Columbia B9136 March 04, 2025

Lecture 6 — Online Multi-Resource Allocation

Lecturer: Will Ma *Scribe: Millend Roy and Ton Meesena*

All the guarantees discussed in the lecture are related to fluid relaxations. The following is the outline of contents covered in this lecture:

1. Online Generalized Assignment via OCRS
   (a) Model Formulation
   (b) Randomized Rounding Algorithm:
      i. Solve FLU
      ii. Do randomized routing on arrival type j to resource i according to FLU
      iii. OCRS for resource i decides whether to accept/reject the route
   (c) Analysis
      i. k_i-Inventory Constraint: connect to the result from $k-$unit prophet OCRS
      ii. General Constraint: connect to the result from OCRS for knapsack
2. Static Assortment Calendar via Prophet Inequality
   (a) Model Formulation & CDLP and its Dual
   (b) Threshold-based Algorithm:
      i. Solve $CDLP$,
      ii. Draw assortment S_t randomly according to $CDLP$
      iii. Discard items in S_t whose fluid revenue per resource is below a threshold
   (c) Analysis: $\mathcal{P}(\mathcal{I}) \geq \mathsf{FLU}(\mathcal{I})/2$
      i. Dynamic Substitution
      ii. Static Substitution
3. Online Personalized Assortment via Online Matching
   (a) Model Formulation & CDLP and its Dual
   (b) Online Algorithm with Shadow Price:
      i. Choose an assortment maximizing expected return under ϕ_t at time t, considering the shadow price.
   (c) Analysis: $\mathcal{P}(\mathcal{I}) \geq \left(1 - \left(1 + \frac{1}{k}\right)^{-k}\right) \mathsf{FLU}(\mathcal{I})$ through primal-dual arguments.
      i. Show that revenue gain is equal to the dual gain
      ii. Show dual feasibility
4. Correlation and Interpretation of Fluid
   (a) Formulations of coupling problem and FLU
   (b) Correlation Gaps
      i. Upper Bound: algorithmic result from prophet $k-$secretary problem
      ii. Lower Bound: hard instance with Bernoulli random variables

1 Online Generalized Assignment via OCRS

1.1 Model

Consider the following instance $\mathcal{I}$ with:

1. **Types:** resources $i \in [m]$ and customer types $j \in [n]$.
2. **Reward**: $r_{ij} \geq 0$ for assigning i to j.
3. **Size:** $s_{ij} \in (0, 1]$ that type j takes when assigned to resource i. A special case is discussed later, where $s_{ij} = \frac{1}{k_i}$, $\forall j$. Here, k_i represents the inventory for resource i which captures the knapsack constraint.
4. **Capacity Constraint**: each resource has a unit capacity, i.e., each resource can be only assigned up to the total size of 1.

For the online arrival process, the following notations are used:

1. **Arrival Rate:** λ_{t_j} probability that arrival at time t has type j, $\forall t \in [T], \forall j \in [n]$.
2. **Total Arrival Rate:** $\sum_j \lambda_{t_j} = 1$, $\forall t$.

So, the process is such that we know the arrival probabilities λ_{t_j} in advance, but the actual realizations of the types of customers are unknown. At each time t, a type j_t is revealed according to the probability distribution $\{\lambda_{t_j}\}_{j=1}^{n}$. Additionally, we have the following assumption:

Assumption 1. The type realizations $\{j_t\}_{t=1}^{T}$ are independent across time.

Therefore, the dynamics of the process can be described as:
For each time period $t = 1, \ldots, T$:

1. A customer type j_t is realized with probability λ_{t_j}.
2. Upon observing type j_t, the system knows (a) the associated size s_{ij_t} and (b) reward r_{ij_t}.
3. The decision maker selects at most one resource $i \in [m]$ to assign the customer to, subject to feasibility constraints (e.g. respecting resource capacities).
4. If assignment occurs to resource i, a reward r_{ij_t} is collected.
5. The system proceeds to the next step with the goal of maximizing the expected total reward.

Remark 2. Although the set of possible types may vary across time, this can be modeled by setting $\lambda_{t_j} = 0$ for any type j that cannot arrive at time t. Similarly, periods with no arrivals can be captured by introducing a "null" type with zero rewards. This keeps the model simple and general without needing additional complexity.

Note that s_{ij} can be also heterogeneous across different customer types. As a special case of the model, if $s_{ij} = \frac{1}{k_i}$ $\forall j$, this is equivalent to resource i having k_i units of inventory. Since the total capacity is 1, and each assignment consumes $\frac{1}{k_i}$, resource i can support at most k_i assignments. This formulation allows us to capture both general knapsack constraints and, in this special case, the classic setting where each resource can serve at most k_i customers.

For this problem, we will show that the same guarantees achieved for $c-$selectable OCRS extend directly to the multi-resource setting. Importantly, introducing multiple resources does not worsen the guarantee; the competitive ratios remain the same as in the single-resource case.

We begin by defining the fluid relaxation of the problem for an instance $\mathcal{I}$ which captures types, rewards, capacities, and arrival rates of the problem. Let z_{ij} represent the number of times resource i is assigned to type j. We have:

$$\begin{aligned}
\mathsf{FLU}(\mathcal{I}) = \max \quad & \sum_{ij} r_{ij} z_{ij} \\
\text{s.t.} \quad & \sum_{j} s_{ij} z_{ij} \leq 1 \quad \forall i \quad \text{(capacity constraints)} \quad (\text{❋}) \\
& \sum_{i} z_{ij} \leq \lambda_j = \sum_{t} \lambda_{t_j} \quad \forall j \quad \text{(arrival constraints)} \quad (\text{♥}) \\
& z_{ij} \geq 0 \quad \forall i, j
\end{aligned}$$

Here, the constraint (❋) refers to the total size for resource i and the constraint (♥) refers to the expected number of arrivals of type j.

