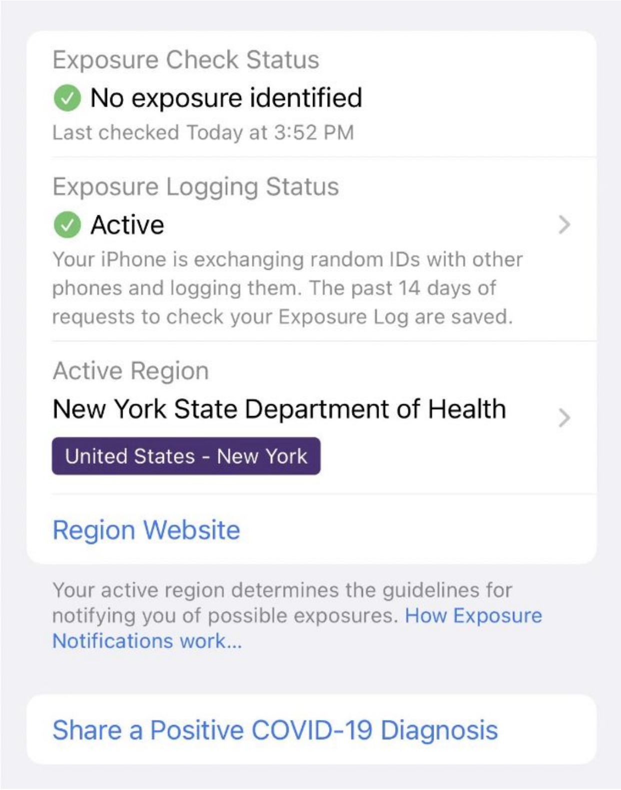
Mechanism Design for Crowdsourcing

Hui Huang, Wenqian, Xinghan, Yufei

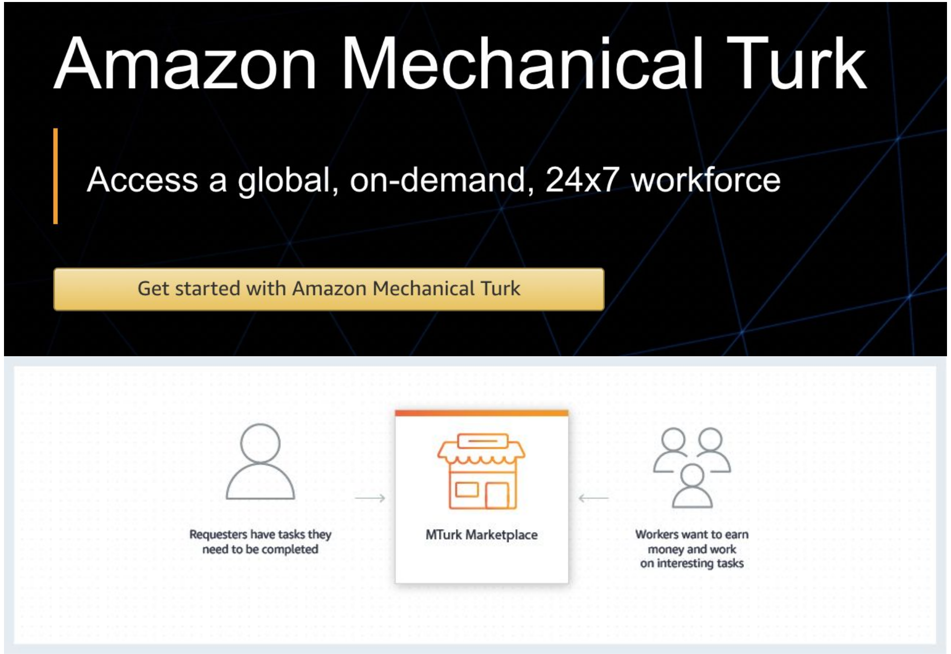
Columbia | Engineering
The Fu Foundation School of Engineering and Applied Science

Background

- Nowadays, **the proliferation of mobile devices** (e.g., smartphones, wearable devices, in-vehicle sensing devices) provides a new opportunity for extending existing crowdsourcing applications to a larger contributing crowd, **making contribution easier and everywhere.**
- Furthermore, today's cellphones are programmable and come with a **rich set of powerful embedded sensors** (such as GPS, WiFi/3G/4G interfaces, accelerometer, digital compass, gyroscope, microphone, and camera). The great potential of mobile sensing offers a variety of novel, **efficient ways to collect data.**.



