Multi-campaign Oriented Spatial Crowdsourcing

Libin Zheng#, Lei Chen#
Hong Kong University of Science and Technology
Hong Kong, China
{lzhengab, leichen}@cse.ust.hk

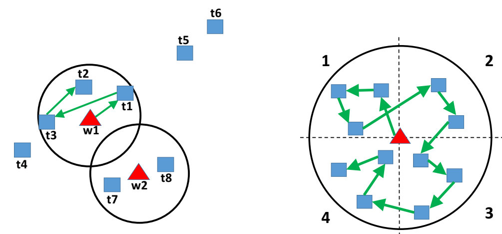
***Abstract*—The system throughput and workers' travel distance are two important factors in spatial crowdsourcing, and improving one of them usually means sacrificing the other. However, existing works either fail to consider the trade-off between these two factors or resolve their conflicts by simply targeting tasks within a bounding circle for each worker. In this paper, we compromise between the throughput and the distance by formulating these two factors as score terms in the objective function. Apart from that, we study the multi-campaign scenario in our problem, which is not uncommon in practical applications while not yet discussed in existing works. The worker diversity of the campaigns is formulated as another score term in the objective function. The problem of multi-campaign oriented spatial crowdsourcing is to maximize the aforementioned score function. We prove the problem is NP-hard and provide several approximation solutions. Extensive experiments have been conducted to validate the devised solutions.**

I. Introduction

The rapid development of mobile networks and devices starts a new age of crowdsouricng, spatial crowdsourcing (SC), by enabling geography-related tasks. Task assignment in spatial crowdsourcing has been widely studied in existing research works [1], [2], [3], [4].

Most existing works of MCS, i.e., mobile crowd sensing, focus on a single campaign for collecting specific data [3], [4]. The scenario of multiple campaigns has not yet been discussed in these works, while it is not uncommon in spatial crowdsourcing platforms (e.g., gMission). As shown in Example 1, a campaign is a task set for a particular purpose. Tasks inside a campaign correspond to its places of interest. Different users of the SC-application may issue multiple campaigns to the server, typically targeting at various data collections. For this multi-campaign scenario, the single-campaign oriented solutions cannot be transferred, because they do not resolve inter-campaign conflicts like a single worker is preferred by multiple campaigns. On the other hand, existing works in spatial crowdsourcing [1], [2] target at matching unit tasks and workers. Their solutions do not address assigning workers to campaigns. A campaign is a set of unit tasks for a particular purpose. In Example 1, each targeted spot corresponds to one unit task in the campaign.

Example 1 (Advertisement evaluation campaign)**.** *A company releases some advertising posters in a number of spots of the city to promote its products. In order to obtain the feedbacks, the company issues an advertisement evaluation campaign, which asks the workers to visit those spots and give their comments on the advertising effects.*



(a) Trade-off between throughput and distance (b) Clockwise region-by-region movement

Fig. 1.

System throughput (number of assigned tasks) and workers' travel distance are two widely concerned factors in spatial crowdsourcing [1]. However, improving the system throughput usually results in larger travel distances for the workers. In some situations where workers are far away from the tasks, blindly pursuing a large throughput would make the workers travel to some remote places. It may be better to have these remote tasks wait for the appearance of nearby workers. Therefore, compromising between these two factors is an important issue. Existing works either fail to consider both factors [2] or simply set bounding circles for workers [1].

Example 2 (Trade-off between throughput and distance)**.** *Figure 1(a) gives an example of eight tasks and two workers. GeoCrowd [1] sets a bounding circle for each worker, which recognizes their nearby tasks. This hard constraint successfully filters out remote tasks t_5 and t_6, but it misses the reachable task t_4. With route $t_1t_2t_3t_4$ for w_1, t_4 can be visited with a short distance. Besides, the method in [1] is route-blind, so there is no penalty for making detours like $t_1t_3t_2$.*

To better compromise between throughput and distance, we formulate them as two score terms in the objective function. On one hand, this formulation helps to filter out distant tasks while keeping reachable tasks visited. On the other hand, it guides to devise travel routes such that workers' travel distance is as small as possible.