We know that $\mathsf{OFF}(\mathcal{I}) \leq \mathsf{FLU}(\mathcal{I})$. This is because $\mathsf{OFF}(\mathcal{I})$ algorithm observes all type realizations and assigns resources optimally and the above constraints hold on every realized path. By linearity of expectation, they also hold in expectation. Therefore, if we define z_{ij} as the expected number of type j matched to resource i under OFF, z_{ij} forms a feasible solution to the $\mathsf{FLU}(\mathcal{I})$ with an objective value equal to the OFF's expected reward. Since the LP's optimal value is at least as large as any feasible solution, we have: $\mathsf{OFF}(\mathcal{I}) \leq \mathsf{FLU}(\mathcal{I})$.

1.2 Algorithm: Randomized Rounding

Algorithm: Randomized Rounding

1. Solve FLU, hereafter let $(z_{ij})_{i,j}$ refer to the optimal solution.
2. Given arrival of type j, "route" to resource i w.p. $\frac{z_{ij}}{\lambda_j}$, $\quad \forall i$ and to nothing w.p. $1-\sum_i \frac{z_{ij}}{\lambda_j}$ (valid probability distribution by (♥)). An arrival being routed is analogous to an agent being *active* in the OCRS context.
3. Once type j is routed to resource i, the OCRS for resource i decides to accept or reject. If it is accepted, we assign this resource i to this type of customer j; otherwise, we assign nothing.

The key idea is to decouple the problem across resources. We first solve a single fluid LP that globally balances the allocation probabilities across all resources. This LP informs how frequently each type should be routed to each resource. After routing, each resource independently applies its own OCRS to decide which arrivals to accept, exactly as in the standard OCRS framework.

Rather than worrying about global coordination after routing, we handle contention at each resource separately through OCRS. To understand the overall guarantee of this algorithm, we will next analyze the performance and then formalize the precise OCRS guarantees needed.

1.3 Analysis

Let $\mathcal{P}(\mathcal{I})$ be the expected return of the randomized rounding algorithm introduced in Subsection 1.2 for instance $\mathcal{I}$. To facilitate the analysis, we introduce the following variables:

1. $D_{ti}^j :=$ indicator that, at time t, the type j arrives and is routed to the resource i. That is,
$$D_{ti}^j \sim \text{Ber}(\lambda_{t_j} \cdot \frac{z_{ij}}{\lambda_j}).$$
2. $X_{ti}^j :=$ indicator that the OCRS accepts type j for resource i at time t if it is routed.

That is, the algorithm assigns resource i to customer type j at time t when $D_{ti}^j X_{ti}^j = 1$. The expected reward from this algorithm can be written as:

$$\begin{aligned}
\mathcal{P}(\mathcal{I}) &= \sum_{i,j} r_{ij} \sum_t \mathbb{E}[D_{ti}^j X_{ti}^j] \\
&= \sum_{i,j} r_{ij} \sum_t \Big[\mathbb{E}[D_{ti}^j] \cdot \underbrace{\mathbb{E}[X_{ti}^j]}_{\geq c_i \ \forall t,j \text{ by OCRS}} \Big] \quad \text{(using independence between routing and assigning decisions)} \\
&\geq \sum_{i,j} r_{ij} \sum_t \lambda_{t_j} \frac{z_{ij}}{\lambda_j} c_i \qquad \text{(since } \lambda_j = \sum_t \lambda_{t_j}\text{)} \\
&= \sum_{i,j} r_{ij} z_{ij} c_i \\
&\geq \big(\min_{i\in[m]} c_i\big) \underbrace{\sum_{i,j} r_{ij} z_{ij}}_{=\mathsf{FLU}}
\end{aligned}$$

Now to understand what c_i is, let us first try to look at a special case of the problem before moving to the general problem.

CASE 1: k_i-inventory constraint
Here, we assume that resource i has k_i units in inventory, i.e.,

$$s_{ij} = \frac{1}{k_i}, \quad \forall j.$$

Our goal is to show that we can design an OCRS for resource i with the guarantee:

$$\mathbb{E}[X_{t_i}^j] \geq c_i, \forall j, \forall t,$$

given that we can assign at most k_i units. For every t, a customer gets routed to resource i w.p. $\sum_j \lambda_{t_j} \frac{z_{ij}}{\lambda_j}$. We can think of a customer being *routed* to i as an agent being *active* in the OCRS.

We know that the expected total number routed to i is

$$\begin{aligned}
&= \sum_t \sum_j \lambda_{t_j} \frac{z_{ij}}{\lambda_j} \\
&= \sum_j z_{ij} \\
&\leq k_i \quad \text{(by the LP constraint (*))}
\end{aligned}$$

Now, consider the resource's perspective: it faces a sequence of requests arriving dynamically over time. At each time step, it must decide whether to accept or reject an incoming request while ensuring that its total accepted requests do not exceed its capacity k_i. *This setup exactly matches the $k-$unit OCRS problem, where a resource must handle a limited number of assignments in an online fashion.*

Thus, the performance guarantee for the resource is directly determined by the OCRS guarantee c_i, which depends on k_i. Specifically:

- If $k_i = 1$, the standard one-unit OCRS guarantee is $c_i = 1/2$.
- For large k_i, concentration properties improve the allocation efficiency, and the guarantee approaches $c_i \approx 1 - \frac{1}{\sqrt{k_i}}$.

