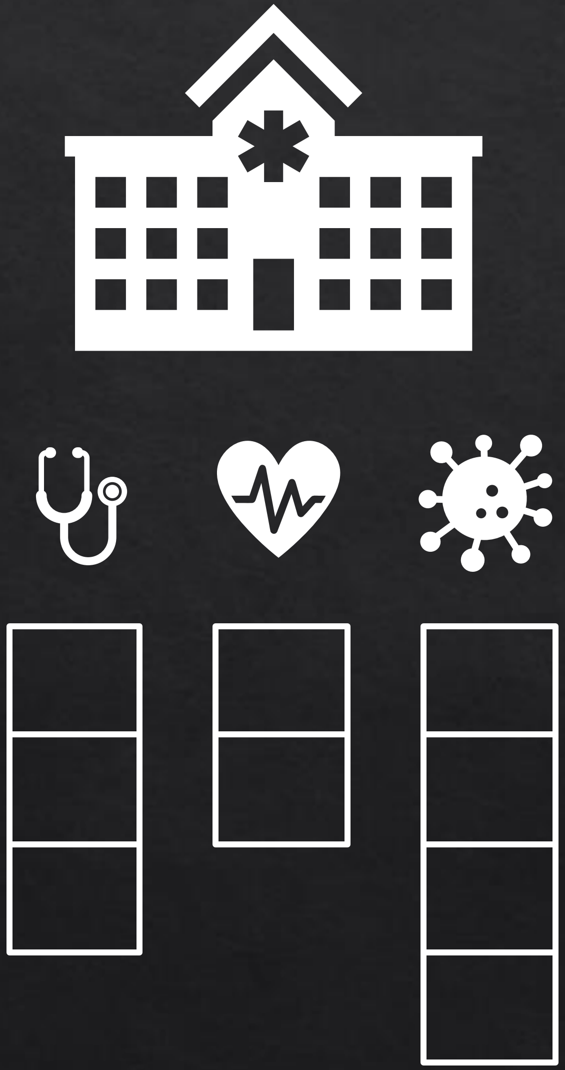
Online Allocation with Priorities and Quotas