In multi-campaign spatial crowdsourcing, we need to additionally consider the pairwise utility among workers assigned to the same campaign. Workers in the same campaign form a group. The utility sum of worker pairs represents the group's pairwise utility. In terms of workers' profile dissimilarity, the pairwise utility of a campaign represents its worker diversity. Worker diversity can be desired by the campaigns. For instance, in Example 1, a diverse worker set enables the company to learn from different consumer classes, and thus offers better support to its business strategy. Without loss of generality, in this paper, we use worker diversity to represent campaigns' pairwise utility. It is clear that there may be other pairwise utility demands, such as social connectivity and profile similarity. They can be applied to the problem using the same expression in the objective function.

In this paper, we formulate the problem of multi-campaign oriented spatial crowdsourcing, considering three common factors: system throughput [1], travel distance [1], [6] and worker diversity [2], [5]. To the best of our knowledge, this is the first time that the multi-campaign scenario has been focused. With an objective function combining the aforementioned three factors, the formulated optimization problem guides its solutions I. to be diversity-aware, grouping diverse workers for campaigns; II. to reach a compromise of throughput and distance flexibly. For the diversity concerned here, we consider the inherent differences among the participating workers. This is rather different from [2] which manages to maximize the expected spatiotemopral diversity of the collected answers. For the trade-off between throughput and distance, the objective function itself is a soft form to provide the guidance. It guides the solutions to not only give up those remote tasks (e.g., t_5 and t_6 in Figure 1(a)), but also devise travel routes to minimize the overall travel distance for reachable tasks (e.g., t_1, t_2, t_3 and t_4 for w_1 in Figure 1(a)).

We show the NP-hardness of the formulated problem by reductions from the *open vehicle routing problem*. Thus, we propose several heuristics. To summarize, we have made the following contributions. We formulate the problem of Multi-Campaign oriented Spatial Crowdsourcing (MCSC) with worker diversity and throughput-and-distance trade-off as objectives, and prove its hardness in Section II. We devise heuristics for worker-to-campaign assignment and worker route planning in Section III and IV respectively. We conduct extensive experiments and analyze the performance of the proposed algorithms in Section V.

In addition to the contributions listed above, we conclude the paper in Section VI.

II. Multi-campaign oriented spatial crowdsouricng

A. Problem formulation

Following most existing works [1], [2], we adopt the setting that the SC-server functions at a sequence of time stamps. In the following, we define the input variables of a time stamp.

Definition 1 (Campaign)**.** *A campaign C_i is a set of tasks $t_{i,j}$ requiring a particular action, such as data collection. Each task $t_{i,j}$ corresponds to conducting the action at the location of interest $l_{i,j}$.*

Definition 2 (Worker)**.** *An online worker w_k reports his/her current location l_k, and capacity z_k which represents the maximum number of tasks he/she would like to receive. For any pair of workers w_{k1} and w_{k2}, there is a dissimilarity measure $s_{k1,k2}$.*

The input contains a set of available campaigns $C\{C_i\}$, including those newly posted and previous uncompleted ones. Each campaign $C_i\{t_{i,j}\}$ consists of a set of pending tasks $t_{i,j}$'s, which have not been assigned in previous time stamps. The input also contains a set of available workers $W\{w_k\}$, including newly online workers and those who completed their previous assigned tasks. Finally, for any pair of objects, there is a distance measure $d_{(\cdot,\cdot)}$.

The output of a time stamp is the planned routes for the workers: $A\{a_k\}$ (we refer to $A\{a_k\}$ as a travel scheme or assignment interchangeably in the rest of this paper). Different from most existing works [1], [2], [6] which simply match workers with tasks, the SC-server in MCSC also plans routes for workers. That is, for each w_k, a_k is a travel route $l_0 l_1 ... l_f$, where l_0 and $l_1, ..., l_f$ are the locations of w_k and his/her assigned tasks $t_{k,1}, ..., t_{k,f}$ respectively.