CASE 2: general constraint

For the general case, $s_{ij} \in (0, 1]$, handling resource assignments becomes more complex. When a type j arrives at time t and is routed to resource i, the resource may face different types j, each with different sizes s_{ij}. So, the OCRS has to handle heterogeneous sizes.

Previously, OCRS guarantees could ignore the identity of the type j because all types consumed the same amount of capacity. However, with varying sizes, OCRS decisions depend on the specific s_{ij}, as some types might fit into the remaining capacity while others cannot.

We define the *effective size* of t for resource $i = \sum_j D_{ti}^j s_{ij}$, where $D_{ti}^j = 1$ if type j at time t is routed to resource i, and zero otherwise. This captures the actual size consumption from type j if it is routed to resource i, and is zero if nothing is routed.

The key constraint from the LP tells us that the total expected effective size routed to resource i is at most 1: $\sum_t \mathbb{E}[\sum_j D_{ti}^j s_{ij}] = \sum_j s_{ij} z_{ij} \leq 1$ (by constraint (✻)).

In this setting, we apply a knapsack OCRS that handles varying item sizes and selects a subset of items to accept. Therefore, $c_i = 1/4$. (same as HW3 problem), meaning the OCRS guarantees to accept a fraction of at least $1/4$ of the fluid benchmark. However, note that this is not the best for knapsack OCRS. A tighter analysis yields an approximate guarantee of $\frac{1}{3+e^{-2}} \approx 31.9\%$.

2 Static Assortment Calendar via Prophet Inequality

2.1 Model

We now consider a different setting from the previous type-based models. In this model, there are no dynamic customer types arriving over time. Instead, at each time step t, we have a choice model that specifies the probability with which each product is selected from the assortment offered. Here, there are no types and we cannot even define OFF. We just have a choice function for time t that says the probability of it being chosen.

For time $t \in [T]$:

1. **Products:** $j \in [n]$ with prices $r_j \geq 0$.
2. **Resources:** $i \in [m]$ with starting inventory $k_i \in \mathbb{N}$
3. **Unit Consumption:** Each product j consumes 1 unit of *single resource* $i_j \in [m]$

4. **Known choice function:** $\phi_t(j, S)$ for each timestep t, where:

   - we assume *substitutability*, i.e. adding more products to an assortment should not increase the likelihood of selling a particular product.

   $$\forall t, \forall j \in S \subseteq S', \quad \phi_t(j, S) \geq \phi_t(j, S')$$

   - we can restrict assortment at time t to $\mathcal{S}_t$,

   $$\mathcal{S}_t = \{S \subseteq [n] : |S| \leq k\}$$

For simplicity, this model focuses on unit consumptions and does not include generalized knapsack-style sizes (though such extensions are possible). Each resource may support multiple products with varying prices. Having multiple products associated with the same resource allows us to model the idea that a single resource can be sold at different prices through different product offerings. This variability becomes important for capturing richer assortment and pricing strategies.

Static Assortment Calendar problem is you pre-commit to assortments $S_1, \cdots, S_T$ ahead of time. Once chosen, these assortments are fixed and cannot be adapted based on realized sales or remaining inventory. You can think of this as scheduling assortments over a long horizon (such as months of a selling season), where products share a common global inventory, and decisions must be made in advance.

Now, lets write the $\mathsf{FLU}(\mathcal{I})$ where, $x_t(S)$ is the probability of offering assortment S at time t:

$$\begin{aligned}
\underbrace{\mathsf{FLU}(\mathcal{I})}_{\text{“CDLP”}} = \max \quad & \sum_{t,S} x_t(S) \sum_{j \in S} r_j \phi_t(j, S) \\
\text{s.t.} \quad & \sum_{t,S} x_t(S) \sum_{j : i_j = i, j \in S} \phi_t(j, S) \leq k_i \quad \forall i \quad (\alpha_i) \quad (\textcolor{blue}{\ast}) \\
& \sum_{S} x_t(S) = 1 \quad \forall t \quad (\beta_t) \quad (\heartsuit) \\
& x_t(S) \geq 0 \quad \forall t, S
\end{aligned}$$

This LP is called the **Choice-Based Deterministic Linear Program (CDLP)** because it captures the demand through choice probabilities and relaxes the problem into a deterministic fractional form. One challenge with the fluid LP (CDLP) is that the number of possible assortments $S \subseteq [n]$ is exponential in n. Therefore, the LP has exponentially many variables and constraints, which raises the natural question: How do we solve such a large LP efficiently?

The dual of the above can be written as:

$$\begin{aligned}
\min \quad & \sum_i k_i \alpha_i + \sum_t \beta_t \\
\text{s.t.} \quad & \beta_t + \sum_{j \in S} \alpha_{ij} \phi_t(j, S) \geq \sum_{j \in S} r_j \phi_t(j, S) \quad \forall t, S \\
& \alpha_i \geq 0 \\
& \beta_t \geq \max_S \sum_{j \in S} (r_j - \alpha_{ij}) \phi_t(j, S) \quad \text{(assortment optimization)}
\end{aligned}$$

We then solve the above primal and dual together using ellipsoid method or typically column generation. Once the dual is solved, we check whether the constraints are valid or not, if not, we add in the variable and try again.

