

Acceptance-Aware Mobile Crowdsourcing Worker Recruitment in Social Networks

Liang Wang, Dingqi Yang, Zhiwen Yu, *Senior Member, IEEE*, Qi Han, *Senior Member, IEEE*, En Wang, Kuang Zhou, and Bin Guo, *Senior Member, IEEE*,

Abstract—With the increasing prominence of smart mobile devices, an innovative distributed computing paradigm, namely Mobile Crowdsourcing (MCS), has emerged. By directly recruiting skilled workers, MCS exploits the power of the crowd to complete location-dependent tasks. Currently, based on online social networks, a new and complementary worker recruitment mode, i.e., socially aware MCS, has been proposed to effectively enlarge worker pool and enhance task execution quality, by harnessing underlying social relationships. In this paper, we propose and develop a novel worker recruitment game in socially aware MCS, i.e., **Acceptance-aware Worker Recruitment (AWR)**. To accommodate MCS task invitation diffusion over social networks, we design a Random Diffusion model, where workers randomly propagate task invitations to social neighbors, and receivers independently make a decision whether to accept or not. Based on the diffusion model, we formulate the AWR game as a combinatorial optimization problem, which strives to search a subset of seed workers to maximize overall task acceptance under a pre-given incentive budget. We prove its NP hardness, and devise a meta-heuristic-based evolutionary approach named *MA-RAWR* to balance exploration and exploitation during the search process. Comprehensive experiments using two real-world data sets clearly validate the effectiveness and efficiency of our proposed approach.

Index Terms—Mobile Crowdsourcing, Worker Recruitment, Social Networks, Memetic Algorithm.

1 INTRODUCTION

WITH the dramatic proliferation of sensor-rich mobile devices and wireless communication technologies, a novel distributed problem-solving paradigm, namely *Mobile Crowdsourcing (MCS)* [1], has become a promising way leveraging the power of the crowd to accomplish location-dependent tasks in the real world. MCS applications usually contain three stakeholders: task owner, participant worker and the MCS platform. Generally, the MCS platform first outsources location-dependent tasks posted by the task owner to potential workers via an open call. The recruited workers then jointly accomplish tasks and obtain some payment as incentive for participation.

Worker recruitment is fundamental to a successful MCS campaign. Typically, MCS applications use a straightforward direct method, i.e., the platform directly selects and recruits appropriate individuals to carry out tasks [2], [3]. Until recently, a novel mode, namely Socially Aware MCS (a.k.a Word-of-Mouth (WoM)-based MCS) [4], [5] has emerged as an extension of the previous method. First of all, socially aware MCS elaborately recruits a limited number of

users as seed workers by harnessing some public information about potential workers, e.g., check-in history. The seed workers then freely recruit more individuals through their mobile social networks to contribute their efforts. In other words, beside executing the MCS tasks themselves, seed workers are granted the rights to privately enlist their social contacts over a larger scale, by exploiting their respective social relationship, such as kinship and friendship.

Compared with the direct mode, socially aware MCS has several advantages. First, strong social ties can motivate users to actively participate in MCS campaign and boost their participation level [6]. For instance, it has been shown in [7] that the higher the sociality, the better the completion ratio of MCS tasks. Second, by utilizing social relationships, users might impose certain pressure on their social contacts to better conduct MCS tasks, and thus enhance its quality [8]. Last, within the acquaintance network, the privacy concerns may be alleviated to some extent. In a nutshell, socially aware MCS can effectively enlarge crowd user pool [9] and facilitate high-quality task execution while preserving users privacy [10].

However, as an integration of MCS paradigm and social networking, socially aware MCS presents many open technical challenges, which must be systematically tackled. First, from a global perspective, how to model the process of MCS task invitation diffusion throughout mobile social networks? Different from common information propagation models in social networks, it is impossible for socially aware MCS to flood task invitations to everybody, due to the need for low intrusiveness, limited incentive budget, and task result quality requirements. Therefore, existing information propagation models cannot be directly adopted for socially aware MCS. Second, from a local perspective, interactions

- L. Wang, Z. Yu and B. Guo are with School of Computer Science, Northwestern Polytechnical University, Xi'an, China, 710129.
- D. Yang is with the State Key Laboratory of Internet of Things for Smart City, Department of Computer and Information Science, University of Macau, Macau SAR, China.
- Q. Han is with the Colorado School of Mines, 1500 Illinois St, Golden, CO 80401.
- E. Wang is with the College of Computer Science and Technology, Jilin University, Changchun, China, 130012.
- K. Zhou is with the School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an, China, 710129.
- *L. Wang is the corresponding author, Email: liangwang0123@gmail.com.

Manuscript received April 19, 2005; revised August 26, 2015.

only happen between the platform and participant workers in typical MCS applications. While in socially aware MCS, users interaction behavior may be an important factor [6] and should be incorporated. Therefore, the role of participant workers becomes more complicated as both inviters and invitees at the same time, and the relevant incentive mechanisms must be tailored accordingly. Finally, from the perspective of participant workers (including task inviters and invitees), various questions must be answered, such as, which social neighbors should be recruited as workers for each inviter, and which task invitations would be accepted by invitees?

Unfortunately, few systematic studies have examined the aforementioned issues. In this paper, we propose and develop a novel worker recruitment game in socially aware MCS, namely **Acceptance-aware Worker Recruitment (AWR)**, to guide the selection process of seed workers. To accommodate the MCS task diffusion process, we design a diffusion model, namely *Random Diffusion*, in which inviters randomly propagate task invitation to social neighbors, and invitation receivers make a decision whether to accept a task or not. By comprehensively investigating key factors: spatial proximity, social ties strength, inviter solicitation, we establish task acceptance estimation and the corresponding incentive mechanism. Afterwards, we formulate a combinatorial optimization problem named **RAWR**, which strives to search a subset of seed workers to maximize overall task acceptance under an incentive budget. Through theoretical analysis, we shown that our **RAWR** problem is NP-hard, and it is difficult to solve it using exact algorithms. Therefore, we propose an effective algorithm **MA-RAWR** to solve this problem; it is built on top of a population-based evolutionary method Memetic Algorithm (MA), by further balancing exploration by global search and exploitation by local refinement. Furthermore, we devise several enhancement strategies to improve its performance, based on problem-specific heuristics knowledge.

Specifically, we make the following contributions.

- We integrate social networks into mobile crowdsensing paradigm to build an acceptance-aware worker recruitment game for the first time.
- We devise a random diffusion model for MCS tasks in social networks, and formalize the acceptance-aware worker recruitment game for users, namely **RAWR** problem.
- Based on a meta-heuristic Memetic algorithm, we devise an effective **MA-RAWR** algorithm to tackle our **RAWR** problem. Moreover, several heuristics knowledge-based enhancement strategies are also proposed to further improve the performance of our **MA-RAWR** algorithm.
- We conduct extensive experiments using real social network data sets, and show the efficiency and effectiveness of our proposed approach.

The rest of this paper is organized as follows. Preliminary concepts are present in Section 2. Our problem definition is provided in Section 3, based on the devised diffusion mode. In Section 4, we propose a unified framework to solve our **RAWR** problem, by utilizing a meta-heuristic memetic algorithm. The extensive experimental results based on real

TABLE 1
Definitions of Notations

Symbol	Explanation
$\mathcal{U} = \{u_i\}$	Mobile Users
$\mathcal{T} = \{< \mathbf{T}_i >, \mathcal{C}_{max}\}$	MCS Tasks
$\mathcal{G} = \{\mathcal{U}, \mathcal{E}, \mathcal{W}\}$	Social Relationship Network
$\mathcal{A}_T^u$	User u 's Acceptance for Task $\mathcal{T}$
$\mathcal{SG}_u$	u 's Diffusion Tree
$\mathcal{A}_T(\mathcal{SG}_u)$	Total Acceptance of $\mathcal{SG}_u$
ℓ_{max}	Maximum Diffusion Distance
$\mathcal{H}_{u_i, u_j}$	Diffusion Path from u_i to u_j
$\mathbf{D} = [d_{i,j}]_{1 \leq i, j \leq \mathcal{U} }$	Diffusion Probability Matrix
$\mathcal{N}_x(\cdot)$	Neighborhood structures

social network data sets are provided in Section 5. Related work is summarized in Section 6. Finally, we briefly conclude this paper.

2 PRELIMINARY CONCEPTS

In this section, we introduce several key concepts related to our problem. Table 1 summarizes the main symbols used throughout the paper.

Definition 1 (Mobile Users). A group of mobile users $\mathcal{U} = \{u_1, \dots, u_n\}$ in social network applications can be recruited as workers in MCS campaign, where each recruited worker can disseminates task invitations to their social acquaintances and contributes effort to complete tasks. For an individual user u_i , some check-in records are available and represented as $\{(tm_{i_1}, loc_{i_1}, \varphi_{i_1}), \dots, (tm_{i_m}, loc_{i_m}, \varphi_{i_m})\}$, (1) $loc_{i_j} \in \mathcal{L}$, $1 \leq j \leq m$, denotes the spatial landmark user u_i has visited at time tm_{i_j} ; (2) $\varphi_{i_j} \in \Gamma$ denotes Point of Interests (POI) category tags associated with landmark loc_{i_j} , such as gym, museum, etc., with Γ representing a full set of POI category tags. $\square$

In addition to individual's check-in history, each user has strong or weak social ties with others. Formally, we define a social network $\mathcal{G} = \{\mathcal{U}, \mathcal{E}, \mathcal{W}\}$, where $e_{i,j} \in \mathcal{E}$ denotes an undirected edge, i.e., a social tie, connecting node u_i and u_j ; $w_{i,j} \in \mathcal{W}$ denotes a normalized weight of edge $e_{i,j}$ and $\mathcal{W} : \mathcal{E} \rightarrow \mathbb{R}$. Hereafter, we use the terms "mobile user" and "node" interchangeably. For the sake of illustration, $w_{i,j} > 0$ implies that there exists social tie $e_{i,j}$; otherwise, $e_{i,j}$ dose not exist. Formally, social tie strength $w_{i,j}$ is represented using Jaccard similarity coefficient as below:

$$w_{i,j} = \frac{N(u_i) \cap N(u_j)}{N(u_i) \cup N(u_j)}, \quad (1)$$

where $N(u_j)$ denotes the node u_j and its neighboring nodes. Intuitively, if user u_i and u_j share more neighboring nodes, the relative social tie would be strong.

