget constraint.

After considering the subjective collaboration likelihood, the actual user utility depends not only on its objective ability but also on its cooperators within the same group. Thus, how to *model each user's actual utility to finish a sensing task* is the first challenge. Moreover, the utility increment of adding a new user to a user group is hard to measure (maybe non-linearly or non-monotonically) because the collaboration likelihood with existing users in the group will affect the new user's actual utility. In this case, the previous greedy recruitment strategies [3], [16] cannot work well anymore. Furthermore, when recruiting a group of users to perform tasks, the platform needs to pay users to cover their consumed resources (e.g., the traveled distance, sensing time and battery power) [17], [18]. We call these consumed resources a recruiting cost. Obviously, the recruiting costs of all selected users cannot exceed the platform's payment budget. In this case, for the single-round scenario, how to *recruit a group of users with both the strong objective ability and good collaboration likelihood under the budget constraint* is the second challenge. Moreover, in the multi-round scenario without enough user information, a conservative strategy is to continue recruiting the previously well-behaved users (exploitation). By contrast, a progressive strategy is to try recruiting uncertain but potential users (exploration). That is, exploitation or exploration alone would lead to bad task competition effect. Therefore, how to *balance the trade-off between exploration and exploitation in the multi-round scenario* is the third challenge. Additionally, the platform does not know each user's actual cost, and each user needs to upload their costs for selection. However, users may report fake costs to obtain more profits; thus, how to *ensure that users report their costs truthfully is the fourth challenge.*

To overcome the above challenges, we first formulate the user ability by constructing a graph, where vertex weights denote the users' objective abilities and edge weights denote users' collaboration likelihood with each other. Second, we convert the single-round user recruitment problem into a min-cut problem [19] and select users based on a modified min-cut algorithm under the budget constraint. Third, we propose the multi-round user recruitment strategy based on the combinatorial multi-armed bandit model (URMB) [20], [21] to balance exploration and exploitation. We prove that the URMB can achieve a tight regret bound. Finally, we present a payment strategy based on the min-cut algorithm to ensure that users report their costs truthfully in both single-round and multi-round scenarios.

Our main contributions are summarized as follows:

- As we know, this is the first work considering both users' objective abilities and the subjective collaboration likelihood when recruiting users under the budget constraint in MCS. Then, in the single-round scenario, we convert the user recruitment problem into the min-cut problem and propose a graph theory based algorithm to find the approximate solution under the budget constraint. We also prove the equivalence of the conversion, the completeness, and approximation ratio of the proposed algorithm.
- For the multi-round user recruitment problem without enough prior knowledge, we propose the multi-round user recruitment strategy based on combinatorial multi-armed bandit model (URMB) to balance exploration and exploitation. We also prove that the proposed strategy achieves a tight regret bound.
- To prevent users from reporting fake costs, we present a payment strategy based on a min-cut algorithm to ensure that users will get the maximum profits when truthful. The strategy proves to achieve truthfulness and individual rationality.
- We conduct extensive simulations based on three real-world datasets to verify the performance of the proposed strategy, and the results show that URMB always outperforms other strategies.

The remainder of the paper is organized as follows. We first review the related works in Section 2. Then, we introduce the system model and formulate the user recruitment problems for the single-round and multi-round scenarios in Section 3. Furthermore, we propose the user recruitment strategy for different scenarios in detail in Section 4. In Section 5, we discuss the theoretical analysis of the proposed strategy. Finally, we evaluate the performance in Section 6 and conclude this paper in Section 7.

2 Related Work

2.1 User Recruitment in MCS

The user recruitment problem in MCS has been studied for years. When recruiting users, some works often consider the user's objective ability such as the ability to cover sensing tasks [3], [16]. In [3], Gao *et al.* consider the joint probability of multiple vehicles and design the truthful user recruitment strategy to select a group of users to ensure that each task is performed with a high probability. Yang *et al.* [16] exploit the historical data to model each user's mobility and predict the probability of users performing tasks. Then, they propose a prediction-based user recruitment framework by using optimal stopping theory. Similarly, based on the user mobility model, He *et al.* [22] propose a greedy approximation algorithm and a genetic algorithm to select users in the vehicular network. Wang *et al.* [23] utilize the content and the context information to characterize each user's preference on different tasks. Then, they predict the probability of users performing tasks and recruit the best user-task pair. However, all of these works assume that users' information such as the historical location information or context information is known to the platform. However, such user information is difficult to obtain in advance, considering the protection of the sensitive information. To this end, Wu *et al.* [24] develop a Thompson Sampling algorithm to learn each user's utility based on the feedback of performing tasks in previous rounds and select a single user in each round. Furthermore, Gao *et al.* [25] propose an extended UCB based user recruitment strategy to learn the utilities of a group of users and solve the exploitation-exploration problem. However, both [24] and [25] ignore each user's subjective collaboration likelihood with others. In summary, these works about user recruitment either assume the user's information (e.g., historical trajectory and interest preferences) is available or ignore the user's subjective collaboration likelihood with others.

2.2 Combinatorial Multi-Armed Bandit Problem

To balance exploration and exploitation, the multi-armed bandit problem (MAB) has been studied for years, and there are several well-known algorithms: ϵ-greedy algorithm [26], upper confidence bound algorithm (UCB) [27] and so on. Li *et al.* [28] argue that in the practical applications, MAB always ignores some important factors in the system. For example, some arms can be played together in combination, or some arms will sleep for a period of time after being played once. To that end, they propose a new combinatorial sleeping MAB model with fairness constraints based on UCB. Additionally, Kim *et al.* [29] investigate the contextual MAB scenario and propose a new contextual MAB algorithm based on the relaxed semiparametric reward model. Kagrecha *et al.* [30] argue that the expected reward in MAB is the risk-neutral metric; then, they investigate the arm selecting framework under the fixed budget based on both the expected reward and conditional risk value. Moreover, Gai *et al.* [20] further extend the MAB to the combinatorial multi-armed bandit problem (CMAB) where multiple variables (arms) can be selected at the same time. They investigate the linearly weighted combination of selected variables and propose the LLR algorithm with bounded regret. Afterward, Chen *et al.* [21] propose a general algorithm CUCB with a large number of nonlinear reward instances to solve the CMAB problem. Hüyük *et al.* [31] investigate the combinatorial Thompson Sampling strategy and achieve a better regret than [21]. However, these CMAB algorithms cannot be directly used to solve the user recruitment problem in MCS because they ignore the arm's influence on each other when determining the reward of each arm.

3 System Model and Problem Formulation

3.1 System Model

In this paper, we consider a general MCS campaign, where a cooperative task τ is published by the platform in some sensing areas at the beginning and will remain there until the end of system time. The MCS campaign consists of multiple rounds represented as $T = \{1, \ldots, t, \ldots, K\}$, and in each round the platform recruits users to perform tasks. The mobile user set is denoted as $U = \{u_1, \ldots, u_M\}$, and each user constantly moves around sensing areas. The recruiting cost of user u_i in round t is denoted as c_i^t. Note that each user reports its recruiting cost at the beginning of the round. Moreover, we adopt the frequency of u_i passing through the sensing areas in round t to denote its objective ability ρ_i^t; then, all objective abilities of users in U can be represented as the vector $\boldsymbol{\rho^t} = \{\rho_1^t, \rho_2^t, \ldots, \rho_M^t\}$. Note that the objective ability of each user actually follows a fixed distribution during a long-term time, which will be verified in the following sections. Additionally, we use a likelihood matrix $\boldsymbol{\alpha} = \{\alpha_{11}, \alpha_{12}, \ldots, \alpha_{ij}, \ldots, \alpha_{MM}\}$ to denote the collaboration likelihood between any two users in U, and $\alpha_{ij} \in [0, 1]$ denotes the collaboration likelihood between u_i and u_j. Generally, if u_i and u_j are familiar and have a good relationship, then the collaboration likelihood between them will be high. However, if they are unfamiliar or familiar but have a bad relationship, then the collaboration likelihood between them is low, and accordingly their task completion effect becomes bad. Obviously, we have $\alpha_{ij} = \alpha_{ji}$ because of the symmetry of the collaboration likelihood between u_i and u_j, and α_{ii} will not be used in the following sections. Note that we obtain the collaboration likelihood between each two users from the real-world datasets which include both the user mobility and user social information (e.g., whether u_i and u_j are friends).

In each single round t, users report their costs, and the platform selects a group of users S^t to perform the task τ, and the task completion effect depends not only on the user's objective ability but also on the collaboration likelihood with each other. We consider that the effect of each user completing the task once is similar. Therefore, we use the quality of data (QoD) [32], [33] of the task, i.e., the number of times the task is performed, to measure the task completion effect of S^t, which is represented as follows:

$$Q^t(S^t) = \sum_{u_i \in S^t} \bar{\alpha}_i^t \cdot \rho_i^t, \tag{1}$$

where $\bar{\alpha}_i^t = \sum_{u_j \in S^t, j \neq i} \alpha_{ij}^t / (|S^t| - 1)$ denotes the average collaboration likelihood between u_i and each user in $S^t / \{u_i\}$.

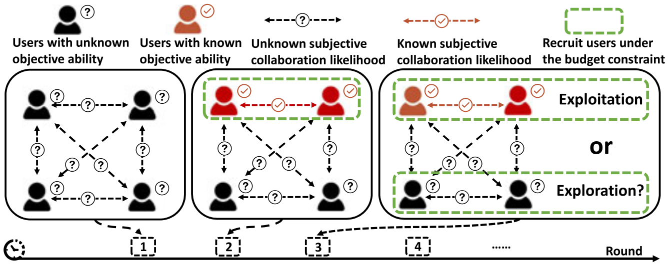


Fig. 2. The illustration for exploration and exploitation.

Let ρ_i^t and α_{ij}^t denote the objective ability of u_i and the collaboration likelihood between u_i and u_j. During the total K rounds, the platform recruits users per round, which is the multi-round scenario, and the total task completion effect of K rounds is represented as $\sum_{t\in T} Q^t(S^t)$.

3.2 Problem Formulation

In each single round t, when the platform knows each user's objective ability and collaboration likelihood, how to recruit a group of users to maximize the task completion effect of the single round under the cost budget constraint is the single-round user recruitment problem, which can be formulated as follows:

$$\text{Maximize} \qquad Q^t(S^t) \tag{2}$$

$$\text{Subject to} \qquad \sum_{u_i\in S^t} c_i^t \leq \mathcal{B}. \tag{3}$$

The constraint (3) indicates that the sum of the recruiting costs of the selected users cannot exceed the cost budget $\mathcal{B}$. Furthermore, in the multi-round scenario, when the platform does not know each user's objective ability and collaboration likelihood, how to estimate users' abilities and maximize the total task completion effect of all rounds is the multi-round user recruitment problem, which can be formulated as follows:

$$\text{Maximize} \qquad \sum_{t\in T} Q^t(S^t) \tag{4}$$

$$\text{Subject to} \qquad \sum_{u_i\in S^t} c_i^t \leq \mathcal{B},\ \forall t \in T. \tag{5}$$

Similarly, the constraint (5) indicates that the budget constraint should be satisfied in each round. In this case, to solve the multi-round user recruitment problem, a conservative strategy is to continue recruiting the previously well-behaved users (exploitation), and a progressive strategy is to try recruiting unknown but potential users (exploration). The two strategies conflict with each other, and the key to solve the problem is to balance the trade-off between exploration and exploitation. An example of the trade-off is shown in Fig. 2, where the red users indicate the known users with accurate objective ability and the black users are the unknown users with the inaccurate objective ability. Moreover, the black and red dashed lines indicate the collaboration likelihood between two users. As shown in Fig. 2 (green lines), when selecting users under the budget constraint, we decide whether to explore more unknown users or exploit more known users in each round to maximum the total task completion effect of K rounds.

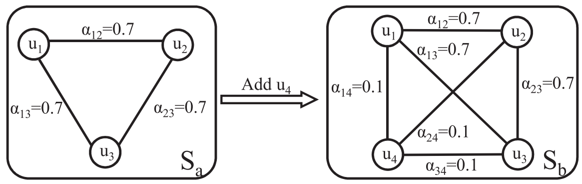


Fig. 3. An example for the single-round recruitment problem.

4 User Recruitment With Objective Ability and Collaboration Likelihood

In this section, we present strategies to solve user recruitment problem in single-round and multi-round scenarios.

4.1 Single-Round User Recruitment

In the single-round scenario, users report their recruitment costs, and we assume that the platform knows each user's ability and collaboration likelihood; then, the platform selects a group of users S based on their abilities under the cost budget constraint to maximize the task completion effect in the current round t. In this subsection, we omit the parameter t in notations. As shown in Eq. (2), the single-round user recruitment problem is non-linear and non-monotonic because the average collaboration likelihood $\bar{\alpha}_i, \forall u_i \in S$ changes when a new user is added into the user group S. An example is shown in Fig. 3, assuming that the objective ability of each user $\rho_1 = \rho_2 = \rho_3 = \rho_4 = 2$, $\alpha_{12} = \alpha_{13} = \alpha_{23} = 0.7$ and $\alpha_{14} = \alpha_{24} = \alpha_{34} = 0.1$, we obtain the QoD of S_a as $Q(S_a) = 4.2$ according to Eq. (1). However, after adding u_4 to S_a, we obtain $Q(S_b) = 3.2 < 4.2$. Therefore, it is difficult to find the optimal group of users to maximize the utility function, and existing greedy-liked user recruitment strategies [3], [16] cannot work effectively anymore. Before the solution, we first prove the NP-hardness of the single-round user recruitment problem in the following theorem.