Unlike previous models where the FLU upper-bounded the OFF optimal, here there is no natural offline benchmark due to the static calendar structure and fixed pre-commitment. However, the FLU still provides a meaningful upper bound on achievable revenue. This makes it valuable for designing approximate algorithms.

2.2 Algorithm

Algorithm for Static Assortment Calendar (with thresholds)

1. Solve CDLP, let $x_t(S)$ refer to optimal solution.
2. Define

$$\text{prob of } j \text{ being chosen at time } t \text{ from FLU:} \quad x_{tj} = \sum_{S: j \in S} x_t(S)\phi_t(j,S)$$

$$\text{prob of } j \text{ being chosen at any time from FLU:} \quad x_j = \sum_t x_{tj}$$

$$\text{benchmark from FLU:} \quad \mathsf{FLU}(\mathcal{I}) = \sum_j r_j x_j$$

3. Draw a random assortment $S_t \sim (x_t(S))_S$ independently for each $t \in [T]$ (which is valid by constraint (♥)).
4. Recall that product j consumes one unit of resource i_j. Here, set thresholds based on *fluid revenue per resource unit:*

$$\tau_i := \sum_{j: i_j = i} \frac{r_j x_j}{2k_i} \quad (\star) \quad \text{"half of fluid revenue per unit from resource } i\text{"}$$

That is, $\mathsf{FLU}(\mathcal{I}) = \sum_i 2k_i\tau_i$.

5. Offer:

$$S_t \setminus \{j : r_j < \tau_{i_j}\}.$$

That is, given an assortment from the sampling, we exclude products whose potential revenue generated is lower than the threshold with respect to the resources they consume.

2.3 Analysis

In this section, we will show that the above algorithm guarantees at least half of the expected revenue from FLU, i.e.,

$$\mathcal{P}(\mathcal{I}) \geq \frac{\mathsf{FLU}(\mathcal{I})}{2}.$$

We start by defining some random variables:

- $X_{t,j}$: Indicator random variable whether product j is sold at time t under the algorithm.
- $N_t(i)$: Number of resource i sold by end of time t.

The expected total revenue on an instance $\mathcal{I}$ can be written as:

$$\begin{aligned}
\mathcal{P}(\mathcal{I}) &= \sum_t \sum_j (r_j - \tau_{i_j} + \tau_{i_j}) \mathbb{E}[X_{t,j}] \\
&= \sum_t \sum_j (r_j - \tau_{i_j})^+ \mathbb{E}[X_{t,j}] + \sum_t \sum_j \tau_{i_j} \mathbb{E}[X_{t,j}] \qquad \text{(offers product } j \text{ only when its price is above } \tau_{i_j}) \\
&= \sum_{t,j} [r_j - \tau_{i_j}]^+ \mathbb{E}[X_{t,j}] + \sum_i \tau_i \underbrace{\sum_t \sum_{j:i_j=i} \mathbb{E}[X_{t,j}]}_{=\mathbb{E}[N_T(i)]} \\
&\geq \sum_{t,j} [r_j - \tau_{i_j}]^+ x_{tj} \mathbb{P}[N_{t-1}(i_j) < k_{i_j}] + \sum_i \tau_i \sum_t \sum_{j:i_j=i} \mathbb{E}[X_{t,j}] \qquad (\dagger)
\end{aligned}$$

Before we continue our proof for the revenue guarantee, we examine two types of substitutions in assortments when some products are stocked out:

1. **Dynamic Substitution:** If a product stocked out before time t, it is removed from the assortment before the customer sees it. Customer t sees

   $$S_t \setminus \{j : r_j < \tau_{i_j}\} \setminus \{j : N_{t-1}(i_j) = k_{i_j}\}$$

   and chooses among the remaining available products that have not stocked out!!

2. **Static Substitution:** The assortment S_t remains fixed regardless of stock levels. Customer t still sees

   $$S_t\{j : r_j < \tau_{i_j}\}$$

   , but if they choose an item that has stocked out, no sale occurs.

Under either substitution model, we argue that we can still earn at least half of the expected revenue from FLU. We consider

$$\begin{aligned}
X_{t,j} &= \sum_{S:j\in S} \mathbb{1}(S_t = S)\mathbb{1}(N_{t-1}(i_j) < k_i)\mathbb{1}(\text{t chooses j}) \\
\mathbb{E}[X_{t,j}] &= \sum_{S:j\in S} x_t(S)\mathbb{P}[N_{t-1}(i_j) < k_i] \underbrace{\mathbb{P}[\text{t chooses j} \,|N_{t-1}(i_j) < k_i]}_{\geq \phi_t(j,S) \text{ under either substitution model due to the substitutability assumption}} \\
&\geq \mathbb{P}[N_{t-1}(i_j) < k_i] \sum_{S:j\in S} x_t(S)\phi_t(j,S) \\
&= x_{tj}\mathbb{P}[N_{t-1}(i_j) < k_i].
\end{aligned}$$