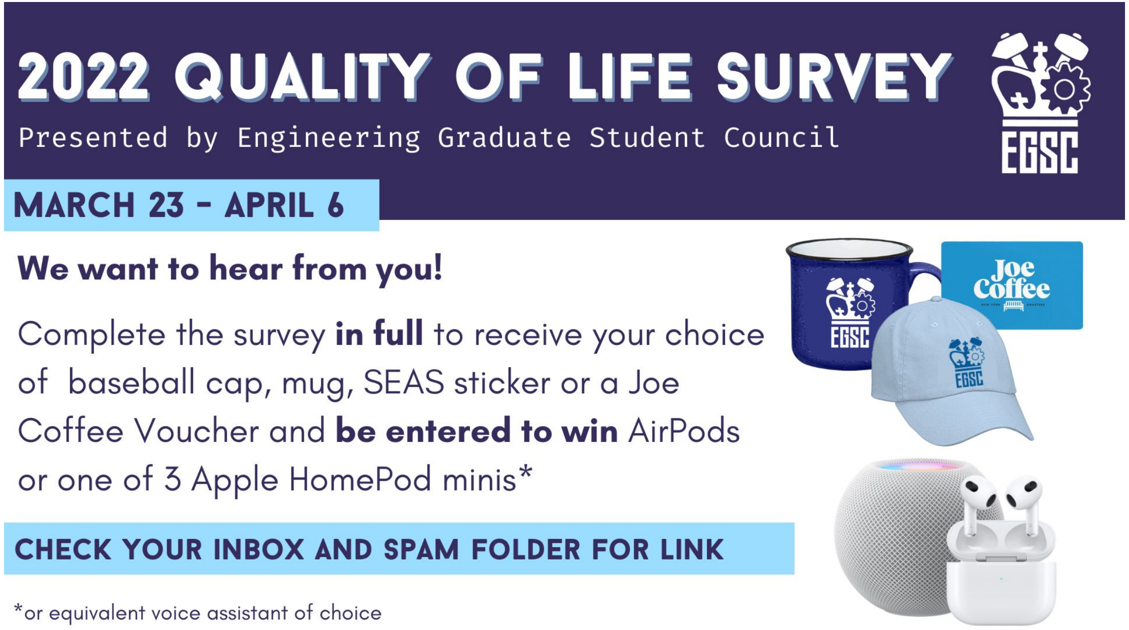
Background



Amazon Mechanical Turk (MTurk) is a crowdsourcing marketplace enabling individuals and businesses (known as Requesters) to engage a 24/7, global distributed workforce (known as Workers) to perform tasks.

A Human Intelligence Task (HIT) is a single, self-contained task a Requester creates on MTurk, an example of a task would be "Identify the red apple in this image of a fruit basket".

Background



Columbia Engineering Graduate Student Council (EGSC) want to conduct some surveys by providing incentives to students.

Background

vs.

- ➢ All of participating users **report** their strategic types, including the tasks they can complete and the bids, to the crowdsourcer (campaign organizer) **in advance**, and then the crowdsourcer selects a subset of users after collecting the information of all users to maximize his/her utility.

- ➢ **In practice**, however, users always **arrive one by one online** in a random order, and user availability changes over time. Therefore, **an online incentive mechanism** is necessary to make decisions, based on the information of users arriving before the present moment, without knowing future information.

Problem Formulation

- Crowdsourcer's info.
  - Each task has unit utility ***u***, one user perform one task
  - Objective function: **maximize # *of hired users/performed tasks***
  - Fix budget: ***B***

- Users' info.
  - **Stochastic arrivals** (online)
  - Cost ***c*** follows an i.i.d distribution (strong but practical)
  - Perform task if the payment is larger than or equal to ***c*** (IR)

Problem Formulation

Online Mechanism: **Allocation rule** + **Payment rule**

$$M = (f, p)$$

- ❑ Posted price Model (Singla, A.)
- ❑ Auction Model (Zhao, D.)