Definition 2 (Mobile Crowdsourcing Task). Suppose there is a set of released mobile crowdsourcing tasks $\mathcal{T} = \{< \mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_l >, \mathcal{C}_{max}\}$, where $\mathbf{T}_i = (loc_i, \vec{P}_i)$, $loc_i \in \mathcal{L}$ is a specific spatial location as longitude and latitude; task topic $\vec{P}_i$ is characterized by a distribution of POI category tags, and will be explained later; $\mathcal{C}_{max}$ is the total incentive budget for all the involved tasks in $\mathcal{T}$, which is pre-specified by task owners. $\square$

Here, task topic $\vec{P}$ is built in the following way. We regard POI categories $\Gamma = \{\pi_1, \pi_2, \pi_3, \dots, \pi_q\}$ as topics in this study, where each element $\pi_i \in \Gamma$ denotes one POI category tag, such as outdoors, entertainment, and so on. Given the set of POI category tags Γ , task topics is represented as a distribution over the POI category tags [11], [12], e.g., $\vec{P} = \{\langle 0.3, \text{outdoors} \rangle, \langle 0.7, \text{entertainment} \rangle\}$.

Task Acceptance Estimation. To balance the workload and ensure a timely response, we stipulate that each user can only undertake one task in $\mathcal{T}$. Hence, when receiving $\mathcal{T}$'s invitation, every user needs to evaluate the tasks contained in $\mathcal{T}$, makes a decision on whether or not to accept one of them to conduct. Actually, many previous works have investigated the factors which might impact users' task acceptance in MCS campaign [13], [14]. Following the insights in them, we assume that each user is rational to make his/her acceptance decision to take one task, say T_i , according to the matching between the task requirements and their own context, e.g., spatial proximity and interest level. If a mobile user decides to undertake one task, they will try their best to accomplish it due to the implicit "social contract" from their referrers [15]. However, constrained by the customized specifications of the MCS tasks, their acceptance would vary from one user to another. It is most natural that mobile users who are close to T_i 's location loc_i and interested in its topic $\vec{P}_i$ have a greater probability to accept the task [9]. In the following, considering the specific requirements of task T_i , **task acceptance estimation** for mobile user u_j , i.e., $\mathcal{A}_{T_i}^{u_j}$, is quantified jointly from two dimensions: spatial proximity and interest level.

1) Spatial proximity: Following the practice in existing research [12], [16], we adopt a widely used decay formulation as below to calculate the likelihood that user u_j would like to visit T_i 's specified location loc_i ,

$$Sp(u_j, T_i) = \alpha * \exp(-\beta * \text{dist}(u_j, loc_i)), \quad (2)$$

where the distance measurement $\text{dist}(\cdot)$ employs Euclidean distance, the amplification factor α and decay ratio β are set to positive real numbers to normalize the spatial proximity Sp . In this work, by comparing different combinations of parameter α and β , we set them as 1 and 4, respectively. Note that, the choices of these two parameters are orthogonal to our method proposed later, and can be adjusted to accommodate different application requirements. What is more, here we select user u_j 's historical location which is the closest to loc_i , due to the skew distribution of users' check-in records.

2) Interest preference: As done in other works, we learn mobile users' interest level directly from their check-ins. Similar to the task topic representation, here the interest level of each user, say u_j , is represented as $\vec{O}_{u_j} = \{o_{j1}, o_{j2}, \dots, o_{jq}\}$, where $o_{jx} \in \vec{O}_{u_j}$ is the weighted value with respect to x -th element in Γ . By calculating the similarity between $\vec{O}_{u_j}$ and task T_i 's topic $\vec{P}_i$, we obtain the matching degree as below:

$$Ip(u_j, T_i) = \frac{\vec{O}_{u_j} * \vec{P}_i}{\|\vec{O}_{u_j}\| * \|\vec{P}_i\|}, \quad (3)$$

where the similarity is measured using Cosine similarity.

Therefore, by integrating the aforementioned factors, we calculate the task acceptance estimation as follows:

$$\mathcal{A}_{T_i}^{u_j} = Sp(u_j, T_i) * Ip(u_j, T_i). \quad (4)$$

So, with respect to the invitation of MCS task set $\mathcal{T}$, the **task acceptance estimation** of user u_j can be represented as below:

$$\begin{aligned} \mathcal{A}_{\mathcal{T}}^{u_j} &= \max_{T_i \in \mathcal{T}} \mathcal{A}_{T_i}^{u_j} \\ &= \max_{T_i \in \mathcal{T}} \{Sp(u_j, T_i) * Ip(u_j, T_i)\}. \end{aligned} \quad (5)$$

In other words, it chooses one task in $\mathcal{T}$ which has the maximum task acceptance estimation as u_j 's task acceptance estimation for $\mathcal{T}$.

Incentive Cost. In MCS campaign, an incentive mechanism is indispensable to compensate users' consumed resources, such as battery depletion, data storage, and network bandwidth. The total incentive cost provided to the recruited workers should not exceed the pre-specified incentive budget $\mathcal{C}_{max}$. Moreover, considering different roles taken by the recruited participants, it is far from practice to employ a simple uniform payment in socially aware MCS, e.g., identical incentive for all the involved workers [17]. Thus, it is necessary to devise incentive mechanism to provide discriminated rewards for different worker roles, i.e., seed worker, invitee worker, which we will discuss in detail later.

3 PROBLEM DEFINITION

With respect to task invitation diffusion process, we formalize the worker recruitment game under a *random diffusion* model, in which seed workers randomly diffuse task invitations to neighbor nodes. Using the diffusion model, we then define our acceptance-aware worker recruitment problem.

3.1 Random Diffusion Model

In our devised *Random Diffusion Model*, mobile users randomly and independently invite their social neighbors to participate into MCS campaign, according to a specific diffusion probability. Here, the strength of social tie, i.e., edge weight in $\mathcal{G}$, is employed as the diffusion probability. As stated in [18], in real world, people tend to cooperate initially with their immediate neighbors, and then with other ones. Following this intuition, for a potential seed node u , a tree structure $\mathcal{SG}_u$ is utilized to model the task diffusion process. Specifically, rooted from seed node u , the diffusion tree $\mathcal{SG}_u$ iteratively expands in a Breadth-First-Search manner level by level, where the level of each "influenced node" u_j equals to ℓ -hop distance $\ell(u, u_j)$. That is, seed node u and potential influenced nodes are hierarchically organized in $\mathcal{SG}_u$. The task diffusion process does not terminate until it reaches a maximum ℓ -hop distance ℓ_{max} [19]. To explain this, considering diffusion latency and limited budget, the task diffusion process should be restricted into a limited number of steps, i.e., ℓ_{max} . As a result, the potential influenced nodes in $\mathcal{SG}_u$ can be represents as: $\mathcal{U}_{df}(u) = \{u_j | \ell(u_j, u) \leq \ell_{max}\}$. In practice, the choice of ℓ_{max} is orthogonal to our proposed approach, and can be tuned according to different application requirements.

Similar to the influence maximization problem in social networks, to achieve the desired seed nodes among all the available mobile users, it is necessary to implement task diffusion simulation process, based on our built random diffusion model. And then, according to the evaluation of simulation results, it is possible to optimally determine the seed nodes. However, during propagation simulation, one issue needs also to be seriously considered, that there might be more than one connecting path from seed node u to the influenced node u_j , due to the stochastic nature. In other words, the task invitation might be delivered via different diffusion paths originated from u and terminated at u_j . For simplicity, the MCS platform fixes the path with the largest diffusion probability as the task diffusion path to the invitee. Formally, $\forall u_j \in \mathcal{U}_{df}(u)$, the task diffusion path $\mathcal{H}_{u,u_j}^*$ can be identified as below:

$$\begin{aligned} \arg \max_{\mathcal{H}_{u,u_j}} p(\mathcal{H}_{u,u_j}) &= \arg \max_{\mathcal{H}_{u,u_j}} \prod_{w_{x,y} \in \mathcal{H}_{u,u_j}} w_{x,y} \\ &\propto \arg \max_{\mathcal{H}_{u,u_j}} \log \prod_{w_{x,y} \in \mathcal{H}_{u,u_j}} w_{x,y} \quad (6) \\ &= \arg \max_{\mathcal{H}_{u,u_j}} \sum_{w_{x,y} \in \mathcal{H}_{u,u_j}} \log w_{x,y}. \end{aligned}$$

where $\mathcal{H}_{u,u_j}$ represents a connecting path from node u to u_j , $p(\mathcal{H}_{u,u_j})$ is its diffusion probability, and $w_{x,y}$ denotes the weight of edge $e_{x,y}$ located on $\mathcal{H}_{u,u_j}$.

To illustrate it, a toy example is demonstrated in Fig. 1, where a social network including 13 nodes and 13 edges is present in the left part. To simplify, only partial edge weights are indicated. Suppose u_1 has been selected as a seed user, its diffusion tree $\mathcal{SG}_{u_1}$ is illustrated in the right part, where ℓ_{max} is set to 3. Starting from seed node u_1 , $\mathcal{SG}_{u_1}$ first examines u_1 's immediate neighbor nodes: $\{u_2, u_3, u_4\}$, where the relevant ℓ -hop distance is 1. With the increasing of ℓ -hop distance, it continues growing along the edges associated with discovered nodes step by step. There exist two connected paths from u_1 to u_7 : $\mathcal{H}_{u_1,u_7}^1 = \langle e_{1,2}, e_{2,7} \rangle$ and $\mathcal{H}_{u_1,u_7}^2 = \langle e_{1,4}, e_{4,7} \rangle$, with diffusion probability 0.06 and 0.3, respectively. $\mathcal{H}_{u_1,u_7}^2$ is finally determined by the MCS platform as the actual diffusion path, due to its larger diffusion probability.

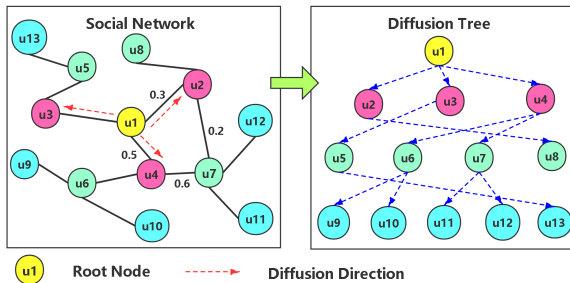


Fig. 1. A Toy Example of Diffusion Tree.

Based on the established diffusion tree, now we are ready to derive the results of worker recruitment in a stochastic manner. More specifically, starting from root node, for each involved edge $e_{i,j}$, if its associated diffusion probability, i.e., edge weight $w_{i,j}$, is not less than a generated

random range from 0 to 1, it means that user u_i might diffuse task invitation to u_j ; otherwise, it will not. For instance, two different worker recruitment results are present in Fig. 2 for seed node u_1 . For example, the recruited workers $\mathcal{U}_{wk}$ in the left part of Fig. 2 are $\{u_1, u_2, u_3, u_5, u_8, u_{13}\}$.