Now, assuming $r_j \geq \tau_{i_j}$, from equation (†), we have:

$$\begin{aligned}
\mathcal{P}(\mathcal{I}) &\geq \sum_i \left(\mathbb{P}[N_T(i) < k_i] \sum_{j:i_j=i} [r_j - \tau_{i_j}]^+ \sum_t x_{tj} + \tau_i \mathbb{E}[N_T(i)] \right) \\
&\geq \sum_i \left(\mathbb{P}[N_T(i) < k_i] \sum_{j:i_j=i} (r_j \sum_t x_{tj} - \tau_i \sum_t x_{tj}) + \tau_i \mathbb{E}[N_T(i)] \right) \\
&= \sum_i \left(\mathbb{P}[N_T(i) < k_i] \left(\underbrace{\sum_{j:i_j=i} r_j x_j}_{=2k_i\tau_i \text{ by } (\star)} - \tau_i \underbrace{\sum_{j:i_j=i} \sum_t x_{tj}}_{\leq k_i \text{ by } (\ast)} \right) + \tau_i \mathbb{E}[N_T(i)] \right) \\
&\geq \sum_i \left(\mathbb{P}[N_T(i) < k_i](2k_i\tau_i - k_i\tau_i) + \tau_i \mathbb{E}[N_T(i)] \right) \\
&= \sum_i \left(\mathbb{P}[N_T(i) < k_i] k_i \tau_i + k_i \tau_i \mathbb{P}[N_T(i) = k_i] \right) \\
&= \sum_i k_i \tau_i \\
&= \frac{\mathsf{FLU}(\mathcal{I})}{2}.
\end{aligned}$$

3 Online Personalized Assortment via Online Matching

3.1 Model

Consider an instance $\mathcal{I}$ with:

1. **Customer:** $t \in [T]$ arrives sequentially.
2. **Items:** $i \in [m]$ with starting inventory of $k_i \in \mathbb{N}$ units and prices $r_i \geq 0$.
3. **Choice function:** $\phi_t(i, S)$ for each customer t revealed at time t.
   - Assuming *substitutability*:

     $$\forall t, \forall j \in S \subseteq S', \quad \phi_t(j, S) \geq \phi_t(j, S').$$

     This assumption is crucial; otherwise, the algorithm must preserve items in very specific ways (e.g., keeping the same number of bread and butter).
   - We can also restrict the assortment at time t to $\mathcal{S}_t$,

     $$\mathcal{S}_t = \{S \subseteq [n] : |S| \leq k\}$$

Note here that we can no longer solve the CDLP in advance to get the threshold since the choice function is revealed at each time t.

Again, $X_t(S)$ is the probability of offering assortment S at time t. We write the $\mathsf{FLU}(\mathcal{I})$:

$$\begin{aligned}
\underbrace{\mathsf{FLU}(\mathcal{I})}_{\text{“CDLP”}} = \max \quad & \sum_{t,S} X_t(S) \sum_{i \in S} r_i \phi_t(i,S) \\
\text{s.t.} \quad & \sum_{t,S} X_t(S) \sum_{i \in S} \phi_t(i,S) \le k_i \quad \forall i \quad (\alpha_i) \\
& \sum_{S} X_t(S) = 1 \quad \forall t \quad (\beta_t) \\
& X_t(S) \ge 0 \quad \forall t, S
\end{aligned}$$

The dual of the above can be written as:

$$\begin{aligned}
\min \quad & \sum_i k_i \alpha_i + \sum_t \beta_t \\
\text{s.t.} \quad & \beta_t + \sum_{i \in S} \alpha_i \phi_t(i,S) \ge \sum_{i \in S} r_i \phi_t(i,S) \quad \forall t, S \qquad (\bullet) \\
& \alpha_i \ge 0
\end{aligned}$$

3.2 Algorithm

Online Personalized Assortment Algorithm

For customer t, we offer the following assortment:

$$\arg\max_S \sum_{i \in S} r_i \big(1 - \underbrace{p\big(\frac{N_{t-1}(i)}{k_i}\big)}_{\text{shadow price}}\big) \phi_t(i,S).$$

Note that the shadow price above is used for decision-making and is not real.

- Assume p(1) =1
- Substitutability implies items that have stocked out are never offered, i.e. the algorithm is valid.

This algorithm is similar to the algorithm for online $k-$matching problem in the sense that we use the number of remaining items as a shadow price to de-prioritize items running low on stocks.

3.3 Analysis

In this section, we will try to show that

$$\mathcal{P}(\mathcal{I}) \ge \left(1 - \left(1 + \frac{1}{k}\right)^{-k}\right) \mathsf{FLU}(\mathcal{I})$$

where $k = \min_i k_i$.

Here, we consider the dual variables α_i, β_t where $\alpha_i = \mathbb{E}[A_i]$, $\beta_t = \mathbb{E}[B_t]$, and

$$A_i = \frac{1}{k_i} \sum_{N=0}^{N_T(i)-1} r_i p\left(\frac{N}{k_i}\right)$$

$$B_t = \max_S \sum_{i \in S} r_i \big(1 - p\big(\frac{N_{t-1}(i)}{k_i}\big)\big)\phi_t(i, S).$$

Next, we want to show:

(a) The expected revenue from the algorithm is equal to the dual objective, i.e., $\mathcal{P}(\mathcal{I}) =$ Dual Obj.

(b) We then approximate dual feasibility, and show that $(\alpha_i/c, \beta_t/c)$ is dual feasible for some constant c.

