SCIENCE CHINA
Information Sciences

• LETTER •

August 2021, Vol. 64 189101:1–189101:2
https://doi.org/10.1007/s11432-019-3061-x

Real-time bottleneck matching in spatial crowdsourcing

Long LI[1], Lingling WANG[2*] & Weifeng LV[1]

[1]*State Key Laboratory of Software Development Environment (SKLSDE Lab) and Beijing Advanced Innovation Center for Big Data and Brain Computing (BDBC), Beihang University, Beijing* 100191*, China;*
[2]*School of Management and Economics, Beijing Institute of Technology, Beijing* 100081*, China*

Received 27 December 2019/Revised 18 March 2020/Accepted 21 July 2020/Published online 21 May 2021

Citation Li L, Wang L L, Lv W F. Real-time bottleneck matching in spatial crowdsourcing. Sci China Inf Sci, 2021, 64(8): 189101, https://doi.org/10.1007/s11432-019-3061-x

Dear editor,

Spatial crowdsourcing (SC) services (e.g., Uber, DiDi, and Meituan) have become popular with smart-phone growth. However, the online matching problems in real-time spatial data are a key issue in SC [1–4]. Unlike the current one-sided online matching study in real-time spatial data [5], which focuses on minimizing the overall cost of the matching, we focus on minimizing the bottleneck cost, i.e., minimizing the maximum distance cost of the matching. The reason why we consider the bottleneck optimization goal is explained in Appendix A. The real-time minimum bottleneck matching (RMBM) problem in SC is defined as follows.

Definition 1 (The RMBM problem)**.** In a 2D space, given a worker set W with specific locations, a set of tasks T released by users whose spatial information is unknown before they appear; the RMBM problem is to find a matching M of W and T to minimize the maximum distance cost of all worker-task matching pairs, $\mathrm{Cost}(M) = \max_{w\in W, t\in T} \mathrm{dis}(w,t)$, where $\mathrm{dis}(\cdot,\cdot)$ is the distance function. And all the following constraints must be satisfied.

- Capacity constraint: a task can only be assigned to one worker and vice versa.
- Cardinality constraint: $|M| = \min(|W|, |T|)$.
- Real-time constraint: when a task occurs, the task must be allocated immediately to a worker as long as there exists an available worker. Otherwise, the task will expire.
- Invariability constraint: every worker-task matching pair cannot be revoked or re-matched.

LLDF algorithm. The RMBM problem is difficult to overcome because the bottleneck cost is very sensitive, i.e., the bottleneck cost is decided by only one single awful worker-task matching pair. To solve the RMBM problem, an online algorithm, local low-density first (LLDF), is proposed. In LLDF, we believe that workers with lower density have higher probabilities of being outliers and lead to a larger bottleneck cost. We, therefore, give the low-density worker a high priority to match for lowering the final bottleneck cost. Information on the idea of LLDF is shown in Appendix B and the description of workers' density is shown below.

Definition 2 (Density)**.** Given a set of workers $W = \{w_0, w_1, \ldots, w_{k-1}\}$ with specific locations, a distance function $\mathrm{DenDis}(\cdot,\cdot)$ in a 2D space and a distance threshold θ ($\theta > 0$), an arbitrary worker w_i's density is $\mathrm{Density}(w_i) = |S|$, where $S = \{\forall w_j \in W | \mathrm{DenDis}(w_i, w_j) \leqslant \theta\}$. For w_i, the larger $\mathrm{Density}(w_i)$ means more workers surrounding around w_i, and w_i's density $\mathrm{Density}(w_i)$ is higher.

We name $\mathrm{DenDis}(\cdot,\cdot)$ as "density distance" and $\mathrm{DenDis}(\cdot,\cdot)$ is the same with the distance function $\mathrm{dis}(\cdot,\cdot)$ in the RMBM problem definition by default. Notice that the different settings of the threshold θ have a considerable influence on the results of workers' density. To make our algorithm adaptively suit different distributions of workers, we use $\kappa \times \mathrm{AvgDenDis}$ as the threshold θ, where AvgDenDis is the average density distance of $\mathrm{DenDis}(\cdot,\cdot)$ of arbitrary two workers and κ is a preset parameter.

In many instances, a simpler distance metric than $\mathrm{dis}(\cdot,\cdot)$ can be used as $\mathrm{DenDis}(\cdot,\cdot)$ to accelerate LLDF. In fact, owing to the adaptive threshold $\kappa \times \mathrm{AvgDenDis}$ used in LLDF, the actual distance between workers does not matter when calculating the density of each worker, and LLDF can work well as long as the density distance function DenDis can roughly explain the relative distance distribution between workers.