Mechanisms - Posted price model

Consider a user comes, the crowdsourcer will post a price to the current user. If the price is larger than the cost of this user, then the user just take the offer. Otherwise the user will just leave.

In the whole process, the crowdsourcer won't collect any information beyond the acceptance or rejection from the user.

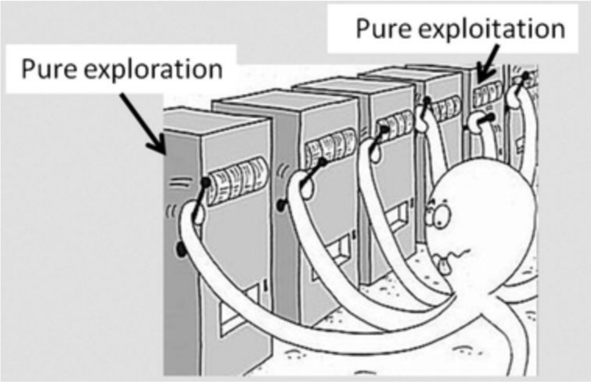
Mechanisms - Posted price model

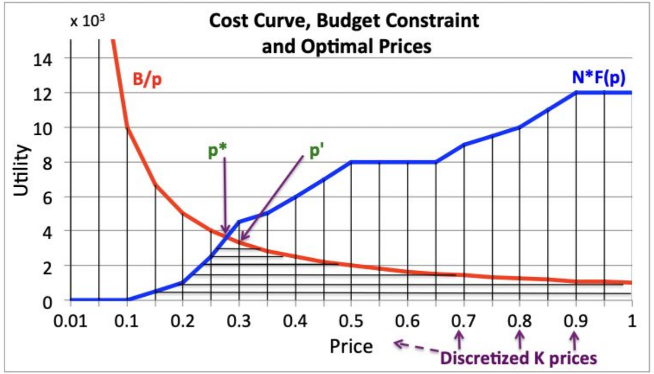
Multi-armed Bandit (MAB) Model

Assume we are playing a Multi-armed Bandit, we discretize the available price into different arms with uncertainty reward (1 if the user take it, 0 o/w).

The uncertainty comes from the random distribution of the user’s cost.

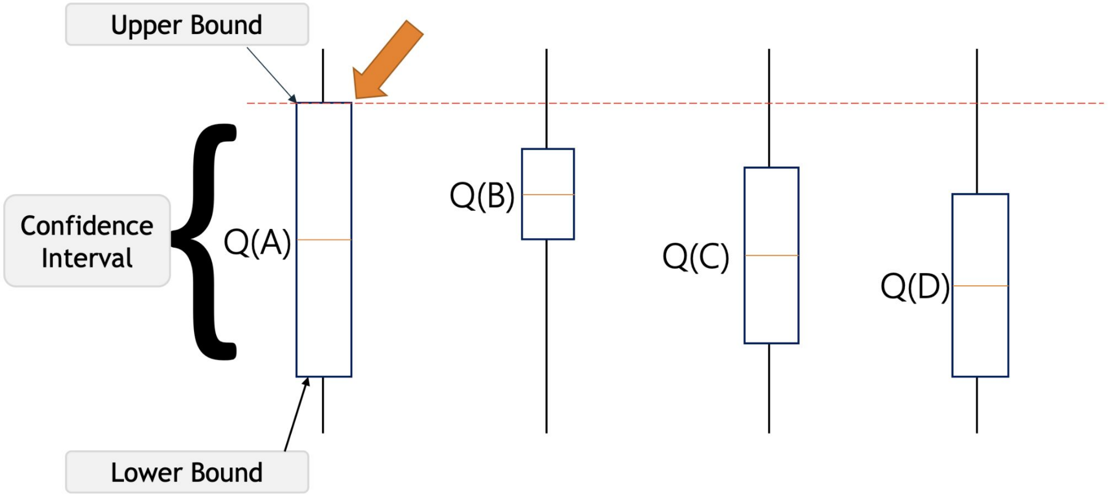
Since we have a fixed budget, we need a tradeoff between exploration and exploitation.





