Multi-Objective Optimization Based Allocation of Heterogeneous Spatial Crowdsourcing Tasks

Liang Wang, Zhiwen Yu, *Senior Member, IEEE*, Qi Han, *Member, IEEE*, Bin Guo, and Haoyi Xiong, *Member, IEEE*

Abstract—With the rapid development of mobile networks and the proliferation of mobile devices, spatial crowdsourcing, which refers to recruiting mobile workers to perform location-based tasks, has gained emerging interest from both research communities and industries. In this paper, we consider a spatial crowdsourcing scenario: in addition to specific spatial constraints, each task has a valid duration, operation complexity, budget limitation, and the number of required workers. Each volunteer worker completes assigned tasks while conducting his/her routine tasks. The system has a desired task probability coverage and budget constraint. Under this scenario, we investigate an important problem, namely heterogeneous spatial crowdsourcing task allocation (HSC-TA), which strives to search a set of representative Pareto-optimal allocation solutions for the multi-objective optimization problem, such that the assigned task coverage is maximized and incentive cost is minimized simultaneously. To accommodate the multi-constraints in heterogeneous spatial crowdsourcing, we build a worker mobility behavior prediction model to align with allocation process. We prove that the HSC-TA problem is NP-hard. We propose effective heuristic methods, including multi-round linear weight optimization and enhanced multi-objective particle swarm optimization algorithms to achieve adequate Pareto-optimal allocation. Comprehensive experiments on both real-world and synthetic data sets clearly validate the effectiveness and efficiency of our proposed approaches.

Index Terms—Heterogeneous spatial crowdsourcing, multi-objective optimization, particle swarm optimization, mobility prediction

1 Introduction

The dramatic proliferation of mobile devices and the easy access to wireless mobile network is enabling a novel distributed problem-solving paradigm, namely *Spatial Crowdsourcing* (SC) [1], [2], which exploits the power of the crowd to perform location based tasks in the real world. Different from traditional crowdsourcing, spatial crowdsourcing tasks are location dependent and workers should physically travel to a specific location to complete the designated tasks. In recent years, spatial crowdsourcing has received increasing attention from both academia (e.g., [3], [4], [5]) and industry (e.g., Picasa [6], Gigwalk [7]).

Among many emerging SC applications, the involved tasks are heterogeneous and complicated, rather than homogeneous and trivial ones. For instance, a SC task may need to search a tagged area for a lost child [8], report real-time traffic flow within the targeted region [9] and so on. Obviously, these tasks are more complex and time-consuming to conduct than others such as checking street signs [10] or taking a photo of a scenic spot [5]. Inspired by the appearance of complex spatial tasks, in this work, we will consider a kind of spatial crowdsourcing, namely *Heterogeneous Spatial Crowdsourcing* (HSC). In addition to spatial location constraints, the complexity of HSC arises from several perspective—task completion duration (i.e., a task must be completed during a specified duration), operation complexity, budget limitation, and the number of required workers. To improve reliability, we aggregate all the returned results that may be redundant. Each individual worker selected for a specific task should independently perform the assigned task, and gain the announced reward.

Consider a scenario of heterogeneous spatial crowdsourcing, six different spatial tasks are released as shown in Fig. 1: t_1, t_2, t_3, t_4, t_5 and t_6. Each task consists of specified physical location, valid time interval, operation complexity, incentive rewards, and the number of required workers. Without loss of generality, the operation complexity is indicated by an estimated time needed to complete the task, and the reward is proportional to the operation complexity. For instance, task t_1 (e.g., computing statistics about traffic flow) defined as a tuple $(loc_1, 8:00 \sim 9:00, 3\,\text{min}, \$3, 3)$ needs three workers to independently conduct it at landmark loc_1 between 8 am and 9:00 am. The task will take roughly three minutes and three dollars will be rewarded to each worker who completes the task within the time duration.

To accomplish spatial tasks, SC applications need to select the best subset of available workers by matching task requirements with worker profiles. Recent studies have established that spatial locations have a significant impact

- *L. Wang is with the School of Computer Science and Technology, Northwestern Polytechnical University and Xi'an University of Science and Technology, Xi'an, Shaanxi 710065, China. E-mail: liangwang0123@gmail.com.*
- *Z. Yu and B. Guo are with the School of Computer Science and Technology, Northwestern Polytechnical University, Xi'an, Shaanxi 710065, China. E-mail: zhiwenyu@nwpu.edu.cn, guobin.keio@gmail.com.*
- *Q. Han is with the Colorado School of Mines, Golden, CO 80401. E-mail: qhan@mines.edu.*
- *H. Xiong is with the Missouri University of Science and Technology Smart CPS Cluster Hires Startup, Rolla, MO 65409. E-mail: haoyi.xiong.fr@ieee.org.*

Manuscript received 5 May 2017; revised 3 Sept. 2017; accepted 2 Nov. 2017. Date of publication 8 Nov. 2017; date of current version 1 June 2018. (Corresponding author: Liang Wang.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2017.2771259

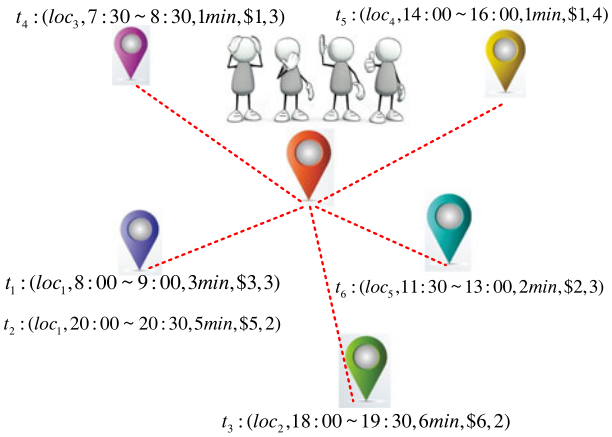


Fig. 1. The scenario of HSC tasks.

on the quality of final results for spatial crowdsourcing [10], [11]. By aligning spatial tasks with workers' daily routines, the disturbance to workers' main business in completing spatial tasks can be lowered, and the incentive cost incurred by longer detours can be decreased [12], [13], [14]. Moreover, the chances of cheating results (i.e., generate false responses without actually visiting the specified task location) will be decreased. However, to accommodate the *Multiple Constraints* in heterogeneous spatial crowdsourcing (i.e., spatial location, valid duration, processing time and number of required workers) is a challenge, as we need to consider not only crowd workers' potential moving trajectory, but also their stay duration. In the paper, we mathematically formulate the heterogeneous SC task allocation problem (*HSC-TA Problem*), which aims to efficiently assign all existing heterogeneous tasks to the right workers, with objectives of *maximizing crowdsourcing quality (i.e., the possibility with which selected workers will perform assigned task is maximized) and minimizing incentive budget (i.e., the total incentive budget paid to all selected workers is minimized) simultaneously*.

In practice, these two objectives are contradictory, if we improve one objective, the other will degrade accordingly. In other words, there really is no an optimal solution that can simultaneously improve the multi-objectives, but a set of Pareto-optimal solutions [16]. Several existing studies have explored the task assignment problem in SC using multi-objetive optimizations. They either strive to optimize one objective with the other objective as a fixed boundary constraint [17], [18], [19] or identify an allocation scheme based on a pre-given tradeoff between these two objectives [20], [21], [22]. However, no matter whether it is a boundary constraint or a tradeoff guidance, they all assumed to be predefined by requestors. We argue that the SC task requirements and available volunteer workers are dynamic and stochastic. Before task allocation, a requestor usually does not have a clear idea about the matching degree between his/her released tasks and volunteer workers. Therefore, it is unrealistic to require requestors to determine a satisfying tradeoff between multiple objectives a priori, not to mention claiming a reasonable boundary to guide the allocation process. Further, these objectives usually do not increase or decrease linearly. Consider two potential allocation schemes, where scheme A: $(Quality = 50\%, Cost = 10\$)$, scheme B: $(Quality = 80\%, Cost = 11\$)$. Assuming that an upper bound of budget cost is given as 10$ in advance, thus scheme A will be provided as an optimal allocation to the requestor, and the potential allocation B will not be examined. In fact, it is cost-efficient to achieve 30 percent increment in crowdsourcing quality with slight increase of cost (i.e., 1$). However, because we have no knowledge of the overall distribution of different solutions, the more desirable solution (i.e., scheme B) may be missed. In practice, this situation is unavoidable, and we will validate it in our experiments later. Hence, a practical approach is to provide all Pareto-optimal solutions (i.e., Pareto set) with respect to involved multi-objectives for requestors to make a reasonable decision. In other words, our job is to discover adequate Pareto-optimal allocation solutions for decision-maker.

As discussed above, we could not employ the prior approaches to solve our problem which is provably a NP-Hard problem. In response to the challenges mentioned above, we make following major contributions.

- We formally define the heterogeneous spatial crowdsourcing task allocation problem in Section 3, under the multiple constraints, with multi-objectives optimization of task coverage and incentive budget simultaneously. To the best of our knowledge, this is the first framework solving heterogeneous spatial task allocation issue, leveraging the participant users' comprehensive mobility perdition in Section 4 to achieve Pareto-optimal solutions.
- In order to solve the multi-objective optimization problem and obtain adequate Pareto-optimal solutions, we propose efficient heuristic approaches, namely multi-round linear weight optimization (MRLWO) algorithm, and enhanced multi-objective particle swarm optimization (EMOPSO) algorithm to tackle the HSC-TA problem in Section 5.
- We conduct extensive experiments using real and synthetic data sets to evaluate our proposed allocation algorithms on small and large-scale environment. The results show the effectiveness and efficiency of our proposed HSC-TA solving approaches in Section 6.

In addtion, Section 2 reviews previous work on spatial crowdsourcing. We conclude the paper in Section 7 .

2 Related Work

More recently, SC has received lots of attention and many new applications have emerged. In SC applications, the requestor posts tasks and recruits workers to complete them. Due to the temporal and spatial constraints of the tasks and background knowledge of different workers, not everyone is qualified to undertake a specific task. Therefore, how to select an appropriate worker from candidate worker pool to perform a specific spatial task is important as it can directly affect the quality of final result [23]. There has been recent work on SC task allocation (e.g., worker selection). In general, two types of task allocation have been studied, they are *Worker Selected Tasks* (WST) and *Server Assigned Tasks* (SAT) [3].