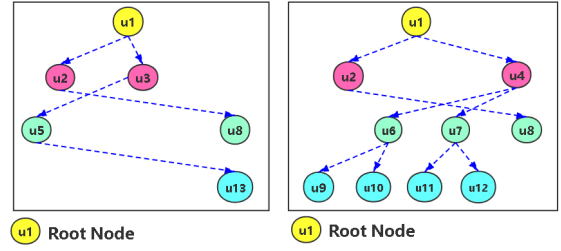


Fig. 2. Different Worker Recruitment Results.

Expected Task Acceptance: Due to the stochastic process, the generated worker recruitment results may vary every trial. In other words, we can not directly fix the performance of random results, in terms of task acceptance $\mathcal{AT}(\mathcal{SG}_u)$ and worker recruitment cost $\mathcal{C}(\mathcal{SG}_u)$. Therefore, we employ expected value to evaluate its performance, as the expected value is a measure of the center of the distribution of the stochastic processes that are the returns. Given MCS task set $\mathcal{T}$, for each node u_j , we define two state variables Θ_x^j and Θ_y^j as follows:

$$\Theta_x^j = \begin{cases} 1, & u_j \text{ has received task invitation;} \\ 0, & \text{otherwise,} \end{cases} \quad (7)$$

$$\Theta_y^j = \begin{cases} 1, & u_j \text{ accepts task after receiving invitation;} \\ 0, & u_j \text{ rejects task after receiving invitation.} \end{cases} \quad (8)$$

Thus, we define the conditional probability of one user u_j accepts the MCS task, given u_j has already received task invitation. Formally, the conditional probability can be represented as: $p(\Theta_y^j = 1 | \Theta_x^j = 1) = \mathcal{A}_{\mathcal{T}}^{u_j}$. Suppose u_i is determined as the seed node, the expected task acceptance of u_j can be derived as below:

$$\begin{aligned} \mathbb{E}(\Theta_y^j) &= \mathbb{E}_{\Theta_x} [\mathbb{E}_{\Theta_y}(\Theta_y^j | \Theta_x^j)] \\ &= \mathbb{E}_{\Theta_x} [p(\Theta_y^j = 1 | \Theta_x^j = 1)] \\ &= \mathbb{E}_{\Theta_x} [\mathcal{A}_{\mathcal{T}}^{u_j}] \\ &= \mathcal{A}_{\mathcal{T}}^{u_j} * p(\Theta_x^j = 1) \\ &= \mathcal{A}_{\mathcal{T}}^{u_j} * p(\mathcal{H}_{u_i,u_j}). \end{aligned} \quad (9)$$

Therefore, for seed node u_i , the overall expected task acceptance estimation of its diffusion tree can be directly calculated as follows:

$$\mathbb{E}[\mathcal{AT}(\mathcal{SG}_{u_i})] = \sum_{u_j \in \mathcal{SG}_{u_i}} p(\mathcal{H}_{u_i,u_j}) * \mathcal{A}_{\mathcal{T}}^{u_j}. \quad (10)$$

Note that, as a source node, u_i has already been invited by the MCS platform to undertake one task. So, in order to provide a unified representation, we suppose that there exists one "cyclic diffusion path" originated and terminated at source node u_i , and its diffusion probability is 1, i.e., $p(\mathcal{H}_{u_i,u_i}) \equiv 1$.

However, there might be more than one seed node to be enlisted, such that $\mathcal{S} = \{u_1, u_2, \dots, u_m\}$. Intuitively,

we need to merge these seed nodes' diffusion trees, i.e., $\mathcal{SG}_S = \mathcal{SG}_{u_1 \cup u_2 \cup \dots \cup u_m}$. In $\mathcal{SG}_S$, each invitee node, i.e., $u_j \in \mathcal{SG}_S \setminus S$, might be invited by more than one seed node in S , i.e., there might exist more than one diffusion path terminated at u_j . Due to the potential propagation path overlapping, we should synthetically considered all the coupled diffusion paths, and derive the corresponding diffusion probability. However, it is too ad hoc and computationally expensive, considering the involved different seed nodes, graph topology, etc.. As a result, here we make an assumption that, for each task invitee, say u_j , all the relevant diffusion paths are independent with each other. Accordingly, u_j 's diffusion probability can be calculated as: $1 - \prod_{u_x \in S} [1 - p(\mathcal{H}_{u_x, u_j})]$, where $1 - p(\mathcal{H}_{u_x, u_j})$ denotes the probability that seed node u_x has not invited u_j . Hence, the above task acceptance estimation should be reformulated as below:

$$\mathbb{E}[\mathcal{A}_T(\mathcal{SG}_S)] = \sum_{u_j \in \mathcal{SG}_S} \underbrace{\left\{ 1 - \prod_{u_x \in S} [1 - p(\mathcal{H}_{u_x, u_j})] \right\}}_{\text{Diffusion Probability}} * \mathcal{A}_T^{u_j}. \quad (11)$$

User Incentive Mechanism: To be fair, when a recruited worker conducts one task, the MCS platform will provide him with the same amount of reward $\mathcal{R}_0$, e.g., $\mathcal{R}_0 = 15$ dollars. Besides, following the design commonly used in influence maximum problem [12], [20], we provide each seed user an extra reward $\mathcal{R}_1$, e.g., $\mathcal{R}_1 = 10$ dollars, to incentivize them to start diffusion trees. In other words, even if seed users have not conducted the task by themselves, they will also obtain extra reward for organizing the diffusion trees. It should however be noted that, for simplicity, here we just incentivize seed nodes for their task invitation diffusion during the stochastic recruitment process. Subsequently, the expected incentive cost is calculated as follows:

$$\mathbb{E}[\mathcal{C}(\mathcal{SG}_S)] = |S| * \mathcal{R}_1 + \sum_{u_j \in \mathcal{SG}_S} \left\{ 1 - \prod_{u_x \in S} [1 - p(\mathcal{H}_{u_x, u_j})] \right\} * \mathcal{A}_T^{u_j} * \mathcal{R}_0, \quad (12)$$

where $|S|$ denotes the number of seed users.

3.2 Problem Definition and Analysis

Given a MCS task set $\mathcal{T}$, we aim to choose a subset of seed workers S among the available mobile users $\mathcal{U}$, such that the expected task acceptance of recruited workers is maximized subject to the constraint that the expected incentive cost is not exceeding the pre-specified budget $\mathcal{C}_{max}$. Formally, our acceptance-aware worker recruitment problem in the random diffusion model (RAWR) can be formulated as a constrained combinatorial optimization problem as follows:

$$\begin{cases} \arg \max_{S \subseteq \mathcal{U}} \mathbb{E}\{\mathcal{A}_T(\mathcal{SG}_S)\} \\ s.t. : \mathbb{E}\{\mathcal{C}(\mathcal{SG}_S)\} \leq \mathcal{C}_{max}. \end{cases} \quad (13)$$

Lemma 1. The RAWR problem is NP-hard.

Proof: We prove it by reducing the Weighted Set Cover problem to the RAWR problem. In the Weighted Set Cover

problem, given a collection of subsets $\{S_1^\#, \dots, S_m^\#\}$ over a universe $U^\#$, where each subset $S_i^\#$ is specified with a weight $w(S_i^\#)$, $1 \leq i \leq m$, it wishes to decide k subsets of maximum total weight $\sum_{i=1}^k w(S_i^\#)$. We map the universe $U^\#$ in the weighted set cover problem to the node set $\mathcal{U}$, and also map each subset $S_i^\#$ to the diffusion tree $\mathcal{SG}_{u_i}$. And the expected task acceptance is regarded as the weight in the Weighted Set Cover problem. Subsequently, the cost of each diffusion tree is set to 1 and the incentive budget $\mathcal{C}_{max}$ in RAWR is set to k . Thus, the Weighted Set Cover problem is equivalent to deciding if there is a k diffusion tree set with the maximum expected task acceptance in the RAWR problem. As the Weighted Set Cover problem is NP-hard, our RAWR problem is also NP-hard. $\square$

Due to its NP hardness, there exists no exact algorithms that can achieve optimal solution in polynomial time. In the following, we would like to further analyze our RAWR problem.

Lemma 2. The task acceptance estimation function $\mathbb{E}[\mathcal{A}_T(\mathcal{SG}_S)]$ is monotonic nondecreasing.

Proof: Without loss of generality, let us consider two seed worker sets S_2 and S_1 , where $S_2 = S_1 \cup u$ and $u \notin S_1$, i.e., $S_1 \subset S_2$. Their respective diffusion trees are represented as $\mathcal{SG}_{S_2}$, $\mathcal{SG}_{S_1}$ and $\mathcal{SG}_u$, respectively. Obviously, the following relationship holds: $\mathcal{SG}_{S_2} = \mathcal{SG}_{S_1 \cup u} = \mathcal{SG}_{S_1} \cup \mathcal{SG}_u$. For the sake of illustration, we define the incremental expectation function as below:

$$\Delta \mathbb{E}(S, u) = \mathbb{E}[\mathcal{A}_T(\mathcal{SG}_S \cup \mathcal{SG}_u)] - \mathbb{E}[\mathcal{A}_T(\mathcal{SG}_S)]. \quad (14)$$

Thus, the incremental value with variables S_2 and S_1 can be formulated as follows:

$$\mathbb{E}[\mathcal{A}_T(\mathcal{SG}_{S_2})] - \mathbb{E}[\mathcal{A}_T(\mathcal{SG}_{S_1})] = \Delta \mathbb{E}(S_1, u). \quad (15)$$

As shown in Fig. 3, there are three different cases for $\mathcal{SG}_{S_1}$ and $\mathcal{SG}_u$: (a) $\mathcal{SG}_u$ is completely contained in $\mathcal{SG}_{S_1}$; (b) $\mathcal{SG}_u$ and $\mathcal{SG}_{S_1}$ are partially overlapped; (c) $\mathcal{SG}_u$ and $\mathcal{SG}_{S_1}$ are independent. The increment $\Delta \mathbb{E}$ could achieve a minimum value in case (a), and a maximum value in case (c). Thus, in the following, we will mainly analyze the situation (a), to obtain its minimum value.

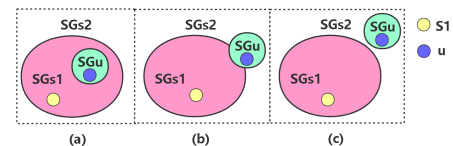


Fig. 3. The Diagram of Lemma 2.