Theorem 1. *The user recruitment problem is NP-hard.*

Proof. As shown in Eqs. (2) and (3), we consider a special case of the single-round user recruitment problem: given a MCS scenario, where the user set is U, the costs of users c_i are uniform, the objective abilities of users $\rho_i \in \mathbb{N}$ and the subjective collaboration likelihood of users $\alpha_{ij} = 1, \forall u_i, u_j \in U$. Here, the user recruitment problem can be equivalently seen as the knapsack problem, a famous well known NP-hard problem: given a set of items, each with a weight c_i and a value ρ_i, determine which item to include in a group so that the total weight is less than or equal to a given limit $\mathcal{B}$ and the total value is as large as possible. Thus, the single-round user recruitment problem is also at least NP-hard. □

To this end, we find an approximate algorithm to address the problem. First of all, we convert the single-round user recruitment problem into the min-cut problem based on graph theory to find the best group of users efficiently through the following steps:

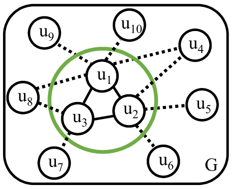


Fig. 4. An example for constructed graph G.

Step 1: Without loss of generality, we construct an edge weighted undirected graph $G=(V,E)$, where each vertex represents a user, denoted as $V=U=\{u_1,\ldots,u_i,\ldots,u_M\}$, and each vertex is assigned a weight c_i. Moreover, assuming that S denotes the solution to the single-round user recruitment problem in Eq. (2), then for the edge between any two vertices u_i and u_j, we assign the edge weight based on their objective abilities and subjective collaboration likelihood:

$$w^*_{ij}=\frac{(\rho_i+\rho_j)\cdot\alpha_{ij}}{|S|-1},\forall u_i,u_j\in U,i\neq j. \tag{6}$$

Note that G is a complete graph because we should consider the collaboration likelihood between any two users. An example of a constructed graph G is shown in Fig. 4, and we only demonstrate a part of the edges for brevity. Then, according to Eqs. (1) and (6), we obtain

$$\sum_{u_i,u_j\in S,i\neq j} w^*_{ij}=\sum_{u_i\in S}\frac{\sum_{u_j\in S,j\neq i}\alpha_{ij}}{|S|-1}\cdot\rho_i=Q(S). \tag{7}$$

Therefore, the single-round user recruitment problem in Eq. (2) is equivalent to the problem of finding the induced subgraph $G'=(V',E')$ with the maximum edge weight, i.e., the maximum weight subgraph problem:

$$\text{Maximize} \quad \sum_{u_i,u_j\in V',i\neq j} w^*_{ij} \tag{8}$$

$$\text{Subject to} \quad \sum_{u_i\in V'} c_i\leq\mathcal{B}. \tag{9}$$

However, when searching for subgraphs with the edge weight w^*_{ij}, we only consider the objective abilities and collaboration likelihood of users and ignore user costs as shown in Eq. (8), which may lead to some extreme cases. For example, if there is a vertex (user) with a good ability (objective ability and collaboration likelihood) and large recruitment cost, we will select it when only considering its ability. However, the vertex will occupy most of the budget and reduce the overall recruitment effect, i.e., we select a low cost-effective vertex. In short, it is difficult to directly find the optimal solution of the maximum weight subgraph problem. To this end, when constructing the graph $G=(V,E)$, we consider not only the vertex weight but also the edge weight to reassign the weight of each edge in E as follows:

$$w_{ij}=\left(\frac{\rho_i}{c_i}+\frac{\rho_j}{c_j}\right)\cdot\alpha_{ij},\forall u_i,u_j\in U,i\neq j. \tag{10}$$

With the new edge weight, the maximum weight subgraph problem in Eq. (8) is no longer equivalent to the single-round user recruitment problem, but its solution approximates the optimal solution of the single-round user recruitment problem. The approximation ratio is proved in Section 5. Next, our main goal is to find the maximum weight subgraph with the modified weight.

Step 2: As shown in Fig. 4, we assume that the subgraph covered by the green circle is the optimal subgraph $G'=(V',E')$ with the maximum modified edge weight, and the total weight of the solid edges is $\sum_{u_i,u_j\in V',i\neq j}w_{ij}$. In other words, as soon as we find these solid edges, the maximum weight subgraph problem is solved. However, it is difficult to find these edges directly, so we tend to find the edges which are connected to the vertices in V' and are not included in E', i.e., the dashed edges in Fig. 4. Furthermore, we discover that the dashed edges are cut by the green circle, which is the cut $c(V',V\setminus V')$ dividing V into two disjoint parts V' and $V\setminus V'$. We call the edges cut by $c(V',V\setminus V')$ cut-set, and the weights are $w(c(V',V\setminus V'))=\sum_{u_i\in V'}\sum_{u_j\in V\setminus V'}w_{ij}$.

Definition 1 (u_s-u_t Cut). *A u_s-u_t cut $c(\mathcal{S},\mathcal{T})$ is a partition of the vertex set V into two vertex sets $\mathcal{S}\cup\mathcal{T}=V$, $\mathcal{S}\cap\mathcal{T}=\emptyset$ and $u_s\in\mathcal{S}$, $u_t\in\mathcal{T}$.*

Definition 2 (Minimum Cut). *The minimum cut is the cut with the minimum $w(c(\mathcal{S},\mathcal{T}))$.*

Let $w_i=\sum_{u_j\in V,j\neq i}w_{ij}$ represent the weight of all edges between u_i and vertices in V. We obtain

$$\sum_{u_i,u_j\in V',i\neq j} w_{ij}=\frac{1}{2}\cdot\left(\sum_{u_i\in V'}w_i-w(c(V',V\setminus V'))\right). \tag{11}$$

Without loss of generality, we convert the maximum weight subgraph problem in Eq. (8) into the minimum cut problem in Eq. (13) as follows:

$$\text{Maximize} \quad \frac{1}{2}\cdot\left(\sum_{u_i\in V'}w_i-w(c(V',V\setminus V'))\right) \tag{12}$$

$$\Leftrightarrow\text{Minimize} \quad w(c(V',V\setminus V'))-\sum_{u_i\in V'}w_i \tag{13}$$

$$\text{Subject to} \quad \sum_{u_i\in V'}c_i\leq\mathcal{B}. \tag{14}$$

For the $\sum_{u_i\in V'}w_i$ in Eq. (13), we regard $-w_i$ as the cost of adding vertex u_i into the subgraph G'. Afterwards, we build a sink vertex u_t and connect all vertices to it, and assign $-w_i$ to the edge between u_i and u_t as $w_{it}=-w_i$. Due to the non-negative property of edge weight, we add a large enough variable $\mathcal{W}=\sum_{u_i\in V}w_i$ to the weight $w_{it}=\mathcal{W}-w_i$. In this way, based on graph theory [19], we can convert the graph $G=(V,E)$ into a new undirected graph $\tilde{G}=(\tilde{V},\tilde{E})$ through four steps:

1) Add source vertex u_s and sink vertex u_t to the vertex set V;
2) Assign the cost 0 to both u_s and u_t;
3) Connect all vertices in V to source u_s (undirected edge) and assign the weight $\mathcal{W}$;
4) Connect all vertices in V to sink u_t (undirected edge) and assign the weight $\mathcal{W}-w_i$.

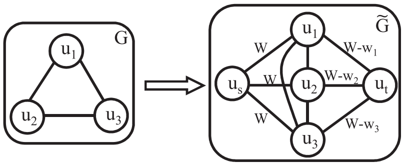


Fig. 5. An example for the conversion from G to $\tilde{G}$.

This conversion process can be formulated as follows:

$$\tilde{V} = V \cup \{u_s, u_t\}, \tag{15}$$

$$c_s = c_t = 0, \tag{16}$$

$$\tilde{E} = \{(i,j)|(i,j) \in E\} \cup \{(u_s, u_i)|u_i \in V\} \cup \{(u_i, u_t)|u_i \in V\}, \tag{17}$$

$$w_{si} = \mathcal{W}, \quad u_i \in V, \tag{18}$$

$$w_{it} = \mathcal{W} - w_i, \quad u_i \in V. \tag{19}$$

An example of the conversion process is shown in Fig. 5. Through the conversion, the problem of finding the maximum weight subgraph in G becomes the problem of finding the minimum u_s-u_t cut in $\tilde{G}$. The equivalence of these two problems will be proven in Section 5.

Algorithm 1. Minimum Cut

Input: $G = (V, E)$, $\mathcal{B}$
Output: cut $c(S', V \setminus S')$
1 $w_0 \leftarrow +\infty$, $S^* \leftarrow \emptyset$, $\mathcal{C}^* \leftarrow 0$;
2 **for** each $u_i \in \{u \in V | u \neq u_s, u \neq u_t\}$ **do**
3 $\quad S \leftarrow S^* \cup \{u_i\}$, $\mathcal{C} \leftarrow \mathcal{C} + c_i$, $\gamma \leftarrow 0$;
4 $\quad$ **while** *True* **do**
5 $\quad\quad$ Search the vertex u_j such that $w(S, u_j) = \max \{w(S, u) | u \in V \setminus S, u \neq u_t, \mathcal{C} + c \leq \mathcal{B}\}$;
6 $\quad\quad$ **if** u_a *exists* **then**
7 $\quad\quad\quad S \leftarrow S \cup \{u_j\}$, $\mathcal{C} \leftarrow \mathcal{C} + c_j$, $\gamma \leftarrow \gamma + 1$;
8 $\quad\quad$ **else**
9 $\quad\quad\quad$ Break the loop;
10 $\quad$ **if** $w(c(S, V \setminus S)) \leq w_0$ **then**
11 $\quad\quad w_0 \leftarrow w(c(S, V \setminus S))$;
12 $\quad\quad S' \leftarrow S$;
13 $\quad$ **if** $\gamma > 4$ **then**
14 $\quad\quad V \leftarrow V \setminus \{u_i\}$, $S^* \leftarrow S^* \cup \{u_i\}$, $\mathcal{C}^* \leftarrow \mathcal{C}^* + c_i$;
15 $\quad\quad$ Back to line 2 ;
16 **return** cut $c(S', V \setminus S')$

After the conversion process, we propose a specific minimum cut algorithm as shown in Algorithm 1 based on Stoer-Wagner algorithm [34]. In Algorithm 1, we first input the converted graph $\tilde{G}$ into the algorithm and initialize some variables (line 1). Then, for each vertex u_i in $V \setminus \{u_s, u_t\}$, we give the initial value $\{u_i\}$ to the vertex set S, value c_i to the total cost $\mathcal{C}$ and value 0 to the variable γ (line 3). Next, in the loop phase (lines 4~9), we search the vertex u_a which is connected to the vertex set S most tightly from the vertex set $V \setminus S$ while ensuring $\mathcal{C} + c_b \leq \mathcal{B}$ (line 5), where $w(S, u_j) = \sum_{u_i \in S} w_{ij}$ is defined as the weights of all edges between the vertex set S and the vertex u_j, and $\mathcal{B}$ is the budget constraint (14). If the vertex u_a exists, we add it to S and add its cost to $\mathcal{C}$ (line 7). If u_a doesn't exist, it means that the budget $\mathcal{B}$ has been exhausted, and the loop phase ends (line 9). Next, we discover the minimum cut in the current round $c(S', V \setminus S')$ (lines 10~12). In lines 13-15, we use Breadth-First Search (BFS) [35] to find the adjacent vertices of u_i. Finally, we get the cut $c(S', V \setminus S')$, which proves to be the minimum u_s -u_t cut in Section 5. Clearly, the computational complexity of Algorithm 1 is less than $O(|V| + |E|)$. In short, as long as we input the converted graph $\tilde{G}$ to Algorithm 1, we will get the minimum u_s-u_t cut of $\tilde{G}$. The solution to the maximum weight subgraph problem in Eq. (8) is exactly the vertex set $V' = S' \setminus \{u_s\}$, and the approximation solution to the single-round user recruitment problem in Eq. (2) is exactly the users in V'.

In summary, we solve the single-round user recruitment problem in Eq. (2) through the following steps:

1) Construct a graph $G = (V, E)$ based on users' abilities (objective abilities and collaboration likelihood) and recruiting costs to convert the single-round user recruitment problem into the maximum weight subgraph problem;
2) Construct another graph $\tilde{G}$ through the process in Eqs. (15), (16), (17), (18), and (19) to convert the maximum weight subgraph problem into the minimum u_s-u_t cut problem in the undirected graph;
3) Propose an algorithm based on graph theory to find the minimum cut. Correspondingly, the single-round user recruitment problem is solved.