Part (a): Dual Objective

At a high level, we want to show that *the expected gain in revenue is equal to the dual gain.* We condition on any values of $(N_{t-1}(i))_{i=1}^M$ and suppose S_t is offered at time t :

- Algorithm's expected revenue from customer t:

$$\sum_{i \in S_t} r_i \phi_t(i, S).$$

- Expected change in dual objective :

$$\begin{aligned}
\beta_t &= \sum_{i \in S_t} r_i \big(1 - p\big(\frac{N_{t-1}(i)}{k_i}\big)\big)\phi_t(i, S) \\
\mathbb{E}[\Delta A_i | (N_{t-1}(i))_i] &= \sum_{i \in S_t} \frac{1}{k_i} r_i p\big(\frac{N_{t-1}(i)}{k_i}\big) \\
\beta_t + k_i \mathbb{E}[\Delta A_i | (N_{t-1}(i))_i] &= \sum_{i \in S_t} r_i \phi_t(i, S)
\end{aligned}$$

Since the expected change in the dual objective is equal to the expected gain in revenue from customer t under any realization $(N_{t-1}(i))_{i=1}^M$, we can conclude that the expected revenue from the algorithm is equal the dual objective with the dual variables (α_i, β_t).

Part (b): Dual Feasibility

Next, we will check the dual feasibility (α_i, β_t). For any t, S, we know that

$$B_t \geq \sum_{i \in S} r_i \big(1 - p\big(\frac{N_{t-1}(i)}{k_i}\big)\big)\phi_t(i, S) \qquad \text{(by being the maximum over } S\text{)}$$

$$A_i \geq \frac{r_i}{k_i} \sum_{N=0}^{N_{t-1}(i)-1} p\left(\frac{N}{k_i}\right)$$

Recall the dual constraint in (•), we have

$$B_t + \sum_{i \in S} A_i \phi_t(i,S) \geq \sum_{i \in S} r_i \phi_t(i,S) \Bigg(\underbrace{1 - p(\frac{N_{t-1}(i)}{k_i}) + \frac{1}{k_i} \sum_{N=0}^{N_{t-1}(i)-1} p\left(\frac{N}{k_i}\right)}_{\text{We want this to be } \geq c(k_i) \text{ for any value of } N_{t-1}(i)} \Bigg)$$
$$\geq \min_i c(k_i) \sum_{i \in S} r_i \phi_t(i,S) \qquad (\star).$$

If the above holds for every realization, then we can conclude that $(\frac{\alpha_i}{\min_i c(k_i)}, \frac{\beta_t}{\min_i c(k_i)})$ would be dual feasible and, by weak duality,

$$\mathsf{ALG}(\mathcal{I}) = \sum_i \alpha_i + \sum_t \beta_t \geq \left(\min_i c(k_i)\right) \mathsf{FLU}(\mathcal{I}).$$

Next, we will show how to choose $p(\cdot)$ for $c(\cdot)$ so we achieve the result in $(\star)$.
Recall that setting $p_l = \left(1 + \frac{1}{k}\right)^{-l} \; \forall l \in [k] \cup \{0\}$ satisfies:

$$1 - p_l + \frac{1}{k} \sum_{l' > l} p_{l'} \geq 1 - \left(1 + \frac{1}{k}\right)^{-k}, \quad \forall l \in [k] \cup \{0\}. \tag{1}$$

We want to set $p(\cdot)$ such that

$$p\left(\frac{N}{k_i}\right) = p_{k_i - N}, \quad \forall N \in [k_i] \cup \{0\}.$$

As a result, we have

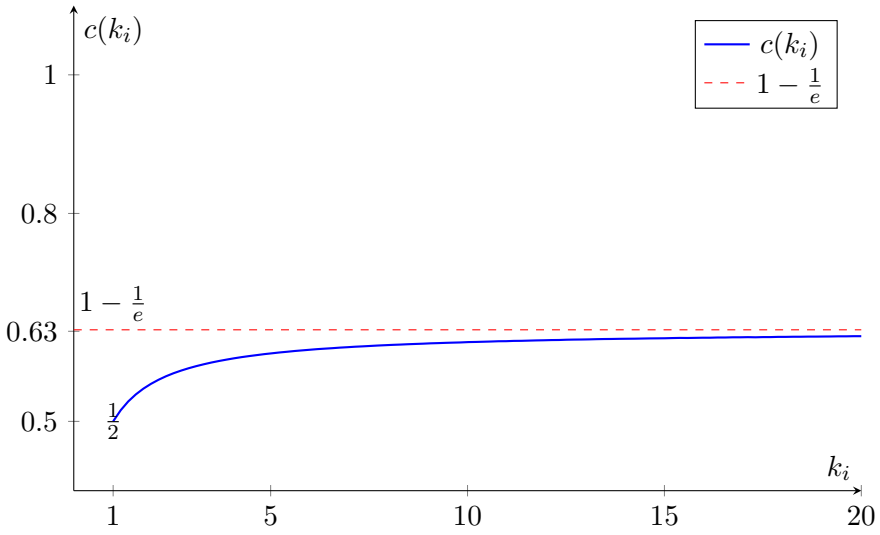
$$\underbrace{1 - p\left(\frac{N_{t-1}(i)}{k_i}\right) + \frac{1}{k_i} \sum_{N=0}^{N_{t-1}(i)-1} p\left(\frac{N}{k_i}\right)}_{\text{Let } N = N_{t-1}(i)} = 1 - p_{k_i - N} + \frac{1}{k_i} \sum_{N'=0}^{N-1} p_{k_i - N'}.$$