For the sake of illustration, we use the symbols $S^\#$ to represent the node set $\mathcal{SG}_{S_1} \setminus (S_1 \cup u)$, i.e., all the recruited nodes except for seed nodes S_1 and node u . Thus, in

situation (a), $\mathbb{E}[\mathcal{A}_{\mathcal{T}}(\mathcal{SG}_{S_1})]$ can be calculated as below:

$$\begin{aligned} \mathbb{E}[\mathcal{A}_{\mathcal{T}}(\mathcal{SG}_{S_1})] &= \left\{ 1 - \prod_{u_x \in S_1} [1 - p(\mathcal{H}_{u_x, u})] \right\} * \mathcal{A}_{\mathcal{T}}^u + \\ &\sum_{u_j \in S_1} \mathcal{A}_{\mathcal{T}}^{u_j} + \sum_{u_j \in S^\#} \left\{ 1 - \prod_{u_x \in S_1} [1 - p(\mathcal{H}_{u_x, u_j})] \right\} * \mathcal{A}_{\mathcal{T}}^{u_j}. \end{aligned} \quad (16)$$

And $\mathbb{E}[\mathcal{A}_{\mathcal{T}}(\mathcal{SG}_{S_2})]$ can be calculated as below:

$$\begin{aligned} \mathbb{E}[\mathcal{A}_{\mathcal{T}}(\mathcal{SG}_{S_2})] &= \mathcal{A}_{\mathcal{T}}^u + \sum_{u_j \in S_1} \mathcal{A}_{\mathcal{T}}^{u_j} + \\ &\sum_{u_j \in S^\#} \left\{ 1 - \prod_{u_x \in S_1} [1 - p(\mathcal{H}_{u_x, u_j})] * [1 - p(\mathcal{H}_{u, u_j})] \right\} * \mathcal{A}_{\mathcal{T}}^{u_j}. \end{aligned} \quad (17)$$

Thus, the increment $\Delta\mathbb{E}$ can be calculated as follows:

$$\begin{aligned} \Delta\mathbb{E}_{min} &= \prod_{u_x \in S_1} [1 - p(\mathcal{H}_{u_x, u})] * \mathcal{A}_{\mathcal{T}}^u + \\ &\sum_{u_j \in S^\#} \left\{ \prod_{u_x \in S_1} [1 - p(\mathcal{H}_{u_x, u_j})] * p(\mathcal{H}_{u, u_j}) \right\} * \mathcal{A}_{\mathcal{T}}^{u_j} \geq 0. \end{aligned} \quad (18)$$

As the minimum increment $\Delta\mathbb{E}_{min}$ is nonnegative, the monotonic nondecreasing property is proved. $\square$

Lemma 3. The task acceptance estimation function $\mathbb{E}[\mathcal{A}_{\mathcal{T}}(\mathcal{SG}_S)]$ is non-submodular.

Proof: Let S_1 and S_2 are two seed worker sets, $S_1 \subset \mathcal{U}$, $S_2 \subset \mathcal{U}$, $S_1 \subset S_2$, and $u \in \mathcal{U} \setminus S_2$. To prove its submodular, the following relationship must hold:

$$\Delta\mathbb{E}(S_1, u) \geq \Delta\mathbb{E}(S_2, u). \quad (19)$$

As demonstrated in Lemma 2, the increment will achieve a minimum value like in Eq. 18. The maximum incremental value can be calculated as below, according to case (c) in Lemma 2.

$$\Delta\mathbb{E}_{max} = \sum_{u_j \in \mathcal{SG}_u \setminus u} p(\mathcal{H}_{u, u_j}) * \mathcal{A}_{\mathcal{T}}^{u_j} + \mathcal{A}_{\mathcal{T}}^u. \quad (20)$$

The following comparison expression can not achieved,

$$\Delta\mathbb{E}_{min}(S_1, u) \geq \Delta\mathbb{E}_{max}(S_2, u), \quad (21)$$

thus its non-submodular property is proved. $\square$

Based on Lemma 2, it is possible to sequentially construct the desired solution, such as greedy-based techniques. However, due to its non-submodular property, there are few performance guarantees to achieve near optimal solution.

4 PROPOSED APPROACHES

Based on the previous discussions, it is difficult to solve our problem using exact algorithms, especially for larger-scale problem settings involving a large number of users. Thus, we turn to population-based stochastic evolutionary optimizers. Evolutionary optimizers utilize the meta-heuristic search to find a near optimal solution within a reasonable

running time, and avoid exhaustively searching all possibilities. In particular, Memetic Algorithm (MA) is more favored for combinatorial optimization, as it integrates both local and global searches. The promising search capability of MA and its variants have been widely validated in many complex problems [21], [22]. In this section, using MA as the basis, we propose an effective MA-RAWR algorithm to tackle our problem, also design several enhancement strategies to further improve the algorithm performance.

4.1 Problem-Specific Heuristic Knowledge

The problem-specific knowledge can be exploited to facilitate problem solving, i.e., *Diffusion Overlapping Effect* and *Diffusion Utility*. Next, we discuss them in detail.

1) *Diffusion Overlapping Effect*. The core issue in RAWR is that there exists an “overlapping effect” between different diffusion trees. In the following, we would like to quantify the effect on the basis of distance measurement. One matrix representation $\mathbf{D} = [d_{i,j}]_{1 \leq i, j \leq |\mathcal{U}|}$ is employed to model all the involved nodes’ diffusion trees. More specifically, one entry $d_{i,j} \in \mathbf{D}$ represents the diffusion probability from node u_i to u_j , within diffusion tree $\mathcal{SG}_{u_i}$. Mathematically, it can be represented as follows:

$$d_{i,j} = p(\mathcal{H}_{u_i, u_j}), \quad (22)$$

where $\mathcal{H}_{u_i, u_j}$ denotes a diffusion path in $\mathcal{SG}_{u_i}$. The main diagonal elements $d_{i,i}$, $1 \leq i \leq |\mathcal{U}|$, of $\mathbf{D}$ are all equal to 1. By this way, the ℓ -hop proximity is transformed into first-order proximity, and each row in $\mathbf{D}$, say $\mathbf{D}(i, \cdot)$, can be regarded as user u_i ’s “task diffusion distribution” over $\mathcal{U}$. Here, we use their respective distributions to measure the diffusion overlapping among users. Formally, we employ the Euclidean distance to measure the overlapping between $\mathcal{SG}_{u_i}$ and $\mathcal{SG}_{u_j}$ as below:

$$DO(\mathbf{D}(i, \cdot), \mathbf{D}(j, \cdot)) = \sqrt{\sum_{k=1}^{|\mathcal{U}|} (d_{i,k} - d_{j,k})^2}. \quad (23)$$

The pairwise distances which denote the difference between diffusion overlapping are calculated one by one, and organized as an adjacency matrix $\mathbf{O} = [o_{i,j}]_{1 \leq i, j \leq |\mathcal{U}|}$, where $o_{i,j} = DO(\mathcal{SG}_{u_i}, \mathcal{SG}_{u_j})$. Obviously, the higher value of $DO(\mathcal{SG}_{u_i}, \mathcal{SG}_{u_j})$, the less diffusion overlapping between u_i and u_j . Note that, as diffusion tree is independent of task $\mathcal{T}$, it can be computed in advance.

2) *Diffusion Utility*. Given a MCS task set $\mathcal{T}$, user u ’s utility can be evaluated according to its diffusion tree $\mathcal{SG}_u$. We first need to calculate each user’s task acceptance estimation $\mathcal{A}_{\mathcal{T}}^{u_i}$, $u_i \in \mathcal{U}$. Subsequently, their respective utility on diffusion trees can be computed as follows:

$$\begin{bmatrix} \mathcal{A}_{\mathcal{T}}(\mathcal{SG}_{u_1}) \\ \vdots \\ \mathcal{A}_{\mathcal{T}}(\mathcal{SG}_{u_n}) \end{bmatrix} = \begin{bmatrix} d_{1,1} & \cdots & d_{1,n} \\ \vdots & \ddots & \vdots \\ d_{n,1} & \cdots & d_{n,n} \end{bmatrix} * \begin{bmatrix} \mathcal{A}_{\mathcal{T}}^{u_1} \\ \vdots \\ \mathcal{A}_{\mathcal{T}}^{u_n} \end{bmatrix}. \quad (24)$$

Therefore, the optimization objective $\mathbb{E}\{\mathcal{A}_{\mathcal{T}}(\mathcal{SG}_S)\}$ can be evaluated directly using the diffusion tree’s utility $\mathcal{A}_{\mathcal{T}}(\mathcal{SG}_{u_i})$, $u_i \in \mathcal{U}$, rather than each user’s acceptance estimation. To make better use of it in our population-based algorithm, we

normalize the diffusion utility in the range from 0 to 1 as below:

$$\widetilde{\mathcal{A}}_{\mathcal{T}}(\mathcal{S}\mathcal{G}_{u_i}) = \frac{\mathcal{A}_{\mathcal{T}}(\mathcal{S}\mathcal{G}_{u_i}) - \min_{u_x \in \mathcal{U}} \mathcal{A}_{\mathcal{T}}(\mathcal{S}\mathcal{G}_{u_x})}{\max_{u_x \in \mathcal{U}} \mathcal{A}_{\mathcal{T}}(\mathcal{S}\mathcal{G}_{u_x}) - \min_{u_x \in \mathcal{U}} \mathcal{A}_{\mathcal{T}}(\mathcal{S}\mathcal{G}_{u_x})} \quad (25)$$

4.2 MA-RAWR Algorithm

By incorporating the above heuristic knowledge, we devise the MA-RAWR algorithm to effectively solve our RAWR problem under the architecture of MA.

Framework: Based on the distribution of $\widetilde{\mathcal{A}}_{\mathcal{T}}(\mathcal{S}\mathcal{G}_{u_i})$, we adopt a dual-chromosome scheme see below to encode potential solutions. We first construct a diverse population of initial solutions, i.e., binary chromosomes, using a decomposition-based heuristic initialization mechanism. We then implement a variable neighborhood search method to achieve local optimum based on our elaborated neighborhood structures $\mathcal{N}_x(\mathcal{S})$. Afterwards, in accordance with the basic workflow of Differential Evolution (DE), we heuristically perform the operations of reproduction, including crossover and mutation, with probability cr and mr , respectively, to produce offspring individuals. To guarantee the incentive budget constraint, i.e., incentive budget $\mathcal{C}_{max}$, a solution feasibility checking and repair operation is necessary for the offspring solutions. For an infeasible solution, users having least diffusion utility will be removed first, until the constraint has been satisfied. Based on the evaluation of newly produced solutions, variable neighborhood search is adaptively triggered to avoid premature convergence at a local minimum. Finally, a selective replacement operation, e.g., tournament selection, is adopted to determine the next generation. This process does not terminate until a stopping condition is satisfied. The workflow of MA-RAWR algorithm is illustrated as below:

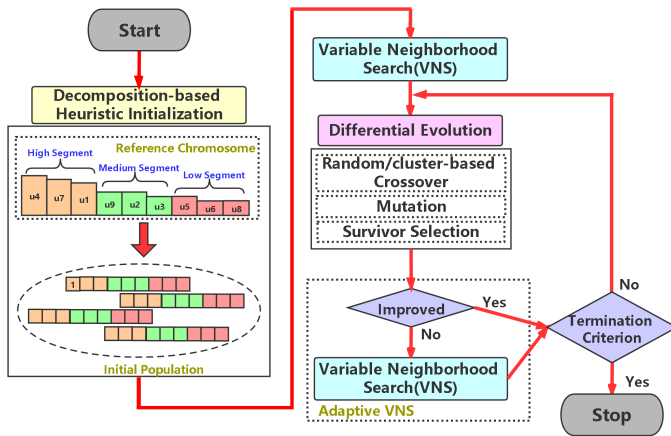


Fig. 4. The Workflow of MA-RAWR Algorithm.