Under WST mode, all tasks are available and visible to all workers, and workers autonomously select their desired tasks to maximize their own benefit [4], [24]. Because tasks are loosely organized by crowd workers, the server can not control the overall quality of spatial crowdsourcing. For instance, many tasks located in hotspot regions are

TABLE 1
Frequently Used Symbols

Symbol	Explanation
$\mathbb{T} = \{t_i\}$	Heterogeneous spatial tasks
$\mathbb{W} = \{w_j\}$	Available volunteer workers
$\mathbb{R} = \{loc_i\}$	Spatial locations
$\mathbb{TW} = \{tw_k\}$	Discrete split time window
QM/AM	Qualification/Allocation Matrix
CpT/DpT	Task Coverage/Cost Discount per Task
$vt_i = [st_i, et_i]$	Valid time interval of task t_i
pt_i	Processing time of task t_i
dm_i	Number of required workers of task t_i
rw_i	Marked rewards of task t_i
S_q^j	Locations visited in tw_k for worker w_j
θ_i^j	Dwell time distribution on loc_i for w_j
γ	Maximum assigned tasks per worker

competitive, while many tasks in non-hotspot regions are starving for long time. Hence, the majority of existing studies adopt the SAT mode, where the server takes charge of SC task allocation process. For each task, according to the matching between SC task requirements and workers' characteristic, the SC-server algorithmically selects suitable workers from the pool of workers [23], [26]. Reddy et al.[23] identify well-suited participant users according to spatial-temporal availability and participation habits. Cheng et al. consider the skill requirements in SC task, and propose heuristic approaches based on the matching of qualified skill [26]. However, the problem settings in those studies differ from ours in several aspects, including heterogeneous SC task requirements, volunteer workers mobility prediction and multi-objective optimization allocation.

Recent studies have established that spatial location plays an important role in the quality of spatial crowdsourcing results (e.g., the willingness to accept assigned tasks, the reliability of returned result report) [10], [11]. In particular, an effective algorithm needs to capture the hidden mobility profile of workers and predict their future location by leveraging workers' historical traces. According to the spatial proximity, the SC server assigns tasks to their neighboring workers [15], [27]. More recently, Kandappu et al. investigate mobile crowdsourcing in campus environment, and the results demonstrate that trajectory-aware task recommendation does outperform the WST-based approach [12], [13]. Moreover, by employing a "task bundling" strategy, the incentive cost per task can be decreased. In this paper, we also adopt the mobility pattern based strategy. However, the existing approaches can not accommodate heterogeneous SC tasks. Existing approaches either choose the Truncated Lévy Walk (TLW) model where constant moving speed and direction are assumed [28], extract limited hotspot regions to connect a popular route with maximum likelihood [12], [13], or directly employ a statistical probability distribution to derive hot location in given time segment [15], [19], [30]. Obviously, these approaches is unrealistic or can not apply into our scenario.

In practice, the crowdsourcing quality and recruitment budget are two important issues to be addressed in spatial crowdsourcing applications. Wang et al. propose a fine-grained multitask allocation framework to maximize task coverage ratio under a given shared budget constraint [19]. In [29], a budget-aware task assignment approach is proposed towards the goal of maximizing the expected quality of returned results within a limited budget. In [30], Xiong et al. propose a task allocation framework, named iCrowd, to achieve dual optimal goals which are k-depth coverage maximization and incentive payment minimality. Although the multi-objectives of result quality and incentive budget are both needed in these studies, one of the objectives is usually handled as a constant constraints given a-priori [18]. A few studies attempt to search a local-optimal allocation solution based on a given tradeoff between these two objectives, instead of providing all Pareto-optimal solutions [17], [20], [21], [22]. Therefore, neither of existing studies can provide a complete picture with respect to spatial crowdsourcing quality and budget incentive.

3 Problem Definition

We formally define the HSC-TA problem in this section. For ease of illustration, we list the notations used frequently in this paper in Table 1.

Definition 1 (Multi-Constraints Heterogeneous Task). *Let $\mathbb{T} = \{t_1, t_2, \ldots, t_n\}$ be a set of n multi-constraints heterogeneous tasks. Each task $t_i (1 \leqslant i \leqslant n)$ is a tuple of five elements: $(loc_i, vt_i, pt_i, rw_i, dm_i)$. It requires dm_i workers to complete within physical location loc_i during time interval $vt_i = [st_i, et_i]$. It takes pt_i time units for each worker to complete t_i and each worker may get paid rw_i.*

For simplicity, we transform continuous temporal dimension into discrete time windows $\mathbb{TW} = \{tw_1, tw_2, \ldots, tw_p\}$. In this paper, the length of each time window tw is set as 30 minutes, and this temporal granularity can be easily tuned to adapt to many other applications. For different spatial task applications, the valid time interval vt may span one or more time windows, such as $[8:00, 8:30]$ or $[13:30, 15:00]$. Accordingly, vt can also be represented in the form of time windows, such as $[tw_i, tw_j]$, $1 \leqslant i \leqslant j \leqslant \frac{24*60}{|tw|}$, where $|tw|$ denotes the granularity of time window tw. For instance, the interval $[13:30, 15:00]$ can be translated as $[tw_{28}, tw_{30}]$ when the granularity of time window tw equals 30 minutes. The number of time windows $\mathbb{TW}$ is 48 (i.e., $\frac{24*60}{30}$).

The task processing time pt is estimated by a requestor and posted when this task is released. In essential, pt is an indicator of task complexity. If the task to be performed is a trivial one (e.g., to check whether the green bins need to be emptied [12]), its completion time will be short; on the contrary, the overhead time will be longer. On the other hand, processing time pt can also reflect the intrinsic cost of task itself (i.e., marked reward) which is irrelevant to which worker is selected to perform it and how to perform it. In the paper, we assume that the announced reward rw_i of task t_i is proportional to its processing time pt_i: $rw_i = \delta * pt_i$, where δ is a constant coefficient.

3.1 Worker Mobility Model

The mobility of volunteer workers incorporates the temporal information, visited spatial locations and the dwell time. To normalize the diversity of mobility over time dimension, we use the same time window concept described previously.

TABLE 2
Task Bid in Multi-Task Allocation Situation

Tasks	t_1	t_2	t_3	t_1, t_2	t_1, t_2, t_3
$\overline{bid}$	\$11.10	\$10.70	\$10.34	\$16.625	\$21.67

Definition 2 (Worker Mobility). *Let $\mathbb{W} = \{w_1, w_2, \ldots, w_m\}$ be a set of m volunteer workers. Worker w_j's mobility behavior in each time window tw_k can be formally represented as $(w_j, tw_k, S_k^j, \Theta_k^j) = (w_j, tw_k, \{loc_1, loc_2, \ldots, loc_q\}, \{\theta_1^j, \theta_2^j, \ldots, \theta_q^j\})$, $1 \leqslant j \leqslant m$, $1 \leqslant k \leqslant \frac{24*60}{|tw|}$. In each time window tw_k, worker w_j sequentially pass through different spatial location in $S_k^j = \{loc_1, loc_2, \ldots, loc_q\}$, and the dwell time associated with loc_* follow a specific probability distribution θ_*^j.*

During each time window tw_k, user w_j may traverse more than one spatial locations, or no locations (i.e., spatial location set S_k^j is empty). The probability distribution of θ_*^j can be derived from the historical traces of worker w_j's.

Intuitively, the definition of *Worker Mobility* captures not only worker's routes, but also the dwell time at each passed location. Using this definition, the mobility regularity can accommodate multi-constraints HSC tasks with stringent spatial and temporal constraints. Similar to previous research works (e.g., [15], [19], [30]), in this paper, we also employ an offline task allocation fashion. Specifically, all the participant workers register in platform as candidates, and their trajectory history are utilized by the platform only for the purpose of task allocation. Based on the predicted moving route, an appropriate subset of tasks is assigned and pushed to each worker before task execution.

3.2 Worker Incentive Model

We establish a worker incentive model under centralized allocation framework by conducting a questionnaire survey. We assume that, by aligning spatial task with workers' movement trajectory, the incentive cost incurred by performing assigned tasks can be reduced. This is because the explicit cost (e.g., traveling cost to accomplish assigned tasks) can be amortized by their own main business. In order to verify this hypothesis, we conduct a survey among 57 students in Northwestern Polytechnical University in Xi'an, China. In the questionnaire, some virtual spatial tasks (such as noise measurement, taking photos) are devised in campus environment. According to whether SC tasks are distributed by following workers' routine route, we ask about two task allocation schemes: random and mobility-aware.

Bid Price of Single Task. Given the same spatial task, each subject is required to publish his bid prices under two different allocation schemes respectively. The average bid prices under random allocation is \$10.45, 1.885 times of the price for the mobility-aware scheme.

Bid Price of Multiple Tasks. To investigate the cost for multiple tasks, we predefined many routine behavior in campus, such as $\{library \rightarrow restaurant \rightarrow public\ garden \rightarrow gymnasium\}$, and three different spatial tasks are located within these landmarks. Assuming that respondents will follow one of these given moving routine in the near future, we incrementally allocate each subject one, two and three tasks, and the respondents are required to bid a price each time.

After analyzing the returned biddings, we found that most respondents' biddings in multi-tasks are less than the sum of all the involved single task's price. Specifically, as shown in Table 2, the average price $\overline{bid}$ of bundling tasks, including task 1 and 2, is \$16.625, which is less than the sum of these two tasks' bids(i.e., $11.10 + 10.70 = 21.80$), by about 28.51 percent; and the $\overline{bid}$ of bundling tasks (i.e., task 1, 2 and 3) is \$21.67 which is less than the sum of all these three tasks' single bids by about 32.57 percent. Moreover, we found that the total incentive costs of multiple spatial tasks will increase exponentially, instead of a straight-forward linear increase. In other words, the more allocated tasks for a worker, the lower the average bid per task. This phenomenon is similar to the theory of scale effect in economics, that the cost per product will be decreased with the increasing of production scale. Moreover, the findings in [12], [31] also support it.

Therefore, in order to decrease the total budget cost, an effective strategy is to assign more tasks per worker under mobility-aware allocation scheme. For each selected worker w_j who is assigned tasks $T_j = \{t_1, t_2, \ldots, t_{n'}\}$, the corresponding incentive cost to be paid can be formulated as

$$Cos(T_j) = \frac{1}{e^{a*(n'-1)}} * \sum_{i=1}^{n'} rw_i = \frac{1}{e^{a*(n'-1)}} * \sum_{i=1}^{n'} \delta * tp_i, \quad (1)$$

where rw_i denotes the incentive reward of task $t_i, 1 \leqslant i \leqslant n'$, and the parameter a is used to control the decay rate, in our case, a is set to be 0.2. In particular, the first term (i.e., $\frac{1}{e^{a*(n'-1)}}$) in Eq. (1) is the decay coefficient which will reduce with the increasing scale of T_j. When one worker is recommended with only one task t_i, the corresponding incentive cost is exactly task t_i's reward rw_i.