4.2 Multi-Round User Recruitment

After solving the single-round user recruitment problem, we focus on the multi-round user recruitment problem. Different from the single-round user recruitment, in the multi-round scenario, the platform does not know users' objective abilities and collaboration likelihood and has to estimate users' abilities to recruit users. Thus, recruiting users under the cost budget constraint in each round to maximize the total task completion effect of all rounds is difficult. To solve the problem, we should consider both the following two points at the same time:

1) Exploration: The platform should recruit suitable users in each round to explore unknown users who have been recruited few times and are estimated inaccurately. Then, the platform update these users' estimated abilities (objective ability and collaboration likelihood) based on the feedback of previous rounds to get close to the actual abilities.
2) Exploitation: In each round, the platform recruits suitable users according to the estimated user abilities to maximize the task completion effect of the current round and finally achieve the maximum total task completion effect of all round.

Before addressing the problem, we first propose a strategy to update the estimated abilities of the explored users based on the feedback of previous rounds.

4.2.1 *Update Strategy*

Specifically, after the recruited users finish the task in each round, the actual objective ability of each user (the frequency of each selected user passing through the task) and the actual QoD (the total number of times the task is performed) are observable, which are called feedback. Moreover, for each user, the frequency of passing through the task is influenced by many independent factors such as personal preferences, traffic conditions and so on. According to the central limit theorem [36], each user's observed objective abilities in different rounds approximately follow a normal distribution, which is verified by the real-world dataset in Section 5. Additionally, users' actual collaboration likelihood does not change over time. Consequently, we can update the estimated objective abilities and collaboration likelihood based on the feedback of previous rounds [24].

Algorithm 2. Update Among Rounds

Input: All the observed data $(\boldsymbol{\rho_o^t}, Q_o^t), t \in [t_1, t_m]$
Output: Updated likelihood matrix $\boldsymbol{\alpha^{t_m+1}}$

1 $\rho_i^{t+1} \leftarrow \sum_{r=1}^{k_{i,t}} \rho_{i,r}/k_{i,t}$;
2 **while** $\eta \frac{\partial J(\boldsymbol{\alpha^{t_m}})}{\partial \alpha_{ij}^{t_m}} \leq \varepsilon, \forall \alpha_{ij}^{t_m}$ in $\boldsymbol{\alpha^{t_m}}$ **do**
3   **for** each $\alpha_{ij}^{t_m}$ in $\boldsymbol{\alpha^{t_m}}$ **do**
4         $\alpha_{ij}^{t_m} \leftarrow \alpha_{ij}^{t_m} - \eta \frac{\partial J(\boldsymbol{\alpha^{t_m}})}{\partial \alpha_{ij}^{t_m}}$;
5 $\boldsymbol{\alpha^{t_m+1}} \leftarrow \boldsymbol{\alpha^{t_m}}$;
6 **return** $\boldsymbol{\rho^{t_m+1}}, \boldsymbol{\alpha^{t_m+1}}$

Update on objective ability: Considering that each user's objective ability follows a normal distribution, we can use the mean of observations to update each user's estimated objective ability. We assume that in the multi-round scenario, a user u_i has been selected $k_{i,t}$ times before round t, and $\rho_{i,r}$ refers to the actual observed objective ability when u_i is recruited for the rth recruitment time (not round r). Furthermore, the user's estimated objective ability ρ_i^{t+1} in round $t+1$ is updated as follows:

$$\rho_i^{t+1} = \frac{\sum_{r=1}^{k_{i,t}} \rho_{i,r}}{k_{i,t}}. \tag{20}$$

Then, the updated estimated objective ability is used for the single-round recruitment in round $t+1$. Note that the update process is conducted in each round according to Eq. (20), and the estimated objective ability of each user will gradually converge to the actual objective ability.

Update on collaboration likelihood: Then, we update the estimated likelihood matrix based on the batch gradient descent (BGD) [37]. First, we define the hypothetical function $\hbar$ to denote the expected value of QoD in round t as follows:

$$\hbar(\boldsymbol{\alpha^t}, \boldsymbol{\rho^t}) = Q^t(S^t) = \sum_{u_i \in S^t} \bar{\alpha}_i^t \cdot \rho_i^t, \tag{21}$$

where $\boldsymbol{\rho^t}$ and $\boldsymbol{\alpha^t}$ denote the estimated objective ability vector and collaboration likelihood matrix in round t, respectively. To denote the difference between the values of the estimated QoD and the actual observed QoD, we define the loss function J in the mth round according to Eq. (1) as follows:

$$J(\boldsymbol{\alpha^{t_m}}) = \frac{1}{2m} \sum_{t=1}^{t_m} (\hbar(\boldsymbol{\alpha^{t_m}}, \boldsymbol{\rho_o^t}) - Q_o^t)^2. \tag{22}$$

Note that the loss function $J(\boldsymbol{\alpha^{t_m}})$ is the total loss of all previous rounds before round t_m, and $\hbar(\boldsymbol{\alpha^{t_m}}, \boldsymbol{\rho_o^t})$ denotes the expected QoD value based on the actual observed objective ability vector $\boldsymbol{\rho_o^t}$ in round t and the estimated likelihood matrix $\boldsymbol{\alpha^{t_m}}$ in round t_m. Q_o^t denotes the observed value of the QoD in round t. Then, we propose the update strategy as shown in Algorithm 2. First, we input the observed data of all rounds before round t_m (including), i.e., $\boldsymbol{\rho_o^t}$ and Q_o^t, $t \in [t_1, t_m]$, where t_1 represents the first round. Moreover, line 1 represents the update on the objective ability vector according to Eq. (20). Then, we enter the loop and update the value of each item $\alpha_{ij}^{t_m}$ in the likelihood matrix $\boldsymbol{\alpha^{t_m}}$ (lines 3-4), where $\frac{\partial J(\boldsymbol{\alpha^{t_m}})}{\partial \alpha_{ij}^{t_m}}$ is the gradient of J, and η denotes the step size (i.e., learning rate). The loop stops when the descent distances of all items are less than the accuracy ε (line 2). Finally, we can get the updated likelihood matrix $\boldsymbol{\alpha^{t_m+1}}$, which will be used for the recruitment process in the next round.

Algorithm 3. Multi-Round User Recruitment

Input: User set U, user objective ability vector $\boldsymbol{\rho}$, likelihood matrix $\boldsymbol{\alpha}$
Output: Total QoD Q after K rounds

$Q \leftarrow 0, t \leftarrow 1, \boldsymbol{\rho^t} \leftarrow \boldsymbol{\rho}, \boldsymbol{\alpha^t} \leftarrow \boldsymbol{\alpha}$;
$r_i \leftarrow 1$ for each $u_i \in U$;
while $t \leq K$ **do**
    $t \leftarrow t+1$;
    for each ρ_i^t *in* $\boldsymbol{\rho^t}$ **do**
        $\tilde{\rho}_i^t \leftarrow \rho_i^t + \sqrt{\frac{3 \ln t}{2 r_i}}$;
    Convert the user set U to graph G through Eq. (6) with $\boldsymbol{\tilde{\rho}^t}, \boldsymbol{\alpha^t}$;
    Convert graph G to graph $\tilde{G}$ through Eqs. 15–19;
    $c(S', V \setminus S') \leftarrow$ Minimum Cut ($\tilde{G}$);
    Select the users in $S' \setminus \{u_s\}$;
    for each $u_i \in S' \setminus \{u_s\}$ **do**
        $r_i \leftarrow r_i + 1$;
    Users perform the task and we observe the actual values of $\boldsymbol{\rho_o^t}$ and Q_o^t at the end of round t;
    $\boldsymbol{\rho^{t+1}}, \boldsymbol{\alpha^{t+1}} \leftarrow$ Update Among Rounds ();
    $Q \leftarrow Q + Q_o^t$;
return Q

4.2.2 Multi-Round Strategy

Next, to address the multi-round user recruitment problem, we propose a multi-round strategy based on the combinatorial multi-armed bandit model (URMB). As mentioned above, to maximize the total task completion effect of K rounds, the platform should both consider exploration and exploitation when recruiting users in each round. Neither pure exploration nor pure exploitation will bring the best results. Therefore, balancing the trade-off of exploration and exploitation is the key of solving the multi-round user recruitment problem. The multi-round user process can be mainly divided into two parts:

1) The single-round user recruitment process in each round based on estimated user abilities (objective abilities and collaboration likelihood) and costs.
2) The update process on user abilities at the end of each round.

Considering these two parts, we propose Algorithm 3 based on CUCB [21]. First, we input the user set U and a little prior knowledge about each user's ability $\boldsymbol{\rho}$ and $\boldsymbol{\alpha}$ as the initial estimates for the first recruitment. Actually, the accuracy of the prior knowledge about user abilities determines the regret bound of the algorithm, which is proven in Section 5. After the initialization process (lines 1-2), the algorithm enters the loop when $t \leq K$ (line 3). In each round t, we conduct the single-round user recruitment process based on Algorithm 1. Note that the objective ability vector $\boldsymbol{\rho}^t$ used for the single-round user recruitment (lines 7-9) includes an adjustment term $\sqrt{3\ln t/2r_i}$ (line 6) to balance the trade-off between exploration and exploitation, where t indicates the current round number, and r_i indicates the number of times the user u_i has been selected before the current round. Specifically, the adjustment item is used to readjust each user's ability in each round to increase the recruitment probability of users who are rarely recruited. Moreover, we regard the prior knowledge about users' objective abilities as each user is selected once and initialize $r_i = 1$ for each user (line 2). Furthermore, we select users in $S' \setminus \{u_s\}$ to perform task τ and update r_i (lines 11-12). Then, at the end of each round, both the actual objective ability of each user (the frequency of passing through the task) and the number of tasks performed are observable. Next, we update the selected users' abilities based on Algorithm 2 and get the latest user ability for the next round (line 14). Note that the update strategy needs all observed data up to the current round t. Afterwards, we add the observed task completion effect QoD Q_o^t of the current round to the total QoD Q. Then, the current round t ends and goes to the next round. Finally, we get the total QoD of K rounds as output.

4.3 Graph-Based Truthful Payment

In both single-round and multi-round scenarios, users report their costs to the platform at the beginning of a round, and the platform recruit users based on their abilities and costs. Existing works [8], [16] always assume that users report their costs truthfully. However, we argue that users may report fake costs to obtain more profits, for example. If a user reports its cost 10 and a fake cost 12, the user can be recruited with both two costs. Then, the user is likely to report a fake cost to get more payment, which leads to a bad recruitment effect. To guarantee users' truthfulness, there have been several works focusing on the untruthful behaviors and devising payment mechanisms both when the platform knows users' abilities [4], [10], [38] and when the platform does not know users' abilities [39], [40]. Specifically, to achieve truthfulness with known user information, Zhou *et al.* [4] propose an online user recruitment and payment methods that makes on-spot decisions upon dynamically arriving users. Wang *et al.* [10] focus on devising an offline privacy-preserving user recruitment and payment methods. Meanwhile, a group-buying-based pricing method is proposed by [38]. To achieve truthfulness with unknown user information, Gao *et al.* [39] propose an auction-based bandits model to learn users' information and decide payments. Further, K-bandits method is proposed in [40] to decide payments for unknown users. Unfortunately, these payment mechanisms ignore users' collaboration likelihood and cannot achieve truthfulness in this paper.

To this end, we present a truthful payment calculation strategy as shown in Algorithm 4 to calculate payments for selected users at the end of round in both single-round and multi-round scenarios. Most existing payment strategies are based on the greedy user recruitment strategies [3], [41], which would not work well in this paper due to the collaboration likelihood. Hence, we propose a novel payment calculation algorithm based on graph theory. The specific steps of the algorithm are included in Algorithm 4. We first input the converted graph G, the cost budget $\mathcal{B}$ like Algorithm 1, and a recruited user to calculate its payment. Then, we initialize some variables. Note that V_s denotes the vertex set without the recruited user u_s (line 1). Next, we conduct the same process like that in Algorithm 1 based on the Stoer-Wagner algorithm (lines 4-12). However, while searching the vertex u_j (line 5), we add it to the vertex set S and calculate a temporary payment p of u_s (line 8), where ρ_s is its objective ability and α_{sk} is its collaboration likelihood with u_j. Then, we get the maximum temporary payment p_s as the final payment of u_s. Note that in the multi-round scenario, the platform allows users to report different costs in each round because in reality user's costs may vary over rounds, and Algorithm 4 can still handle this situation.