The pseudo code of MA-RAWR is shown in Algorithm 1. In the following, we describe the subcomponents of MA-RAWR algorithm in detail.

Dual-Encoding Solution Representation: With respect to the solution representation, we adopt a dual-encoding scheme, i.e., reference and binary chromosomes, to encode potential solutions. The reference chromosome $\mathcal{S}_{rf}$ is a full

permutation of all the users, in a descending order of their respective diffusion utilities $\widetilde{\mathcal{A}}_{\mathcal{T}}(\mathcal{S}\mathcal{G}_{u_i})$. While the binary chromosome, i.e., real solution $\mathcal{S}$, is organized with the same order of the reference chromosome, in which each entry indicates whether its corresponding node is chosen by $\mathcal{S}$ or not. Suppose one solution $\mathcal{S}$, if i -th user in $\mathcal{S}_{rf}$ is selected as seed node, its corresponding entry in solution chromosome, i.e., i -th element, is set to 1; otherwise, it is 0. Intuitively, nodes with a larger diffusion utility, will be placed at the top of reference chromosome. In this way, the reference chromosome which is regarded as the index of real solutions implicitly embeds the diffusion utility distribution. During the evolutionary process, the binary chromosome evolves to achieve better solutions via modifying its binary variables.

Decomposition-based Heuristic Initialization: Due to the large search space, an evolutionary process starting from initial solutions constructed by a fully random initialization might fail to evolve to satisfactory solutions or even fail to reach feasible solutions. In other words, it is impossible to generate competitive initial individuals that are too far away from the promising solution region. Consequently, it is more admirable to explore the problem-specific heuristic knowledge, and devise a heuristic driven initialization mechanism. Inspired by the cooperative coevolution techniques [23], we design a decomposition-based heuristic initialization method in the following.

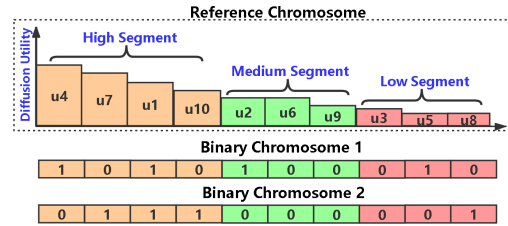


Fig. 5. The Toy Example of Solution Representation.

The basic idea is that, according to the overall dis-

ALGORITHM 1: MA-RAWR Algorithm

Input: Task set: $\mathcal{T}$, Mobile users: $\mathcal{U}$, Diffusion probability matrix: $\mathbf{D}$, Diffusion utility: $\mathcal{A}_{\mathcal{T}}(\mathcal{S}\mathcal{G})$, Population size: PS ;

Output: Optimal solution: $\mathcal{S}^*$;

```

1 Initialize  $(PS) \rightarrow \{\mathcal{S}_i\}, 1 \leq i \leq PS$ ;
2 Variable Neighborhood Search  $(\mathcal{S}_i, \mathcal{N}_x(\cdot)) \rightarrow \{\mathcal{S}_i\}$ 
  / * Algorithm 2 * /;
3 while termination criterion not satisfied do
4   Random/Cluster-based Crossover  $(\mathcal{S}_i, cr) \rightarrow$ 
     Offspring / * Algorithm 3 * /;
5   Mutation  $(\mathcal{S}_i, mr) \rightarrow$  Offspring;
6   Survivor Selection  $(\mathcal{S}_i, \text{Offspring}) \rightarrow \{\mathcal{S}'_i\}$ ;
7   if no improvement achieved then
8     Variable Neighborhood Search  $(\mathcal{S}'_i, \mathcal{N}_x(\cdot))$ 
        $\rightarrow \{\mathcal{S}'_i\}, 1 \leq i \leq \lfloor \frac{1}{4} PS \rfloor$  / * Algorithm 2 * /;
9   end
10   $\{\mathcal{S}'_i\} \rightarrow \{\mathcal{S}_i\}, 1 \leq i \leq PS$ ;
11 end
```

tribution of diffusion utility over all the nodes, different nodes represent different levels of optimality, and require different attention from our designed optimizer. Here, we decompose reference chromosome into different segments labeled as *High Segment*, *Medium Segment* and *Low Segment*, respectively, according to their diffusion utilities. To make it more clear, a toy example with 10 possible users and two binary chromosomes is present in Fig. 5. For instance, one node whose normalized value of diffusion utility is more than 0.7 would be classified into *High Segment*. Based on the decomposition, we conduct a bias sampling on these partitioned segments according to a predefined proportion. For instance, we may select 5, 3, and 1 nodes from *High Segment*, *Medium Segment* and *Low Segment*, respectively. Note that, the sample proportion can be adjusted according to different problem specifications. Within each segment, we repeatedly sample nodes with probability corresponds to their diffusion utility distribution $\mathcal{A}_T(\mathcal{SG}_u)$. In other words, nodes with a higher diffusion utility will be selected with a higher probability. Note that, the sampled nodes can not form a complete and feasible solution, instead, they are just a fragment with limited size.

Afterwards, starting from these sampled nodes, we parallelly perform a random walk procedure on the adjacency matrix $\mathcal{O}$ to extend the initial solution. For example, suppose u_6 is a sampled node, one of its linked nodes $\{u_1, u_3, u_5\}$ would be selected as the next incumbent one, according to the probability distribution $\{o_{1,6}, o_{3,6}, o_{5,6}\}$. To be specific, the probability that node u_1 will be chosen is calculated as: $1 - o_{1,6} / (o_{1,6} + o_{3,6} + o_{5,6})$. This process will terminate until the expected incentive budget becomes exhausted. Finally, the extended fragments with each segment will be jointed to form a complete feasible solution $\mathcal{S}_i, 1 \leq i \leq PS$, where PS denotes the scale of the population.

Adaptive Variable Neighborhood Search: Originating from a specific solution and exploiting multiple neighborhood structures, Variable Neighborhood Search (VNS) [24] repeatedly applies a local search strategy to search for a better solution than the current one. In view of the high dimensional solution representation, it is expensive and also redundant to examine all the potential neighborhoods of incumbent solution. Thus, by harnessing the decomposed segments in solution chromosomes, we define different neighborhood structures as following.

1) 1-Flip $\mathcal{N}_1(\mathcal{S})$: For a candidate solution $\mathcal{S}$, it randomly selects a bit in *High Segment* (or *Medium Segment*), and flips its value, i.e., if the selected bit equals to 1, then set it to 0; otherwise, set to 1.

2) 2-Swap $\mathcal{N}_2(\mathcal{S})$: It selects a 0-valued and a 1-valued bit in *High Segment* (or *Medium Segment*), and exchange their values. That is, a determined seed node will be replaced by a non-determined node in *High/Medium Segment*.

3) 2-Swap $\mathcal{N}_3(\mathcal{S})$: Different from $\mathcal{N}_2(\cdot)$, the 1-valued and 0-valued bits are chosen from *High Segment* and *Medium Segment*, respectively.

4) Hybrid 2-swap $\mathcal{N}_4(\mathcal{S})$: It conduct $\mathcal{N}_2(\cdot)$ operations on *High Segment* and *Medium Segment* simultaneously.

By switching between different neighborhood structures above, local search is used to enhance the exploitation, i.e., to search the better one of the incumbent solution. The steps of *VNS algorithm* are shown in Algorithm 2. First, a pertur-

bation operation named shaking is conducted to randomly generate a solution from incumbent solution's x^{th} neighborhood $\mathcal{N}_x(\mathcal{S})$. Then, the function *FindBestNeighbor* is used to find the best neighbor within the limited region of $\mathcal{N}_x(\mathcal{S})$. The process will iteratively exploit the neighborhood structures until all these neighborhoods have been examined. VSN is conducted on every individual in each evolutionary iteration. To improve our algorithm's efficiency, the adaptive VNS procedure is implemented. More specifically, if no improvement has been achieved in current generation, VSN procedure will be triggered to be implemented on partial solutions, e.g., the top 25% solutions $\lfloor \frac{1}{4}PS \rfloor$.

ALGORITHM 2: VSN Algorithm

Input: Neighborhood Structures: $\mathcal{N}_x(\cdot), 1 \leq x \leq x_{max}$,
Solution: $\mathcal{S}_i$;
Output: Best Found Solution $\mathcal{S}_i$;

```

1  $1 \rightarrow x$ ;
2 Generate random solution  $(\mathcal{S}_i, \mathcal{N}_x(\cdot)) \rightarrow \mathcal{S}'_i$ ,  

 $\mathcal{S}'_i \in \mathcal{N}_x(\mathcal{S}_i) / *$  Shaking Stage  $*$  /;
3 while  $x \leq x_{max}$  do
4   FindBestNeighbor $(\mathcal{S}'_i, \mathcal{N}_x(\cdot)) \rightarrow \mathcal{S}''_i$   

    $/ *$  Local Search Stage  $*$  /;
5   if  $\mathbb{E}\{\mathcal{A}_T(\mathcal{SG}_{\mathcal{S}''_i})\} < \mathbb{E}\{\mathcal{A}_T(\mathcal{SG}_{\mathcal{S}_i})\}$  then
6      $\mathcal{S}''_i \rightarrow \mathcal{S}_i$  and  $1 \rightarrow x$ ;
7   else
8      $x + 1 \rightarrow x$ ;
9   end
10 end
```

Reproduction and Survivor Selection: Here, we take the routine of Differential Evolution (DE) to reproduce new offsprings. The reproduction process comprises two steps: crossover and mutation operations. With a certain crossover rate cr , it firstly selects two breakpoints, namely *start point* and *end point*, upon the reference chromosome. And then, a pair of solutions selected as "parent chromosomes" will conduct a two-point crossover operation, resulting in new offspring individuals. Inspired by niching method, we devise a cluster-based crossover strategy by utilizing solution similarity clustering, except for the classical random crossover. First, all the solutions are clustered based on similarity calculated upon individual's chromosome fragment of *High Segment* and *Medium Segment*, using hamming distance. In this way, the promising fragment with high possibility of optimality, i.e., *High Segment* and *Medium Segment*, could be inherited by offspring individuals. Subsequently, the two-point crossover is implemented on those "parent chromosomes" selected from intra-cluster and inter-cluster, respectively. With respect to the breakpoints, *start* and *end point*, cluster-based crossover only exchanges the fragment of *High Segment* or *Medium Segment*, instead of completely randomly selected points. The crossover operation is illustrated in Fig. 6. All these steps in the *ClusCross* algorithm are shown in Algorithm 3.