Mechanisms - Posted price model

Upper Confidence Bound (UCB)



As the acquisition function, UCB takes both mean and variance of the arms into consideration. This give us a natural tradeoff between exploration and exploitation.

Mechanisms - Posted price model

Upper Confidence Bound (UCB)

But the traditional UCB setting does NOT consider the existence of budget, which will lead the algorithm to always choose to play the highest price arm.

Hence adjustments need to be made.

$$\widetilde{V_i^t} = \min\left\{\widetilde{F_i^t}, \frac{B}{N \cdot p_i}\right\}$$

B: remaining budget
N: # times selecting this arm
pi: corresponding price

while $B^t > c_{\min}$ & $t < N$ **do**

$\widetilde{F_i^t} = F_i^t + \sqrt{\frac{2 \cdot \ln(t)}{N_i^t}}$; Traditional UCB

$\widetilde{V_i^t} = \min\left\{\widetilde{F_i^t}, \frac{B}{N \cdot p_i}\right\}$; Adjusted Budget

$i^t = \arg\max_i \widetilde{V_i^t}$ $s.t.$ $p_i \leq B^t$;

`/* ties broken by picking lowest` i `*/` ;

Offer price $p^t = p_{i^t}$ to worker w^t;

Observe acceptance decision y^t;

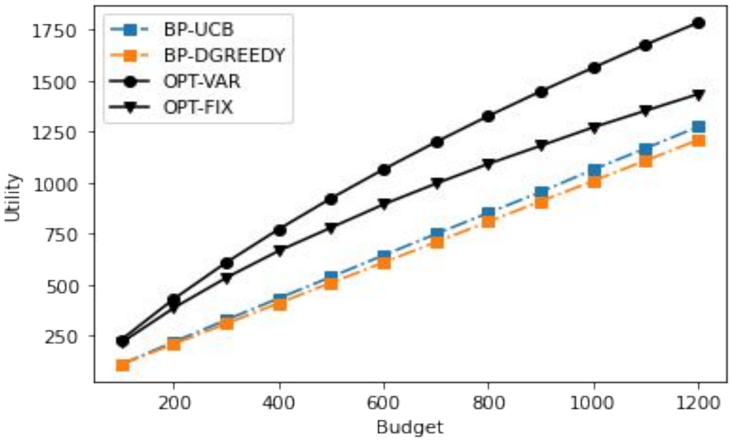
Update $F_{i^t}^t = F_{i^t}^t + \frac{(y^t - F_{i^t}^t)}{(N_{i^t}^t + 1)}$; $N_{i^t}^t = N_{i^t}^t + 1$;

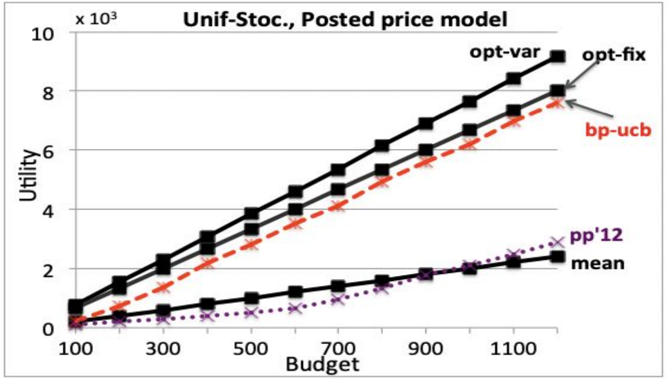
Update $U = U + y^t$; $B^{t+1} = B^t - p^t \cdot y^t$; $t = t + 1$;

