The Benefits of Delay to Online Decision-Making

Yaqi Xie (Chicago Booth)
Will Ma (Columbia GSB)
Linwei Xin (Chicago Booth)

Introduction: Delaying Real-Time Decisions

- Real-time decisions are usually irrevocable for many online decision-making problems
- One common practice is delaying real-time decisions
- In online retailing: there is typically a time delay between when the order is received and when it gets picked and assembled for shipping
- In ride-hailing, Uber has been implementing a batched matching algorithm to match riders with drivers in batches
- This type of wait-and-then-match to increase market thickness has been applied to many other contexts, including kidney exchange and online games

Downside of Delay

Decisions cannot be delayed forever

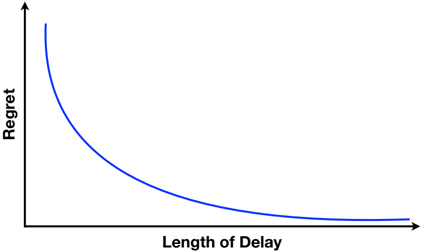
- Online retailing: long delay can lead to warehouse congestion that causes delivery delay
- Ride-hailing: riders can leave after a long wait

Research Objective: to study this fundamental trade-off and theoretically characterize the benefits of delaying real-time decisions

Main Results

Theorem (informal)

For a broad family of online decision-making problems, the gap between a proposed online policy with delay and an offline optimal hindsight policy decays **exponentially** fast in the length of delay.



Main takeaways:

- A little delay is all we need
- Our results support many practices that delay is usually not too long

Literature Review - Order Fulfillment Problem

- Xu, Allgor and Graves (2009): seminal paper on the order fulfillment problem
- Research on this topic in recent years: Acimovic and Graves (2015), Jasin and Sinha (2015), Andrews et al. (2019), Zhao, Wang and Xin (2020), DeValve et al. (2021), Amil, Makhdoumi and Wei (2022), Ma (2022)
- Acimovic and Farias (2019): a recent tutorial on the fulfillment optimization problem

Literature Review - Batching

- Xu, Allgor and Graves (2009): examine the practical benefits of reactively batching orders together by analyzing real data
- Wei, Jasin and Kapuscinski (2021): study the optimal time to delay fulfillment and how to proactively batch orders
- Wang et al. (2022): order holding to consolidate multiple orders placed by the same person or to catch order cancellations
- Batching in other contexts: Gurvich and Ward (2014), Ashlagi et al. (2019), Akbarpour, Li and Gharan (2020), Ashlagi and Roth (2021), Chen, Elmachtoub and Lei (2021), Feng and Niazadeh (2021), Kerimov, Ashlagi amd Gurvich (2021)
- Advance demand information in inventory control (e.g., Gallego and Özer 2001, Lu, Song and Yao 2003) and queueing (e.g., Spencer, Suda and Xu 2014, Xu 2015, Xu and Chan 2016)

Literature Review - Constant Regret

- Expected regret in online resource allocation problems: Jasin and Kumar (2012), Arlotto and Gurvich (2019), Bray (2019), Bumpensanti and Wang (2020), Vera and Banerjee (2021)
- Vera and Banerjee (2021): develop a general technique (called the Bayes selector) that can lead to an online policy with constant expected regret for a large class of online resource allocation problems

Model (Vera and Banerjee 2021)

- M different types of resources indexed by i
  - Initial budget vector $B = (B_1, \ldots, B_M)$
- N different types of customers, and there is a permissible set of actions S_j for each type j
- Each action $s \in S_j$ requires a_{is} units of resource i and earns reward r_s
  - S_j contains the "reject" option
  - Partial allocation is not allowed
- At each time t, a customer of type j arrives with probability λ_j
  - No arrival is possible, with probability $\lambda_0 = 1 - \sum_{j=1}^{N} \lambda_j$
- The goal is to maximize the total expected T-period reward

Examples

- Multisecretary problem: only one resource type
- Online packing (or quantity-based network revenue management): each customer type is associated with one action that consumes possibly multiple resources
- Online matching: each customer type can be satisfied by using one resource from a given set of options
- Order fulfillment with delivery deadline and trucking capacities:
  - each order type is associated with a deadline (e.g., same-day delivery)
  - capacities on each warehouse-destination pair
- Omnichannel fulfillment: warehouse/store inventory, online/offline customers