The role of mutation operator is to perturb individuals in the current generation, resulting in an enhanced diversity. With a given mutation rate mr , two solutions are randomly selected as parents, e.g., $\mathcal{S}_i$ and $\mathcal{S}_j$. A difference

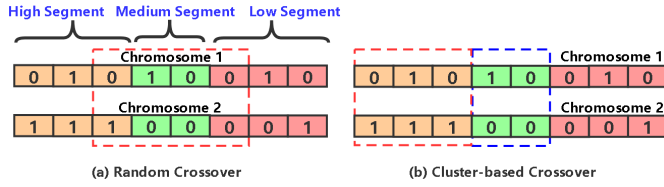


Fig. 6. The Diagram of Crossover Operation.

vector is calculated by subtracting these two solutions, i.e., $\Delta S = S_i - S_j$. The best solution in current generation S_{best} is then integrated with the difference vector to produce a newly offspring solution: $S_{new} = S_{best} + \Delta S$. Meanwhile, a repair operation might be applied to modify S_{new} into a feasible one if necessary. In the stage of *Survivor Selection*, the fittest individuals are selected from the current generation solutions and newly generated offsprings to survive into the next generation. The tournament selection scheme is adopted to repeatedly choose those individuals. After the competition, the fittest solutions survives and enters the next generation.

Complexity Analysis: In the MA-RAWR algorithm, we need to continually evaluate each solution in terms of acceptance and incentive cost estimation. The computational complexity of the solution evaluation is $O(|\mathcal{G}_S|)$, where $|\mathcal{G}_S|$ denotes the number of nodes contained in diffusion tree $\mathcal{G}_S$. In the following, we discuss the time complexity of each major component of MA-RAWR algorithm in each generation. The heuristic initialization component costs $O(PS \times S_{ave} \times |\mathcal{G}_S|)$ to construct PS initial solutions, where S_{ave} denotes the average size of selected nodes in solution, i.e., $S_{ave} = \frac{C_{max}}{E[C(\mathcal{G}_S)]}$. The computational overhead of VSN in Algorithm 2 is dominated by the function FindBestNeighbor. More specifically, it costs $O(LH + LM)$, $O(\overline{S}_{HS} \times \overline{S}_{MS} + \overline{S}_{MS} \times \overline{S}_{MS})$, $O(\overline{S}_{HS} \times \overline{S}_{MS})$, and $O(\overline{S}_{HS} \times \overline{S}_{HS} \times \overline{S}_{MS} \times \overline{S}_{MS})$ within these four neighbor structures, respectively, where LH and LM denote the size of *High Segment* and *Medium Segment*, respectively, and $\overline{S}_{HS}$ and $\overline{S}_{MS}$ represents the size of selected nodes and no selected nodes in *High Segment*, i.e., $\overline{S}_{HS} + \overline{S}_{MS} = LH$. The reproduction consumes $O(PS^2 \times cr)$ and $O(PS \times mr)$ in crossover and mutation operations, and the survivor selection takes $O(PS)$.

ALGORITHM 3: ClusCross Algorithm

Input: Population solution: $\{S_i\}$, $1 \leq i \leq PS$,
Crossover probability: cr , Cluster size: k ;
Output: Offspring individuals;
1 Population Clusteing $(\{S_i\}, k) \rightarrow \mathcal{B} : \{\mathcal{B}_1, \dots, \mathcal{B}_k\}$;
2 Select solution pair $(\{S_i\}, \mathcal{B}) \rightarrow (S_{i_1}, S_{i_2})$;
3 $S_{i_1}, S_{i_2} \in \mathcal{B}_x, 1 \leq x \leq k$ / * **Intra-Cluster** * /;
4 $S_{i_1} \in \mathcal{B}_x, S_{i_2} \in \mathcal{B}_y, x \neq y$ / * **Inter-Cluster** * /;
5 **if** $cr \geq rand()$ **then**
6 Two-point segment crossover $(S_{i_1}, S_{i_2}) \rightarrow$
 (S'_{i_1}, S'_{i_2}) ;
7 **end**

5 EVALUATION AND DISCUSSION

In this section, we systematically evaluate the performance of our proposed technique. Our experiments are conducted on a standard server (Windows 10), with Intel(R) Core(TM) i7-8550U CPU, and 16 GB main memory.

5.1 Experimental Settings

Data Sets: We use two real-world geo-social networks in which users share their check-ins. The two networks are directed graphs based on friend relationship. The first data set is gathered from Foursquare in New York, in which 5,100 nodes, 11,933 edges and 706,344 check-ins are included. And a part of check-in records are visualized in Fig. 7. The second data set is gathered from Gowalla in Boston, in which 145,381 nodes, 546,335 edges and 8,427,156 check-ins are included.

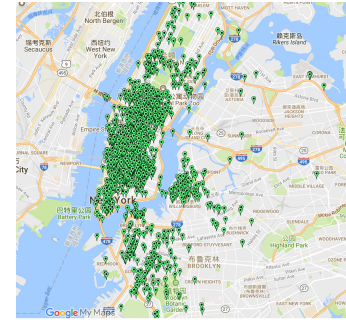


Fig. 7. The spatial distribution of check-ins in New York city.

Baseline Algorithms: To the best of our knowledge, there is little work directly related to our studied problem. In the most relevant study [9], it utilizes generalized *Greedy* algorithm to recruit workers in social-network-assisted MCS, where a feasible solution is constructed by iteratively adding of nodes based on marginal benefit. We employ it as a baseline algorithm for comparison. Moreover, we also adopt two state-of-the-art optimization approaches which are used to tackle monotone non-submodular maximization problems with a monotone cost constraint, i.e., *EAMC* algorithm (AAAI 2020) [25] and *POMC* algorithm (IJCAI 2017) [26]. For a fair comparison, we also incorporate our heuristic knowledge in above baselines. Specifically, in each iteration, one solution in population is selected according to its fitness evaluation, i.e., task acceptance estimation, instead of uniformly at random. With respect to the bit-wise mutation operation, it flips each bit of the incumbent solution based on its diffusion utility, rather than independently with an equal probability.

Parameter Settings: To construct one MCS task, firstly we randomly choose a landmark contained in our collected check-in data sets, e.g., Foursquare and Gowalla, to specify its spatial location. And the task topic is built directly from the established POI category tag set Γ . The maximum ℓ -hop distance ℓ_{max} is set to 3. For our incentive mechanism, the relevant parameters are set as follows: $R_0 = 15$, $R_1 = 10$. With respect to the termination criterion in MA-RAWA algorithm, we set a maximum evolutionary generation MG to 60. The population size PS is 50. For the solution chromosome segments, *High Segment*, *Medium Segment* and *Low Segment*

are partitioned at the top 5% fragment, 6%~15% fragment and the remainder fragment of the reference chromosome, respectively. The size of clusters in the crossover operation is 8, and the crossover rate cr and mutation rate mr are set to 0.50 and 0.20, respectively.

5.2 Experimental Results and Analyses

First of all, we conduct experiments to validate our proposed approach's performance in terms of optimization effectiveness and search efficiency. Since the baseline algorithms, including *POMC* and *EAMC*, and our proposed *MA-RAWR* are all randomized approaches, we repeat each of them 10 times independently, and report the average results. Specifically, for Foursquare social network, the node pool is fixed as 5,100, and the budget constraint C_{max} varies from 8,000 to 10,000 with an increment of 500. While for Gowalla data set, considering its large scale of more than 140,000 users, we employ a weighted-Monte Carlo sampling technique to reduce the dimensionality of the problem space, on the basis of each node's diffusion utility. To be specific, we sample 12,000 nodes, and vary the incentive budget C_{max} from 16,000 to 24,000 with an increment of 2,000.

Fig. 8 (a) and (b) present the experimental results of optimization effectiveness, using Foursquare and Gowalla data set, respectively. First, the expected task acceptance grows with the increase of the incentive budget C_{max} . The reason is that more workers can be recruited under a larger incentive budget, and the overall expected acceptance would be improved. Furthermore, our proposed *MA-RAWR* approach achieves the best performance, followed by *EAMC*, *POMC* and generalized *Greedy* algorithms. By integrating exploration and exploitation strategies, our proposed *MA-RAWR* algorithm can explore promising regions of the solution space, while enabling the local refinement. While for *EAMC* and *POMC* algorithms, even if the problem-specific heuristics are used, their performance still can not compete with *MA-RAWR* algorithm. The underlying reason is that the bit-wise mutation operation results in a slowly evolving process, especially for a large search space. Limited by the greedy nature, generalized *Greedy* algorithm achieve the worst performance, which is consistent with the finding in prior works.

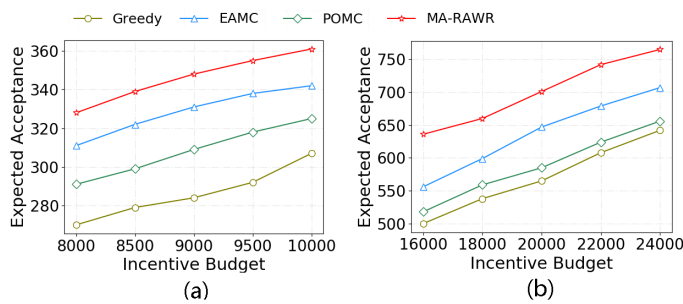


Fig. 8. The optimization effectiveness of different approaches.

In addition, we study the search efficiency of all the involved approaches, and the running time are presented in Fig. 9. We observe that our *MA-RAWR* algorithm yields the shortest running time, followed by *EAMC*, *POMC* and gen-

eralized *Greedy*. Actually, benefit from the heuristic knowledge and global exploration, the *MA-RAWR* algorithm offers fast convergence speed. During the *MA-RAWR* algorithm's run, it is found that the most time consuming component is variable neighborhood search, because it requires to repeatedly exploit neighborhood space and evaluate the task acceptance estimation. Furthermore, compared to other algorithms, our proposed *MA-RAWR* is less sensitive to the incentive budget C_{max} , as it does not sequentially construct solutions. Due to the random bit-wise mutation operation, the solution population in the *POMC* algorithm expands rapidly. Thus, its evolution over time is rather slow. By adopting a bin structure, i.e., solutions contain the same size of nodes, the search efficiency of the *EAMC* algorithm has been improved compared with *POMC* algorithm. However, its execution time is still longer than that of our *MA-RAWR* algorithm. Generalized *Greedy* algorithm has the longest running time, because it requires to traverse all the involved nodes in each iteration. In conclusion, compared to the second best approach, i.e., *EAMC* algorithm, the expected acceptance of our *MA-RAWR* algorithm increases about 7.77%, while the average running time decreases by about 28.2%.