Sid Banerjee

Matt Eichhorn

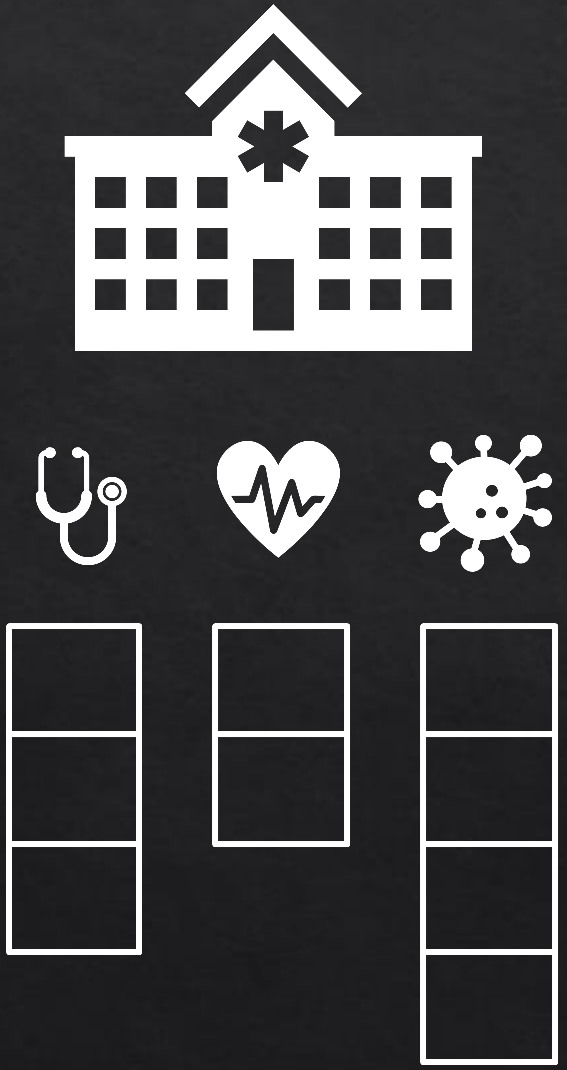
David Kempe

Motivation: Hospital Triage



1. At start of night shift, hospital has available rooms

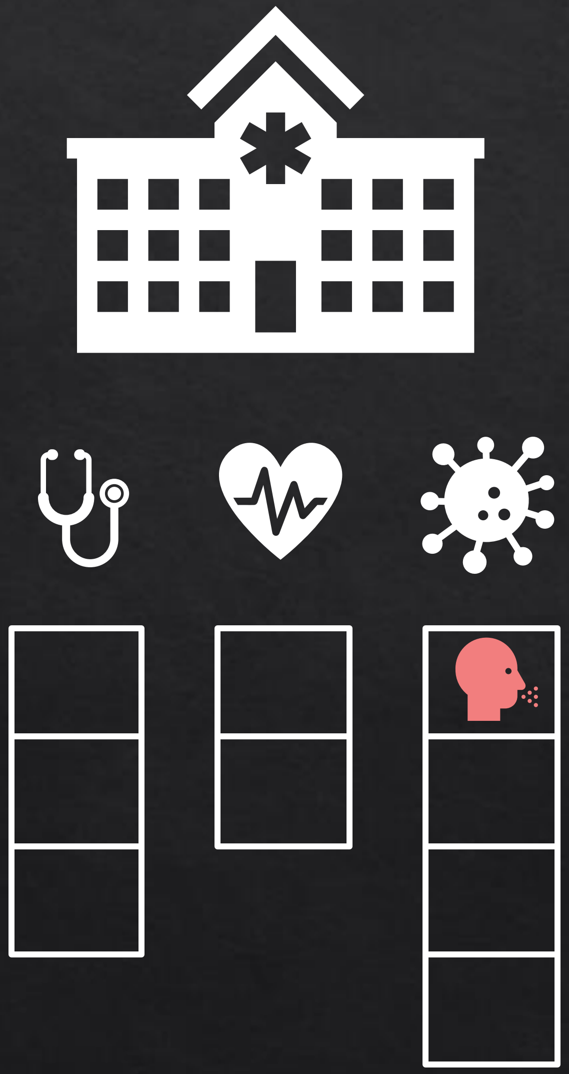
Motivation: Hospital Triage



1. At start of night shift, hospital has available rooms

2. Patients arrive sequentially at hospital

Motivation: Hospital Triage



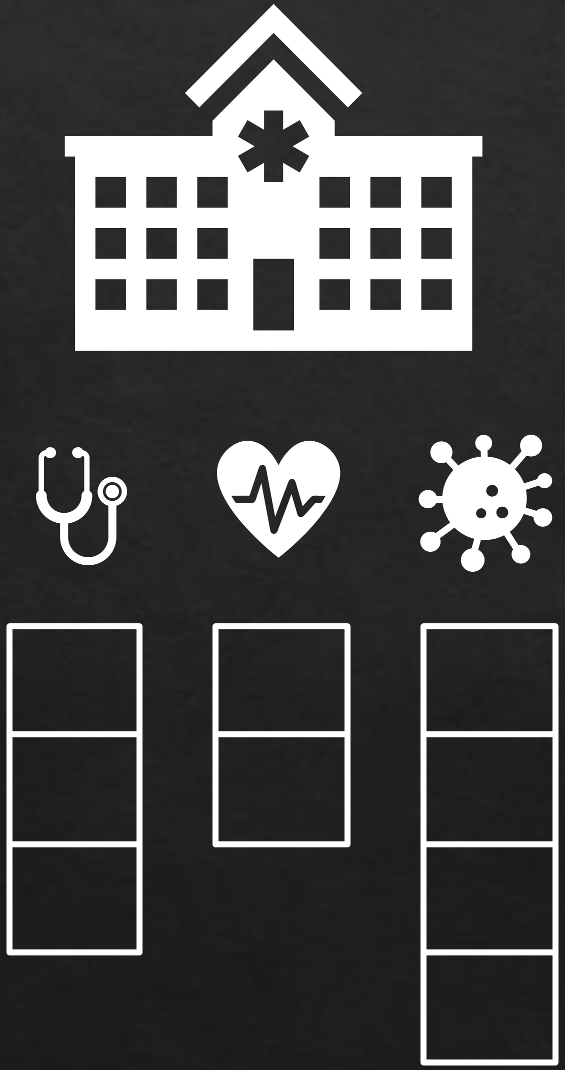
1. At start of night shift, hospital has available rooms