3.3 Heterogeneous SC Task Allocation

The HSC-TA problem we study in this work aims to select suitable workers to perform heterogeneous spatial tasks considering two metrics: crowdsourcing quality represented by *Task Coverage* and incentive budget represented by *Task Cost*. The task coverage is the probability that spatial tasks are accomplished by selected workers, whereas the task cost indicates the total budget to be paid by the task requestor. Next, we will quantify these two criterions.

Task Coverage. Using crowd workers' predicted mobility, the existing spatial tasks can be matched with appropriate workers. Assuming that task t_i is assigned to worker w_j, as w_j is expected to pass t_i's specific location loc_i during its valid period vt_i, but it can not guarantee that w_j will surely complete t_i. This is because: 1) the fact that w_j will visit loc_i during the time interval vt_i is only in a probabilistic sense; 2) w_j's potential dwelling time on loc_i may not always satisfy processing time constraints tp_i. Thus, whether the selected worker w_j is capable of accomplishing task t_i is quantified by a probability, which depends on the likelihood that w_j visit loc_i during vt_i and the corresponding stay time distribution θ_i^j. We define this quantified probability as the *task coverage*.

To evaluate the task coverage quality of an allocation (e.g., assign task t_i to worker w_j), we formulate this metric denoted by $Cov(t_i, w_j)$ as follows:

$$Cov(t_i, w_j) = x_{i,j} * P(loc_i, vt_i | w_j) * P(\theta_i^j \geqslant pt_i), \quad (2)$$

where $x_{i,j}$ denotes a binary variable: set to 1 if task t_i is assigned to worker w_j, and 0 if not; $P(loc_i, vt_i | w_j)$ represents

the likelihood that user w_j will visit loc_i during time interval $vt_i = [st_i, et_i]$, and $P(\theta_i^j \geqslant pt_i)$ is the probability that the dwelling time w_j will stay at loc_i is no less than processing time constraint pt_i.

As the valid time duration vt_i may span more than one time windows, $P(loc_i, vt_i|w_j)$ can be rewritten as

$$P(loc_i, vt_i|w_j) = 1 - \prod_{tw_k \in vt_i} (1 - P(loc_i, tw_k|w_j)), \quad (3)$$

where $P(loc_i, tw_k|w_j) = P(loc_i \in S_q^j, tw_k|w_j)$ denotes the probability that worker w_j will visit loc_i during time window tw_k across the predicted moving route.

Task Cost. As shown above, under mobility-aware allocation, if we assign more tasks to one worker, the total incentive cost will decrease exponentially. But in order to balance the workload and avoid monopoly among workers, each worker should have a maximum capability to perform spatial tasks, in the paper, we use a upper bound threshold γ to restrict the maximum number of assigned tasks per worker.

According to Eq. (1), if worker w_j is assigned a set of spatial tasks T_j, the corresponding incentive cost can be formulated as

$$Cos(T_j) = \frac{1}{e^{0.2*(|T_j|-1)}} * \sum_{i=1}^{|T_j|} \delta * tp_i, \quad (4)$$

where $|T_j| = \sum_{i=1}^{n} x_{i,j}$ denotes the number of tasks assigned to user w_j, $|T_j| \leqslant \gamma$.

Definition 3 (HSC-TA Problem). *Given n multi-constraints heterogeneous spatial tasks in $\mathbb{T}$, and m participant workers in $\mathbb{W}$, the problem of HSC-TA is to assign each task $t_i \in \mathbb{T}$ with a set, W_i, of workers $w_j \in \mathbb{W}$, such that:*

1. *the total budget cost of all spatial tasks which are paid to workers $w_j \in \mathbb{W}$ is minimized, and*
2. *the task probability coverage of all spatial tasks assigned to workers is maximized;*

and subject to the following two constraints:

1. *each worker $w_j \in \mathbb{W}$ is allocated a spatial task $t_i \in \mathbb{T}$ such that his/her arrival time at location loc_i will fall into the duration of valid time interval $vt_i = [st_i, et_i]$,*
2. *the dwell time of user w_j stay at loc_i will be no less than the processing time pt_i.*

Formally, our HSC-TA problem is to concurrently optimize both objectives as follow:

$$\begin{cases} max: & \sum_{t_i \in \mathbb{T}} \sum_{w_j \in \mathbb{W}} x_{i,j} * P(loc_i, vt_i|w_j) * P(\theta_i^j \geqslant pt_i), \\ min: & \sum_{j=1}^{m} \frac{1}{e^{0.2*(|T_j|-1)}} * \sum_{i=1}^{|T_j|} \delta * tp_i, \end{cases} \quad (5)$$

subject to: $\forall\ w_j, |T_j| \leqslant \gamma$, and $\forall\ t_i, |W_i| \leqslant dm_i$, where T_j is the set of spatial tasks assigned to worker w_j, and W_i is the set of selected workers to perform task t_i.

3.4 Hardness Analysis of HSC-TA Problem

HSC-TA is an optimization problem with two objectives. Given n multi-constrained heterogeneous spatial tasks and m available workers, in the worst case, the possible task-worker pair allocation solution can be numerous, thus the searching all possible allocation solutions can be computationally infeasible ($O((n+1)^m)$). We can prove that our HSC-TA problem is an NP-hard problem. Consider a simplified version of the problem, in which we aim to maximizing the objective *Task Coverage*, and the residual objective (i.e., *Task Cost*) is transformed into a constraint conditions by adding a fixed bound, such as C. For each allocation solution, there is a utility associated with *Task Coverage*, and the corresponding *Task Cost* can not exceed the given budget bound C. This is an NP-hard problem, because it can be reduced from the well-known Knapsack problem [32]. Therefore, our HSC-TA problem is also NP-hard.

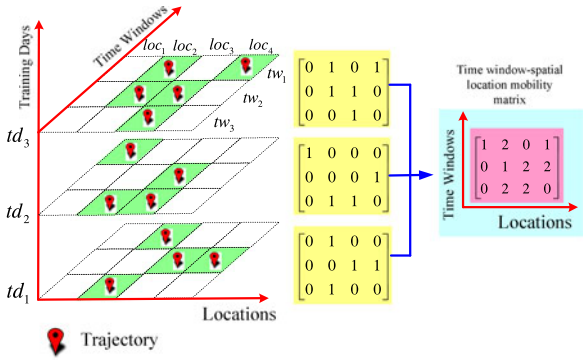


Fig. 2. Worker mobility pattern in three-order tensors.

Since the HSC-TA problem involves conflicting bi-objectives to be optimized, and the associated multi-constraints should also be covered by candidate workers' potential mobility behavior, it is infeasible to directly employ existing approximation approaches to solve this problem. Hence, with regard to the NP-hardness and our HSC-TA problem, we will devise heuristic algorithms to efficiently search Pareto-optimal solutions.

4 Worker Mobility Prediction

In this section, we construct a mobility prediction model to accommodate the multi-constraints in our HSC-TA problem.

Given a set of moving trajectories of all available workers, for each training day $\mathbb{TD}$, we map each worker's trace records onto a two-dimensional space comprised of $\mathbb{TW}$ split time windows and $\mathbb{R}$ locations. And then, it counts $\lambda_{w,td,tw,loc}$. In each training day ($td \in \mathbb{TD}$), if worker ($w \in \mathbb{W}$) reports his position within location ($loc \in \mathbb{R}$) during time window ($tw \in \mathbb{TW}$), $\lambda_{w,td,tw,loc} = 1$; otherwise, $\lambda_{w,td,tw,loc} = 0$.

For each worker w, we can construct a three-order tensor, including training day $\mathbb{TD}$, time window $\mathbb{TW}$, and spatial location $\mathbb{R}$. For example, as shown in the left of Fig. 2, one worker w's historical mobility trace within the period of $\mathbb{TD}$ training days (i.e.,from td_1 to td_3) is visualized, where $tw_k (k = 1, 2, 3)$ represents discrete time windows and $loc_i (i = 1, 2, 3, 4)$ denotes spatial locations. With regard to each training day, a time window-spatial location array can be filled with the value of $\lambda_{w,td,tw,loc}$ by counting the existing historical traces. In the left of Fig. 2, the rectangles in green denotes that worker w has visited location loc within time window tw and in training day td. By combining all two-dimensional arrays in the direction of $\mathbb{TD}$, we can obtain a composite time window-spatial location mobility matrix $\mathbb{V}$,

as shown in the right of Fig. 2, which indicates the mobility distribution regularity of worker w. In the composite matrix $\mathbb{V}$, every element $v_{k,i}$ is calculated as follows:

$$v_{k,i} = \sum_{td \in \mathbb{TD}} \lambda_{w,td,tw,loc}. \tag{6}$$

In order to discriminate worker's interesting locations during each time window, for composite matrix $\mathbb{V}$, we compute its average value ϕ, and define the time window-spatial location pairs having no less than ϕ as target locations (i.e., $v_{k,i} \geqslant \phi$). The threshold ϕ is formulated as follows:

$$\phi = \frac{\sum_{td \in \mathbb{TD}} \sum_{tw \in \mathbb{TW}} \sum_{loc \in \mathbb{R}} \lambda_{w,td,tw,loc}}{\sum_{td \in \mathbb{TD}} \sum_{tw \in \mathbb{TW}} \sum_{loc \in \mathbb{R}} sgn(\lambda_{w,td,tw,loc})}, \tag{7}$$

By compare $v_{k,i}$ with the average value ϕ, a set of interesting locations S can be obtained. In Eq. (7), we select a sgn function instead of the number of entries in $\mathbb{V}$, due to the observation that the trained user-oriented matrix $\mathbb{V}$ is usually sparsely distributed. The realistic reason is that users often visit a small number of spatial locations relative to the total number of spatial locations, and fine spatiotemporal granularity would also cause sparse expansion [24], [25].

After interesting locations are identified with Eq. (7), we then reconstruct sequential mobility by the refined time window-spatial location pairs S. Before doing this, we will elaborate two observations.

(1) Within time window tw, more than one location may be identified as interesting, such as S_{tw}, as a worker may regularly visit many regions during tw. However, not all the interesting locations in S_{tw} can be predicted as the potential visited regions. The reason is that these interesting locations may belong to different trips due to mobility diversity. Thus, we must extract a subset of locations which are most likely to be visited frequently for a specific worker in the future.

(2) Consider any two successive time windows, tw_k and tw_{k+1}, the hidden relationship of movement sequence which is defined by road network constraints, spatial neighborhood, and so on, should be concerned. By incorporating this sequential feature, a more realistic mobility schedule can be derived, rather than to construct a predicted mobility route by simply linking interesting locations in S_{tw_k} and $S_{tw_{k+1}}$ [12], [13].