Algorithm 4. Graph-Based Truthful Payment

Input: $G = (V, E)$, $\mathcal{B}$, the selected user u_s
Output: Payment p
1 $V_s \leftarrow V \setminus \{u_s\}$, $w_0 \leftarrow +\infty$, $S^* \leftarrow \emptyset$, $\mathcal{C}^* \leftarrow 0$, $p_s \leftarrow 0$;
2 **for** each $u_i \in \{u \in V_s | u \neq u_s, u \neq u_t\}$ **do**
3 $\quad S \leftarrow S^* \cup \{u_i\}$, $\mathcal{C} \leftarrow \mathcal{C}^* + c_i$, $\gamma \leftarrow 0$;
4 $\quad$ **while** *True* **do**
5 $\quad\quad$ Search the vertex u_j such that $w(S, u_j) = \max\{w(S, u) | u \in V_s \setminus S, u \neq u_t, \mathcal{C} + c \leq \mathcal{B}\}$;
6 $\quad\quad$ **if** u_j *exists* **then**
7 $\quad\quad\quad S \leftarrow S \cup \{u_j\}$, $\mathcal{C} \leftarrow \mathcal{C} + c_j$, $\gamma \leftarrow \gamma + 1$;
8 $\quad\quad\quad p \leftarrow \rho_s \cdot \sum_{u_k \in S} \alpha_{sk} /\ (w(S, u_j) - \sum_{u_k \in S} \rho_k \cdot \alpha_{sk}/c_k)$;
9 $\quad\quad\quad$ **if** $p \geq p_s$ **then**
10 $\quad\quad\quad\quad p_s \leftarrow p$;
11 $\quad\quad\quad$ **else**
12 $\quad\quad\quad\quad$ Break the loop;
13 $\quad$ **if** $\gamma > 4$ **then**
14 $\quad\quad V_s \leftarrow V_s \setminus \{u_i\}$, $S^* \leftarrow S^* \cup \{u_i\}$, $\mathcal{C}^* \leftarrow \mathcal{C}^* + c_i$;
15 $\quad\quad$ Back to line 2 ;
16 **return** p_s

5 Theoretical Analysis

In this section, we present the theoretical analysis to demonstrate the properties of the single-round strategy and multi-round strategy in URMB.

5.1 Analysis on Single-Round Strategy

For the conversion process from graph G to $\tilde{G}$ in Eqs. (15), (16), (17), (18), and (19), we convert the problem of finding the maximum weight induced subgraph of graph G into the problem of finding the minimum u_s-u_t cut of $\tilde{G}$.

Lemma 1. *Any u_s-u_t cut $c(\mathcal{S}, \mathcal{T})$ of the converted graph $\tilde{G}$ one-to-one corresponds to a subgraph G' of graph G, where $\mathcal{S}$ and $\mathcal{T}$ are two vertex sets of $\tilde{G}$.*

Proof. We assume that $c(\mathcal{S},\mathcal{T})$ is a u_s-u_t cut of $\tilde{G}$, $u_s \in \mathcal{S}$, $u_t \in \mathcal{T}$. Since the cut is a partition of $\tilde{G}$ into $\mathcal{S}$ and $\mathcal{T}$, we can always find only one subgraph $G' = (V', E')$ of G such that $V' = \mathcal{S} \setminus \{u_s\}$. □

Theorem 2. *The cut-set of cut $c(S', V \setminus S')$ found by Algorithm 1 is the solution to the problem in Eq. (8).*

Proof. The maximum weight subgraph problem in Eq. (8) is to find a subgraph $G' = (V', E')$ to maximize $\sum_{u_i,u_j \in V', i\neq j} w_{ij}$.

Assuming that Algorithm 1 finds a cut $c(\mathcal{S},\mathcal{T})$ in $\tilde{G}$, according to Lemma 1, we have $V' = \mathcal{S} \setminus \{u_s\}$ and $V \setminus V' = \mathcal{T} \setminus \{u_t\}$. Thus, we get

$$
\begin{aligned}
w(c(\mathcal{S},\mathcal{T})) &= \sum_{u_i \in \mathcal{S}, u_j \in \mathcal{T}} w_{ij} \\
&= \sum_{u_i \in V'} w_{it} + \sum_{u_j \in V \setminus V'} w_{sj} + \sum_{u_i \in V', u_j \in V \setminus V'} w_{ij} \\
&= \sum_{u_i \in V'} (\mathcal{W} - w_i) + \sum_{u_j \in V \setminus V'} \mathcal{W} + \sum_{u_i \in V', u_j \in V \setminus V'} w_{ij} \\
&= |V| \cdot \mathcal{W} + \sum_{u_i \in V'} \Big(- w_i + \sum_{u_j \in V \setminus V'} w_{ij} \Big) \\
&= |V| \cdot \mathcal{W} + \sum_{u_i \in V'} \Big(- \sum_{u_j \in V'} w_{ij} \Big) \qquad (23) \\
&= M \cdot \mathcal{W} - \sum_{u_i, u_j \in V', i\neq j} w_{ij}, \qquad (24)
\end{aligned}
$$

where Eq. (23) holds because w_i represents the weights of all edges between u_i and other vertices in V, and $\sum_{u_j \in V'} w_{ij}$ denotes the weights of all edges between u_i and all other vertices in $V \setminus V'$. Meanwhile, G' is the induced subgraph. Thus, $-w_i + \sum_{u_j \in V \setminus V'} w_{ij} = -\sum_{u_j \in V'} w_{ij}$. As $M \cdot W$ is a constant term, according to Eq. (24), as long as we find the minimum u_s-u_t cut $c(\mathcal{S},\mathcal{T})$, we can get the maximum total edge weights $\sum_{u_i,u_j \in V', i\neq j} w_{ij}$, where the vertex set V' is actually the solution to the maximum weight subgraph problem in Eq. (8). □

Theorem 3. *The cut $c(\mathcal{S},\mathcal{T})$ found by Algorithm 1 is a u_s-u_t cut of the converted graph $\tilde{G} = (\tilde{V}, \tilde{E})$.*

Proof. In the conversion process from G to $\tilde{G}$, we assign the weight $W = \sum_{u_i \in V} w_i$ to the edge between u_s and every vertex in V. Thus, for the initial vertex u_i (line 3), we have $w_{is} = W > w_{ij}, \forall u_j \in \tilde{V} \setminus u_s$. Then, according to Algorithm 1 (line 6), the first searched vertex must be u_s, and u_s will be added into $\mathcal{S}$. Meanwhile, the limitation in the search process $u_b \neq u_t$ ensures that $u_t \notin \mathcal{S}$. Therefore, the cut found by Algorithm 1 is a u_s-u_t cut. □

Theorem 4. *Algorithm 1 achieves the approximation ratio as c_{min}/c_{max} for the single-round user recruitment problem in Eq. (8), where c_{min} and c_{max} are the minimum and maximum recruiting costs among all users in U.*

Proof. Assuming that the cut $c(S', V \setminus S')$ is found by Algorithm 1, then according to Theorems 1 and 2, we know the vertex set $S' \setminus \{u_s\}$ is the solution of the maximum weight subgraph problem in Eq. (8). Let $S^\dagger = S' \setminus \{u_s\}$ denote the solution. Next, we prove the approximation ratio of Algorithm 1 through the following two step:

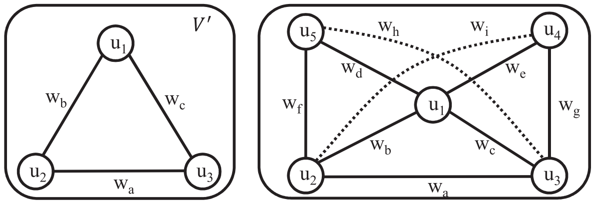


Fig. 6. Illustration for optimality analysis.

Step 1: First, we will prove that vertex set $S^\dagger$ is the optimal solution of the maximum weight subgraph problem in Eq. (8). Meanwhile, since the weights of edges between u_s and any vertex u_i (except u_t) are equal, we ignore the vertex u_s in the following process for simplicity.

Suppose that $S^\dagger$ is the optimal solution. When $|S^\dagger| = 2$, let $S^\dagger = \{u_1, u_2\}$. In Algorithm 1, when the initial vertex is u_1 (line 3), we assume that the next added vertex is u_3 ($u_3 \neq u_2$), which means $w_{13} \geq w_{12}$ and $c_1 + c_3 \leq \mathcal{B}$. According to line 5, $\{u_1, u_3\}$ is better than $\{u_1, u_2\}$, which contradicts the hypothesis that $S^\dagger = \{u_1, u_2\}$ is the optimal set. Thus, the next added vertex will be u_2, which means that Algorithm 1 will find the optimal set $S^\dagger = \{u_1, u_2\}$.

When $|S^\dagger| = 3$, we assume that the optimal set is $S^\dagger = \{u_1, u_2, u_3\}$, and the corresponding edge weights are w_a, w_b, and w_c, as shown in Fig. 6. Note that the following process meets the budget constraint. Let $w_a \geq w_b \geq w_c$. In the worst case, if Algorithm 1 does not find $S^\dagger$ in the process when u_2 or u_3 is the initial vertex (line 3), then we will prove that $S^\dagger$ will be found when u_1 is the initial vertex. If we assume that in the first iteration in lines 4∼9, the initial vertex is u_2 and the next added vertex is an arbitrary vertex u_5, then we have $w_f > w_a$; let $w_f = w_a + \varepsilon_1$, $\varepsilon_1 \in \mathbb{R}^+$. Since the optimal set $S^\dagger = \{u_1, u_2, u_3\}$, then the weight of $\{u_1, u_2, u_5\}$ is $w_f + w_b + w_d < w_a + w_b + w_c$ and we get $w_d < w_c - \varepsilon_1$. For the same reason, if the weight of $\{u_2, u_3, u_5\}$ is $w_a + w_f + w_h < w_a + w_b + w_c$, then $w_h < w_b + w_c - w_a - \varepsilon_1$. Similarly, in the second iteration, if the initial vertex is u_3 and the next added vertex is an arbitrary vertex u_4, let $w_g = w_a + \varepsilon_2$, $\varepsilon_2 \in \mathbb{R}^+$, we get $w_e < w_b - \varepsilon_2$ and $w_i < w_b + w_c - w_a - \varepsilon_2$. Subsequently, when the initial vertex is u_1, the first vertex added to S will be u_2 (line 5) because $w(S, u_2) = w_b > w(S, u_3) = w_c > w(S, u_5) = w_d$ and $w_b > w(S, u_4) = w_e$. Now $S = \{u_1, u_2\}$ and the next added vertex must be u_3 because that $w(S, u_3) - w(S, u_4) = w_a + w_c - (w_e + w_i) > 2(w_a - w_b + \varepsilon_2) > 0$ and $w(S, u_3) - w(S, u_5) = w_a + w_c - (w_f + w_d) > 0$. Therefore, Algorithm 1 will find the optimal user set $S^\dagger$.

Similarly, when $|S^\dagger| > 3$, according to Algorithm 1, once the initial vertex is determined $S \leftarrow \{u_i\}$, the second added vertex u_j is searched by BFS (lines 13∼15). Specifically, we can combine u_i and u_j into a single vertex u_k such that $w_{kx} = w_{ix} + w_{jx}$, $\forall u_x \in V$ in the following adding vertex process, i.e., $S = \{u_i, u_j\} \rightarrow S^* = \{u_k\}$. It does not affect the subsequent algorithm process because $w(S, u_x) = \sum_{u_y \in S} w_{yx} = w_{ix} + w_{jx} = w_{kx} = w(S^*, u_x)$. Consequently, the case of $|S^\dagger| > 3$ can be converted into the case of $|S^\dagger| = 3$ (line 13), whose optimality has been proven. In conclusion, the optimality is proved.

Step 2: We have proved that $S^\dagger$ is the optimal solution to the maximum weight subgraph problem in Eq. (8). However, according to Eqs. (6), (7), and (10), we have

$$\sum_{u_i,u_j\in S,i\neq j} w_{ij} \propto \sum_{u_i\in S} \frac{\sum_{u_j\in S,j\neq i}\alpha_{ij}}{|S|-1}\cdot\frac{\rho_i}{c_i} \neq \sum_{u_i\in S}\frac{\sum_{u_j\in S,j\neq i}\alpha_{ij}}{|S|-1}\cdot\rho_i = Q(S). \tag{25}$$

Clearly, $S^\dagger$ is not the optimal solution to the single-round user recruitment problem. Assuming that $S_{opt} = \{u_1^*, u_2^*, \dots, u_i^*, \dots, u_m^*\}$ is the optimal solution to the single-round user recruitment problem in Eq. (2), we have

$$\frac{Q(S^\dagger)}{min\{c_i\}} \geq \frac{\rho_1\cdot\bar{\alpha}_1}{c_1}+\frac{\rho_2\cdot\bar{\alpha}_2}{c_2}+\cdots+\frac{\rho_n\cdot\bar{\alpha}_n}{c_n} \geq \frac{\rho_1^*\cdot\bar{\alpha}_1^*}{c_1^*}+\frac{\rho_2^*\cdot\bar{\alpha}_2^*}{c_2^*}+\cdots+\frac{\rho_m^*\cdot\bar{\alpha}_m^*}{c_m^*} \geq \frac{Q(S_{opt})}{max\{c_i\}}. \tag{26}$$

The first inequality holds because we have proved that $S^\dagger$ is the optimal solution to the maximum weight subgraph problem in Eq. (8), and therefore has the largest value of $\sum_{u_i\in S^\dagger}\frac{\rho_i\cdot\bar{\alpha}_i}{c_i}$. Consequently, we get $\frac{c_{min}}{c_{max}} \leq \frac{Q(S^\dagger)}{Q(S_{opt})} \leq 1$, where c_{min} and c_{max} are the minimum and maximum recruiting costs among all users in U, respectively. Hence, Algorithm 1 achieves the approximation ratio $\frac{c_{min}}{c_{max}}$. □

5.2 Analysis on Multi-Round Strategy

Lemma 2. *For an arbitrary user set S, the expected value of QoD is monotonically non-decreasing with the objective ability vector $\boldsymbol{\rho}$, i.e., $\hbar(\boldsymbol{\alpha},\boldsymbol{\rho}) \leq \hbar(\boldsymbol{\alpha},\boldsymbol{\rho}')$ if $\rho_i \leq \rho_i'$, $\forall u_i \in S$.*