2. Patients arrive sequentially at hospital

3. Upon Arrival, patient either:

- Receives a room (for entire night)

Motivation: Hospital Triage

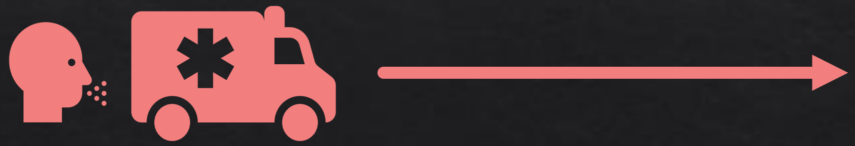


1. At start of night shift, hospital has available rooms

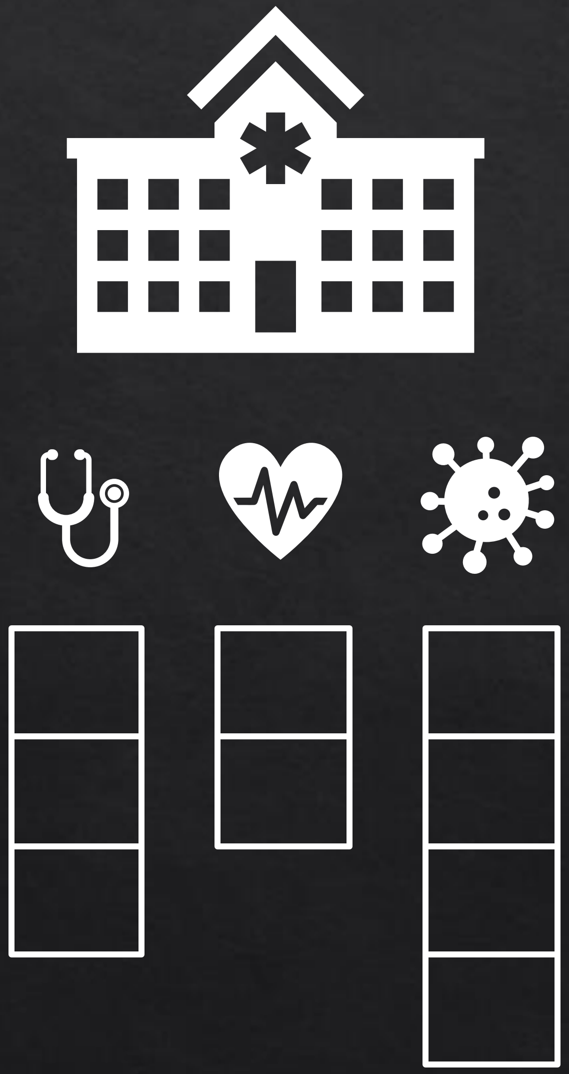
2. Patients arrive sequentially at hospital

3. Upon Arrival, patient either:

- Receives a room (for entire night)
- Is turned away



Motivation: Hospital Triage



1. At start of night shift, hospital has available rooms

2. Patients arrive sequentially at hospital

3. Upon Arrival, patient either:

   - Receives a room (for entire night)
   - Is turned away

4. How should hospital allocate rooms to:

   - Treat as many people as possible,
   - Never turn away the most critical patients?