Based on these two observations, we design a worker mobility prediction approach. First, in each time window tw_k, we group the interesting locations according to the co-occurrence frequencies throughout the existing trace history, for instance $\{S^1_{tw_k}, S^2_{tw_k}, \ldots, S^h_{tw_k}\}$. One subset $S^*_{tw_k}$ with maximum occurrence probability (i.e., the elements in $\mathbb{V}$) will be chosen as target potential locations corresponding to worker w's predicted moving trajectory in time window tw_k. This process can be formalized as follows.

$$S^*_{tw_k} = \underset{1 \leqslant j \leqslant h}{arg} : max\ P(S^j_{tw_k}), \tag{8}$$

Afterwards, by leveraging a first-order Markov transition model learned from trace history, we prune the extracted $S^*_{tw_k}$ ($tw_k \in \mathbb{TW}$) in each time window, and connect these subsets consecutively to generate complete mobility prediction. Specifically, we implement the operation represented formally as: $P(\tilde{S}_{tw_{k+1}} | \tilde{S}_{tw_k}) \geqslant \sigma$, where $S^*_{tw_{k+1}} \subseteq \tilde{S}_{tw_{k+1}}$, $S^*_{tw_k} \subseteq \tilde{S}_{tw_k}$, and σ denotes a predefined threshold value. It means that the transition probability from locations in S_{tw_k} to locations in $S_{tw_{k+1}}$ must be no less than the given setting σ. When the prune operation is finished, the complete mobility behavior of a certain worker w, such as $\{\tilde{S}_{tw_1}, \tilde{S}_{tw_2}, \ldots, \tilde{S}_{tw_p}\}$ can be obtained.

Our worker mobility prediction not only includes the time window and spatial region information, but also the probability distribution of the dwell time θ. By counting all the historical dwell time at each location ($loc \in \mathbb{R}$) per worker w, it is not difficult to get θ^w_{loc}. Thus, worker w's mobility behavior with time window tw can be described as: $(w, tw, \{loc_1, loc_2, \ldots, loc_q\}, \{\theta^w_1, \theta^w_2, \ldots, \theta^w_q\})$.

5 Mobile Crowdsourcing Task Allocation

Our objective is to allocate tasks that best match predicted mobility of volunteer workers to simultaneously maximize *Task Coverage* and minimize *Task Cost*. For such a multi-objective optimization problem, there exists a set of solutions, namely non-dominated solution (i.e., Pareto solution) [16]. Among the Pareto solutions, without additional subjective preference information, all Pareto-optimal solutions are considered equally good. In the following, we will elaborate our proposed heuristic algorithms to achieving adequate Pareto-optimal solutions in detail.

5.1 Problem Reduction

With regard to the two objectives, the goal related to *Task Cost* strives to *decrease* the total incentive cost incurred by recruiting workers to conduct spatial tasks; while *Task Coverage* aims to *increase* the allocation probability coverage of spatial tasks. In order to unify the optimization direction of these two objectives, we reverse the objective of *Task Cost* into another form, namely *Discount per Task* (*DpT*). This metric indicates the discount we can achieve per task by assigning more tasks to each worker. As the number of existing spatial tasks is determined, *DpT* is equivalent to the total discount achieved by all the tasks, and is just the opposite of *Task Cost*. Thus, the *Discount per Task* can be achieved by dividing the total discount with the total number of assigned spatial tasks ($\sum_{i=1}^{n} dm_i$) and can be represented as follows:

$$DpT = \frac{1}{\sum_{i=1}^{n} dm_i} * \sum_{w_j \in \mathbb{W}} \left(1 - \frac{1}{e^{0.2*(|T_j|-1)}}\right) \sum_{t_i \in T_j} \delta * tp_i. \tag{9}$$

where T_j denotes the set of spatial tasks assigned to worker w_j, and $|T_j|$ represents the scale of T_j.

The smaller *Task Cost*, the larger the *Discount per Task*. In other words, these two metrics are consistent in the aspect of decreasing the budget cost. Accordingly, it is necessary to scale the *Task Coverage* metric to match the new metric *DpT*. We hence devise a *Coverage per Task* (*CpT*) to represent the allocation probability coverage per task. Thus, our optimization goals are transformed into the following form:

$$max : \frac{\sum_{t_i \in \mathbb{T}} \sum_{w_j \in \mathbb{W}} x_{i,j} * P(loc_i, vt_i | w_j) * P(\theta^j_i \geqslant pt_i)}{\sum_{i=1}^{n} dm_i}, \tag{10}$$

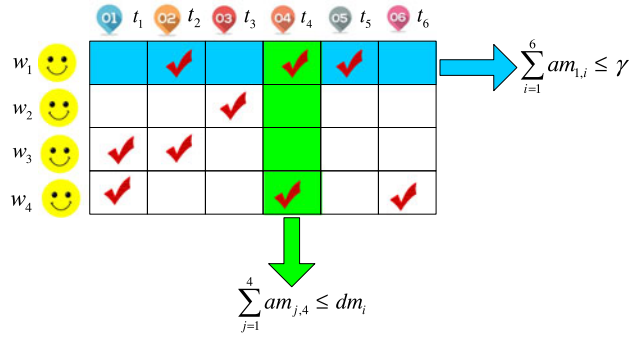


Fig. 3. The structure of allocation matrix AM.

and

$$max: \frac{\sum_{w_j \in \mathbb{W}} \left(1 - \frac{1}{e^{0.2*(|T_j|-1)}}\right) * \sum_{t_i \in T_j} \delta * tp_i}{\sum_{i=1}^{n} dm_i}, \quad (11)$$

subject to: $\forall\ w_j, |T_j| \leqslant \gamma$, and $\forall\ t_i, |W_i| \leqslant dm_i$. Based on the predicted worker mobility, the part of $P(loc_i, vt_i|w_j)$ in Eq. (10) can be calculated according to Eq. (3), such that

$$P(loc_i, vt_i|w_j) = 1 - \prod_{tw_k \in vt_i} (1 - P(loc_i \in S_{tw_k}^{w_j}, tw_k|w_j)), \quad (12)$$

where $S_{tw_k}^{w_j}$ is the predicted mobility locations of worker w_j in time window tw_k.

In the first objective (Eq. (10)), the role of $P(loc_i, vt_i|w_j) * P(\theta_i^j \geqslant pt_i)$ is to compute the qualification of worker w_j to the given heterogeneous spatial tasks t_i (i.e., w_j may pass t_i's specified location loc_i during its valid interval vt_i, and his dwelling time may be lager than t_i's processing time pt_i). If the value of qualification is more than zero, then worker w_j is qualified to perform task t_i; otherwise he is not. To simplify the problem, we propose a matrix structure QM (i.e., Qualification Matrix) to denote the qualifications of volunteer workers $\mathbb{W}$ to perform each spatial task $t_i, 1 \leqslant i \leqslant n$. Specifically, the rows in QM represent each worker in $\mathbb{W}$, and the columns correspond to spatial tasks in $\mathbb{T}$. If worker w_j is qualified to conduct task t_i, the corresponding entry $qm_{j,i} = P(loc_i, vt_i|w_j) * P(\theta_i^j \geqslant pt_i)$ will be more than zero; otherwise, it will be zero. By examine each worker-task pair and calculate the corresponding qualification individually, the matrix QM can be obtained.

Mathematically speaking, the established matrix QM is indeed the feasible space of task allocation solution. In other words, we only need to search the desired allocation solutions in this qualification indicator QM. To represent possible allocation solution, we employ a binary matrix AM (i.e., Allocation Matrix), in which the structure is similar to QM. If task t_i is assigned to worker w_j, the corresponding entry $am_{j,i}$ equals 1; otherwise, $am_{j,i} = 0$. The benefit of this structure is that the constraints of task upper bound γ and duplicates of each task dm_i can be be validated by the sum of each row/column in AM, as shown in Fig. 3. In other words, one possible allocation solution to HSC-TA problem is just an AM binary matrix.

Based on qualification indicator QM, our HSC-TA problem can be reduced into the form of searching a feasible binary matrix AM. Specifically, we can transform our problem into the form of matrix multiplication of these two matrices QM and AM. Therefore, we can obtain the reduced representation of our HSC-TA problem as follows:

$$arg: \max_{AM} \left\{ \frac{QM \bullet AM}{\sum_{i=1}^{n} dm_i}, \frac{\sum_{j=1}^{m} \left[1 - \frac{1}{e^{0.2*(\sum AM(j,:)-1)}}\right] \delta * Tp \bullet AM(j,:)}{\sum_{i=1}^{n} dm_i}, \right. \quad (13)$$

subject to: $\forall\ j, \sum AM(j,:) \leqslant \gamma$, and $\forall\ i, \sum AM(:,i) \leqslant dm_i$.

In above Eq. (13), Tp is a n-length vector in which the entry is each task t_i's processing time tp_i, that is $Tp = [tp_1, tp_2, \ldots, tp_n]$; and the symbol "$\bullet$" denotes dot product operation.

5.2 Single Objective Benchmark Algorithms

Previous studies investigate only one optimization objective, or a primary and a secondary objective to search for optimal allocation solutions. We next also employ this strategy to design algorithms that will be used as a baseline for comparison with other schemes. We designed two greedy-based strategies – *CoverFirst optimization* and *CostFirst Optimization*.

5.2.1 CoverFirst Optimization

The *CoverFirst Optimization* focuses on the *CpT* objective by selecting those workers with higher qualifications for each task by turns. In the process, the elements $qm_{j,i}, 1 \leqslant j \leqslant m$, in each column (corresponding to each task t_i) is sorted in descending order. The top-dm_i workers (the number of required workers for task t_i) will be chosen as the target ones to complete tasks. Note that if one selected worker has been assigned maximum tasks γ, he/she will be dropped and the next qualified worker will be chosen as a substitute one, until the requirement dm_i is satisfied. Moreover, in order to allocate as many tasks as possible, we give assignment priority to those tasks with a small number of qualified candidate workers. In essence, the *CoverFirst* algorithm is a deterministic approach, its time complexity of *CoverFirst* is $O(n * m * log_2 m)$, where n and m denote the number of tasks and workers, respectively.

5.2.2 CostFirst Optimization

The *CostFirst Optimization* focuses on the *DpT* objective by selecting workers who are qualified to more candidate tasks. Using matrix QM, this strategy essentially selects the rows with most nonzero entries. In the process, we iteratively calculate and update the rank of workers with respect to the number of tasks they are qualified for. In each iteration, the worker who has the most qualified tasks is selected as a winner, the selected worker and assigned tasks will be removed in QM. The process is continued until all the tasks are assigned or no qualified users are available for the remaining tasks. As a greedy-based approximate strategy, *CostFirst* optimization is constrained by the homogeneous distribution of qualification indicator QM, parameters γ and dm_i. In worst case, the time complexity of *CoverFirst strategy* is $O(\sum_{x=n-\sum dm_i+1}^{n} x * log_2 x)$.