Proof. The monotonicity is obvious according to Eq. (21). □

Lemma 3. *For an arbitrary user set S, $\hbar(\boldsymbol{\alpha},\boldsymbol{\rho})$ in Eq. (21) achieves bounded smoothness such that for any two objective ability vectors $\boldsymbol{\rho}$ and $\boldsymbol{\rho}'$, if $\max_{u_i\in S}|\rho_i-\rho_i'| \leq \vartheta$, where ϑ is a constant, we have $|\hbar(\boldsymbol{\alpha},\boldsymbol{\rho})-\hbar(\boldsymbol{\alpha},\boldsymbol{\rho}')| \leq \hbar(\boldsymbol{\alpha},\boldsymbol{\vartheta})$, where $\boldsymbol{\vartheta}$ indicates a new objective ability vector such that $\rho_i=\vartheta, \forall u_i \in S$.*

Proof. Since $\hbar(\boldsymbol{\alpha},\boldsymbol{\rho}) = \sum_{u_i\in S}\bar{\alpha}_i\cdot\rho_i$, if $\max_{u_i\in S}|\rho_i-\rho_i'| \leq \vartheta$, we have $|\hbar(\boldsymbol{\alpha},\boldsymbol{\rho})-\hbar(\boldsymbol{\alpha},\boldsymbol{\rho}')| = \sum_{u_i\in S}\bar{\alpha}_i\cdot|\rho_i-\rho_i'| \leq \sum_{u_i\in S}\bar{\alpha}_i\cdot\vartheta = \hbar(\boldsymbol{\alpha},\boldsymbol{\vartheta})$, proved. □

Theorem 5. *The multi-round recruitment strategy achieves the regret of Algorithm 3 in K rounds at most $(\frac{6\ln K}{f^{-1}(\Delta_{min})^2}+\frac{\pi^2}{3}+1)\cdot\Delta_{max}\cdot\Delta_{ratio}$.*

Proof. Suppose that the function $f(\boldsymbol{\rho}) = \hbar(\boldsymbol{\alpha}^{\mathbf{0}},\boldsymbol{\rho})$, where $\boldsymbol{\alpha}^{\mathbf{0}}$ denotes the initial prior knowledge of likelihood matrix. Thus, $f(\boldsymbol{\rho})$ achieves monotonicity and bounded smoothness due to Lemmas 2 and 3. According to [21], we have

$$\begin{aligned} Reg(K) &= K\cdot\frac{c_{min}}{c_{max}}\cdot opt - \sum_{t=0}^{K} Q^t(S^t) \\ &\leq K\cdot\frac{c_{min}}{c_{max}}\cdot opt - \Delta_{ratio}\cdot\sum_{t=0}^{K} f(\boldsymbol{\rho}^t) \\ &\leq \left(\frac{6\ln K}{f^{-1}(\Delta_{min})^2}+\frac{\pi^2}{3}+1\right)\cdot\Delta_{max}\cdot\Delta_{ratio}. \end{aligned} \tag{27}$$

$\Delta_{min} = \frac{c_{min}}{c_{max}}\cdot opt - \max_{S^t}\{Q^t(S^t)\}$, $\Delta_{max} = \frac{c_{min}}{c_{max}}\cdot opt - \min_{S^t}\{Q^t(S^t)\}$ and $\Delta_{ratio} = \min_{S^t}\{\frac{\hbar(\boldsymbol{\alpha}^t,\boldsymbol{\rho}^t)}{f(\boldsymbol{\rho}^t)}\}$ for $t\in[0,K]$, where c_{min}/c_{max} is the approximation ratio of Algorithm 1, opt denotes the QoD value of the optimal user group with the real likelihood matrix, and S^t is the user group selected by Algorithm 1 in round t. So the regret bound is proved. □

5.3 Analysis on Payment Strategy

Lemma 4. *Algorithm 1 is cost-monotone.*

Proof. Suppose that a user u_i is recruited by the platform, according to Algorithm 1, we obtain

$$w(S,u_i) \geq w(S,u), \forall u \in V\setminus S, u\neq u_t. \tag{28}$$

If u_i decreases its cost to c_i^* $(\leq c_i)$, we have $w(S,u_i^*) = \sum_{u_j\in S} w_{ij}^* = \sum_{u_j\in S}(\rho_i/c_i^*+\rho_j/c_j)\cdot\alpha_{ij} \geq w(S,u_i)$. Thus,

$$w(S,u_i^*) \geq w(S,u_i) \geq w(S,u), \forall u\in V\setminus S, u\neq u_t. \tag{29}$$

Then, u_i will still be recruited according to Algorithm 1, i.e., Algorithm 1 is cost-monotone when recruiting users. □

Lemma 5. *The payment got by Algorithm 4 is the critical value.*

Proof. We call a value of a user critical if and only if it meets the following two conditions: 1) If a user's cost is greater than its critical value, then the user will not be recruited; 2) If a user's cost is no less than its critical value, then the user will be recruited [42]. Next, we prove that the payment calculated by Algorithm 4 is the critical value of a recruited user. Assume that a user u_i with the cost c_i is recruited by the platform and gets the payment p_i. We keep the other conditions unchanged and let u_i report the fake cost c_i^* to re-execute the recruitment process.

1) When $c_i \leq c_i^* \leq p_i$, we prove that u_i can still be recruited. Assume that u_j is the user who is added to the vertex set S (line 7 in Algorithm 1) after u_i. Then, when u_i reports the fake cost c_i^*, we obtain

$$\begin{aligned} w(S,u_i) &= \sum_{u_k\in S} w_{ik} = \sum_{u_k\in S}(\rho_i/c_i^*+\rho_k/c_k)\cdot\alpha_{ik} \\ &\geq \rho_i/p_i\cdot\sum_{u_k\in S}\alpha_{ik} + \sum_{u_k\in S}\rho_k/c_k\cdot\alpha_{ik} \\ &\geq w(S,u_j) - \sum_{u_k\in S}\rho_k/c_k\cdot\alpha_{ik} + \sum_{u_k\in S}\rho_k/c_k\cdot\alpha_{ik} \\ &= w(S,u_j), \end{aligned} \tag{30}$$

where the last inequality holds because we have $p_i \geq \rho_i\cdot\sum_{u_k\in S}\alpha_{ik}/(w(S,u_j)-\sum_{u_k\in S}\rho_k\cdot\alpha_{ik}/c_k)$ according to Algorithm 4 (lines 8-10). Thus, we obtain $w(S,u_i) \geq w(S,u_j)$ and $\mathcal{C}+c_i \leq \mathcal{B}$. According to Algorithm 1, u_i will be recruited by the platform with the cost.

2) When $c_i^* \geq p_i$, we prove that u_i will never be recruited. According to Eq. (30), we obtain $w(S,u_i) \leq w(S,u_j)$ when $c_i^* \geq p_i$, which means that in the current iteration the platform attempts to recruit u_j rather than u_i. Similarly, in each iteration of Algorithm 1 (lines 4-9), there is always a user who will defeat u_i just like u_j because $p_i = \max\{p\}$ according to Algorithm 4 (lines 9-10).

In summary, if reporting a cost that is no less than the payment, the user will be recruited, otherwise the user will not be recruited. Thus, the payment calculated by Algorithm 4 is the critical value. □

Theorem 6. *Algorithm 4 ensures that users obtain the maximum profits when they report their costs truthfully, i.e., truthfulness.*

Proof. Based on Lemmas 4 and 5, we have proved that Algorithm 1 is *cost-monotone* and the payment calculated by Algorithm 4 is the *critical value*. According to Myerson's principle [42], the payment can ensure that users will get the maximum profits only when they report their costs truthfully.

Additionally, as shown in Lemma 5, we have proved that when a user reports a fake cost between its cost and payment $c_i \leq c_i^* \leq p_i$, it will also be recruited and paid. Actually, each user will still report its cost rather than a fake cost because the payment received by the user is always p_i as long as the user is recruited and does not depend on the reported cost. In addition, according to Lemma 4, a user with a low cost is more likely to be recruited. Thus, each user prefers to reporting their cost truthfully to be recruited as much as possible. Truthfulness is proven. □

Theorem 7. *Algorithm 4 achieves individual rationality.*

Proof. Assume that u_i is a recruited user and u_j is the user who is added to the vertex set S (line 7 in Algorithm 1) after u_i. According to Algorithm 4, the payment of u_i satisfies

$$\begin{aligned} p_i &\geq \rho_i \cdot \sum_{u_k \in S} \alpha_{ik} \Big/ \left(w(S, u_j) - \sum_{u_k \in S} \rho_k \cdot \alpha_{ik}/c_k \right) \\ &\geq \rho_i \cdot \sum_{u_k \in S} \alpha_{ik} \Big/ \left(\sum_{u_k \in S} \rho_i/c_i \cdot \alpha_{ik} \right) = c_i. \end{aligned} \tag{31}$$

The last inequality holds because u_i is added to S before u_j, which means that $w(S, u_i) \geq w(S, u_j)$ in the current iteration, i.e., $w(S, u_j) \leq \sum_{u_k \in S} (\rho_i/c_i + \rho_k/c_k) \cdot \alpha_{ik}$. Therefore, the payment is always no less than the cost, and individual rationality is proven. □

6 Performance Evaluation

6.1 Datasets and Settings

In the simulations, we adopt three widely-used real-world datasets: Brightkite [43], Gowalla [43], and Foursquare [44]. Brightkite includes 58,228 nodes (users), 214,078 edges (collaboration likelihood), and a total of 4,491,143 check-in records over the period of Apr. 2008 - Oct. 2010. Similarly, Gowalla consists of 196,591 nodes, 950,327 edges, and 6,442,890 check-in records. Foursquare contains 456,988 check-in records made by 10,162 users. These three datasets mainly provide two kinds of information: one is the user's movement information, which can be used for modeling the user's objective ability, and the other is the social relationship between users, which is suitable for measuring the collaboration likelihood between users. For example, if two users are friends, the collaboration likelihood between them is high. Therefore, these three datasets are suitable for our simulation. As shown in Fig. 7, the blue nodes denote some users' geographic locations over a period of time in three datasets respectively, and the red circles indicate the areas with concentrated check-ins.

(a) Brightkite (b) Gowalla (c) Foursquare

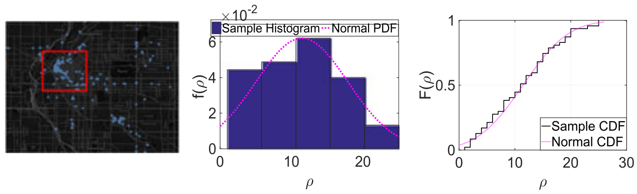
Fig. 7. The distribution of check-ins of three datasets.

TABLE 1
List of Key Notations

Notation	Description
U, T	The set of users, rounds
$\mathcal{B}$	The budget of each round
S^t	The set of selected users (vertices) in round t
$Q(S^t)$	The number of times the task is performed by users in S^t
ρ_i	The objective ability of u_i
c_i	The cost of recruiting u_i
α_{ij}	The collaboration likelihood between u_i and u_j
$\bar{\alpha}_i$	The average collaboration likelihood among u_i and others
$k_{i,t}$	The number of times that user u_i has been recruited before round t
$\rho_{i,r}$	The actual observed objective ability of u_i at the time of its rth recruitment
$\boldsymbol{\rho}^t, \boldsymbol{\alpha}^t$	The estimated objective ability vector and collaboration likelihood matrix in round t
w_{ij}	The weight of the edge between u_i and u_j
w_i	The total weights of the edges among u_i and all other vertices in V
$w(S, u)$	The weights of all edges between the vertex set S and vertex u
$\hbar, J$	The hypothetical function and loss function in the batch gradient descent
$\boldsymbol{\rho}_o^t, Q_o^t$	The actual observed value of the objective ability vector and QoD in round t

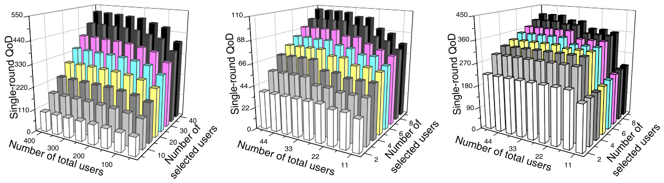
TABLE 2
Simulation Parameters

Parameter	Value
Number of total users U	$10 \sim 50$
Number of rounds T	$40 \sim 200$
Budget $\mathcal{B}$	$40 \sim 200$
Cost distribution c	$1 \sim 60$
Accuracy ε	0.001
Step size η	0.1



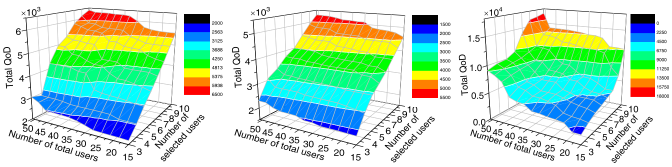
(a) Distrubution (b) PDF (c) CDF

Fig. 8. Test for normal distribution.