Letting $l = k_i - N$ and using eq. (1) gives

$$1 - p_l + \frac{1}{k_i} \sum_{l' > l} p_{l'} \geq 1 - \left(1 + \frac{1}{k_i}\right)^{-k_i} = c(k_i)$$

and we set

$$c(k) = 1 - \left(1 + \frac{1}{k}\right)^{-k}.$$



4 Correlation & Interpretation of FLU

In this section, we present another interpretation of FLU as a problem of choosing a correlation. Consider the following online allocation problem where we can pick at most one item ($k = 1$) and the value distribution $\vec{F}$ is corresponding to

$$V_1 = \begin{cases} 2, & \text{w.p. } \frac{1}{2} \\ 1, & \text{w.p. } \frac{1}{2} \end{cases}$$

and V_2 is i.i.d. We have derived the optimal offline policy with $\mathsf{OFF}(\vec{F}) = 1.75$. That is, the optimal offline algorithm gets a value of 2 w.p. $\frac{3}{4}$ and 1 w.p. $\frac{1}{4}$. The fluid relaxation from this instance yields $\mathsf{FLU}(\vec{F}) = 2$. Here, the fluid implies that it always get a value of 2, by *correlating* the two valuations. We will formalize this notion of correlating valuation distributions and show that, under an ideal polytope, FLU, is the same as finding the best correlating!

4.1 Formalization of Coupling Problem

Here, we define:

1. **Agents:** $t \in [T]$ with valuations drawn from $(F_t)_{t\in[T]}$;
2. **Feasible Family:** $\mathcal{F}$ (e.g., set of ways of selecting at most k out of T agents).
3. **Set of Selected Agents:** $S \subseteq [T]$ that must satisfy $S \in \mathcal{F}$.

Now, the expected reward from the optimal offline policy can be written as:

$$\mathsf{OFF}(\vec{F}) = \mathbb{E}_{\vec{V}\sim\vec{F}}\left[\max_{S\in\mathcal{F}} \sum_{t\in S} V_t\right].$$

This can be very general, for instance, $[T]$ can be a set of edges in the graph, $\mathcal{F}$ is the set of all matching, and we want to find a maximum matching.

Let $\Gamma(F_1, \ldots, F_T)$ be the space of all joint distributions over $(\mathbb{R}_{\geq 0})^T$ with marginals $F_1, \ldots, F_T$ for any $F_1, \ldots, F_T \in \Delta(\mathbb{R}_{\geq 0})$. We can think of these distributions as ways of *correlating* or *coupling*

these marginal distributions. Then, we claim the following relationship between the offline policy and the fluid relaxation:

Claim

Given an ideal polytope (i.e., $P = conv(\{\mathbb{1}_S : S \in F\})$, the best possible coupling is the same as FLU, that is,

$$\sup_{\vec{F} \in \Gamma(F_1,\ldots,F_T)} \mathsf{OFF}(\vec{F}) = \mathsf{FLU}(F_1, \ldots, F_T). \tag{2}$$

We note that the value of OFF depends on the correlation of the distributions while the FLU only depends on the marginals and not on the correlation. Basically, the claim is that if we choose the best way of correlating, we can recover FLU. We also note that FLU also depends on the polytope P. The larger P is, the more relaxed FLU becomes. So, the above claim might not be true once $P \supsetneq conv(\{\mathbb{1}_S : S \in F\}$.

We can prove this claim through LP formulations. For simplicity, we start with discrete-distribution settings:

- **Possible valuations:** $r_1, \ldots, r_n \geq 0$;
- **Marginal Distributions:** F_t is given by PMF $(f_{tj})_{j=1}^n$ satisfying $\sum_j f_{tj} = 1$ for all $t \in [T]$;

The *coupling problem* can be formulated as:

- **Type Vector:** $\vec{j} = [j_1, j_2, \ldots, j_T] \in [n]^T$; That is, there are n possible classes, and each class comes with a valuation. For $t \in [T]$, j_t indicates the type and the valuation of customer t.
- $x(\vec{j})$ is a probability that we choose type vector $\vec{j}$, which is the probability that $(V_t = r_{j_t} \quad \forall t)$.

Coupling Problem

Given $(F_1, \ldots, F_T)$, we can write the coupling problem as:

$$\begin{aligned}
\sup_{\vec{F} \in \Gamma(F_1,\ldots,F_T)} \mathsf{OFF}(\vec{F}) = \max_{x(\cdot)} & \sum_{\vec{j} \in [n]^T} x(\vec{j}) \cdot \left(\max_{S \in \mathcal{F}} \sum_{t \in S} r_{j_t} \right) && \text{(coupling objective)} \\
\text{s.t.} & \sum_{\vec{j}: j_t = J} x(\vec{j}) = f_{tJ} \quad \forall t \in [T], J \in [n] && \text{(coupling constraint)} \\
& x(\vec{j}) \geq 0 \quad \forall \vec{j} \in [n]^T.
\end{aligned}$$

That is, we try to find a coupling that *preserves* the marginals; and, for each realization of the type vector $\vec{j} \in [n]^T$ with probability $x(\vec{j})$, the maximum reward, i.e., $\max_{S \in \mathcal{F}} \sum_{t \in S} r_{j_t}$, is attained.

The fluid relaxation can be written as an LP as:

FLU

$$\begin{aligned}
\mathsf{FLU}(F_1, \cdots, F_T) = \max & \sum_{t,j} r_j z_{tj} \\
\text{s.t.} & \sum_{S \in \mathcal{F}} x(S) = 1, \\
& \sum_{S \in \mathcal{F}: t \in S} x(S) = x_t, \quad \forall t \\
& \sum_j z_{tj} \le x_t \quad \forall t \\
& 0 \le z_{tj} \le f_{tj}, \quad \forall t, j \\
& x(S) \ge 0 \quad \forall S \in \mathcal{F}
\end{aligned}$$

where

- $x(S) := \mathbb{P}[\text{select } S]$
- $x_t := \mathbb{P}[\text{customer } t \text{ is selected}]$
- $z_{tj} := \mathbb{P}[\text{customer } t \text{ is selected with valuation } r_j]$.