Three commonly concerned factors in spatial crowdsourcing: system throughput [1], travel distance [1], [6] and worker diversity [2], [5], are formulated as objectives in MCSC.

System throughput. The number of assigned tasks measures the throughput of the system. For a travel scheme A, we denote its number of assigned tasks as N.

Travel distance. For a travel scheme A, we denote its overall travel distance as D, i.e., $D = \sum_{a_k \in A} d_{a_k}$, where d_{a_k} represents the travel distance of route a_k.

Worker diversity. We measure a group's pairwise utility with $\sum_{(w_{k1},w_{k2}) \in W_{C_i}} s_{k1,k2}$. In terms of diversity, $s_{k1,k2}$ represents the profile dissimilarity between the two workers. Subsequently the overall worker diversity of an assignment A is:

$$S = \sum_{C_i \in C} \sum_{(w_{k1},w_{k2}) \in W_{C_i}} s_{k1,k2}$$

where W_{C_i} is the assigned workers of campaign C_i.

To compromise between throughput and distance, we make N and D positive and negative score terms respectively in the objective function $f(A) = N - \lambda_D D$, where λ_D is a regularization term. The reciprocal of λ_D, $\frac{1}{\lambda_D}$, represents the maximum moving distance per task. Taking a worker-task pair w_k-t_j as an example, we assume their distance at a moment is d and t_j is unvisited. Then visiting t_j at that moment brings the gain: $\Delta(N - \lambda_D D) = 1 - \lambda_D * d$. This gain would be negative if d is larger than $\frac{1}{\lambda_D}$. Therefore, task t can be visited by w_k if $d < \frac{1}{\lambda_D}$. For instance, in Figure 1(a), after w_1 travels to t_3, t_4 would be available assuming $d_{l_{t3},l_{t4}} < \frac{1}{\lambda_D}$. In this case, the optimal route of w_1 is $l_0 \rightarrow l_{t_1} \rightarrow l_{t_2} \rightarrow l_{t_3} \rightarrow l_{t_4}$. In contrast, tasks which require a moving distance larger than $\frac{1}{\lambda_D}$, such as t_5 and t_6 in Figure 1(a), should not be assigned in current time stamp. They should wait for the appearance of nearby workers.

We further add S to the objective function, multiplied by a regularization term λ_S:

$$\begin{aligned} f(A) &= N - \lambda_D D + \lambda_S S \\ &= \sum_{C_i} N_i - \lambda_D D_i + \lambda_S \sum_{C_i} \sum_{w_{k1},w_{k2} \in C_i} s_{w_{k1},w_{k2}} \end{aligned} \quad (1)$$

For setting proper normalization values of λ_S and λ_D, please refer to our technical report [7]. Now we formally define the MCSC problem.

Definition 3 (Multi-campaign oriented spatial crowdsourcing (MCSC))**.** *Given a campaign set $C\{C_i\}$, a worker set $W\{w_k\}$, dissimilarity measures of workers $s_{(\cdot,\cdot)}$'s and travel distances $d_{(\cdot,\cdot)}$'s, the problem of task assignment and route management in multi-campaign oriented spatial crowdsourcing is to find the scheme A^* such that the objective function* (1) *is maximized, with the following constraints:*
I. The number of assigned workers of a campaign cannot be more than its available tasks, i.e., $\forall C_i,\ |W_{C_i}| \le |C_i|$.
II. The number of assigned tasks of a worker cannot exceed its capacity, i.e., $\forall w_k,\ |a_k| \le |z_k|$. ($|a_k|$ represents the number of visited tasks in route a_k).
III. A worker is assigned to at most one campaign.