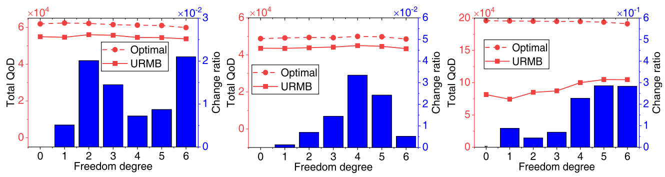
(a) Brightkite (b) Gowalla (c) Foursquare

Fig. 9. single-round QoD with the number of selected users.



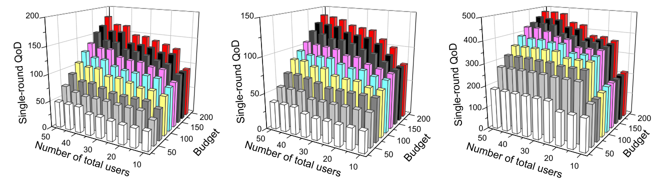
(a) Brightkite (b) Gowalla (c) Foursquare

Fig. 10. multi-round QoD with the number of selected users.



(a) Brightkite (b) Gowalla (c) Foursquare

Fig. 11. Freedom degree.



(a) Brightkite (b) Gowalla (c) Foursquare

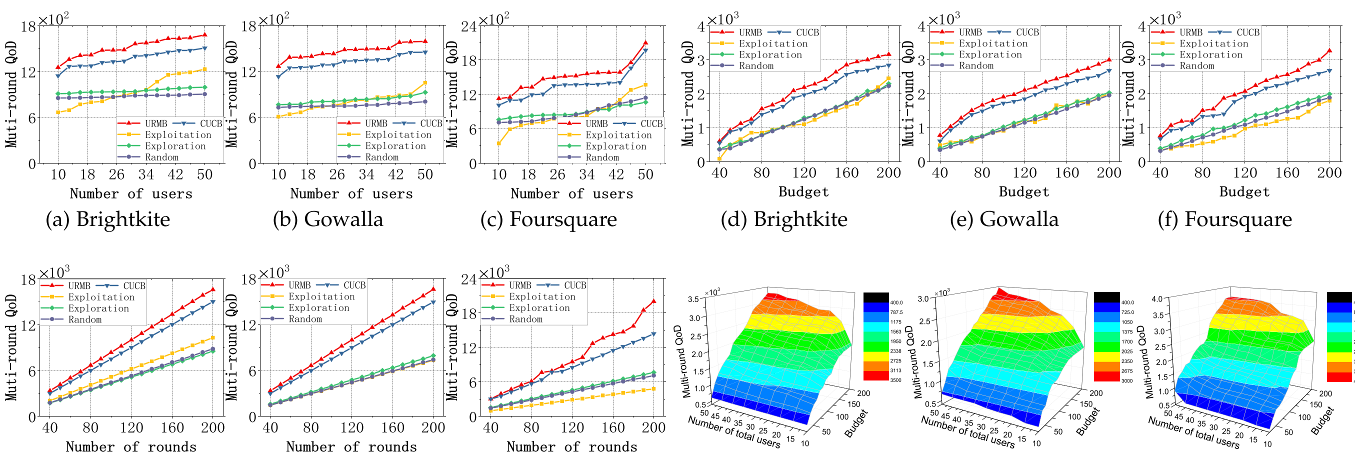
Fig. 12. Single-round QoD with the budget.

We introduce simulation settings as follows: (1) Round set T: we divide each user's check-in records in the datasets into K parts evenly according to time: Apr. 2008 - Oct. 2010 for Brightkite, Foursquare, and Dec. 2009 - Oct. 2010 for Gowalla. Each part corresponds to a round. (2) User objective ability vector $\boldsymbol{\rho}$: inspired by the settings in [45], we define that the sensing duration is 8:00 am $\sim$ 6:00 pm on each day, and use the frequency of each user passing through the sensing area to represent its objective ability ρ_i in round t. For convenience, users' objective abilities are denoted by a vector $\boldsymbol{\rho}$. (3) Real likelihood matrix $\boldsymbol{\alpha_r}$: A ground truth of the likelihood matrix is significant in the simulations to evaluate the effectiveness of proposed algorithms. Without loss of generality, we set the collaboration likelihood of a friend relationship to [0.5,1] uniformly, as well as the likelihood of a non-friend relationship to [0,0.5) uniformly inspired by the settings in [12]. Note that the generated matrix $\boldsymbol{\alpha_r}$ is only used as ground truth rather than prior knowledge. (4) Initial likelihood matrix $\boldsymbol{\alpha^0}$: without loss of generality, we set the initial likelihood matrix uniformly, i.e., each item in it is set to [0,1] uniformly and repeat simulations 1000 times to observe the average results. (5) The number of total users M: we filter the three datasets to pick users who have at least 1000 check-in records, and obtain a total of 580 users for Brightkite, 53 users for Gowalla, and 51 users for Foursquare. Thus, we change M in the range $[10, 50]$ inspired by the settings in [45]. (6) Budget $\mathcal{B}$: we change the budget from 40 to 200. (7) cost c of each user: in simulations, we generate each user's cost from $[1, 60]$ inspired by the settings in [3]. To fully evaluate the effectiveness of proposed algorithms, we consider three popular distributions of user costs: uniform, concave, and convex distributions as shown in Fig. 14. Through pre-simulations, we decide the default values of these parameters to clearly display the simulation results, i.e., $K = 200$, $M = 30$, and $B = 100$. The values of key simulation parameters are presented in Table 2.

6.2 Baselines and Metrics

We mainly compare URMB with following algorithms:

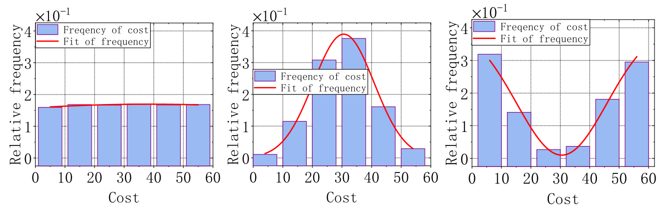
- *Optimal*: it searches globally to find the optimal group of users based on the real abilities.
- *CUCB* [21]: the state-of-the-art CMAB method, which uses upper bound confidence based reward to



(a) Brightkite (b) Gowalla (c) Foursquare (d) Brightkite (e) Gowalla (f) Foursquare

(g) Brightkite (h) Gowalla (i) Foursquare (j) Brightkite (k) Gowalla (l) Foursquare

Fig. 13. Multi-round QoD.



(a) Uniform (b) Concave (c) Convex

Fig. 14. Different distributions of user costs.

balance the trade-off of exploration and exploitation and requires an oracle to select super arms in each round. Yet, this method does not consider collaboration likelihood. To adjust this method to our case, let it select users with initial likelihood matrix $\boldsymbol{\alpha}^0$ and use Algorithm 1 in this paper as its oracle.

- *Exploitation*: it only selects users with strong current abilities, and ignores the exploration for users with poor current abilities but high potential, i.e., Algorithm 3 without the adjustment term in line 6.
- *Exploration*: it only selects users with the least number of exploration, i.e., the users with small $k_{i,t}$ values in Eq. (20).
- *Random*: it selects users randomly until the budget constraint $\mathcal{B}$ is exhausted.

Note that all the baselines are under the budget constraint $\mathcal{B}$. Afterwards, we use following metrics to evaluate the compared algorithms: (1) Single-round QoD Q_s for the single-round scenario, (2) Multi-round QoD Q for the multi-round scenario, (3) Running time for algorithms, (4) Regret according to Eq. (27), and (5) The loss J in Eq. (22).

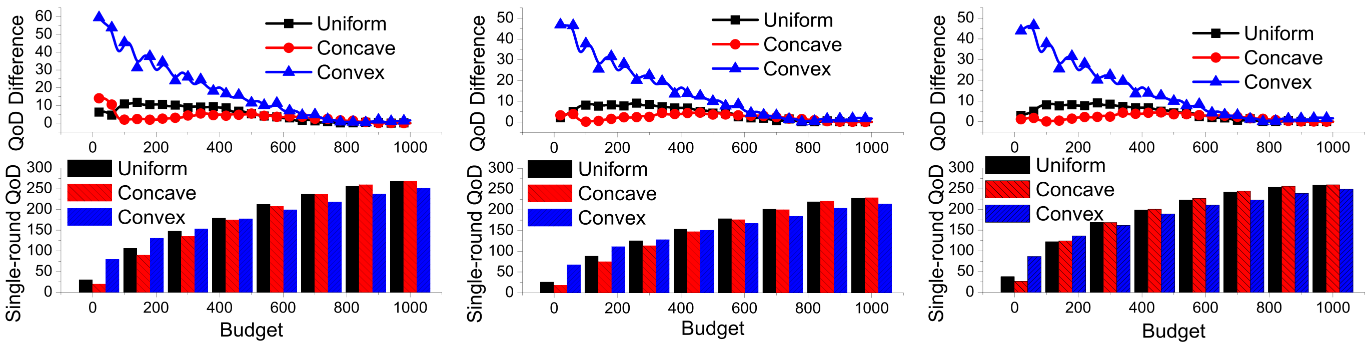
6.3 Evaluation Results

Before evaluating the performance of the URMB, we first verify that users' objective abilities (frequencies of passing through the sensing areas) over a period time follow a normal distribution, because only then can we update each user's estimated objective ability according to Eq. (20). As shown in Fig. 8, we adopt a total of 2094 check-ins of a user u_i in Brightkite in Fig. 8a (Denver), where the red rectangle indicates the sensing area and blue nodes denote its locations over a period of time. Based on the check-in data, we can get the user's objective ability in each round, which can be expressed as the vector $\boldsymbol{\rho_i}$, and $|\boldsymbol{\rho_i}| = K$. The frequency histogram graph and the probability density function (PDF) of the normal distribution with the mean and standard deviation of $\boldsymbol{\rho_i}$ are shown in Fig. 8b, and the cumulative distribution functions (CDF) are shown in Fig. 8c. Clearly, the objective abilities in K rounds of each user follow the normal distribution. Next, we conduct simulations and evaluate the performance of URMB in the single-round and multi-round scenarios.

In previous work [1], we assume that the platform can only recruit a fixed number of users, i.e., the number of selected users $|S^t|$ in Eq. (2) is fixed. In this case, we conduct extensive simulations and part of the results are shown in Figs. 9, 10, and 11. Specifically, Fig. 9 illustrates that the single-round QoD shows an upward trend with the increase of the number of total users and the number of selected users. Similarly, Fig. 10 shows the changing trend of the multi-round QoD. This indicates that our strategy URMB can handle the situation when the number of selected users $|S^t|$ is fixed. Then, we further illustrate that even if the number of selected users $|S^t|$ changes dynamically in each round, URMB can still achieve a good performance in the actual multi-round recruitment process in Fig. 11. The freedom degree represents the degree that the actual number of selected users deviates from the fixed number $|S^t|$.

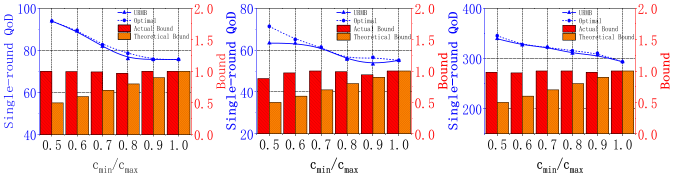
In fact, the scenario shown in Figs. 9, 10, and 11 is a special case of this paper, i.e., the cost of each user is the same and equal to 1, and the number of selected users actually corresponds to the budget $\mathcal{B}$. In this paper, we further consider the scenario where user costs are different. First, we generate each user's cost based on the uniform distribution as shown in Fig. 14a and conduct simulations to verify the performance of URMB in the single-round scenario under the budget constraint.

Impact of Parameters on the Single-Round QoD. Based on three datasets, we change the values of budget $\mathcal{B}$ and the number of total users M, and conduct simulations in the single-round scenario ($K = 1$). For Brightkite in Fig. 12a, we find that when M changes from 10 to 50 and $\mathcal{B}$ changes from 40 to 200, the single-round QoD Q_s shows a gradual upward trend, which is reasonable because when M increases, the solution space of Algorithm 1 is larger than before so that Algorithm 1 can find the better user group. Moreover, when $\mathcal{B}$ rises, the single-round QoD increases because Algorithm 1 can recruit more users than before when budget is larger. Similarly, we conduct simulations in Figs. 12b for Gowalla and 12c for Foursquare, and get similar results.



(a) Brightkite (b) Gowalla (c) Foursquare

Fig. 15. Cost distribution in single-round scenario.



(a) Brightkite (b) Gowalla (c) Foursquare

Fig. 16. Upper Bound in single-round scenario.

TABLE 3
Running Time of Three Datasets

Running time (ms)	$\mathcal{B}$ (URMB)				$\mathcal{B}$ (Optimal)			
	40	80	120	160	40	80	120	160
Brightkite	7	8	8	19	4	7	43	725
Gowalla	9	9	10	22	4	6	50	685
Foursquare	8	9	10	30	3	5	52	642

Impact of Parameters on the Multi-Round QoD. At first, when $\mathcal{B} = 100$ and $K = 200$, we change the number of total users M from 10 to 50 and conduct simulations. The results in Figs. 13a, 13b, and 13c show that the multi-round QoDs of all strategies increase monotonically with M, and URMB has advantages over other strategies. What's more, CUCB is the second-best strategy since it considers both exploitation and exploration when selecting users, but it ignores the impact of the collaboration likelihood of each user. Meanwhile, the results of Exploitation are not good because pure exploration ignores the users who have great potential but are not explored much. Obviously, Exploration and Random strategies have close and low QoDs because they do not tend to select users with good abilities.