The constraint $z_{tj} \le f_{tj}$ implies that the chance of t selected when it is with type j is at most the chance that it is realized to be type j. One can show that these two LP formulations are provide the same optimal values!

4.2 Correlation Gap

Next, we define the **Correlation Gap** as:

$$\text{CorrGap} := \sup_{F_1, \cdots, F_T \in \Delta(\mathbb{R}_{\ge 0})} \frac{\sup_{\vec{F} \in \Gamma(F_1, \cdots, F_T)} \mathsf{OFF}(\vec{F})}{\mathsf{OFF}(\underbrace{F_1 \times \cdots \times F_T}_{\text{independent}})},$$

which is always at least 1. *The correlation gap demonstrates the weakness of using* FLU *as a benchmark!* If this gap is small, it means that FLU provides a bound close to OFF. Note that the correlation gap also depends on the feasibility family $\mathcal{F}$ considered.

Recall that our competitive ratio compared to FLU can be written as:

$$\begin{aligned}
\text{CompRatio} := \inf_{\vec{F} \in \Delta(\mathbb{R}_{\ge 0})^T} \frac{\sup_\pi \pi(\vec{F})}{\mathsf{FLU}(F_1, \cdots, F_T)} &\le \inf_{\vec{F} \in \Delta(\mathbb{R}_{\ge 0})^T} \frac{\mathsf{OFF}(\vec{F})}{\mathsf{FLU}(F_1, \cdots, F_T)} \\
&= \frac{1}{\text{CorrGap}}.
\end{aligned}$$

This inequality is still true even when we do not work on an ideal polytope as $\mathsf{FLU} \ge \sup_{\vec{F} \in \Gamma(F_1, \cdots, F_T)} \mathsf{OFF}(\vec{F})$.

Bounds on Correlation Gaps

Suppose $\mathcal{F} = \{S \subseteq [T] : |S| \leq k\}$. We start with an upper on the correlation gap for this setting. From the Prophet $k-$Secretary result, there exists a policy that guarantees:

$$\text{CorrGap} \leq \frac{1}{1 - e^{-k}\frac{k^k}{k!}} \approx \frac{1}{1 - \frac{1}{\sqrt{2\pi k}}}.$$

Next, we will derive a lower bound on CorrGap using the following hard instance:

$$V_t = \begin{cases} 1 & \text{w.p. } \frac{k}{T} \\ 0 & \text{otherwise} \end{cases} \qquad \forall t \in [T].$$

We can correlate these valuations so that $\mathsf{FLU} = k$. Moreover, we know that

$$\mathsf{OFF}(\underbrace{F_1 \times \cdots \times F_T}_{\text{independent distributions}}) = \mathbb{E}\left[\min\{\text{Binomial}\left(T, \frac{k}{T}\right), k\}\right]$$

since the optimal offline policy will just accept any non-zero realization until it reaches k, giving an expected reward of $\mathbb{E}\left[\min\{\text{Binomial}\left(T, \frac{k}{T}\right), k\}\right]$.

Therefore,

$$\frac{1}{\text{CorrGap}} = \inf_{\vec{F} \in \Delta(\mathbb{R}_{\geq 0})^T} \frac{\mathsf{OFF}(\vec{F})}{\mathsf{FLU}(F_1, \cdots, F_T)} \leq \frac{\mathbb{E}\left[\min\{\text{Binomial}\left(T, \frac{k}{T}\right), k\}\right]}{k}$$

Taking limit as $T \to \infty$, the right hand size is approaching $1 - e^{-k}\frac{k^k}{k!}$ and thus

$$\text{CorrGap} \geq \frac{1}{1 - e^{-k}\frac{k^k}{k!}}.$$

Bibliographical notes. The randomized rounding framework for online generalized assignment originates from Alaei et al. (2012), with the modern treatment using OCRS and the results for knapsack coming from Jiang et al. (2022). The static assortment calendar problem and analysis come from Ma et al. (2021). The online personalized assortment problem comes from Golrezaei et al. (2014), and the analysis presented here uses Kalyanasundaram and Pruhs (2000). Related to the final section on interpreting the Fluid relaxation using correlation, see Natarajan et al. (2009) for a reference on the coupling problem, and see Yan (2011) for a reference on the correlation gap of selecting k items.

References

S. Alaei, M. Hajiaghayi, and V. Liaghat. Online prophet-inequality matching with applications to ad allocation. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 18–35, 2012.

N. Golrezaei, H. Nazerzadeh, and P. Rusmevichientong. Real-time optimization of personalized assortments. *Management Science*, 60(6):1532–1551, 2014.

J. Jiang, W. Ma, and J. Zhang. Tight guarantees for multi-unit prophet inequalities and online stochastic knapsack. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1221–1246. SIAM, 2022.

B. Kalyanasundaram and K. R. Pruhs. An optimal deterministic algorithm for online b-matching. *Theoretical Computer Science*, 233(1-2):319–325, 2000.

W. Ma, D. Simchi-Levi, and J. Zhao. Dynamic pricing (and assortment) under a static calendar. *Management Science*, 67(4):2292–2313, 2021.

K. Natarajan, M. Song, and C.-P. Teo. Persistency model and its applications in choice modeling. *Management Science*, 55(3):453–469, 2009.

Q. Yan. Mechanism design via correlation gap. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 710–719. SIAM, 2011.