Having constraint I is because at most $|C_i|$ workers are needed to perform a campaign's tasks. Subsequently larger campaigns are allowed to have a larger number of workers. Constraint III means that workers attend campaigns one by one. They need to finish the current campaign before starting a new campaign. This simulates that workers in Amazon's Mechanical Turk[1] (AMT) perform tasks batch by batch (same for gMission), where a batch is a task set of the same topic. It is not difficult to understand the benefits of adopting this mode in the multi-campaign scenario. A campaign naturally defines a type of tasks. Workers obtain proficiency by focusing on one campaign at a time.

A solution of the MCSC problem includes assigning workers to campaigns and planning their routes. Taking Figure 1(a) as an example where we assume w_1 and w_2 are assigned to the campaign with tasks $t_{1\sim 8}$. A fair planning is to have $\rightarrow l_{t_1} \rightarrow l_{t_2} \rightarrow l_{t_3} \rightarrow l_{t_4}$ and $\rightarrow l_{t_7} \rightarrow l_{t_8}$ for w_1 and w_2 respectively.

We prove the hardness of the problem in [7]. Thus, we propose several heuristics. A solution of MCSC consists of worker-to-campaign assignment and route planning of workers, which are addressed in Section III and IV respectively.

III. WORKER-TO-CAMPAIGN ASSIGNMENTS

In this section, we discuss assigning workers to campaigns with two different heuristics.

A. Diversity based assignment

The heuristic described here, Diversity Based Assignment (DBA), aims to separate the workers into $|C|$ groups such that S is maximized. This corresponds to the *minimum k-cut problem* [8], which is NP-hard. For efficiency concern, we propose the Greedy grouping method for DBA, which consists of two phases.

Initialization. It randomly samples $|C|$ workers from W, each of which represents a group G_h $(1 \le h \le |C|)$.

[1]https://www.mturk.com/mturk/welcome

Greedy assignment. It randomly permutates the remaining workers and assigns them one by one. For each worker, it calculates his/her overall dissimilarity to the groups:

$$s_{w,G_h} = \sum_{w' \in G_h} s_{w,w'}$$

and assigns the worker to the group with maximum dissimilarity.

Theorem 1. *The Greedy grouping of DBA achieves an approximation factor of $\frac{1}{|C|}\left(1 - \frac{|C|(|C|-1)}{|W|(|W|-1)}\right)$ for maximization of S.*

Proof: Please refer to our technical report [7]. ∎

The output of the Greedy grouping is $|C|$ groups, which are matched with the campaigns via the Hungarian algorithm [9].

B. Hybrid-benefit based assignment

The Hybrid-benefit based assignment, HBA, assigns workers one by one, jointly considering the gain of $N - \lambda_D D$ and $\lambda_S S$. In each step, it assigns the worker w_{k^*} to the campaign C_{i^*} such that this pair brings the maximum immediate gain among remaining w_k-C_i pairs.

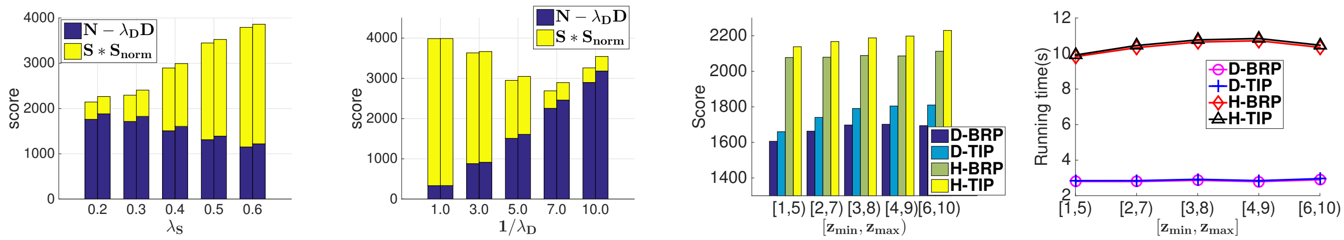
The gain of assigning w_k to C_i, i.e., $\Delta f(W_{C_i} \cup \{w_k\}) = f(W_{C_i} \cup \{w_k\}) - f(W_{C_i})$, contains two components, $\Delta\lambda_S S_i$ and $\Delta(N_i - \lambda_D D_i)$. The diversity gain $\Delta\lambda_S S_i$ can be directly calculated as:

$$\Delta\lambda_S S_i = \lambda_S \sum_{w_{k'} \in W_{C_i}} s_{w_k, w_{k'}}$$

For $\Delta(N_i - \lambda_D D_i)$, it is affected by how we plan routes for W_{C_i}& w_k to perform tasks of C_i. Exactly calculating $\Delta(N_i - \lambda_D D_i)$ would lead to unbearable time costs. We propose a faster technique, which costs linear amount of time, to roughly estimate $\Delta(N_i - \lambda_D D_i)$. As a result, the overall time complexity of HBA is $O\left((|C| + |\widehat{C_i}|)|W|^2 + |C||W||\widehat{C_i}|\right)$.

We assume Euclidean distance between tasks and workers. We only consider tasks inside the circle $\mathcal{C}(l_k, r = \frac{1}{2\lambda_D})$ for w_k, which ensures positive gain from individual tasks. We want to plan a route for w_i to visit the tasks in the circle, and the value of $N - \lambda_D D$ of this route would be the estimation result of $\Delta(N_i - \lambda_D D_i)$. We plan the route by randomly choosing the next task, which only needs a random permutation operation and costs linear amount of time. We refine the route by moving region by region, as shown in Figure 1(b). By planning the route region by region, we bound the distance of any two adjacent points with $\sqrt{2}r$ (except the inter-region travel distance), which is smaller than that of planing the route in the whole circle: $2r$.

It is obvious that a larger radius, r, can accommodate more tasks and thus better estimate $\Delta(N_i - \lambda_D D_i)$. Meanwhile, we need to guarantee that visiting tasks in this circle would not bring negative gain. We show in our technical report [7] that the radius can be enlarged to $\frac{1}{\frac{7+\sqrt{2}}{5}\lambda_D}$ while ensuring positive gains.



(a) Varying λ_S: H-BRP (left bar) and H-TIP (right bar)

(b) Varying $\frac{1}{\lambda_D}$: H-BRP (left bar) and H-TIP (right bar)

(c) Varying $[z_{min}, z_{max})$: Score

(d) Varying $[z_{min}, z_{max})$: Time cost

Fig. 2. Effect of λ_S, λ_D and $[z_{min}, z_{max})$.

IV. Route planning.

We leave the first heuristic (named BRP) for route planning in our technical report [7] as it is relatively simple. The second heuristic, Task-Insertion Planning (TIP), maintains the intermediate devised routes of workers, and inserts the tasks one by one. Given the tasks $\{t_j\}$ and the assigned workers $\{w_k\}$ of campaign C_i, TIP keeps searching the least costly insertion pair t_j-$a_{k,loc}$ and performing the insertion. $a_{k,loc}$ refers to the insertion after location loc in the route a_k. TIP uses a binary search tree to maintain the costs and rankings of $a_{k,loc}$'s for each t_j. After each insertion, it updates the trees of the un-assigned tasks. The time cost of TIP can be further optimized with a lazy update for the trees. Please refer to our technical report [7] for more details. TIP has an average-case time complexity of $O\left(|C_i|^2 \log |C_i|\right)$.

V. Experiments

In this section, we conduct experimental comparisons among D-BRP, D-TIP, H-BRP and H-TIP. Note that D-BRP (H-BRP) refer to the combination of DBA (HBA) and BRP. The experimental settings are summarized in [7].