5.3 Multi-Round Linear Weight Optimization Algorithm

The main idea of this linear weight algorithm is to integrate multi-objectives into a single comprehensive objective by

dot multiplying predefined preferences [16], [20]. Specifically, we construct an integrated utility function which concerns the optimization objectives of maximizing task coverage CpT and maximizing cost discount DpT. The utility function can be formulated as follows:

$$UF = \alpha * CpT + (1 - \alpha) * DpT, \tag{14}$$

where the role of weight factor α is to balance these two objectives, CpT and CpT. Considering the different dimensions of the objectives, we adopt a model in [21] to unify their scales as shown in Eq. (15), where $f(x)$ is the value of objective function, $f^{max}(x)$ and $f^{min}(x)$ are maximum and minimum values of objective $f(x)$, respectively.

$$f^*(x) = \frac{f(x) - f^{min}(x)}{f^{max}(x) - f^{min}(x)}. \tag{15}$$

Based on the feasible solution space (i.e., QM matrix), we propose a greedy allocation strategy which incrementally selects a worker to suitable tasks. In each iteration, the greedy-based algorithm computes all the available workers' utility by Eq. (14) from current existing tasks. As integrated the dual optimization objectives, for a specific worker w_j, the number of assigned spatial tasks is not always the more the better. For each volunteer worker w_j, based on his/her task-qualification distributions (i.e., the entries in $QM(j,:)$), it is necessary to examine all the possible allocation solutions (i.e., the maximum possible allocations can achieve $\sum_{x=1}^{\gamma} \frac{n!}{x!*(n-x)!}$), and choose one with maximum utility as the target sub-solution for w_j. After ranking all the obtained utilities of current available candidates, the worker who has maximal utility will be chosen as the winner in current iteration. In other words, there exists two maximal utility selection process: one is to select the maximum utility for w_j's all possible assignments, and the other is to select the maximum utility among all the candidate workers. After the winning worker is determined, the recorded matrices QM and AM will be updated accordingly. This process continues until *all tasks in pool have been assigned* or *the final chosen maximal utility in current iteration remains zero*.

Based on our greedy linear weight approach, we propose a multi-round linear weight optimization (*MRLWO*) algorithm to investigate more approximate Pareto-optimal solutions under different preference combinations. For example, we vary the weight factor α from 0 to 1 with 0.1 increment, and then it will produce 11 combinations of preferences , such as $(0.2, 0.8)$, $(0.6, 0.4)$, $(0.7, 0.3)$, etc. For each preference combination, we use the greedy linear weight strategy to achieve a corresponding allocation solution, finally 11 approximate allocation solutions will be derived. In worst case, the time complexity of *MRLWO* algorithm is $O(|\alpha| * \sum dm_i * m * \sum_{x=1}^{\gamma} \frac{n!}{x!*(n-x)!} * log_2 m * \sum_{x=1}^{\gamma} \frac{n!}{x!*(n-x)!})$, where the part of $O(m * \sum_{x=1}^{\gamma} \frac{n!}{x!*(n-x)!} * log_2 m * \sum_{x=1}^{\gamma} \frac{n!}{x!*(n-x)!})$ is corresponding to the aforementioned two maximal utility selection process, and $|\alpha|$ is determined by the selected number of preference combinations. For example, in the above mentioned scenario (i.e., α from 0 to 1 with 0.1 increment), the value of $|\alpha|$ is 11.

Note that the obtained solutions are not all Pareto-optimal solutions, since in each iteration, the greedy-based linear weight approach can achieve a near-optimal solution instead of an optimal solution. Moreover, the linear weight strategy can not guarantee to find Pareto solutions for multi-objective optimization problem, especially for non-convex solution space [33], [34]. Thus, it is needed to compare the final obtained allocation solutions, and pick up the Pareto-optimal solutions by dominance comparison. The pseudo code of *MRLWO* is shown in Algorithm 1, and its workflow is visualized in Fig. 4.

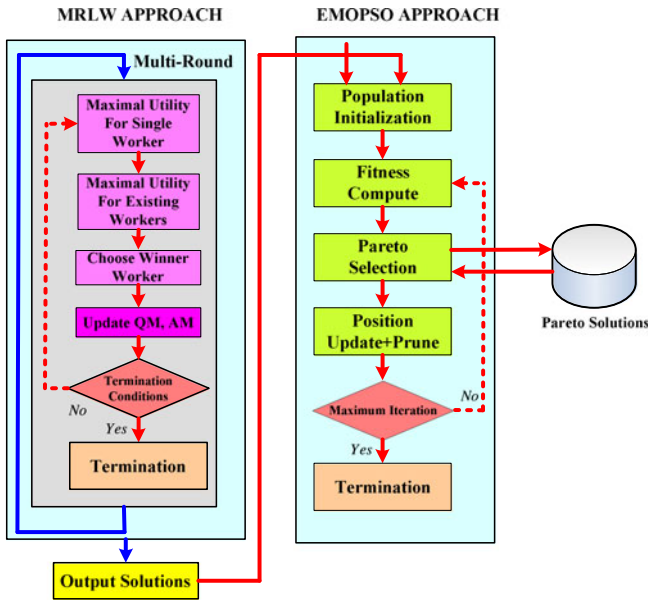


Fig. 4. The workflow of *MRLWO* and *EMOPSO* algorithm.

Algorithm 1. *MRLWO* Algorithm

Input: QM: Qualification matrix, γ: Maximum number of assigned tasks per worker.
Output: Task allocation matrix AM. initialization AM as a zero matrix;
for *each preference group* $(\alpha_k, (1 - \alpha_k))$ **do**
  repeat
    for *each row in* QM (*worker* w_j) **do**
      score UFs for w_j's possible assignments;
      choose the maximal UF of w_j;
    end
    select the worker with maximum UF as winner;
    check winner's γ constraint;
    update AM and remove QM's worker-task pairs;
  until *termination conditions satisfied;*
end

5.4 Enhanced Multi-Objective Particle Swam Optimization Algorithm

In essence, the traditional linear weight strategy cannot deal with non-convex optimization problem. Moreover, how to determine the best set of weight combinations to well match the unknown Pareto solution distribution remains a difficult problem. Moreover, with the increase of the scale of spatial tasks and workers, the solution space will expand exponentially, which is too large for traditional optimization strategies.

In order to explore the overall solution space, we turn to a heuristic method, particle swarm optimization [35], in

order to achieve as much as possible Pareto-optimal solutions for a task requestor to choose. The particle swarm optimization (PSO) is an evolutionary algorithm which iteratively tries to improve a candidate solution with regard to a given objective function. It drives a population of candidate solutions (i.e., particles) to move around in the search-space based on particle's position and velocity.

As the search space in our problem is huge (i.e., hundreds of concurrent tasks and thousands of available users), the initial population produced from feasible solution space QM randomly is very unlikely to be close to the target Pareto-solution frontier, and it is difficult for a traditional PSO to get satisfying convergence performance and final solutions. In order to effectively accelerate convergence and achieve satisfying solutions, we propose to employ the output solutions obtained by our *MRLWO* approach as part of initial population directly, and adjust the traditional PSO to our HSC-TA problem, namely Enhanced Multi-Objective Particle Swarm Optimization (*EMOPSO*). By this way, the final results of *EMOPSO* are at least as good as the *MRLWO* algorithm theoretically.

The workflow of the *EMOPSO* algorithm is illustrated in Fig. 4. In the first step, the population of *EMOPSO* is initialized in a random fashion according to the original search space QM. For each task t_i, workers who are qualified to perform it are chosen as the candidate ones. It then randomly selects a group of workers among the candidate ones while considering the given duplicate scale dm_i and maximum assigned task threshold γ. In addition to the randomly generated population, the results from the *MRLWO*algorithm are also included as the input of initialized population.

The fitness value (i.e., these two objectives CpT and DpT) of population is calculated by Eq. (13). The Pareto solutions are then chosen from existing population by comparing these two objective values. If one solution can not be dominated by any other solutions, it is regarded as a Pareto solution in current iteration, and its corresponding fitness will be stored in an external storage. In each iteration, the newly generated Pareto solution will also be compared with the ones recorded in external storage. The purpose is to ensure that there is no non-Pareto solutions in external storage after updating operation in each iteration.

Afterwards, each particle's position pos will be updated by the guidance of a stochastically selected Pareto solution pos_g and its local best known position pos_l. The updated position pos^* can be formulated as follows:

$$pos^* = pos + c_1 * vel + c_2 * R_1 * (pos_l - pos) + c_3 * R_2 * (pos_g - pos), \quad (16)$$

where c_1 denotes an inertia weight of particle's velocity vel (in the first iteration, vel is initialized to zero), c_2 and c_3 are two positive constants, and R_1, R_2 are random values in the range $[0, 1]$. Note that in the process of position update, the form of solution AM will be reshaped into vector format, in our case it can be represented as 1 row and $m * n$ columns.

Finally, as our required final solution are 0-1 integer variable, by the position update process, the new position may be a continuous variable which is not a desirable solution for our problem. Therefore, it is necessary to prune this kind of generated position into the form of 0-1 variable, and the resultant solution must also satisfy the constraints of $\sum AM(j,:) \leqslant \gamma$ and $\sum AM(:,i) \leqslant dm_i$. According to the mechanism formulated in Eq. (17), we translate the newly generated variable $am_{i,j}$ into 0-1 integer value. The basic idea is that if $am_{i,j}$ is close to 0 or 1, the probability that it is really 0 or 1 is largest, otherwise it would be opposite value. In addition, we also examine the constraints conditions, and modify the new solution accordingly.

$$am_{i,j} = \begin{cases} 1, & 0.5 < am_{i,j} \leqslant 1.5 \ or \ am_{i,j} < -0.5 \\ 0, & |am_{i,j}| \leqslant 0.5 \ or \ am_{i,j} > 1.5. \end{cases} \quad (17)$$

The procedure will continue until it reaches the maximum iteration. As a stochastic algorithm, the time efficiency is influenced by many factors, such as initialization, potential solution space, parameter settings, and so on. By analyzing the workfolw, the time complexity of our proposed *EMOPSO* algorithm is $O(it * (sz * (sz-1) + as * (as-1)) + 2 * sz)$, where it and sz denote the iteration times and population size, and as represents the average scale of Pareto-optimal solutions recorded in external storage at each iteration. The pseudo code of *EMOPSO* is shown in Algorithm 2, and its workflow is visualized in Fig. 4.