The final question is how to give the lower density worker a higher priority to match. When a task t_i arrives, we calculate the average distance Avg_{t_i} of $\mathrm{dis}(\cdot,\cdot)$ between t_i and all available workers. Then we match t_i to the available worker with the minimum density within the range of $\eta \times \mathrm{Avg}_{t_i}$ away from t_i. Note that η is a preset parameter. Further, Algorithm 1 shows the whole procedure of LLDF.

Complexity and competitive analysis. For each new arriving task, the space and time complexity of LLDF is $O(|W|)$. For initialization, the space and time complexity of LLDF is $O(|W|)$ and $O(|W|^2)$, respectively. We also analyze the competitive ratio's lower bound of LLDF in the adversarial

* Corresponding author (email: linglingwang12@126.com)

model as shown in Theorem 1. The competitive ratio in the adversarial model is defined in Appendix C and the proof of Theorem 1 is shown in Appendix D.

Algorithm 1 LLDF

Input W, T;
Output A feasible matching M;
(i) Initialization:
$M \leftarrow \emptyset$;
$\text{AvgDenDis} \leftarrow \frac{\sum_{p=0,q=0}^{p=|W|-1,q=|W|-1} \text{DenDis}(w_p, w_q)}{|W|^2}$;
$\text{Density} \leftarrow [0, 0, \ldots, 0]_{|W|}$;
for $p = 0$ to $|W| - 1$ **do**
  $\text{DenSet} \leftarrow \{\forall u | u \in W \text{ and } \text{DenDis}(w_p, u) \leqslant \kappa \times \text{AvgDenDis}\}$;
  $\text{Density}[p] \leftarrow |\text{DenSet}|$;
end for
(ii) A task t_i arrives:
$\pi \leftarrow \{\text{dis}(t_i, w_0), \text{dis}(t_i, w_1), \ldots, \text{dis}(t_i, w_{|W|-1})\}$;
$\text{Avg}_{t_i} \leftarrow \frac{\sum_{j=0}^{j=k-1} \pi_j}{k}$;
$\text{Cand} \leftarrow \{\forall u | u \in W \text{ and } \text{dis}(t_i, u) \leqslant \eta \times \text{Avg}_{t_i}\}$;
$w_x \leftarrow$ the worker in Cand with the minimum density;
$M \leftarrow (t_i, w_x)$;
$W \leftarrow W - w_x$;
return M;

Theorem 1. The competitive ratio of LLDF is at least $\eta \cdot 2^{\lfloor k - \log k - 1 \rfloor}$, where $k = |M| = \min(|T|, |W|)$.

To validate the efficiency and effectiveness of LLDF, four existing algorithms, Greedy [6, 7], Permutation [6, 7], Balance [7, 8], and Greedy-HST [5, 9] are used as baseline algorithms to compare with LLDF. All the four baseline algorithms are the state-of-the-art algorithms for solving the general online bottleneck matching problem or the one-sided online minimum matching problem in real-time SC. Experiments on synthetic and real datasets show that LLDF is effective and considerably outperforms smaller bottleneck costs and that LLDF is also efficient in terms of both memory and running time. More details about the experiments are shown in Appendix E.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. U11811463).

Supporting information Appendixes A–E. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Tong Y, Zhou Z, Zeng Y, et al. Spatial crowdsourcing: a survey. VLDB J, 2020, 29: 217–250
2 Tong Y, She J, Ding B, et al. Online mobile micro-task allocation in spatial crowdsourcing. In: Proceedings of the 32nd IEEE International Conference on Data Engineering, 2016. 49–60
3 Li Y, Fang J, Zeng Y, et al. Two-sided online bipartite matching in spatial data: experiments and analysis. Geoinformatica, 2020, 24: 175–198
4 Tong Y, Zeng Y, Ding B, et al. Two-sided online micro-task assignment in spatial crowdsourcing. IEEE Trans Knowl Data Eng, 2020. doi: 10.1109/TKDE.2019.2948863
5 Tong Y, She J, Ding B, et al. Online minimum matching in real-time spatial data. Proc VLDB Endow, 2016, 9: 1053–1064
6 Kalyanasundaram B, Pruhs K. Online weighted matching. J Algorithms, 1993, 14: 478–488
7 Anthony B M, Chung C. Online bottleneck matching. J Comb Optim, 2014, 27: 100–114
8 Kalyanasundaram B, Pruhs K R. The online transportation problem. SIAM J Discrete Math, 2000, 13: 370–383
9 Meyerson A, Nanavati A, Poplawski L. Randomized online algorithms for minimum metric bipartite matching. In: Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm, 2006. 954–959