Model with K-Period Delay

- The first customer is indexed by T and the last customer is indexed by 1
- The decision for the arrival at time t is still made at t but with advance information of arrivals at $t-1, \ldots, t-K$
- The decision is based on:
  - realized demand $D_j[t, t-K]$ over time $[t, t-K]$ for type j
  - forecasted demand $\lambda_j[t-K-1, 1]$ over time $[t-K-1, 1]$ for type j
- The classical setting without delay is captured by setting $K = 0$
- REG $\triangleq V^{\text{off}}(\mathcal{I}) - V^{\text{on}}(\mathcal{I})$
  - Additive expected loss of an online policy compared to the optimal offline given instance $\mathcal{I}$
  - The offline policy can also be regarded as a delay policy (i.e., $K = T - 1$)

Differences from Delay in Two-Sided Marketplaces

- Each single decision is delayed by K, whereas the decisions are made by batches in online marketplaces
- There is a fundamental trade-off between increasing market thickness and mitigating the risk that participants may abandon the market
  - jobs can become obsolete in two-sided marketplaces, whereas jobs never expire in our setting (e.g., order fulfillment)
- Supply is fixed and cannot be replenished in our context, whereas there are both new arrivals as supply and demand in two-sided marketplaces.

A Special Case: Multisecretary Problem

- Only one type of resource
  - Initial budget B (i.e., B available positions)
- The action set is the same for each candidate type: accept or reject
- Accepting type j requires one position and earns reward r_j
- The goal is to maximize the total reward (interpreted as the total capability of accepted candidates)

Solving an Online LP at Time t

$$
\begin{aligned}
\max_{x_j, x_{\emptyset j} \geq 0} \quad & \sum_j r_j x_j \qquad \text{(online LP with delay } K\text{)} \\
\text{s.t.} \quad & \sum_j x_j \leq B_t \\
& x_j + x_{\emptyset j} = D_j[t] + D_j[t-1, t-K] + \lambda_j[t-K-1, 1] \quad \forall j
\end{aligned}
$$

- $\lambda_j = \mathbb{E}[D_j]$
- $x_j, x_{\emptyset j}$: number of accepted and rejected type j candidates
- The classical online LP is equivalent to our current LP with $K = -1$ (interpreted as making a decision before seeing the arrival at t).

Algorithm

Algorithm 1: Algorithm for the Multisecretary Problem with Delay

for $\underline{t = T, T-1, \ldots, K+2}$ **do**

  Observe $\Theta_t, \Theta_{t-1}, \ldots, \Theta_{t-K}$;

  Solve the online LP with delay K and obtain an optimal solution $\{X_{t,j}, X_{t,\emptyset j}\}_j$;

  if $\underline{X_{t,\Theta_t} \geq X_{t,\emptyset\Theta_t}}$ **then**

    Accept Θ_t and update $B_{t-1} \leftarrow B_t - 1$;

  else

    Reject Θ_t and let $B_{t-1} \leftarrow B_t$;

  end

end

Observe $\Theta_{K+1}, \ldots, \Theta_1$ and accept B_{K+1} candidates with highest r_{Θ_t} for $t = K+1, \ldots, 1$.

Main Result I

Theorem

For any instance and $K \geq 0$, the expected regret of Algorithm 1 has the following upper bound:

$$\text{REG} \leq \rho_0^K \cdot \frac{2 \cdot r_{\max} \cdot e^{-\lambda_{\min}}}{\lambda_{\min}^2}.$$

- $\rho_0 \triangleq \lambda_{\min} e^{-\lambda_{\min}} + (1 - \lambda_{\min})$, $\lambda_{\min} \triangleq \min_{j \in [N]} \lambda_j$, $r_{max} \triangleq \max_{j \in [N]} r_j$
- When $K = 0$, our bound $\frac{2 \cdot r_{max} \cdot e^{-\lambda_{\min}}}{\lambda_{\min}^2}$, tightens the one $\frac{2 \cdot r_{max}}{\lambda_{\min}^2}$ implied by Vera and Banerjee (2021) by a factor of $e^{-\lambda_{\min}}$ $(\leq \rho_0)$

Intuition: Benefits of Delay

$$\max_{x_j, x_{\emptyset j} \geq 0} \quad \sum_j r_j x_j \qquad \text{(online LP without delay)}$$

$$\text{s.t.} \quad \sum_j x_j \leq B_t$$

$$x_j + x_{\emptyset j} = D_j[t] + \lambda_j[t-1, t-K] + \lambda_j[t-K-1, 1] \quad \forall j$$

$$\max_{x_j, x_{\emptyset j} \geq 0} \quad \sum_j r_j x_j \qquad \text{(online LP with delay } K\text{)}$$

$$\text{s.t.} \quad \sum_j x_j \leq B_t$$

$$x_j + x_{\emptyset j} = D_j[t] + D_j[t-1, t-K] + \lambda_j[t-K-1, 1] \quad \forall j$$

Turn forecasted demands into realized demands!