Formalizing the Setting

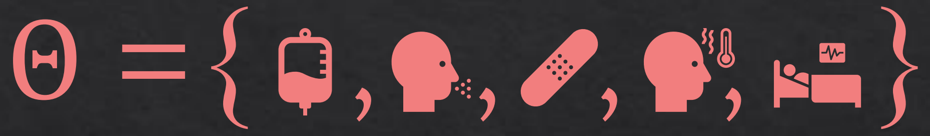
T = Number of arriving agents (<u>known</u>)

Formalizing the Setting

T = Number of arriving agents (known)

Agent Types: $\theta_t \in \Theta$

- Types drawn i.i.d. from known distribution P_Θ
- Each agent has unit demand for an allocation



Formalizing the Setting

T = Number of arriving agents (known)

Agent Types: $\theta_t \in \Theta$

- Types drawn i.i.d. from known distribution P_Θ
- Each agent has unit demand for an allocation

$\Theta = \{$  $\}$

Categories: $c \in C$ with

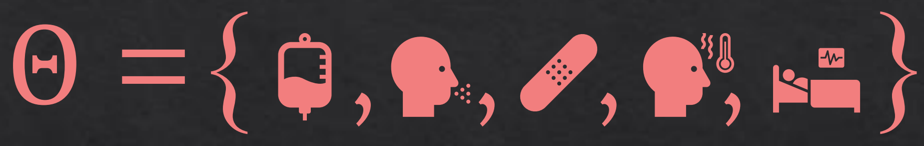
- Quota q_c of units to allocate
- Eligible Types $E_c \subseteq \Theta$
- Priority Order $\succcurlyeq_c$ over eligible types
  - $\theta \succcurlyeq_c \theta'$ means θ has priority over θ' in c

Formalizing the Setting

T = Number of arriving agents (known)

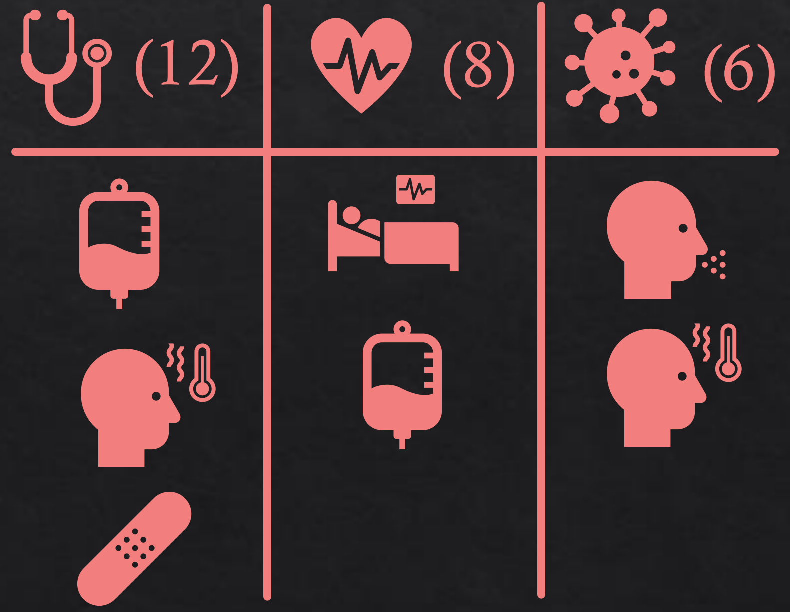
Agent Types: $\theta_t \in \Theta$

- Types drawn i.i.d. from known distribution P_Θ
- Each agent has unit demand for an allocation



Categories: $c \in C$ with

- Quota q_c of units to allocate
- Eligible Types $E_c \subseteq \Theta$
- Priority Order $\succcurlyeq_c$ over eligible types
  - $\theta \succcurlyeq_c \theta'$ means θ has priority over θ' in c



The Online Allocation Problem

When agent θ_t arrives, select their allocation from $C \cup \{\perp\}$

The Online Allocation Problem

When agent θ_t arrives, select their allocation from $C \cup \{\perp\}$

Allocations Should:

- Respect Eligibility: Only give categories types they approve of
- Respect Quotas: Do not over-allocate a category