Algorithm 2. *EMOPSO* Algorithm

Input: sz: population size, it: iteration times, Mobility Pattern Set MP.
Output: A set of allocation matrices SAM.
$SAM = \varnothing$;
Initialize population in pos, pos_l and vel;
repeat
  population fitness calculation with two objectives;
  for *Each particle in population* **do**
    compare with any other and pick out Pareto ones;
    record discovered Pareto solutions into SAM;
    update local best position pos_l;
  end
  delete non-Pareto solutions in SAM;
  randomly choose pos_g from SAM;
  update population particle position;
  prune newly generated particile;
until *meet iteration times* it;

6 Performance Evaluation

We systematically evaluate the performance of our proposed techniques using a real-world data set and a synthesized data set. Our experiments and latency measurements are conducted on a standard server (Windows), with Intel (R) Xeon(R) CPU E5-2620, 2.40 GHz and 32 GB main memory.

6.1 Experimental Setup and Baselines

Data Set We utilize WTD (Wireless Topology Discovery) data published by researchers at University of California San Diego (UCSD) as the real-world data set. In campus environment, WTD records all access points (AP) sensed by users every 20 seconds. Each record consists of the following information: user id, sample time, AP id, RSS and AC/ battery power. Due to the sparse distribution in WTD data

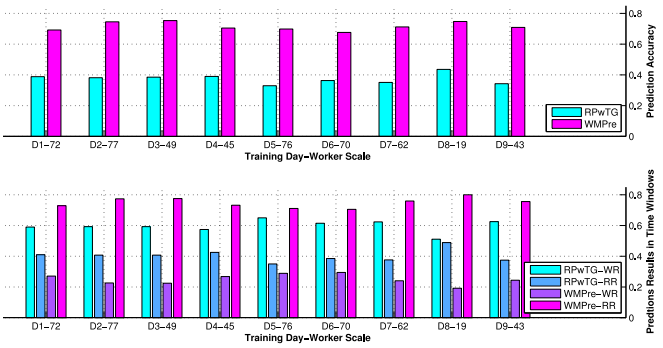


Fig. 5. Mobility prediction results comparison.

set, we filter out those APs and users having too few records, and obtain a final data set which contain 68 APs and 274 mobile users.

The real-world trace usually show an inhomogeneous and sparse distribution over spatio-temporal dimension. Thus, there may not be enough qualified workers to spatial tasks within certain spatial locations and temporal intervals. In other words, not all existing spatial tasks can be allocated to workers. In order to comprehensively verify our proposed approaches on a larger scale, we synthesize a set of simulation data set based on the real data.

Baseline Algorithms. To the best of our knowledge, there is no work focusing on the heterogeneous task allocation in SC paradigm with multi-objective optimization like ours. Therefore, we evaluate the performance of our proposed MRLWO algorithm, EMOPSO algorithm and the two baseline strategies—*CoverFirst* and *CostFirst*. In addition, we employ a joint optimization approach based on combinatorial fractional programming (*CFP*) in [17], in which it iteratively selects the best candidate from available workers, and the utility function is formulated as $\frac{Task\ Coverage}{Tast\ Cost}$. To coordinate the involved optimization objectives, we adapt the utility function of *CFP* strategy as $CpT * DpT$. Note that in our experiment, both proposed algorithm and the baselines are with $\delta = 1$ for the fair comparison.

6.2 Performance of Worker Mobility Prediction

During the mobility prediction model training, we employ a 30 days period sampling data set to learn the hidden mobility pattern. A subsequent 9 days sampling data is used as the testing data set. As each AP has a unique identification number, we can represent a worker' mobility with APs. Moreover, the location of heterogeneous spatial tasks is also related to AP's id. To verify the performance of our proposed worker mobility prediction approach (*WMPre* for short), we employ a moving route prediction algorithm in recent studies [12], [13], which directly connect consecutive hotspot regions with maximum probability by leveraging a transition graph model (*RPwTG* for short). The corresponding results are reported in Fig. 5, and the parameter of σ is set to 0.3. In Fig. 5, the x axis represent the testing day and corresponding workers. For instance, 'D1-72' means that there are 72 volunteer workers in the first testing day. As shown in the top figure of Fig. 5, our proposed worker mobility prediction approach yields a more accurate prediction accuracy than the baseline prediction approach with roughly 2 times.

More specifically, we analyze the distribution of predicted mobility over 48 time windows $\mathbb{TW}$. The results are reported in the bottom figure of Fig. 5, in which 'WR' and 'RR' denote the percentage of wrong and right predicted mobility in time windows. Obviously, for our *WMPre* prediction, the ratio of right predicted tws is higher than *RPwTG*'s; while the ratio of wrong predicted tws is smaller than *RPwTG*'s. The hidden reason lies in two aspects: 1) *RPwTG* only choose one location as hotspot in each tw; while our *WMPre* adaptively determine the hotspot locations in each tw according to mobility distribution and co-occurrence group partition. 2) Without considering the movement sequential relationship, *RPwTG* directly connects hotspots in consecutive time windows to produce a route prediction. However, some hotspots in the predicted route may never appear on one path in historical traces. In contrast, the *WMPre* algorithm incorporates moving transition sequence in the prediction process. Thus, the accurate predicted mobility derived from our *WMPre* algorithm can better adapt multi-constraints heterogeneous spatial tasks' requirements.

6.3 Performance of HSC Task Allocation

Results of Using Real Data. We initialize 100 heterogeneous spatial tasks $\mathbb{T}$ and distribute them around 68 different AP, and then generate a random integer dm_i (i.e., number of required workers for each task) varying from 2 to 4 for each spatial task. As a result, the final number of independent spatial tasks is 314 (i.e., $\sum_{i=1}^{100} dm_i$). The valid duration vt_i of each task spans consecutive time windows with random length, and the maximum length can not exceed 8 time windows (i.e., 4 hours). Each task's processing time pt_i randomly varies from 10 sec. to 600 sec. As the majority of workers are usually active in a few locations, for instance library, classroom, their corresponding mobility exhibits an intrinsically sparse distribution. For instance, after computing the qualification indicator QM based on the spatial tasks $\mathbb{T}$, we find that the ratio of zero entries in QM is up to 98.39 percent. Obviously, in this case, there are not enough existing workers to perform all the generated heterogeneous spatial tasks $\mathbb{T}$. Thus, a metric *Task Allocated Ratio* should be employed to evaluate these allocation strategies from another perspective. Several allocation experiments are conducted for spatial tasks $\mathbb{T}$, in which the maximum number of assigned tasks per worker (i.e., γ) is set to 3.

The results of allocation solution obtained by different optimization approaches are reported in Fig. 6. The number of obtained allocation solutions for multi-objective driven algorithms, such as *MRLWO* and *EMOPSO*, are 7 and 24 (i.e., non-replicative Pareto-optimal solutions); while single objective-based solving approaches, including *CoverFirst*, *CostFirst* and *CFP* algorithms, are all 1. Obviously, as multi-objective driven approaches strive to discover enough Pareto-optimal solutions. Yet *CoverFirst*, *CostFirst*, or *CFP* aims to achieve an extreme optimal solution. And among these three single objective-based solving approaches, the performance of *CFP*, *CoverFirst* and *CostFirst* are comparable and non-dominated. Moreover, the Pareto-optimal solutions of *EMOPSO* lie at the outermost of these two objectives' two-dimensional space in Fig. 6. Note that the objectives indices *DpT* and *CpT*, are calculated as the achieved discount or coverage divided by all the existing spatial tasks, not only the allocated tasks. Thus in

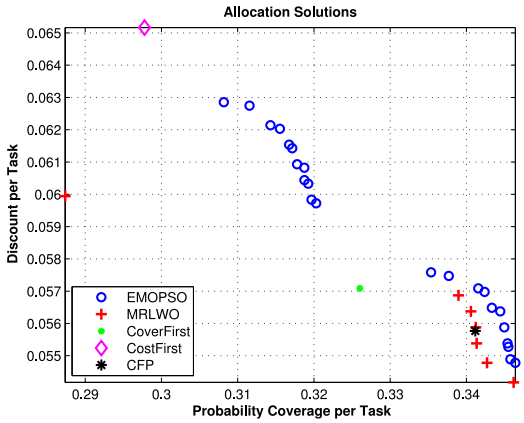


Fig. 6. The allocation solution results in real data set.

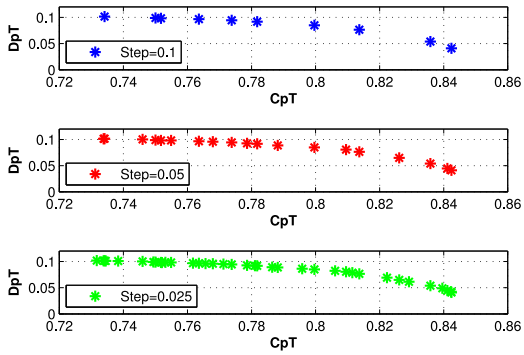


Fig. 7. Different preference combinations of *MRLWO*.

incompletely allocation situations, the corresponding *DpT* and *CpT* is relatively small, as the non-allocated spatial tasks' coverage and discount are all zero.

The *Task Allocated Ratio* of these obtained allocation solutions are also investigated. The corresponding values of *CoverFirst*, *CostFirst* and *CFP* algorithms are 0.584, 0.601 and 0.553, respectively. And the average value of *Task Allocated Ratio* for *MRLWO* and *EMOPSO* algorithms are 0.5548 and 0.6043, respectively. Obviously, the *Task Allocated Ratio* of *MRLWO* is highest among the rest approaches, and this advantage is hold in most cases.

Results of Using Synthetic Data. We comprehensively investigate the performance of our proposed *MRLWO*, *EMOPSO* approaches on a larger scale using synthetic data. Among the synthesized data sets, the average ratio of zero entries in qualified indicator matrix QM is 95.753 percent, the rest of entries follow a uniform distribution varying from 0 to 1.0. In larger-scale synthetic data sets, as there are enough qualified candidate workers for each released task, we will not concern the ratio of allocated tasks hereafter.

We first report the experimental results achieved by *MRLWO* algorithm. In the experiments, the number of independent tasks expanded by 125 original published heterogeneous spatial tasks is 369 (i.e., $\sum_{i=1}^{125} dm_i = 369$), and there are 700 available volunteer workers. In addition, the parameter of γ is set to 4. By traversing preference combinations with variable step being 0.1, *MRLWO* algorithm is conducted repeatedly for 11 rounds, and achieves corresponding 11 allocation solutions shown in Table 3. Among the obtained results, there exists 10 Pareto solutions which are labeled in last column. Obviously, by selecting different weight combinations, *MRLWO* can achieve Pareto-optimal allocation solutions focusing on one of these optimization objectives.