Proof Sketch: Exponential Convergence

$$\max_{x_j, x_{\emptyset j} \geq 0} \quad \sum_j r_j x_j \qquad \text{(online LP with delay } K\text{)}$$

$$\text{s.t.} \quad \sum_j x_j \leq B_t$$

$$x_j + x_{\emptyset j} = D_j[t, t-K] + \lambda_j[t-K-1, 1] \quad \forall j$$

$$\max_{x_j, x_{\emptyset j} \geq 0} \quad \sum_j r_j x_j \qquad \text{(offline LP)}$$

$$\text{s.t.} \quad \sum_j x_j \leq B_t$$

$$x_j + x_{\emptyset j} = D_j[t, t-K] + D_j[t-K-1, 1] \quad \forall j$$

- Use concentration inequality with exponential bounds

Proof Sketch: Exponential Convergence (cont.)

- Let $\left\{X_{t,j},\ X_{t,\emptyset j}\right\}$ and $\left\{X^*_{t,j},\ X^*_{t,\emptyset j}\right\}$ be the optimal solutions
- Coupling argument: bounding the probability of incorrect decisions at each time t before time $K+1$
- When incorrect acceptance for Θ_t (i.e., customer type at t) occurs,
  - $X_{t,\Theta_t} \geq X_{t,\emptyset\Theta_t}$ and $X^*_{t,\Theta_t} = 0$
- When incorrect rejection for Θ_t occurs,
  - $X_{t,\Theta_t} < X_{t,\emptyset\Theta_t}$ and $X^*_{t,\emptyset\Theta_t} = 0$
- Those conditions suggest that incorrect decisions occur only if $\left\{X_{t,j}, X_{t,\emptyset j}\right\}$ deviates from $\left\{X^*_{t,j}, X^*_{t,\emptyset j}\right\}$
- Use Hoeffding's inequality to bound the probability of deviations

Algorithm for the General Model

Algorithm 2: Algorithm for the General Resource Allocation Problem with Delay

for $t = T, T-1, \ldots, K+2$ **do**

  Update the feasible action sets $\{S_{t,j}\}_{j \in [N]}$;

  Observe $\Theta_t, \Theta_{t-1}, \ldots, \Theta_{t-K}$;

  Solve the online LP with delay K and obtain an optimal solution $\{X_{t,s}\}_{s \in S_t}$;

  Choose action $s_t \leftarrow \arg\max_{s \in S_{t,\Theta_t}} \{X_{t,s}\}$, breaking ties arbitrarily;

  Update $B_{t-1,i} \leftarrow B_{t,i} - a_{is_t}$ for each $i \in [M]$;

end

Update $\{S_{K+1,j}\}_{j \in [N]}$ and observe $\Theta_{K+1}, \Theta_K, \ldots, \Theta_1$; follow the offline optimal hindsight policy for the last $K+1$ allocations.

Regret Analysis

$$
\begin{aligned}
\max \quad & \sum_{s\in S_t} r_s x_s \qquad \textcolor{blue}{\text{(online LP with delay } K\text{)}} \\
\text{s.t.} \quad & \sum_{s\in S_t} a_{is} x_s \le B_{t,i}, \quad \forall i \\
& \sum_{s\in S_{t,j}} x_s = D_j[t] + D_j[t-1, t-K] + \lambda_j[t-K-1, 1], \quad \forall j
\end{aligned}
$$

- Use LP's Lipschitz continuity on right-hand-side vector

$$\|X_t^* - X_t\|_\infty \le \kappa_t \, \|D[t-K-1, 1] - \lambda[t-K-1, 1]\|_\infty$$

- The feasible action set S_t shrinks over time t, and the Lipschitz constant κ_t is non-increasing

Main Result II

$a_{\max} \triangleq \max_{i\in[M],s\in S} a_{is}$, $r_{\max} \triangleq \max_{s\in S} r_s$, $\lambda_{\min} \triangleq \min_{j\in[N]} \lambda_j$, $\lambda_{\max} \triangleq \max_{j\in[N]} \lambda_j$, and $C_{\max} \triangleq \max_{j\in[N]} \left\{\left|S_j \setminus s_{\emptyset j}\right|\right\}$, $\kappa_{\max} \triangleq \kappa_T$,

$$\rho_1 \triangleq \exp\left(-2\lambda_{\min}^2\right), \quad \rho_2 \triangleq \lambda_{\min} \exp\left(-\frac{\lambda_{\min}}{\kappa_{\max}^2(C_{\max}+1)^2}\right) + (1-\lambda_{\min})$$