The Online Allocation Problem

When agent θ_t arrives, select their allocation from $C \cup \{\perp\}$

Allocations Should:

- Respect Eligibility: Only give categories types they approve of
- Respect Quotas: Do not over-allocate a category
- Respect Priorities: If an agent is allocated in category c, all higher-priority agents in c should also be allocated

The Online Allocation Problem

When agent θ_t arrives, select their allocation from $C \cup \{\perp\}$

Allocations Should:

- Respect Eligibility: Only give categories types they approve of
- Respect Quotas: Do not over-allocate a category
- Respect Priorities: If an agent is allocated in category c, all higher priority agents in c should also be allocated
- Be Pareto Efficient: Allocate to the maximal extent possible

Priorities as a Constraint

Only allocate if it cannot cause a priority violation, maximize number of allocations

Priorities as a Constraint

Only allocate if it cannot cause a priority violation, maximize number of allocations

$$\frac{\alpha_{\ (T/2)}}{\begin{matrix} a \\ b \\ c \end{matrix}}$$

$$p_a = p_b = p_c = \frac{1}{3}$$

Priorities as a Constraint

Only allocate if it cannot cause a priority violation, maximize number of allocations

$$\alpha\ (T/2)$$

$$a$$

$$b$$

$$c$$

$$p_a = p_b = p_c = \frac{1}{3}$$

- **At Start:**
  - Only allocate to a agents
  - Must guard against possibility of all as in future

Priorities as a Constraint

Only allocate if it cannot cause a priority violation, maximize number of allocations

α $(T/2)$

a

b

c

$p_a = p_b = p_c = \frac{1}{3}$

- **At Start:**
  - Only allocate to a agents
  - Must guard against possibility of all as in future
- **After $\approx \frac{3T}{4}$ Rounds:**
  - Have space for all remaining agents
  - Can start accepting b agents
  - Cannot accept c agents (b agents were rejected)
  - $\approx \frac{T}{12}$ quota goes unfilled

Priorities as a Constraint

Theorem: If our allocations must be priority respecting in hindsight, then any online policy must incur $\Omega(T)$ loss in efficiency in the worst case.

This linear scaling in the loss is undesirable (ideally, we would like constant).

Priorities as a Constraint

Theorem: If our allocations must be priority respecting in hindsight, then any online policy must incur $\Omega(T)$ loss in efficiency in the worst case.

This linear scaling in the loss is undesirable (ideally, we would like constant).

New Idea: Multi-Objective Optimization Problem

- Priority Loss (Δ_P): Number of priority violations

  $$\#\{\, t \,:\, t \text{ unallocated, but there is } t' \text{ with } \theta_{t'} \prec_c \theta_t \text{ allocated in } c \,\}$$

- Efficiency Loss (Δ_E):

  # Allocations in Offline Optimum - # Allocations made by Algorithm

Aside: Computing the Offline Optimum

LP in variables $x_{\theta,c}$ = # agents of type θ allocated in category c

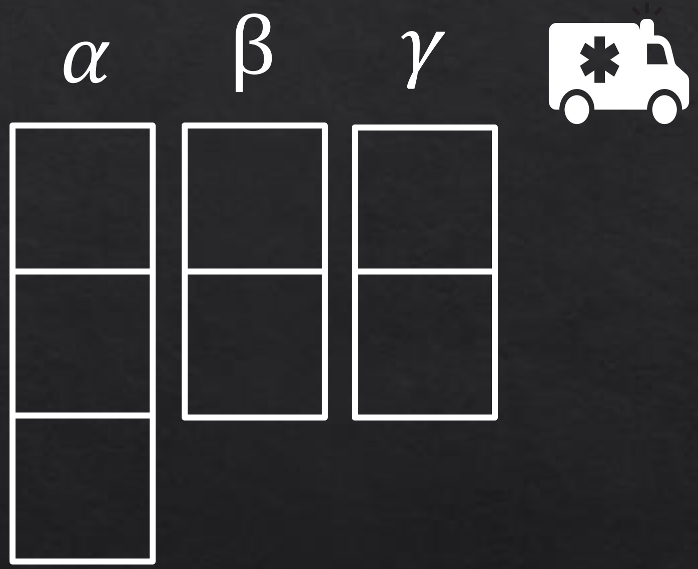
Maximize	$\sum_{\theta\in\Theta}\sum_{c\in C}(1-\delta_{\theta,c})x_{\theta,c}$		**(Pareto efficiency)**
Subject to	$\sum_{\theta\in\Theta} x_{\theta,c} \le q_c$	$\forall\, c \in C$	**(quotas)**
	$\sum_{c\in C} x_{\theta,c} \le N_\theta$	$\forall\, \theta \in \Theta$	**(unit demand)**
	$x_{\theta,c} = 0$	$\forall\, c \in C, \theta \notin E_c$	**(eligibility)**
	$x_{\theta,c} \ge 0$	$\forall\, \theta \in \Theta, c \in C$	