end

Mechanisms - Posted price model

Our implementation vs. Paper's results

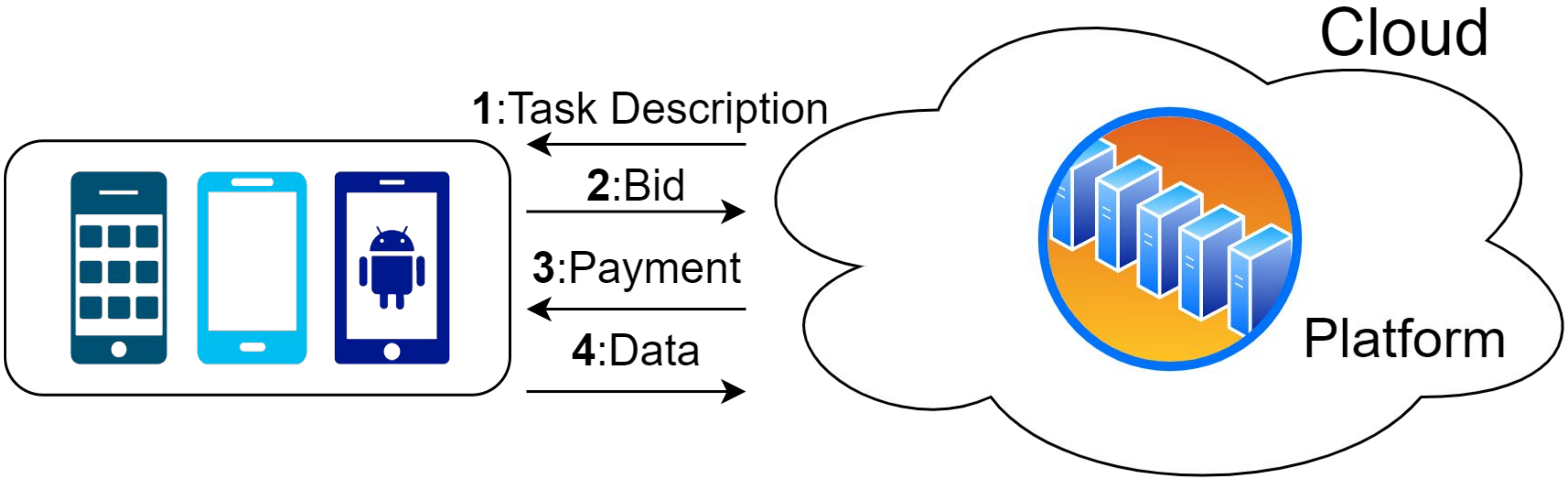




(b) Uniform-Stoc., Posted price model

BP-DGREEDY: Posted price online algo. with truthful biddings
OPT-VAR: Offline algo. with variable price (literally the optimum)
OPT-FIX: Offline algo. with fix price

Mechanisms - Auction model



Mechanisms - Auction model

- Divide all ***T*** stages and budget ***B*** into $\left(\lfloor \log_2(T) \rfloor + 1\right)$ parts for each stage.

Algorithm 1: Online Mechanism under Zero Arrival-departure Interval Case (*OMZ*)

Input: Budget constraint B, deadline T

$(t, T', B', \mathcal{S}', \rho^*, \mathcal{S}) \leftarrow (1, \frac{T}{2^{\lfloor \log_2 T \rfloor}}, \frac{B}{2^{\lfloor \log_2 T \rfloor}}, \emptyset, \epsilon, \emptyset)$;

while $t \leq T$ **do**

    if *there is a user i arriving at time step t* **then**

        if $b_i \leq V_i(\mathcal{S})/\rho^* \leq B' - \sum_{j \in \mathcal{S}} p_j$ **then**

            $p_i \leftarrow V_i(\mathcal{S})/\rho^*$; $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$;

        else $p_i \leftarrow 0$;

        $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{i\}$;

    end

    if $t = \lfloor T' \rfloor$ **then**

        $\rho^* \leftarrow$ **GetDensityThreshold**$(B', \mathcal{S}')$;

        $T' \leftarrow 2T'$; $B' \leftarrow 2B'$;

    end

    $t \leftarrow t + 1$;

end

Mechanisms - Auction model

- Allocation:
  - Marginal density not less than threshold ϱ
  - Total payment lower than stage budget ***B'***
- Payment:
  - Threshold value

Algorithm 1: Online Mechanism under Zero Arrival-departure Interval Case (*OMZ*)