By leveraging determined variable step, *MRLWO* can exploit the potential solution space with corresponding granularity. We conduct experiments by varying different preference combinations with variable step being 0.1, 0.05 and 0.025. And the *MRLWO* algorithm would be conducted repeatedly with 11, 21 and 41 rounds. The corresponding experimental results are reported in Fig. 7, in which the achieved Pareto-optimal solutions under these three conditions are 10, 17 and 34, respectively. Note that, due to space limitation, the coordinates are abbreviated as DpT and CpT. Consistent with the running rounds, the obtained Pareto-optimal solutions also increase with the decrement of variable step. In practice, the number of Pareto-optimal results are usually less than the raw achieved solutions, as some non-Pareto solutions and redundant solutions exist. This is because the allocation solutions are non-uniformly distributed over feasible solution space. As a determined optimization strategy, *MRLWO* employ a fixed weight step setting, and the partitioned preference combinations can not match well with the unknown feasible solution distribution.

We next study the performance of *EMOPSO*. Because the potential feasible solution space is too large, for *EMOPSO* , the selection of initialization mechanisms has a significant impact on the convergence rate of particle population (i.e., allocation solutions). We also investigate this influence factor by devising three different initialization mechanisms: (1) initialized only by the outputs of *MRLWO*, (2) initialized by *MRLWO* and random population (a hybrid way), (3) initialized only by randomly generated population. In the experiments, there are 310 independent tasks, 600 available workers and γ equals 4. The parameters of *EMOPSO* are set as follows: population size of particle sz is set to be 700, iteration times it equals 40, and the number of Pareto-optimal solutions obtained by *MRLWO* is 22. The corresponding results are reported in Fig. 8. Among these three different initialization mechanisms, the convergence performance of the first one (i.e., only by *MRLWO*' outputs) is best, followed by the hybrid initialization and random initialization. The number of achieved allocation solutions from these three mechanisms is 126, 34 and 9, respectively. These results indicate that, by leveraging the outputs of *MRLWO*, the performance of *EMOPSO* can be significantly improved. This is because that the initialized population based on *MRLWO*'s results can directly lead to the Pareto frontier. Consider the

TABLE 3
The Allocation Solutions of *MRLWO* Approach

Preferences: $(\alpha, (1-\alpha))$	CpT	DpT	Pareto
(0.0,1.0)	0.5062	0.0735	**No**
(0.1,0.9)	0.7341	0.1015	Yes
(0.2,0.8)	0.7500	0.0988	Yes
(0.3,0.7)	0.7517	0.0981	Yes
(0.4,0.6)	0.7636	0.0965	Yes
(0.5,0.5)	0.7738	0.0943	Yes
(0.6,0.4)	0.7818	0.0916	Yes
(0.7,0.3)	0.7996	0.0848	Yes
(0.8,0.2)	0.8136	0.0764	Yes
(0.9,0.1)	0.8358	0.0538	Yes
(1.0,0.0)	0.8424	0.0409	Yes

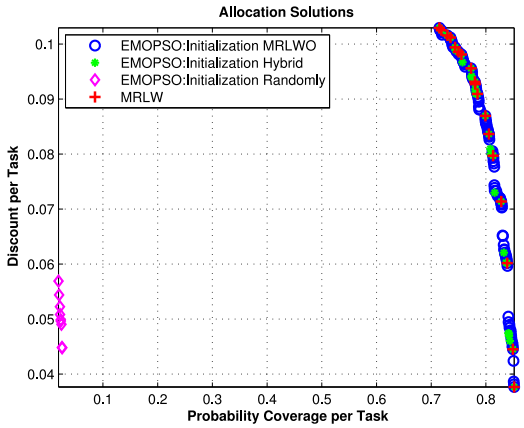


Fig. 8. Different initialization mechanisms of *EMOPSO*.

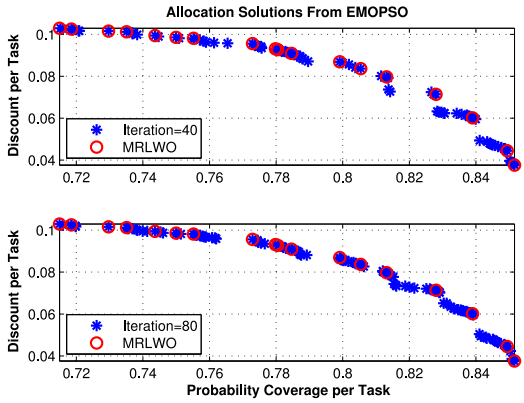


Fig. 10. Different iteration times of *EMOPSO*.

scale of original allocation results for *MRLWO* (i.e., 22), the discovered Pareto-optimal solutions expand to 126. This demonstrates that the *EMOPSO* algorithm has more solution space search and optimizing ability. Moreover, we can see that there exist clear performance gap between random initialization mechanism and other two mechanisms, under the same parameter settings. This is because the potential solution space is too huge for a brute force strategy to enumerate all possible solutions.

In addition, we study the dependency relationships between *MRLWO*'s outputs and *EMOPSO*'s results. Specifically, by varying the weight steps in *MRLWO*, we examine the final results achieved by *EMOPSO*. The corresponding experimental results are reported in Fig. 9, in which the variable steps are set to 0.25, 0.1, 0.05 and 0.025 respectively. The obtained Pareto-optimal allocation solutions are 89, 94, 97 and 127 respectively. It demonstrates that when the variable step is set to a small value, *EMOPSO* can usually obtain more final results. However, this dependency does not change linearly by increasing the variable step.

We also investigate the impact of iteration times on the performance of final obtained Pareto-optimal solutions, by varying the iteration times on 40 and 80, respectively. The settings of spatial tasks, available workers, maximum assigned tasks per worker are all the same as in the previous experiments. Fig. 10 shows the corresponding results, the achieved Pareto-optimal solutions are 91 and 136, respectively. The results demonstrate that, in most cases, the performance of *EMOPSO* can be improved with the increase of iteration times. This is because *EMOPSO* can access more feasible solution space when iteration time increases. However, as *EMOPSO* algorithm is not a deterministic method, the performance does not necessarily improve by increasing the iteration times.

In Fig. 11, we provide the allocation solutions generated by all the different strategies. The top figure in Fig. 11, shows that single-objective based algorithms, such as *CoverFirst* and *CostFirst*, can achieve extreme results which all rise up to the maximum values of *CpT* objective and *DpT* objective (i.e., the boundary of these two objectives). In particular, for *CostFirst*, although the obtained result's *DpT* objective is maximum, its corresponding *CpT* is worst among all achieved results. One possible reason for this is the Long Tail Theory. Each worker can be mainly qualified to limited spatial tasks constrained by his routine activity. In the process of allocation, *CostFirst* does not distinguish these two kind of spatial tasks. The bottom figure of Fig. 11 show all the achieved results, with the exception of *CostFirst*. We observe that the results derived by *EMOPSO* can cover most cases, while the results of *MRLWO* are somewhat sparse to represent all possible Pareto-optimal solutions.

Among the achieved solutions, we take two solutions (i.e., *CostFirst*'s result and *CFP*'s result in the top portion of Fig. 11) as an example to explain the given constraints optimization approach [17], [18], [19]. Specifically, with respect to the *DpT* objective, *CostFirst*'s solution is more than *CFP*'s only by 1.7 percent; while *CostFirst*'s *CpT* metric is less than

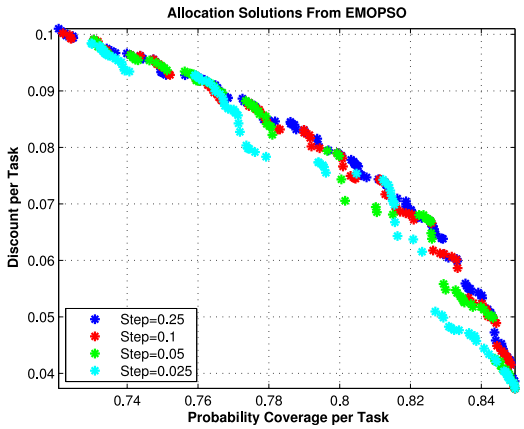


Fig. 9. The dependency relationships between *MRLWO* and *EMOPSO*.

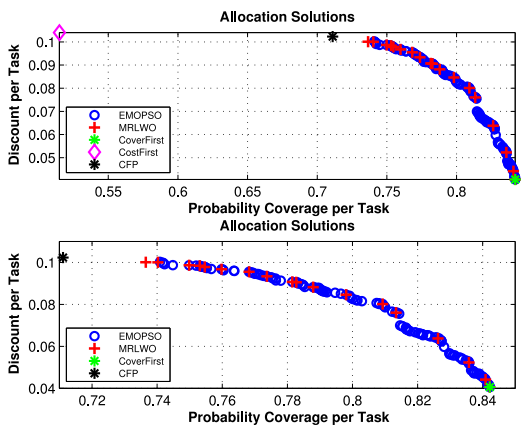


Fig. 11. The achieved pareto solutions by different approaches.

TABLE 4
Time Efficiency Comparison

	CoverFirst	*CostFirst*	*CFP*	*MRLWO*
Case:1	1 sec.	1 sec.	11 sec.	48 sec./round
Case:2	1 sec.	2 sec.	72 sec.	452 sec./round

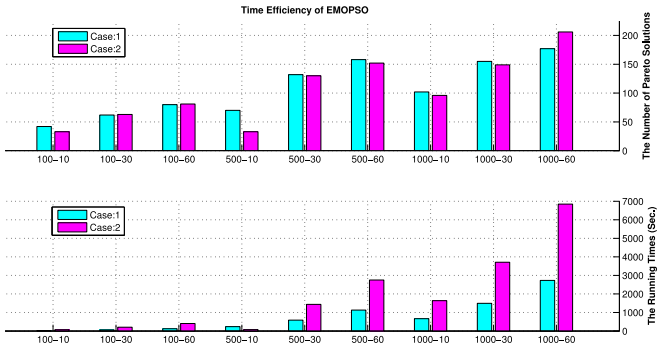


Fig. 12. The time efficiency of *EMOPSO*.

CFP's by roughly 38.09 percent. In other words, for *CostFirst*'s obtained allocation solution, if we would pay an extra small 1.7 percent incentive cost, we can improve the task coverage performance by nearly 38.09 percent. Obviously, it is cost-effective for a requestor to choose *CFP*'s allocation solution. For a given constraint strategy, when the requestor is required to predefine a bound with respect to one objective, such as *DpT*, it is impossible to determine a right boundary to best match the potential solution space. For example, if the requestor determines the bound of *DpT* as *CostFirst*'s measurement, a better solution (i.e., *CFP*'s result) will be missed.