$\delta_{\theta,c}$ chosen so $\delta_{\theta,c} \le \delta_{\theta',c} \Leftrightarrow \theta \succcurlyeq_c \theta'$ **(priorities)**

Example Execution

$\Theta = \{a,b,c,d,e\}$ $\quad T = 9$

α (3)	β (2)	γ (2)
a	a	e
b	d	d
c		

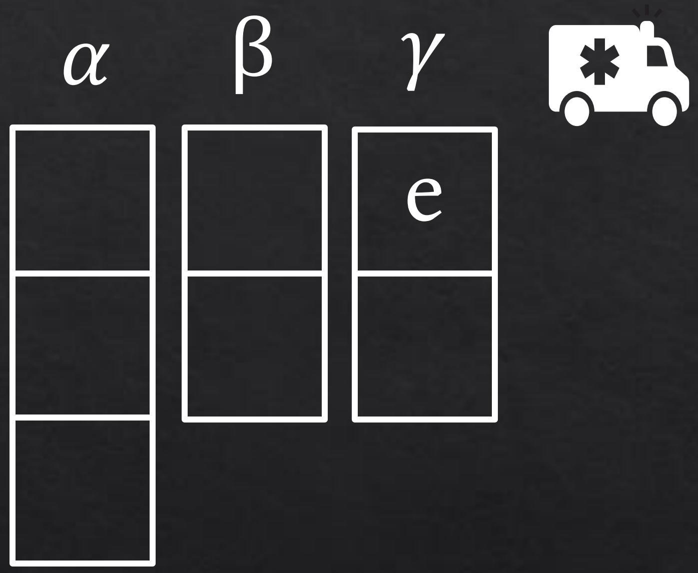


Arrival Sequence:

Example Execution

$\Theta = \{a,b,c,d,e\}$ $T = 9$

α (3)	β (2)	γ (2)
a	a	e
b	d	d
c		

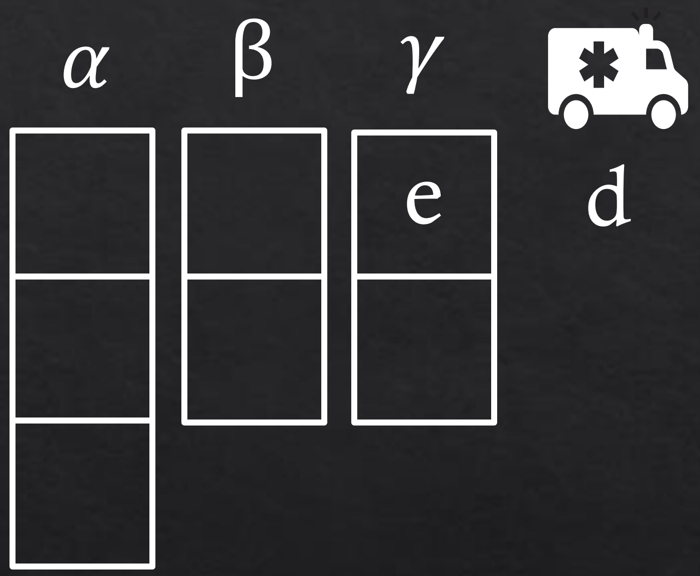


Arrival Sequence: e

Example Execution

$\Theta = \{a,b,c,d,e\}$ $\quad T = 9$

α (3)	β (2)	γ (2)
a	a	e
b	d	d
c		

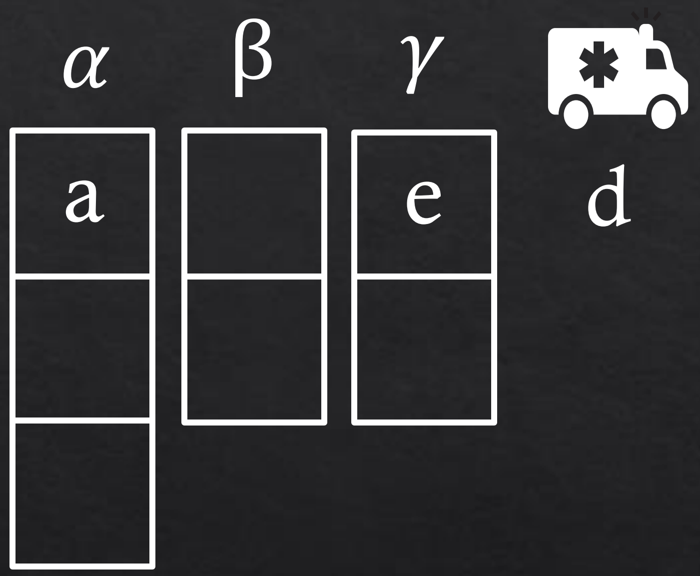


Arrival Sequence: e d

Example Execution

$\Theta = \{a,b,c,d,e\}$ $T = 9$

α (3)	β (2)	γ (2)
a	a	e
b	d	d
c		

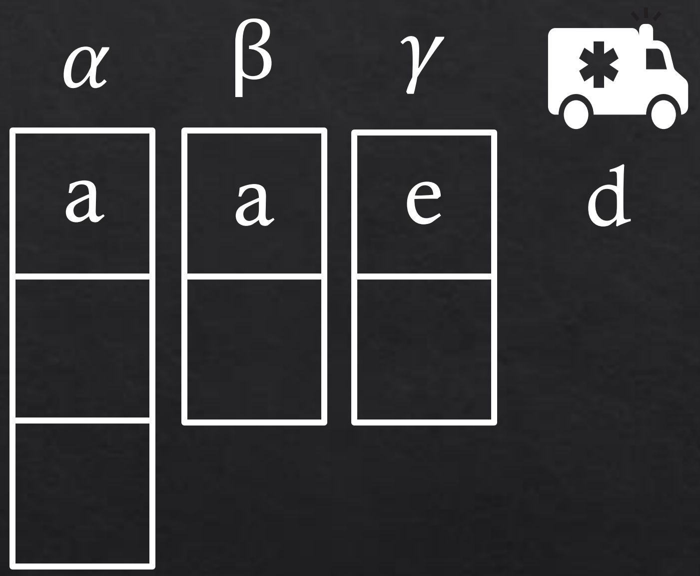


Arrival Sequence: e d a

Example Execution

$\Theta = \{a,b,c,d,e\}$ $T = 9$

α (3)	β (2)	γ (2)
a	a	e
b	d	d
c		

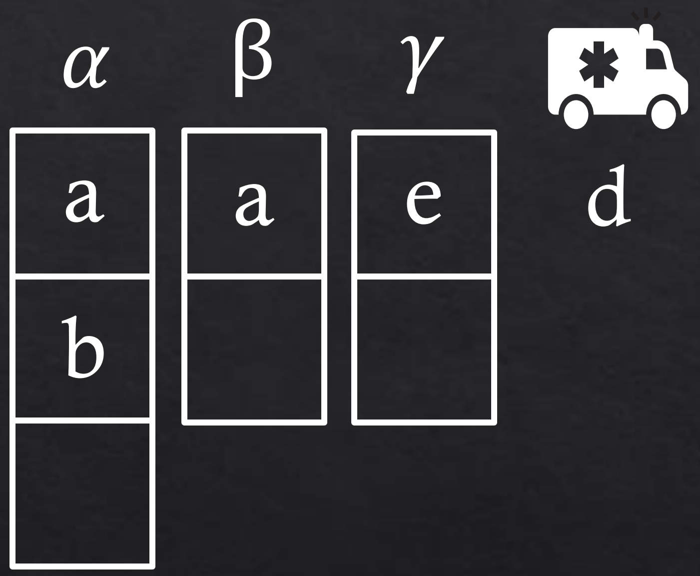


Arrival Sequence: e d a a

Example Execution

$\Theta = \{a,b,c,d,e\}$ $T = 9$

α (3)	β (2)	γ (2)
a	a	e
b	d	d
c		

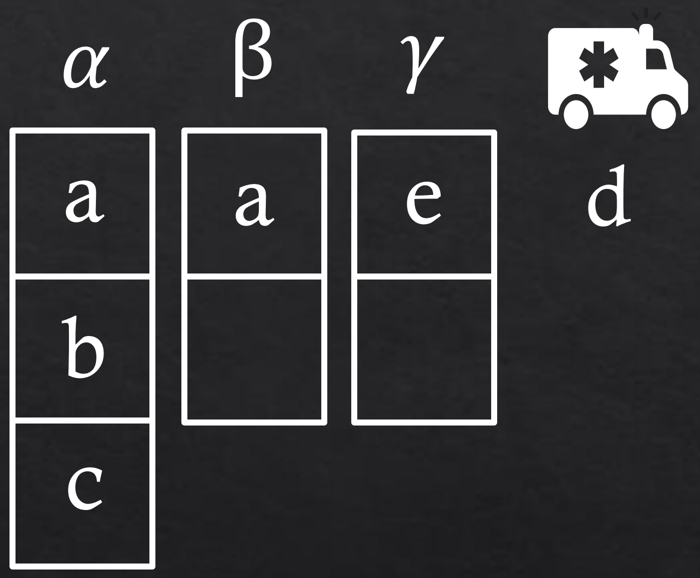


Arrival Sequence: e d a a b

Example Execution

$\Theta = \{a, b, c, d, e\}$ $T = 9$

α (3)	β (2)	γ (2)
a	a	e
b	d	d
c		



Arrival Sequence: e d a a b c