Input: Budget constraint B, deadline T

$(t, T', B', \mathcal{S}', \rho^*, \mathcal{S}) \leftarrow (1, \frac{T}{2^{\lfloor \log_2 T \rfloor}}, \frac{B}{2^{\lfloor \log_2 T \rfloor}}, \emptyset, \epsilon, \emptyset)$;

while $t \leq T$ **do**

    if *there is a user i arriving at time step t* **then**

        if $b_i \leq V_i(\mathcal{S})/\rho^* \leq B' - \sum_{j \in \mathcal{S}} p_j$ **then**

            $p_i \leftarrow V_i(\mathcal{S})/\rho^*$; $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$;

        else $p_i \leftarrow 0$;

        $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{i\}$;

    end

    if $t = \lfloor T' \rfloor$ **then**

        $\rho^* \leftarrow$ **GetDensityThreshold**$(B', \mathcal{S}')$;

        $T' \leftarrow 2T'$; $B' \leftarrow 2B'$;

    end

    $t \leftarrow t + 1$;

end

Mechanisms - Auction model

- Learn a density threshold used for future decision

- Dynamically increases the sample size and stage-budget for the next stage

Algorithm 1: Online Mechanism under Zero Arrival-departure Interval Case (*OMZ*)

Input: Budget constraint B, deadline T

$(t, T', B', \mathcal{S}', \rho^*, \mathcal{S}) \leftarrow (1, \frac{T}{2^{\lfloor \log_2 T \rfloor}}, \frac{B}{2^{\lfloor \log_2 T \rfloor}}, \emptyset, \epsilon, \emptyset)$;

while $t \leq T$ **do**

  if *there is a user i arriving at time step t* **then**

    if $b_i \leq V_i(\mathcal{S})/\rho^* \leq B' - \sum_{j \in \mathcal{S}} p_j$ **then**

      $p_i \leftarrow V_i(\mathcal{S})/\rho^*$; $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$;

    else $p_i \leftarrow 0$;

    $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{i\}$;

  end

  if $t = \lfloor T' \rfloor$ **then**

    $\rho^* \leftarrow$ **GetDensityThreshold**$(B', \mathcal{S}')$;

    $T' \leftarrow 2T'$; $B' \leftarrow 2B'$;

  end

  $t \leftarrow t + 1$;

end

Mechanisms - Auction model

- **Greedy method**:

  (learn from past auctions)

  - Sort users by marginal density
  - If user *i*'s marginal density

    $$V_i(J)/b_i$$

    is larger than the total average value of selected users including *i*

    $$V(J \cup \{i\})/B',$$

    then we will add him to the new selected user set ***J.***

Algorithm 2: GetDensityThreshold

Input: Stage-budget B', sample set $\mathcal{S}'$

$\mathcal{J} \leftarrow \emptyset$; $i \leftarrow \arg\max_{j \in \mathcal{S}'}(V_j(\mathcal{J})/b_j)$;

while $b_i \leq \frac{V_i(\mathcal{J})B'}{V(\mathcal{J} \cup \{i\})}$ **do**

  $\mathcal{J} \leftarrow \mathcal{J} \cup \{i\}$;

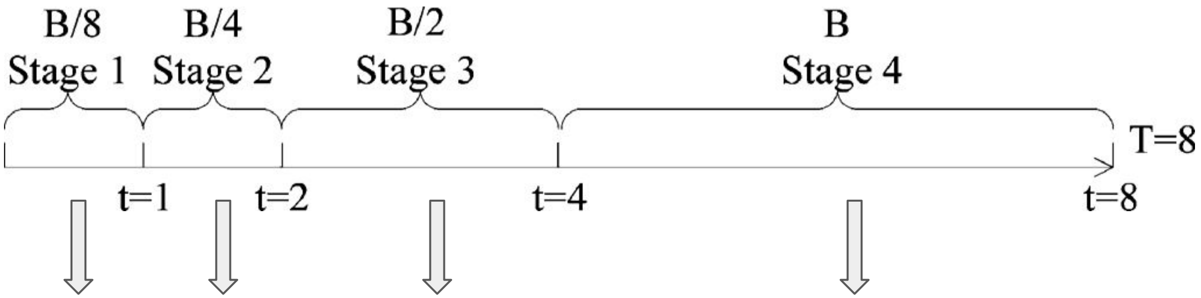
  $i \leftarrow \arg\max_{j \in \mathcal{S}' \setminus \mathcal{J}}(V_j(\mathcal{J})/b_j)$;

end

$\rho \leftarrow V(\mathcal{J})/B'$;