We vary the two regularization terms to examine their effects on $N - \lambda_D D$ and $S * S_{norm}$. Since DBA is not affected by neither λ_D nor λ_S, we show only the results of HBA-based methods, i.e., H-BRP and H-TIP. Figure 2(a) and 2(b) show the score changes with the blue and yellow area of a bar representing the values of $N - \lambda_D D$ and $S * S_{norm}$ respectively. For each bar group, the left and right bars refer to the results of H-BRP and H-TIP respectively. It can be found that for both methods, $S * S_{norm}$ increases while $N - \lambda_D D$ decreases when λ_S becomes larger. The reason is straightforward: with a larger λ_S, HBA is guided to achieve a larger S. Meanwhile, some tasks are sacrificed, resulting in a lower $N - \lambda_D D$. Likewise, when $\frac{1}{\lambda_D}$ increases from 1 km to 10 km, more tasks can be visited. Therefore, $N - \lambda_D D$ expands and $S * S_{norm}$ shrinks gradually.

The experimental results of varying workers' capacity range are shown in Figure 2(c) and 2(d). When workers are given a higher capacity range, all methods obtain larger scores. Besides, both H-TIP and H-BRP obtain a large score improvement from D-TIP and D-BRP, which indicates HBA is more effective than DBA in worker assignment. It can also be found that under same assignment method, TIP achieves a higher score than BRP. This suggests that task-insertion based planning is more effective.

The running times are around 10 seconds and 3 seconds for HBA-based and DBA-based methods respectively. HBA is slightly more costly than DBA.

VI. Conclusion

In this paper, we formulate the problem of multi-campaign oriented spatial crowdsourcing. To the best of our knowledge, this is the first work which generalizes tasks to campaigns. We formulate an objective function considering three factors, i.e., worker diversity, system throughput and workers' travel distance. We propose methods to assign workers to campaigns and plan routes for them. Experiments have been conducted to evaluate the performance of the methods.

VII. Acknowledgment

The work is partially supported by the Hong Kong RGC GRF Project 16207617, National Grand Fundamental Research 973 Program of China under Grant 2014CB340303, the National Science Foundation of China (NSFC) under Grant No. 61729201, Science and Technology Planning Project of Guangdong Province, China, No. 2015B010110006, Webank Collaboration Research Project, and Microsoft Research Asia Collaborative Research Grant.

References

[1] L. Kazemi and C. Shahabi, "Geocrowd: enabling query answering with spatial crowdsourcing," in *Proc. of the SIGSPATIAL GIS*. ACM, 2012, pp. 189–198.

[2] P. Cheng, X. Lian, Z. Chen, R. Fu, L. Chen, J. Han, and J. Zhao, "Reliable diversity-based spatial crowdsourcing by moving workers," *Proc. of the VLDB*, vol. 8, no. 10, pp. 1022–1033, 2015.

[3] D. Zhang, H. Xiong, L. Wang, and G. Chen, "Crowdrecruiter: selecting participants for piggyback crowdsensing under probabilistic coverage constraint," in *Proc. of the UbiComp*. ACM, 2014, pp. 703–714.

[4] W. Guo and S. Wang, "Mobile crowd-sensing wireless activity with measured interference power," *IEEE Wireless Communications Letters*, vol. 2, no. 5, pp. 539–542, 2013.

[5] T. Wu, L. Chen, P. Hui, C. J. Zhang, and W. Li, "Hear the whole story: Towards the diversity of opinion in crowdsourcing markets," *Proc. of the VLDB*, vol. 8, no. 5, pp. 485–496, 2015.

[6] P. Cheng, X. Lian, L. Chen, and C. Shahabi, "Prediction-based task assignment in spatial crowdsourcing," in *Proc. of the ICDE*. IEEE, 2017, pp. 997–1008.

[7] L. Zheng and L. Chen, "Multi-campaign oriented spatial crowdsourcing," http://www.cse.ust.hk/%7Elzhengab/MCSC.pdf.

[8] H. Saran and V. V. Vazirani, "Finding k cuts within twice the optimal," *SIAM Journal on Computing*, vol. 24, no. 1, pp. 101–108, 1995.

[9] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.