Next, when $M = 50$ and $K = 200$, we change the budget $\mathcal{B}$ from 40 to 200 and conduct simulations. From the results in Figs. 13d, 13e, and 13f, we can conclude that the multi-round QoDs of all strategies are on the rise with the increase of $\mathcal{B}$, and URMB always outperforms other strategies. However, the QoD differences among these strategies are not obvious because when $\mathcal{B}$ is large enough, the number of selected users are close to M, therefore, the user groups selected by these strategies are similar. Especially when all the M users are selected, strategies get the same value of QoD. Then, we consider the number of total users and the budget together as shown in Figs. 13j, 13k, and 13l, where the multi-round QoD shows an upward trend with the increase of M and $\mathcal{B}$.

Subsequently, we improve the number of rounds K when $M = 50$ and $\mathcal{B} = 100$ in Figs. 13g, 13h, and 13i. The multi-round QoDs of all strategies rise linearly. What's more, we find that the larger the number of rounds, the greater the difference between URMB and CUCB because more rounds mean more feedback, which makes the performance of the update strategy (Algorithm 2) better so that the platform can estimate each user's objective ability and collaboration likelihood more accurately and select the better user group.

Afterwards, we not only consider the uniform distribution of user costs, but also consider the concave distribution in Fig. 14b and the convex distribution in Fig. 14c. Next, we conduct simulations based on three different distributions.

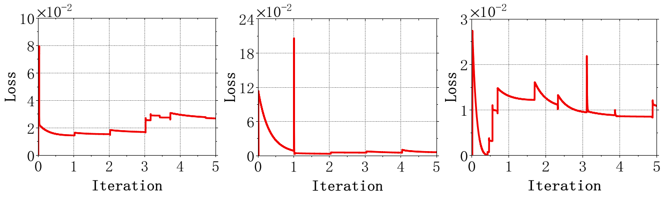
Impact of the Cost Distribution on the Single-Round QoD. As shown in Fig. 14, in the simulations, we generate the cost of each user $c_i \in [1, 60]$ from three distributions: uniform, concave, and convex distributions. Then we change the $\mathcal{B}$ from 40 to 200 and observe the simulation results as shown in Fig. 15. For the Brightkite in Fig. 15a, the upper part of the graph represents the single-round QoD difference between URMB with the edge weight in Eq. (6) and URMB with the edge weight in Eq. (10), which actually reflects the difference between only considering user ability and considering both the user ability and user cost. It is obvious that the QoD difference shows a downward trend with the increase of $\mathcal{B}$. Specifically, when the budget is small, considering both the ability and cost is better because some users with strong ability but high cost can't be recruited. When the budget is large, these users can also be added. Thus, the difference increasingly decreases. Furthermore, we can find that the QoD difference of the convex distribution is always larger than other distributions because in the convex distribution, the user cost is either very high or very low. Therefore, it is likely that a user has strong ability but high costs. The lower part of Fig. 15a illustrates the change trend of the single-round QoD of the three distributions. We find that at first the convex distribution is better than the other two distributions. This is because, as mentioned above, the convex distribution polarizes user costs, so there will be many good users and many poor users. Later, as the budget becomes larger, the QoDs of the three distributions are similar, and the concave distribution is better. This is because the concave makes user costs concentrated, and when the budget is enough, the user's ability has a greater impact on the recruitment result, so the single-round QoD is better than others. Similar results are shown in Figs. 15b for Gowalla and Fig. 15c for Foursquare.

Next, we evaluate the running time, approximation ratio and regret bound of URMB based on three datasets.

Impact of the User Cost on the Approximation Ratio. In Theorem 3, we have proved that URMB in the single-round scenario (Algorithm 1) achieves the approximation ratio c_{min}/c_{max}, where c_{min} and c_{max} are the minimum and

TABLE 4
Regret on Three Datasets

regret/optimal	$\mathcal{B}$ ($M = 50$)							M ($\mathcal{B} = 100$)						
	40	60	80	100	120	140	160	30	32	34	36	38	40	42
Brightkite	0.075	0.069	0.061	0.096	0.071	0.062	0.085	0.167	0.146	0.130	0.133	0.140	0.129	0.124
Gowalla	0.098	0.086	0.072	0.064	0.054	0.071	0.078	0.186	0.170	0.113	0.112	0.120	0.143	0.115
Foursquare	0.391	0.373	0.376	0.375	0.359	0.391	0.359	0.236	0.403	0.360	0.374	0.235	0.314	0.280



(a) Brightkite (b) Gowalla (c) Foursquare

Fig. 17. Loss in multiple rounds.

maximum recruiting costs among all users in U. As shown in Fig. 16, we change the c_{min}/c_{max} from 0.5 to 1 where user costs follow the uniform distribution, and conduct the simulations while $\mathcal{B}=100$, $M=30$. Through the results in Fig. 16a, we can find that the actual bound of URMB is always less than the theoretical bound c_{min}/c_{max} and close to 1, which indicates the effectiveness of URMB. Moreover, the single-round QoDs of both optimal and URMB strategies decline slightly because we rise the value of c_{min}/c_{max} by increasing the value of c_{min} while c_{max} is unchanged, which leads to a larger average user cost.

Impact of the Budget on the Running Time. The computational complexity is a critical metric for the algorithm. In order to evaluate the performance of URMB in the single-round scenario, we keep the number of total users $M=300$ unchanged, and change $\mathcal{B}$ from 20 to 100 to conduct the simulations of URMB and optimal strategies. As shown in Table 3, for all three datasets, when $\mathcal{B}$ is small, the running time of the optimal strategy is similar to that of URMB because the solution space is relatively small. However, when $\mathcal{B}$ grows, the running time of the optimal strategy will rise explosively, but the running time of URMB goes up steadily. Consequently, URMB is efficient to find the approximate solution.

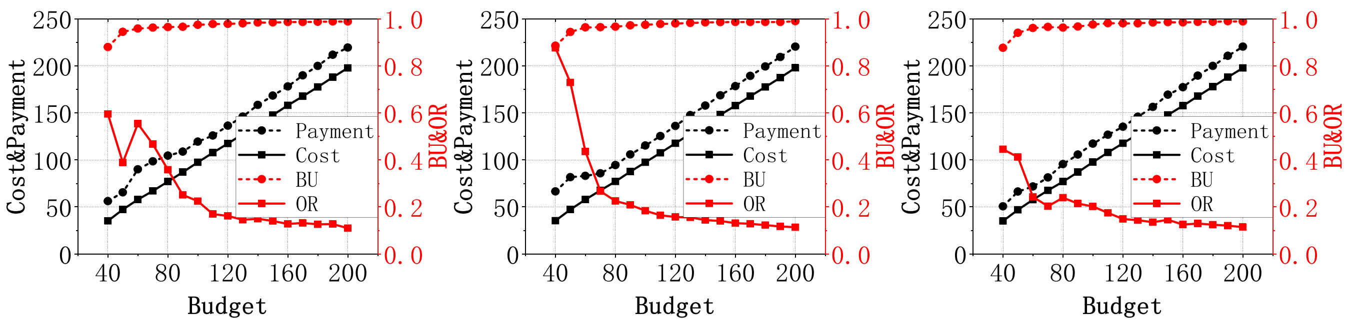
Impact of the Number of Rounds and Budget on the Regret. We change the values of $\mathcal{B}$ from 40 to 200 when $M=50$, and $K=100$, and conduct simulations. The value of the regret in Eq. (27) is related to some factors: the number of rounds K, the difference between *opt* and the expected QoD in each round, the difference between the initial likelihood matrix $\boldsymbol{\alpha}_0$ and the real likelihood matrix $\boldsymbol{\alpha}_r$, and the approximation ratio of Algorithm 1. Without loss of generality, we show the ratio of regret to optimal in Table 4. The values of the regret ratio for Brightkite and Gowalla are small, while the values for Foursquare are large because the difference between $\boldsymbol{\alpha}_0$ and $\boldsymbol{\alpha}_r$ in Foursquare is large, but all of them satisfy the regret bound in Theorem 4.

Impact of the Iteration on the Loss. In the simulations, we record the values of the loss function J in Eq. (22) as shown in Fig. 17, which is actually the difference between the current likelihood matrix $\boldsymbol{\alpha}$ and the real likelihood matrix $\boldsymbol{\alpha}_r$. In Fig. 17, the x-axis represents the number of iterations in BGD, and the y-axis represents the value of loss function J. We find that the value of J often rises suddenly and then gradually decreases. This is because the sudden increase in J indicates that the new observed feedback is added after a round, which makes the difference between expected QoD and observed QoD larger. Moreover, the value of J decreases because the value of $\boldsymbol{\alpha}$ is approaching that of $\boldsymbol{\alpha}_r$ through the update process as shown in Algorithm 2, which indicates that our update strategy is effective.

Finally, we evaluate the performance of the payment calculation strategy in Algorithm 4 from three aspects: overpayment ratio, truthfulness and individual rationality.

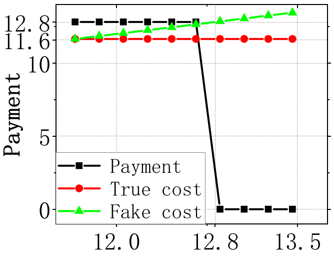
Evaluation on Overpayment Ratio and Budget Utilization. Overpayment ratio (OR) $\sum_{u_i\in S}(p_i-c_i)/\sum_{u_i\in S}c_i$ is an important metric to evaluate whether the payment is reasonable, and budget utilization (BU) indicates the utilization ratio of the budget $\mathcal{B}$, i.e., $\sum_{u_i\in S}c_i/\mathcal{B}$. As shown in Fig. 18, as the budget increases, OR continues to decrease and is basically less than 0.2, which illustrates that the payment is close to the cost, and the platform does not need to spend too much payment to ensure truthfulness and individual rationality, i.e., Algorithm 4 is designed reasonably. When the budget is small, OR is relatively large because when the budget is small, a few users are selected. However, Algorithm 4 determines the payment based on other users' information, so the payment is not accurate. In addition, BU gradually grows and approaches 1 as the budget increases because when the budget is large, more combinations of users can be recruited, so the budget is used better. It can be seen that our algorithm can make good use of the budget $\mathcal{B}$.

Evaluation on Truthfulness. To evaluate the truthfulness, after conducting single-round user recruitment, we randomly select a selected user called the winner and an unselected user called the loser. For the winner whose real cost is 11.6 and payment determined by Algorithm 4 is 12.8 (i.e., the critical value), we make the winner report a fake cost to the platform and re-execute the user recruitment process maintaining other conditions unchanged. The results are shown in Fig. 19a, where we discover that the payment for the winner is always greater than the fake cost when the fake cost increases from 11.6 to 12.8. However, when the fake cost is greater than the critical value 12.8, we discover

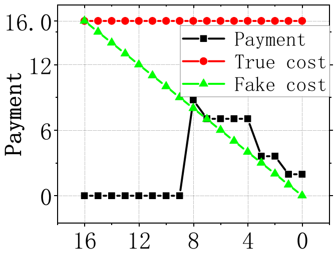


(a) Brightkite (b) Gowalla (c) Foursquare

Fig. 18. Payment, overpayment ratio and budget utilization.



(a) Winner



(b) Loser

Fig. 19. Truthfulness of the winner and loser.

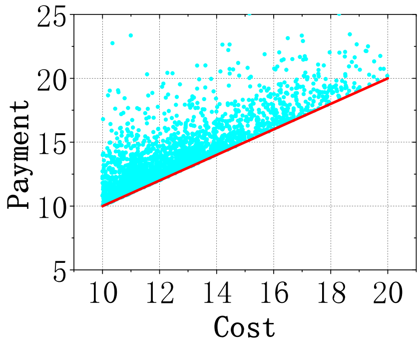


Fig. 20. Individual rationality.

that the winner will not be selected and get the payment 0, which validates Lemma 5. Similarly, for the loser whose real cost is 16.0 and payment is 0, we make the loser report a fake cost from 16.0 to 0, and the results are shown in Fig. 19b. We discover that when the fake cost is greater than 8.0, the loser can be selected by the platform and get a payment. In addition, the payment for the loser is always close to the fake cost, which indicates that the payment is well designed. Moreover, even though the loser can be selected through reporting the fake cost, the payment it gets is always less than its real cost. Thus, Algorithm 4 can effectively avoid malicious competition from users while achieving truthfulness.

Evaluation on Individual Rationality. In the simulations, we record each user's real cost and payment, and results are shown in Fig. 20, where each point represents a selected user, the x-axis indicates the cost, the y-axis indicates the payment, and the red line represents the function of $payment = cost$. The results validate Theorem 7. We discover that all points are in the upper area of the red line, which means that the payments of all selected users are not less than their costs, i.e., individual rationality. Moreover, most of the points are close to the red line, which means that calculated payments are close to users' real costs. In addition, based on three datasets, we change the budget from 40 to 200 and observe the total cost and total payment of selected users as shown in Fig. 18. We find that the total payment is not less than the total cost, which means Algorithm 4 achieves individual rationality.