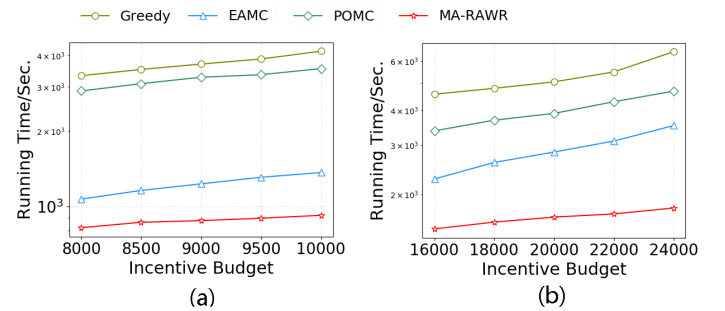


Fig. 9. The search efficiency of different approaches.

Second, to validate the scalability of our proposed approach, we perform experiments on the large-scale Gowalla social network. To be specific, the number of sampled nodes varies from 10,000 to 14,000 with an increment of 1,000, under the budget constraint $C_{max} = 20,000$. The corresponding results, including optimization effectiveness and search efficiency, are reported in Fig. 10. Clearly, with the increase of the number of involved nodes, the expected task acceptance of all the approaches grows accordingly, because more high-profile nodes might be included and selected as seed nodes. With respect to the search efficiency, the performance of *MA-RAWR* algorithm is significantly better than other baseline algorithms. With the increase of the number of involved nodes, the scale of the solution population/bins also grows. It thus requires more running time for *EAMC* and *POMC* algorithms to construct their final solutions.

We next investigate the impact of the related parameters including ℓ_{max} , R_0 and R_1 . Using the first Foursquare data set, we conduct experiments under the conditions of 5,100 nodes and 10,000 incentive budget, by varying parameter ℓ_{max} from 2 to 4 with an increment of 1. The experimental results are present in Fig. 11 (a). We observe that, with the increase of ℓ_{max} , the expected task acceptance grows accordingly. The possible reason is that, a larger value of

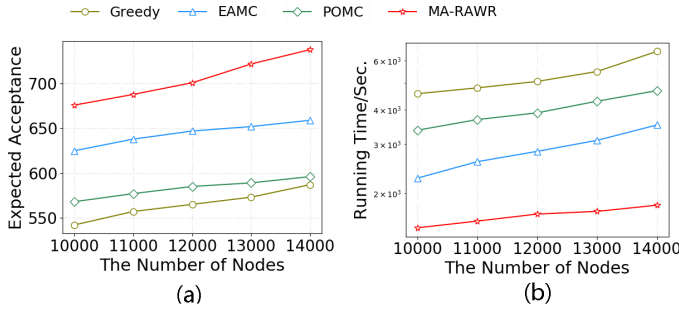


Fig. 10. The scalability of different approaches.

parameter ℓ_{max} enlarges the depth of the diffusion tree. Consequently, each seed node's diffusion utility, i.e., task acceptance of the diffusion tree, will increase.

We also examine the impact of parameters of $\mathcal{R}_0$ and $\mathcal{R}_1$ in our incentive mechanism, by studying different combinations of them, including (1) $\mathcal{R}_0 = 15$ and $\mathcal{R}_1 = 10$; (2) $\mathcal{R}_0 = 15$ and $\mathcal{R}_1 = 15$; (3) $\mathcal{R}_0 = 10$ and $\mathcal{R}_1 = 15$. The experimental results are reported in Fig. 11 (b). We can see that the selection of incentive parameters, including $\mathcal{R}_0$ and $\mathcal{R}_1$, affect the performance of all these studied approaches. Specifically, a larger value of the incentive parameters results in the decrease of expected task acceptance, due to the fact that the recruitment cost per worker becomes higher.

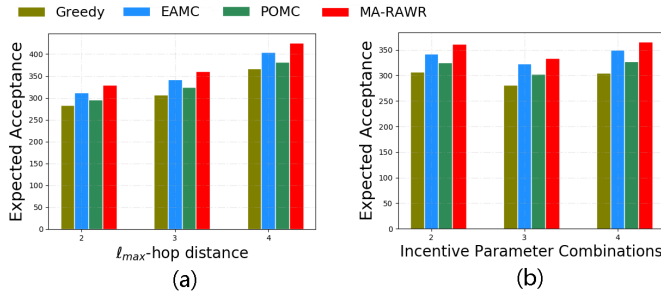


Fig. 11. The impact of different parameters.

Finally, we show the evolution process of our MA-RAWR algorithm in Fig. 12. We observe that the expected task acceptance grows with the increase of the number of evolutionary generations. Thanks to the variable neighborhood search, it enables MA-RAWR to escape from local optima and thus improve its acceptance estimation.

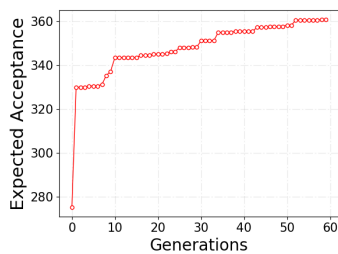


Fig. 12. The evolution process of MA-RAWR algorithm.

6 RELATED WORK

In this section, we provide a brief review of the related research on the following topics.

Information Propagation in Social Networks: With the prevalence of online social networks, information or influence can spread quickly through the “word-of-mouth” propagation among social neighbors [27]. Kempe et al. are the first to formally define influence spread models [28], and prove the NP hardness of Influence Maximization (IM) problem, i.e., selecting a subset of high-profile nodes which can influence the largest number of nodes in the social network. Due to its NP hardness, an exhaustive search by verifying all the possible solutions is not practical. As a result, low-complexity methods are much needed to achieve near optimal solutions in a reasonable time. For example, a greedy-based optimization approach is proposed with a factor approximation ratio in [28]. Leskovec et al. devise a lazy-forward heuristic approach, namely CELF, by harnessing the sub-modularity property [29]. Borgs et al. propose a concept of random reverse reachable set, and devise a near-linear time approach RIS [30]. Although the worker recruitment game studied in this work also requires to diffuse information, i.e., MCS task invitation, over social networks, our goal is to identify high-profile nodes based on MCS task specifications and improve their acceptance, rather than flooding nodes as much as possible.

Worker Recruitment in MCS: Recent years have seen an increasing interest in MCS as exemplified by lots of surveys and tutorials. In MCS, one of the important issues is worker recruitment, which enlists appropriate workers to undertake specified tasks. Under desirable time budget, Chen et al. make stochastic MCS recommendation based on workers' historical trajectories [31]. In [13], Zheng et al. are the first to take workers' rejection into consideration, and strive to maximize workers' acceptance to enhance the system throughput. In [32], the worker recruitment problem is investigated in the environment of vehicle-based MCS. Kang et al. consider a quality-aware online MCS task assignment problem, with the goal of optimizing the overall task quality [33]. By considering individual task quality assurance, worker recruitment issue for multiply MCS tasks are studied in [34]. Jiang et al. investigate a batch allocation problem for crowdsourcing tasks with overlapping skill requirements [35]. In [36], an unknown worker recruitment problem in MCS is studied, where the workers' sensing qualities are unknown a priori. By transform the problem into a combinatorial multi-armed bandit problem, the authors propose an extended UCB based algorithm. The above-mentioned research works all focus on direct worker recruitment, where the factors of social network structures, task invitation propagation, and task acceptance have not been considered and investigated.

Moreover, there are a few research efforts concentrating on integrating MCS campaign with social networks. In [37], [38], the authors study a MCS task assignment problem in mobile social networks, where the task owner moves around, and sends the task invitation to another mobile user when they encounter with each other. The contextual workers in social networks are utilized to perform complex MCS tasks by autonomous coordination, and improve the

reliability of crowd participants [39]. In [40], a strategic social team problem is studied, where a team of workers that are socially connected work together in collaboration to solve complex tasks. A socially aware task selection problem in MCS is studied, where each user independently selects tasks to undertake, and shares the incentive reward with others who execute the same tasks [41]. In [42], user interaction behavior is modeled as a Stackelberg game to incentivize workers to make a maximum contribution via word of mouth. A dynamic incentive mechanism, namely SocialRecruiter, is proposed to encourage workers to spread tasks on social networks [43]. However, these works focus on worker collaboration models [39], [40] or incentive mechanisms [41], [42], which are different from our problem. The most relevant work is [9]. By leveraging an influence propagation process, the authors formulate a social-network assisted worker recruitment problem in MCS paradigm. However, it differs from our work in the following aspects: 1) from the perspective of optimization objective, it strives to maximize the spatiotemporal coverage, while ours is to optimize workers' expected task acceptance; 2) from the perspective of propagation model, it directly extends an existing independent cascade model/linear threshold model to model influence propagation, i.e., flooding task invitations to all the nodes during the diffusion process. Instead, we devise a specialized and more practical MCS task invitation diffusion model with limited hop distances, where users who receive a task invitation independently make decision whether to accept a task or not. Thus, it makes our problem more practical and substantially different from the previous work, and hence it calls for new approaches.

7 CONCLUSION AND FUTURE WORK

In this paper, we propose and study a novel worker recruitment game in socially aware MCS, namely acceptance-aware worker recruitment problem. To better accommodate MCS task diffusion over social networks, we specifically devise a random diffusion model by considering users interaction and autonomous decisions. We then formulate a combinatorial optimization problem to search a subset of seed workers, with the goal of maximizing overall task acceptance under given incentive budget constraints. This problem is proved to be NP hard, and we propose a meta-heuristic-based evolutionary approach. Extensive experiments show the effectiveness of our proposed approach on two real-world social network data sets.

In the future work, we will further investigate how to enhance the robustness of our system in terms of the involved factors, i.e., spatial proximity and interest preference. The reason lies in that the final performance of our system will be impacted by the low accuracy of task acceptance estimation, resulting from spatial proximity and interest preference calculation. Clearly, to cope with this situation, one possible approach might be to increase the incentive budget. However, the method from the dimension of monetary incentives might not be economical for the task owners. So, we would like to solve it on the technical side. Additionally, another metric, namely task selection diversity, might be applied to our system, to avoid the

skewed task selection issue, i.e., a very few tasks have been selected by most users.

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China (No. 2018YFB2100800), the National Natural Science Foundation for Distinguished Young Scholars (No. 61725205), the National Natural Science Foundation of China (No. U2001207, 61972319, 61772428), and the Fundamental Research Funds for the Central Universities (No.31020180QD139).