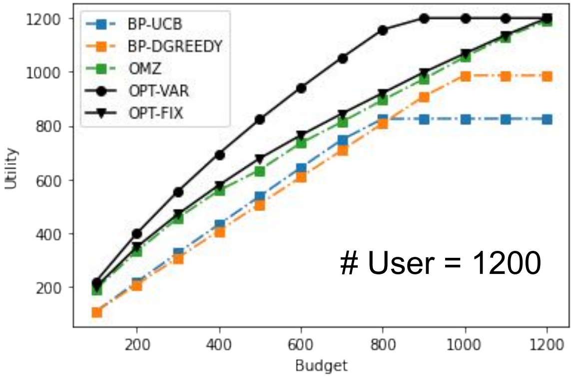
return ρ/δ;

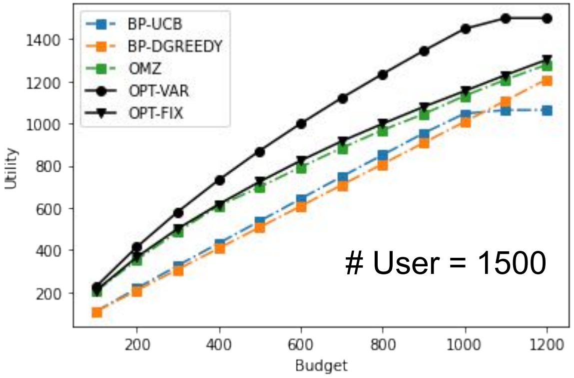
Mechanisms - Auction model

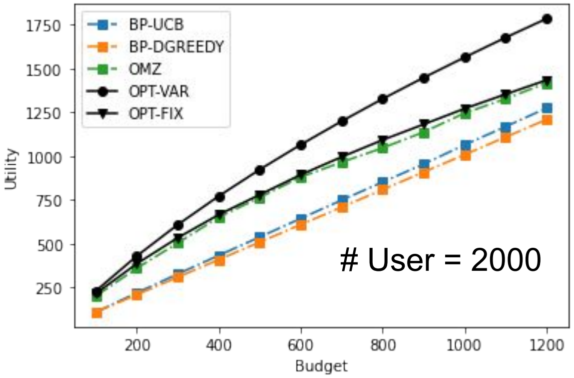


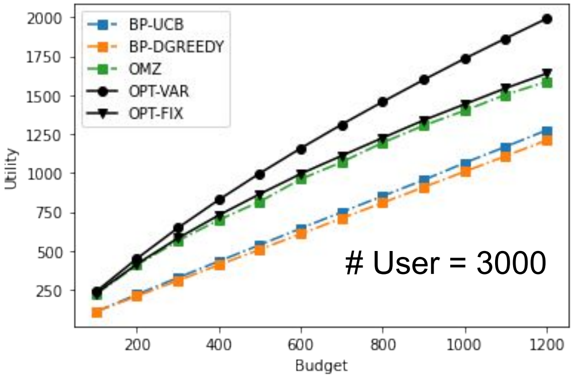
ϱ: 1/2 1/2 1/4 1/4

Numerical Results









Future Discussion

- ❖ Conduct more numerical experiments to compare two mechanisms under different parameter settings.

- ❖ Investigate the performance gap between the two mechanisms.

- ❖ Further improve the mechanisms
  - ➢ Use Thompson sampling to replace the UCB in MAB
  - ➢ Find a better way to include the budget information into UCB
  - ➢ Use regret minimization method to update the density in OMZ

Reference

[1] Singla, A., & Krause, A. (2013, May). Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *Proceedings of the 22nd international conference on World Wide Web* (pp. 1167-1178).

[2] Zhao, D., Li, X. Y., & Ma, H. (2014). Budget-feasible online incentive mechanisms for crowdsourcing tasks truthfully. *IEEE/ACM Transactions on Networking*, *24*(2), 647-661.