7 Conclusion

In this paper, we argue that when selecting a group of users under the budget constraint to perform a cooperative task, we should consider not only a user's objective ability but also its collaboration likelihood with others. In the single-round scenario, we convert the recruitment problem into the min-cut problem based on each user's ability and cost, and propose an algorithm based on graph theory to find the approximate solution. Furthermore, in the multi-round scenario, we propose the multi-round user recruitment strategy based on the combinatorial multi-armed bandit model under the budget constraint (URMB) to balance the trade-off between exploration and exploitation, and prove that it achieves a tight regret bound. Then, we propose a graph-based payment strategy to achieve truthfulness and individual rationality. Finally, we conduct extensive simulations based on three real-world datasets, and the results show that URMB always outperforms other strategies.

Acknowledgments

Part of this paper appeared in IEEE SECON 2020[1]

References

[1] H. Wang, Y. Yang, E. Wang, W. Liu, Y. Xu, and J. Wu, "Combinatorial multi-armed bandit based user recruitment in mobile crowdsensing," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 179–188.
[2] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *Commun. Mag. IEEE*, vol. 48, no. 9, pp. 140–150, Sep. 2010.
[3] G. Gao, M. Xiao, J. Wu, L. Huang, and C. Hu, "Truthful incentive mechanism for nondeterministic crowdsensing with vehicles," *IEEE Trans. Mobile Comput.*, vol. 17, no. 12, pp. 2982–2997, Dec. 2018.
[4] R. Zhou, Z. Li, and C. Wu, "A truthful online mechanism for location-aware tasks in mobile crowd sensing," *IEEE Trans. Mobile Comput.*, vol. 17, no. 8, pp. 1737–1749, Aug. 2018.
[5] Y. Gao, *et al.*, "Mosaic: A low-cost mobile sensing system for urban air quality monitoring," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, 2016, pp. 1–9.
[6] P. Zhou, Y. Zheng, and M. Li, "How long to wait?: Predicting bus arrival time with mobile phone based participatory sensing," in *Proc. 10th Int. Conf. Mobile Syst. Appl. Serv.*, 2012, pp. 379–392.
[7] Y. Liu *et al.*, "Vernier: Accurate and fast acoustic motion tracking using mobile devices," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 754–764, Feb. 2021.
[8] E. Wang, Y. Yang, J. Wu, W. Liu, and X. Wang, "An efficient prediction-based user recruitment for mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 17, no. 1, pp. 16–28, Jan. 2018.
[9] H. Chen, B. Guo, Z. Yu, and Q. Han, "Crowdtracking: Real-time vehicle tracking through mobile crowdsensing," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7570–7583, Oct. 2019.
[10] Z. Wang, J. Li, J. Hu, J. Ren, Z. Li, and Y. Li, "Towards privacy-preserving incentive for mobile crowdsensing under an untrusted platform," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 2053–2061.
[11] Y. Liu, J. Han, and J. Wang, "Rumor riding: Anonymizing unstructured peer-to-peer systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 3, pp. 464–475, Apr. 2011.
[12] Z. Wang, Y. Huang, X. Wang, J. Ren, Q. Wang, and L. Wu, "Socialrecruiter: Dynamic incentive mechanism for mobile crowdsourcing worker recruitment with social networks," *IEEE Trans. Mobile Comput.*, vol. 20, no. 5, pp. 2055–2066, May 2021.
[13] J. Wang, J. Tang, D. Yang, E. Wang, and G. Xue, "Quality-aware and fine-grained incentive mechanisms for mobile crowdsensing," in *Proc. IEEE 36th Int. Conf. Distrib. Comput. Syst.*, 2016, pp. 354–363.
[14] X. Gao, S. Chen, and G. Chen, "Mab-based reinforced worker selection framework for budgeted spatial crowdsensing," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 3, pp. 1303–1316, Mar. 2022.
[15] H. Zhao, M. Xiao, J. Wu, Y. Xu, H. Huang, and S. Zhang, "Differentially private unknown worker recruitment for mobile crowdsensing using multi-armed bandits," *IEEE Trans. Mobile Comput.*, vol. 20, no. 9, pp. 2779–2794, Sep. 2021.

[16] Y. Yang, W. Liu, E. Wang, and J. Wu, "A prediction-based user selection framework for heterogeneous mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 11, pp. 2460–2473, Nov. 2019.

[17] X. Wang, R. Jia, X. Tian, X. Gan, L. Fu, and X. Wang, "Location-aware crowdsensing: Dynamic task assignment and truth inference," *IEEE Trans. Mob. Comput.*, vol. 19, no. 2, pp. 362–375, Feb. 2020.

[18] H. Gao, C. H. Liu, J. Tang, D. Yang, P. Hui, and W. Wang, "Online quality-aware incentive mechanism for mobile crowd sensing with extra bonus," *IEEE Trans. Mobile Comput.*, vol. 18, no. 11, pp. 2589–2603, Nov. 2019.

[19] A. V. Goldberg, "Finding a maximum density subgraph," EECS Dept., Univ. California, Berkeley, California, Tech. Rep. UCB/CSD-84–171, 1984.

[20] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1466–1478, Oct. 2012.

[21] W. Chen, Y. Wang, and Y. Yuan, "Combinatorial multi-armed bandit: General framework and applications," in *Proc. 30th Int. Conf. Int. Conf. Mach. Learn.*, 2013, pp. 151–159.

[22] Z. He, J. Cao, and X. Liu, "High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 2542–2550.

[23] Z. Wang, J. Zhao, J. Hu, T. Zhu, Q. Wang, J. Ren, and C. Li, "Towards personalized task-oriented worker recruitment in mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 20, no. 5, pp. 2080–2093, May 2021.

[24] Y. Wu, F. Li, L. Ma, Y. Xie, T. Li, and Y. Wang, "A context-aware multiarmed bandit incentive mechanism for mobile crowd sensing systems," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7648–7658, Oct. 2019.

[25] G. Gao, J. Wu, M. Xiao, and G. Chen, "Combinatorial multi-armed bandit based unknown worker recruitment in heterogeneous crowdsensing," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 179–188.

[26] M. Tokic, "Adaptive ϵ-greedy exploration in reinforcement learning based on value differences," in *Proc. Annu. Conf. Artif. Intell.*, 2010, pp. 203–210.

[27] R. Agrawal, "Sample mean based index policies by O(log n) regret for the multi-armed bandit problem," *Adv. Appl. Probability*, vol. 27, pp. 1054–1078, 12 1995.

[28] F. Li, J. Liu, and B. Ji, "Combinatorial sleeping bandits with fairness constraints," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 1702–1710.

[29] G. Kim and M. C. Paik, "Contextual multi-armed bandit algorithm for semiparametric reward model," in *Proc. Mach. Learn. Res. Int. Conf. Mach. Learn.*, 2019, pp. 3389–3397.

[30] A. Kagrecha, J. Nair, and K. P. Jagannathan, "Distribution oblivious, risk-aware algorithms for multi-armed bandits with unbounded rewards," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 11269–11278.

[31] A. Hüyük and C. Tekin, "Analysis of thompson sampling for combinatorial multi-armed bandit with probabilistically triggered arms," in *Proc. 23nd Int. Conf. Artif. Intell. Statist.*, 2019, pp. 1322–1330.

[32] Z. Zheng, Y. Peng, F. Wu, S. Tang, and G. Chen, "ARETE: On designing joint online pricing and reward sharing mechanisms for mobile data markets," *IEEE Trans. Mobile Comput.*, vol. 19, no. 4, pp. 769–787, Apr. 2020.

[33] Y. Liu, K. Liu, and M. Li, "Passive diagnosis for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 4, pp. 1132–1144, Aug. 2010.

[34] M. Stoer and F. Wagner, "A simple min-cut algorithm," *J. ACM*, vol. 44, no. 4, pp. 585–591, 1997.

[35] J. Edmonds and R. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *J. ACM*, vol. 19, no. 2, pp. 248–264, 2003.

[36] B. V. Gnedenko, A. N. Kolmogorov, J. L. Doob, and P.-L. Hsu, *Limit Distributions for Sums of Independent Random Variables*. Reading, MA, USA: Addison-Wesley, 1968.

[37] C. J. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. N. Hullender, "Learning to rank using gradient descent," in *Proc. J. Mach. Learn. Res. Int. Conf. Mach. Learn.*, 2005, pp. 89–96.

[38] Y. Zhang and X. Zhang, "Price learning-based incentive mechanism for mobile crowd sensing," *ACM Trans. Sensor Netw.*, vol. 17, no. 2, pp. 1–24, 2021.

[39] G. Gao, H. Huang, M. Xiao, J. Wu, Y. Sun, and S. Zhang, "Auction-based combinatorial multi-armed bandit mechanisms with strategic arms," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.

[40] P. Zhou, X. Wei, C. Wang, and Y. Yang, "k-level truthful incentivizing mechanism and generalized k-MAB problem," *IEEE Trans. Comput.*, early access, Aug. 18, 2021, doi: 10.1109/TC.2021.3105831.

[41] J. Xu, Z. Rao, L. Xu, D. Yang, and T. Li, "Incentive mechanism for multiple cooperative tasks with compatible users in mobile crowd sensing via online communities," *IEEE Trans. Mobile Comput.*, vol. 19, no. 7, pp. 1618–1633, Jul. 2020.

[42] R. B. Myerson, "Optimal auction design," *Math. Oper. Res.*, vol. 6, no. 1, pp. 58–73, 1981.

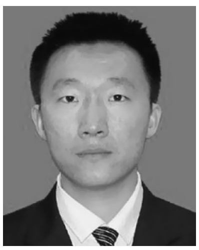
[43] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2021, pp. 1082–1090.

[44] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. Magnenat-Thalmann, "Time-aware point-of-interest recommendation," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2013, pp. 363–372.

[45] J. Wang, F. Wang, Y. Wang, D. Zhang, L. Wang, and Z. Qiu, "Social-network-assisted worker recruitment in mobile crowd sensing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 7, pp. 1661–1673, Jul. 2019.

Hengzhi Wang received the bachelor's degree in software engineering from Jilin University, Changchun, China, in 2017, where he is currently working toward the doctoral degree in computer science and technology. His current research interests include mobile crowdsensing and privacy preserving in mobile computing.

Yongjian Yang received the BE degree in automatization from the Jilin University of Technology, Changchun, China, in 1983, the ME degree in computer communication from the Beijing University of Post and Telecommunications, Beijing, China, in 1991, and the PhD degree in software and theory of computer from Jilin University, Changchun, China, in 2005. He is currently a professor and a PhD supervisor with Jilin University, the vice dean of Software College, Jilin University, the director of the Key Lab under the Ministry of Information Industry, the standing director of Communication Academy, and a member of the Computer Science Academy of Jilin Province. His research interests include network intelligence management, wireless mobile communication and services, and wireless mobile communication.



En Wang received the BE degree in software engineering, the ME degree in computer science and technology, and the PhD degree in computer science and technology from Jilin University, Changchun, in 2011, 2013, and 2016, respectively. He is currently an associate professor with the Department of Computer Science and Technology, Jilin University. He is also a visiting scholar with the Department of Computer and Information Sciences, Temple University, Philadelphia. His current research interests include the efficient utilization of network resources, scheduling and drop strategy in terms of buffer-management, energy-efficient communication between humancarried devices, and mobile crowdsensing.

Wenbin Liu received the BS degree in physics in 2012, and the ME degree in 2016 from the Department of Software, Jilin University, Changchun, where he is currently working toward the PhD degree with the Department of Computer Science and Technology. His current research focuses on mobile crowdsensing.

Yuanbo Xu received the BE degree in computer science and technology, the ME degree in computer science and technology, and the PhD degree in computer science and technology from Jilin University, Changchun, in 2012, 2015, and 2019, respectively. He is currently a postdoctoral with the Department of Artificial Intelligence, Jilin University. He is also a visiting scholar with Rutgers, New Jersey. He has authored or coauthored some research results on journals, such as TMM, TNNLS, and conference as ICDM. His research interests include applications of data mining, recommender system, and mobile computing.

Jie Wu (Fellow, IEEE) is currently the associate vice provost of international affairs with Temple University. He is also the director of the Center for Networked Computing and Laura H. Carnell professor with the Department of Computer and Information Sciences. Prior to joining Temple University, he was a program director of National Science Foundation and was a distinguished professor with Florida Atlantic University. He has regularly authored or coauthored in scholarly journals, conference proceedings, and books. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. He serves on several editorial boards, including the *IEEE Transactions on Service Computing* and *Journal of Parallel and Distributed Computing*. He was the general co-chair or chair of IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, and ACM MobiHoc 2014, and also the program co-chair of IEEE INFOCOM 2011 and CCF CNCC 2013. He was an IEEE Computer Society distinguished visitor, ACM distinguished speaker, and the chair of IEEE Technical Committee on Distributed Processing (TCDP). He is a CCF distinguished speaker. He was the recipient of 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.