REFERENCES

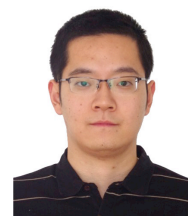
- [1] L Wang, Z Yu, Q Han, et al. Compact Scheduling for Task Graph Oriented Mobile Crowdsourcing. *IEEE Transactions on Mobile Computing*, 2020, DOI: 10.1109/TMC.2020.3040007
- [2] Y Gong, L Wei, Y Guo, et al. Optimal task recommendation for mobile crowdsourcing with privacy control. *IEEE Internet of Things Journal*, 2015, 3(5): 745-756.
- [3] L Wang, Z Yu, Q Han, B Guo, et al. Multi-objective optimization based allocation of heterogeneous spatial crowdsourcing tasks. *IEEE Transactions on Mobile Computing*, 2017, 17(7): 1637-1650.
- [4] Y Wang, W Dai, B Zhang, J Ma, et al. Word of mouth mobile crowdsourcing: Increasing awareness of physical, cyber, and social interactions. *IEEE MultiMedia*, 2017, 24(4): 26-37.
- [5] Y Wang, W Dai, Q Jin, J Ma. BciNet: A biased contest-based crowdsourcing incentive mechanism through exploiting social networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018, DOI: 10.1109/TSMC.2018.2837165.
- [6] J Nie, J Luo, Z Xiong, et al. An Incentive Mechanism Design for Socially Aware Crowdsensing Services with Incomplete Information. *IEEE Communications Magazine*, 2019, 57(4): 74-80.
- [7] S Chessa, A Corradi, L Foschini, et al. Empowering mobile crowdsensing through social and ad hoc networking. *IEEE Communications Magazine*, 2016, 54(7): 108-114.
- [8] A Mani, I Rahwan, A Pentland. Inducing peer pressure to promote cooperation. *Scientific reports*, 2013, 3: 1735.
- [9] J Wang, F Wang, Y Wang, D Zhang, et al. Social-Network-Assisted Worker Recruitment in Mobile Crowd Sensing. *IEEE Transactions on Mobile Computing*, 2018, 18(7): 1661-1673.
- [10] C Gao, J Liu. Network-based modeling for characterizing human collective behaviors during extreme events. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2016, 47(1): 171-183.
- [11] L Wang, Z Yu, D Yang, et al. Efficiently Targeted Billboard Advertising Using Crowdsensing Vehicle Trajectory Data. *IEEE Transactions on Industrial Informatics*, Vol. 16, No. 2, February 2020, pp. 1058-1066.
- [12] L Wang, Z Yu, F Xiong, et al. Influence Spread in Geo-Social Networks: A Multiobjective Optimization Perspective. *IEEE Transactions on Cybernetics*, 2019, DOI: 10.1109/TCYB.2019.2906078.
- [13] L Zheng, L Chen. Maximizing acceptance in rejection-aware spatial crowdsourcing. *IEEE Transactions on Knowledge and Data Engineering*, 2017, 29(9): 1943-1956.
- [14] Y Zhao, K Zheng, H Yin, et al. Preference-aware Task Assignment in Spatial Crowdsourcing: from Individuals to Groups. *IEEE Transactions on Knowledge and Data Engineering*, 2020, DOI: 10.1109/TKDE.2020.3021028.
- [15] IR Macneil. The new social contract. New Haven, CT: Yale University Press, 1980.
- [16] X Wang, Y Zhang, W Zhang, et al. Efficient distance-aware influence maximization in geo-social networks. *IEEE Transactions on Knowledge and Data Engineering*, 2016, 29(3): 599-612.
- [17] MH Cheung, F Hou, et al. Make a difference: Diversity-driven social mobile crowdsensing. In *Proceedings of 2017 IEEE Conference on Computer Communications*. IEEE, 2017: 1-9.
- [18] PC Pinto, P Thiran, M Vetterli. Locating the source of diffusion in large-scale networks. *Physical review letters*, 2012, 109(6): 068702.
- [19] J Yang, J Liu. Influence maximization-cost minimization in social networks based on a multiobjective discrete particle swarm optimization algorithm. *IEEE Access*, 2017, 6: 2320-2329.
- [20] HT Nguyen, MT Thai, TN Dinh. A billion-scale approximation algorithm for maximizing benefit in viral marketing. *IEEE/ACM Transactions On Networking*, 2017, 25(4): 2419-2429.

- [21] T Huang, YJ Gong, S Kwong, et al. A Niching Memetic Algorithm for Multi-Solution Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, 2019, DOI: 10.1109/TEVC.2019.2936440.
- [22] S Wang, J Liu, Y Jin. Finding influential nodes in multiplex networks using a memetic algorithm. *IEEE transactions on cybernetics*, 2019, DOI: 10.1109/TCYB.2019.2917059.
- [23] MA Potter, KAD Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary computation*, 2000, 8(1): 1-29.
- [24] P Hansen, N Mladenović, et al. Variable neighbourhood search: methods and applications. *Annals of Operations Research*. 2010, 175: 367-407. doi:10.1007/s10479-009-0657-6.
- [25] C Bian, C Feng, C Qian, Y Yu. An Efficient Evolutionary Algorithm for Subset Selection with General Cost Constraints. In *Proceedings of AAAI*. 2020, DOI: <https://doi.org/10.1609/aaai.v34i04.5726>.
- [26] C Qian, JC Shi, Y Yu, K Tang. On Subset Selection with General Cost Constraints. In *Proceedings of IJCAI*. 2017, 17: 2613-2619.
- [27] Y Li, J Fan, Y Wang, et al. Influence maximization on social graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2018, 30(10): 1852-1872.
- [28] D Kempe, J Kleinberg, et al. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2003: 137-146.
- [29] J Leskovec, A Krause, C Guestrin, et al. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2007: 420-429.
- [30] C Borgs, M Brautbar, J Chayes, et al. Maximizing social influence in nearly optimal time. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2014: 946-957.
- [31] C Chen, SF Cheng, HC Lau, et al. Towards city-scale mobile crowdsourcing: Task recommendations under trajectory uncertainties. In *Proceedings of Twenty-Fourth International Joint Conference on Artificial Intelligence*. 2015: 1113-1119.
- [32] Z He, J Cao, X Liu. High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility. In *Proceedings of 2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015: 2542-2550.
- [33] Y Kang, X Miao, K Liu, et al. Quality-aware online task assignment in mobile crowdsourcing. In *Proceedings of 12th International Conference on Mobile Ad Hoc and Sensor Systems*. IEEE, 2015: 127-135.
- [34] J Wang, Y Wang, D Zhang, et al. Multi-task allocation in mobile crowd sensing with individual task quality assurance. *IEEE Transactions on Mobile Computing*, 2018, 17(9): 2101-2113.
- [35] J Jiang, B An, Y Jiang, et al. Batch allocation for tasks with overlapping skill requirements in crowdsourcing. *IEEE Transactions on Parallel and Distributed Systems*, 2019, 30(8): 1722-1737.
- [36] G Gao, J Wu, M Xiao, et al. Combinatorial Multi-Armed Bandit Based Unknown Worker Recruitment in Heterogeneous Crowdsensing. In *Proceedings of the IEEE INFOCOM*, 6-9 July 2020
- [37] M Xiao, J Wu, L Huang, et al. Multi-task assignment for crowdsensing in mobile social networks. In the *Proceedings of 2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015: 2227-2235.
- [38] M Xiao, J Wu, L Huang, et al. Online task assignment for crowdsensing in predictable mobile social networks. *IEEE Transactions on Mobile Computing*, 2016, 16(8): 2306-2320.
- [39] J Jiang, B An, Y Jiang, D Lin. Context-aware reliable crowdsourcing in social networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017, 50(2): 617-632.
- [40] W Wang, Z He, P Shi, W Wu, Y Jiang, et al. Strategic social team crowdsourcing: Forming a team of truthful workers for crowdsourcing in social networks. *IEEE Transactions on Mobile Computing*, 2018, 18(6): 1419-1432.
- [41] S Chen, M Liu, S Sun, et al. Socially Aware Task Selection Game for Users in Mobile Crowdsensing. In *Proceedings of GLOBECOM* 2018. IEEE, 2018: 1-7.
- [42] F Zeng, R Wang, J Wu. How Mobile Contributors Will Interact With Each Other in Mobile Crowdsourcing With Word of Mouth Mode. *IEEE Access*, 2019, 7: 14523-14536.
- [43] Z Wang, Y Huang, X Wang, et al. SocialRecruiter: Dynamic Incentive Mechanism for Mobile Crowdsourcing Worker Recruitment

with Social Networks. *IEEE Transactions on Mobile Computing*, 2020, DOI: 10.1109/TMC.2020.2973958.



Liang Wang received the Ph.D. degree in computer science from Shenyang Institute of Automation (SIA), Chinese Academy of Sciences, Shenyang, China, in 2014. He is currently an Associate Professor with Northwestern Polytechnical University, Xi'an, China. His research interests include ubiquitous computing, mobile crowd sensing, and data mining.



Dingqi Yang received his Ph.D. in Computer Science from Pierre and Marie Curie University (Paris VI) and Institut Mines-TELECOM/TELECOM SudParis. He is currently an Associate Professor with the State Key Laboratory of Internet of Things for Smart City, Department of Computer and Information Science, University of Macau, Macau SAR, China. His research interests lie in ubiquitous computing, and social media data analytics.



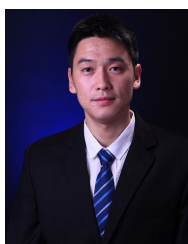
Zhiwen Yu received the Ph.D. degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2006. He is currently a Professor and the Dean of the School of Computer Science, Northwestern Polytechnical University, Xi'an, China. He was an Alexander Von Humboldt Fellow with Mannheim University, Germany, and a Research Fellow with Kyoto University, Kyoto, Japan. His research interests include ubiquitous computing and HCI.



Qi Han received the Ph.D. degree in computer science from the University of California, Irvine, CA, USA, in August 2005. She is a Professor of computer science with the Colorado School of Mines, Golden, CO, USA. Her research interests include mobile crowd sensing and ubiquitous computing.



En Wang received his Ph.D. in Computer Science and Technology from Jilin University, Changchun, China, in 2016. He is currently an Associate Professor with Jilin University, Changchun, China. He is also a Visiting Scholar with Temple University, Philadelphia, PA, USA. His current research interests include the energy-efficient communication between human-carried devices, and mobile crowdsensing.



Kuang Zhou received the doctor degree in University of Rennes 1 in 2016. Now he is an assistant professor at School of Mathematics and Statistics in Northwestern Polytechnical University, Xi'an, China. His research work focuses on uncertain statistical mining, imprecise probability and graph data clustering.



Bin Guo received the Ph.D degree in computer science from Keio University, Minato, Japan, in 2009. He was a postdoctoral researcher with the Institut TELECOM SudParis, Evry, France. He is currently a professor with Northwestern Polytechnical University, Xi'an, China. His research interests include ubiquitous computing, mobile crowd sensing, and HCI.