Theorem 1: The competitive ratio in adversarial model CR_A of the DemCOM algorithm has no bounds. The competitive ratio in random order model CR_{RO} of the DemCOM algorithm is same as that of the greedy algorithm for the tradition online task assignment (TOTA) problem.

Proof: Under the worst case of DemCOM algorithm, all inner crowd workers arrive after all the requests. As a result, no requests are served by inner crowd workers. Meanwhile, no outer crowd worker would like to serve the requests with minimum outer payment $v_r'm$ returned by Algorithm 2. In this case, all requests are rejected, and the total revenue is 0. However, in the offline matching scenario, the optimal case is that all requests are served by inner crowd workers, and the total revenue is $\sum_{i=1}^{|R|} v_{r_i}$. Thus, the competitive ratio in adversarial model CR_A of the DemCOM algorithm has no bounds.

Notice that Lines 1-6 are same as the greedy algorithm of TOTA problem. Only the requests that cannot be served by the inner crowd workers are assigned to the outer crowd workers using DemCOM algorithm. So the requests that are assigned to the inner crowd workers are same as that calculated by the greedy algorithm of TOTA problem. Meanwhile, the optimal case of the offline matching is still $\sum_{i=1}^{|R|} v_{r_i}$. In other words, all requests are served by the inner crowd workers. Thus,

$$\begin{aligned}\frac{\mathbb{E}[MaxSum(M)]}{MaxSum(OPT)} &= \frac{\mathbb{E}[MaxSum(M_{inner}) + MaxSum(M_{outer})]}{MaxSum(OPT)} \\ &= \frac{\mathbb{E}[MaxSum(M_{inner})] + \mathbb{E}[MaxSum(M_{outer})]}{MaxSum(OPT)}\end{aligned}$$

When requests are assigned to the outer crowd workers, the worst case is that no outer crowd worker would like to serve any request with the minimum outer payment, i.e., $\mathbb{E}[MaxSum(M_{outer})] = 0$. That is,

$$\frac{\mathbb{E}[MaxSum(M)]}{MaxSum(OPT)} \geq \frac{\mathbb{E}[MaxSum(M_{inner})]}{MaxSum(OPT)}$$

which is the CR_{RO} of the greedy algorithm of TOTA problem. Thus, the Theorem 1 holds. ∎

Theorem 2: The competitive ratio of the RamCOM algorithm can reach $\frac{1}{8e}$.

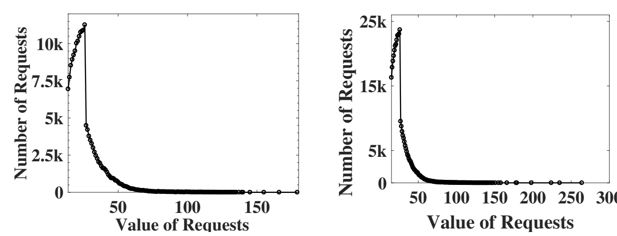
Proof: According to the discussion in Section II-B, the offline version of COM problem can be transformed into a weighted bipartite graph matching problem. We denote this weighted bipartite graph as G, and let $G_{[e^i,e^{i+1})}$ be a subgraph of G, which only contains the edges whose weights are between e^i and e^{i+1}. Let $G_{[e^{i-1},e^i)}$ be a of G subgraph, which only contains the edges whose weight is between 1 and e^i. Let $|OPT_{[e^i,e^{i+1})}|$ be the number of optimal offline matches of $G_{[e^i,e^{i+1})}$; and $|M_{[e^i,e^{i+1})}|$ be the number of matches of inner crowd workers returned by RamCOM. Let $|OPT_{[1,e^i)}|$ be the number of offline optimal matches of $G_{[1,e^i)}$; and $|M_{out_{[1,e^i)}}|$ be the number of matches of outer crowd workers returned by RamCOM.

For each $i \in [1, \theta]$, since only the requests whose values are larger than e^i can be assigned to inner crowd workers, the matches $M_{[1,e^i)}$ whose edge weights lie in the interval $[1, e^i)$ are all matches of outer crowd workers. On the other hand, the matches $M_{[e^i,\max(v_r)]}$ whose edge weights lie in the interval $[e^i, \max(v_r)]$ are the matches of (1) inner crowd workers and (2) outer crowd workers with remained revenue $(v_r - v_r'e) > e^i$.

$$\begin{aligned}&\mathbb{E}[MaxSum(M)] \\ =\ & \frac{1}{\theta}\left(\sum_{i=1}^{\theta} MaxSum(M_{[e^i,\max(v_r)]}) + \sum_{i=1}^{\theta} MaxSum(M_{out_{[1,e^i)}})\right) \\ =\ & \frac{1}{\theta}\sum_{i=1}^{\theta} MaxSum(M_{in[e^i,\max(v_r)]} + M_{out[e^i,\max(v_r)]}) \\ & + \frac{1}{\theta}\sum_{i=1}^{\theta} MaxSum(M_{out_{[1,e^i)}}) \\ \geq\ & \frac{1}{\theta}\sum_{i=1}^{\theta}(e^i|M_{in[e^i,e^{i+1})}| + e^i|M_{out[e^i,e^{i+1})}|) \\ & + \frac{1}{\theta}\sum_{i=1}^{\theta} MaxSum(M_{out_{[1,e^i)}})\end{aligned}$$

For any edge in $OPT_{[e^i,e^{i+1})}$, at least one of the two vertices of this edge must be matched in $(M_{in[e^i,e^{i+1})} \cup M_{out[e^i,e^{i+1})})$, so $|M_{in[e^i,e^{i+1})}| + |M_{out[e^i,e^{i+1})}| \geq \frac{1}{2}|OPT_{[e^i,e^{i+1})}|$. On the other hand,$MaxSum(M_{out_{[1,e^i)}}) \geq \frac{1}{e}OPT_{[1,e^i)}$. Thus,

$$\begin{aligned}&\mathbb{E}[MaxSum(M)] \\ \geq\ & \frac{1}{\theta}\sum_{i=1}^{\theta}(e^i\frac{|OPT_{[e^i,e^{i+1})}|}{2}) + \frac{1}{e\theta}\sum_{i=1}^{\theta} MaxSum(OPT_{[1,e^i)}) \\ \geq\ & \frac{1}{2e\theta}\sum_{i=1}^{\theta} MaxSum(OPT_{[e^i,e^{i+1})}) + \frac{1}{e\theta}\sum_{i=1}^{\theta} MaxSum(OPT_{[1,e^i)}) \\ \geq\ & \frac{1}{2e\theta} MaxSum(OPT)\end{aligned}$$



(a) Food Delivery Service (b) Intelligent Transportation

Fig. 1. Distribution of Request Values in Real-world Applications

The parameter θ is related to the maximal value of all the requests. Fig. 1 show the distribution of request values in the real-world applications. Fig. 1(a) is the value distribution of Ele.me [1], which is a spatial crowdsourcing platform providing food delivery services; Figure 1(b) is the value distribution of DiDi [2], which is a spatial crowdsourcing platform providing intelligent transportation. From these figures, we can see that the values of more than 95% requests are smaller than 50. Thus, in the real application, in more than 95% cases, $\theta \leq 4$. Thus, the competitive ratio of RamCOM algorithm can be larger than $\frac{1}{8e}$. ∎

REFERENCES

[1] "Ele.me," https://www.ele.me/home/.

[2] "Didi," https://www.didiglobal.com/.