$$\rho \triangleq \max\{\rho_1, \rho_2\}$$

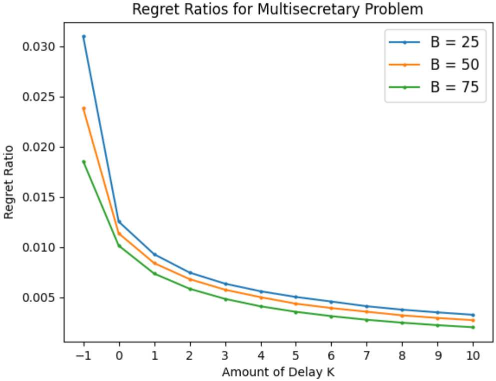
Theorem

For any instance and $K \geq 0$, the expected regret is upper bounded by:

$$a_{\max} \cdot M \cdot r_{\max} \cdot \left(\left\lceil \frac{1}{\lambda_{\min}} \right\rceil \cdot C_{\max} \cdot \rho_1^{-2\frac{C_{\max}-1}{\lambda_{\min}}} + \frac{\kappa_{\max}^2(C_{\max}+1)^2}{\lambda_{\min}^2} \cdot \exp\left(\frac{4\lambda_{\max} C_{\max}}{\kappa_{\max}^2(C_{\max}+1)^2}\right) \right) \cdot \rho^K.$$

Numerical Experiments: Multisecretary Problem

- $(r_j) = (10, 20, 30, 40, 50, 60, 70, 80, 90, 100)$, $T = 100$, $B \in \{25, 50, 75\}$
- Randomly generate 100 sets of distribution $\{\lambda_j\}$ by random points in the simplex $\sum_j \lambda_j = 1$



JD.com's RDC-FDC Fulfillment Network

- FDC (Front Distribution Center):
  - closer to end customers for reducing shipping distances
  - limited capacities (both inventory and fulfillment)
- RDC (Regional Distribution Center):
  - upper-layer warehouses with larger capacities
  - replenishing several lower-layer FDCs
  - "back-up" option when FDCs run out of inventory

Numerical Experiments: Order-Fulfillment Problem

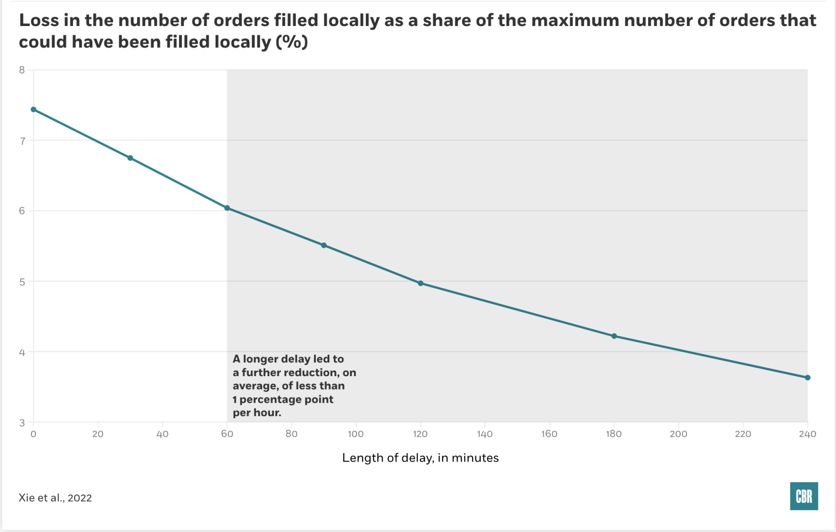
- One FDC with finite inventory and one RDC with infinite inventory
- B_i: initial inventory of SKU i at the FDC
- Order type j: requests a_{ij} units of SKU i
- All orders are delayed by the same absolute time (i.e., by K minutes, not by K orders), so that system congestion will not affect delay duration
- The goal is to maximize the total number of orders entirely fulfilled by the FDC
- Dataset from the 2020 MSOM Data Driven Research Challenge (Shen et al. 2020)

Benefits of Delay: A Toy Example

- There are four SKUs: A, B, C, D
- Each has one unit of inventory at the FDC
- There are three orders arriving in sequence: $\{B, C\}$, $\{A, B\}$, and $\{C, D\}$
- A greedy-type online policy without delay fulfills only the first order from the FDC, whereas a delayed policy fulfills the late two orders from the FDC

Simulation Results

Simulation Results



Thanks for Your Attention!

Image Source: CBR

The manuscript is available on SSRN:
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4248326