Finally, we investigate the efficiency of proposed approaches. There are two different problem cases: case 1 with 700 volunteer workers and 360 independent spatial tasks, case 2 with 1,000 volunteer workers and 600 independent spatial tasks. The corresponding experimental results are listed in Table 4. Among these four involved algorithms, as the baseline methods (i.e., *CoverFirst* and *CostFirst*) generate deterministic allocation solutions, the corresponding time complexity is low. For example, the iterations of *CoverFirst* method is identical to the size of published spatial tasks. The time efficiency of *CFP* and *MRLWO* is greater and increases as the numbers of tasks and workers increase.

We also investigate the operation efficiency of *EMOPSO* with the above two problem cases, by varying the population size and iteration time as the numbers of tasks and workers increase. The operation efficiency is listed in Fig. 12, in which the x axis represent the population size and iteration times. For instance, '100-60' denotes that the corresponding population size is 100, and the iteration times equals to 60. As expected, the runtime of *EMOPSO* raises as the population size and iteration times increase. Meanwhile, the scale of obtained Pareto-optimal solutions also increases. This is simply because *EMOPSO*'s search capacity and optimization space will expand, thus more Pareto-optimal solutions can be achieved. Moreover, in lager-scale problem scenarios, the runtime will increase accordingly. For instance, the average overhead time to obtain one Pareto-optimal solution in case 1 and case 2 are 5.52 sec. and 13.02 sec. The hidden reason is that in larger-scale problem scenarios, the potential feasible solution space expands. And the corresponding time complexity is sufficient in real applications as the task allocation process is offline.

7 Conclusion

In this paper, we study the problem of the heterogeneous spatial crowdsourcing task allocation problem (HSC- TA), which assigns crowdsourcing tasks with multiple spatio-temporal constraints to mobile volunteer workers such that the quality of task coverage is maximized and the total incentive cost is minimized simultaneously. In order to align workers mobility with spatial task requirements, we devise a worker mobility prediction model to determine the feasible solution space of the HSC-TA problem. To solve the multi-objective optimization problem, we develop a greedy approximation approach (i.e., MRLWO) and a heuristic evolutionary approach (i.e., EMOPSO) that can efficiently retrieve the potential allocation solution feasible space, and achieve adequate Pareto-optimal solutions. Extensive experiments show the effectiveness of our proposed approaches on both real and synthetic data sets.

Acknowledgments

This research is partly partially supported by the National Basic Research Program of China (No. 2015CB352400), and the National Natural Science Foundation of China (No. 61402360, 61402369).

References

[1] A. Alfarrarjeh, T. Emrich, and C. Shahabi, "Scalable spatial crowdsourcing: A study of distributed algorithms," in *Proc. IEEE Int. Conf. Mobile Data Manag.*, 2015, pp. 134–144.
[2] Y. Zhao and Q. Han, "Spatial crowdsourcing: Current state and future directions," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 102–107, Jul. 2016.
[3] L. Kazemi and C. Shahabi, "GeoCrowd: Enabling query answering with spatial crowdsourcing," in *Proc. Int. Conf. Advances Geographic Inf. Syst.*, 2012, pp. 189–198.
[4] D. Deng, C. Shahabi, and U. Demiryurek, "Maximizing the number of worker's self-selected tasks in spatial crowdsourcing," in *Proc. ACM Sigspatial Int. Conf. Advances Geographic Inf. Syst.*, 2013, pp. 324–333.
[5] P. Cheng, et al., "Reliable diversity-based spatial crowdsourcing by moving workers," *Proc. VLDB Endowment*, vol. 8, no. 10, pp. 1022–1033, 2015.
[6] Picasa. (2015). [Online]. Available: http://picasa.google.com
[7] Gigwalk. (2017). [Online]. Available: http://www.gigwalk.com/
[8] H. Shin, et al., "Cosmic: Designing a mobile crowd-sourced collaborative application to find a missing child in situ," in *Proc. 16th Int. Conf. Human-Computer Interaction Mobile Devices Services*, 2014, pp. 389–398.
[9] E. Kamar, S. Hacker, and E. Horvitz, "Combining human and machine intelligence in large-scale crowdsourcing," in *Proc. 11th Int. Conf. Auton. Agents Multiagent Syst.*, 2012, pp. 467–474.
[10] J. Thebault-Spieker, L. G. Terveen, and B. Hecht, "Avoiding the south side and the suburbs: The geography of mobile crowdsourcing markets," in *Proc. 18th ACM Conf. Comput. Supported Cooperative Work Social Comput.*, 2015, pp. 265–275.
[11] Y. Chon, N. D. Lane, Y. Kim, F. Zhao, and H. Cha, "Understanding the coverage and scalability of place-centric crowdsensing," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2013, pp. 3–12.
[12] T. Kandappu, et al., "Tasker: Behavioral insights via campus-based experimental mobile crowd-sourcing," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 392–402.
[13] T. Kandappu, et al., "Campus-scale mobile crowd-tasking: Deployment and behavioral insights," in *Proc. 19th ACM Conf. Comput.-Supported Cooperative Work Social Comput.*, 2016, pp. 800–812.

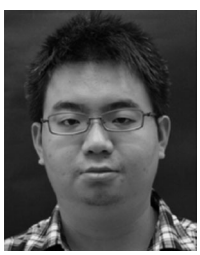
[14] L. Wang, et al., "Mobile crowd sensing task optimal allocation: A mobility pattern matching perspective," *Frontiers Comput. Sci.*, doi: 10.1007/s11704-017-7024-6.
[15] B. Guo, Y. Liu, W. Wu, Z. Yu, and Q. Han, "ActiveCrowd: A framework for optimized multitask allocation in mobile crowdsensing systems," *IEEE Trans. Human-Mach. Syst.*, vol. 47, no. 3, pp. 392–403, Jun. 2017.
[16] K. Deb, S. Karthik, and H. Jussi, "Multi-objective optimization," in *Decision Sciences: Theory and Practice*. Boca Raton, FL, USA: CRC Press, 2016. 145–184.
[17] U. ul Hassan and E. Curry, "Efficient task assignment for spatial crowdsourcing: A combinatorial fractional optimization approach with semi-bandit learning," *Expert Syst. Appl.*, vol. 58, pp. 36–56, 2016.
[18] H. To, L. Fan, L. Tran, and C. Shahabi, "Real-time task assignment in hyperlocal spatial crowdsourcing under budget constraints," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, 2016, pp. 1–8.
[19] J. Wang, et al." Fine-grained multitask allocation for participatory sensing with a shared budget," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1395–1405, Dec. 2016.
[20] R. B. Messaoud, et al., "Fair QoI and energy-aware task allocation in participatory sensing," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2016, pp. 1–6.
[21] Y. Liu, B. Guo, Y. Wang, W. Wu, Z. Yu, and D. Zhang, "Taskme: Multi-task allocation in mobile crowd sensing," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 403–414.
[22] Y. Liu and X Li, "Heterogeneous participant recruitment for comprehensive vehicle sensing," *PloS ONE*, vol. 10, no. 9, 2015, Art. no. e0138898.
[23] S. Reddy, D. Estrin, and M. Srivastava, "Recruitment framework for participatory sensing data collections," in *Proc. Int. Conf. Pervasive Comput.*, 2010, pp. 138–155.
[24] F. Alt, A. S. Shirazi, A. Schmidt, U. Kramer, and Z. Nawaz, "Location-based crowdsourcing: Extending crowdsourcing to the real world," in *Proc. 6th Nordic Conf. Human-Comput. Interaction: Extending Boundaries*, 2010, pp. 13–22.
[25] W. Liang, Z. Yu, B. Guo, T. Ku, and F. Yi, "Moving destination prediction using sparse dataset: A mobility gradient descent approach, *ACM Trans. Knowl. Discovery Data*, vol. 11, no. 3, 2017, Art. 37.
[26] P. Cheng, X. Lian, L. Chen, J. Han, and J. Zhao, "Task assignment on multi-skill oriented spatial crowdsourcing," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2201–2215, Aug. 2016.
[27] 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.
[28] S. Hachem, A. Pathak, and V. Issarny, "Probabilistic registration for large-scale mobile participatory sensing," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, 2013, pp. 132–140.
[29] C. Miao, H. Yu, Z. Shen, and C. Leung, "Balancing quality and budget considerations in mobile crowdsourcing," *Decision Support Syst.*, vol. 90, pp. 56–64, 2016.
[30] H. Xiong, D. Zhang, G. Chen, L. Wang, V. Gauthier, and L. E. Barnes, "iCrowd: Near-optimal task allocation for piggyback crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 15, no. 8, pp. 2010–2022, Aug. 2016.
[31] M. Musthag and D. Ganesan, "Labor dynamics in a mobile micro-task market," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2013, pp. 641–650.
[32] S. Martello and P. Toth, "Algorithms for knapsack problems," *North-Holland Math. Stud.*, vol. 132, pp. 213–257, 1987.
[33] I. Das and J. E. Dennis, "A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems," *Structural Multidisciplinary Optimization*, vol. 14, no. 1, pp. 63–69, 1997.
[34] E. M. Kasprzak and K. E. Lewis, "Pareto analysis in multiobjective optimization using the collinearity theorem and scaling method," *Structural Multidisciplinary Optimization*, vol. 22, no. 3, pp. 208–218, 2001.
[35] J. Kennedy, "Particle swarm optimization," *Encyclopedia of Machine Learning*. New York, NY, USA: Springer, 2011, pp. 760–766.

Liang Wang received the PhD degree in computer science from the Shenyang Institute of Automation (SIA), Chinese Academy of Sciences, Shenyang, China, in 2014. He is currently a postdoctoral researcher with Northwestern Polytechnical University, and an associate professor with the Xi'an University of Science and Technology, Xi'an, China. His research interests include ubiquitous computing, mobile crowd sensing, and data mining.

Zhiwen Yu received the PhD degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2006. He is currently a professor and the vice-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. He is a senior member of the IEEE.

Qi Han received the PhD degree in computer science from the University of California, Irvine, CA, in August 2005. She is an associate professor of computer science in the Colorado School of Mines, Golden, CO. Her research interests include mobile crowd sensing and ubiquitous computing. He is a member of the IEEE.

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



Haoyi Xiong received the PhD degree in computer science from Institute Mines-Téléecom, TELECOM SudParis, and Université Pierre et Marie Curie, France. He is currently an assistant professor in the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO. His research interests include mobile crowdsensing and ubiquitous computing. He received the Best Paper Award from IEEE UIC 2012, and is currently a member of the IEEE.