Example Execution

$\Theta = \{a,b,c,d,e\}$ $\quad T = 9$

α (3)	β (2)	γ (2)
a	a	e
b	d	d
c		

α	β	γ	
a	a	e	d
b	a		
c			

Arrival Sequence: e d a a b c a

Example Execution

$\Theta = \{a,b,c,d,e\}$ $T = 9$

α (3)	β (2)	γ (2)
a	a	e
b	d	d
c		

α	β	γ	
a	a	e	d
b	a		a
c			

Arrival Sequence: e d a a b c a a

Example Execution

$\Theta = \{a,b,c,d,e\}$ $\quad T = 9$

α (3)	β (2)	γ (2)
a	a	e
b	d	d
c		

α	β	γ	
a	a	e	d
b	a		a
c			b

Arrival Sequence: e d a a b c a a b

Example Execution

$\Theta = \{a,b,c,d,e\}$ $T = 9$

α (3)	β (2)	γ (2)
a	a	e
b	d	d
c		

α	β	γ	
a	a	e	d
b	a		a
c			b

Optimal Allocation:

α	β	γ	
a	a	e	c
b	a	d	b
a			

Arrival Sequence: e d a a b c a a b

Example Execution

$\Theta = \{a,b,c,d,e\}$ $T = 9$

α (3)	β (2)	γ (2)
a	a	e
b	d	d
c		

α	β	γ	🚑
a	a	e	d
b	a		a
c			b

Optimal Allocation:

α	β	γ	🚑
a	a	e	c
b	a	d	b
a			

Arrival Sequence: e d a a b c a a b

$\Delta_P = 2 \quad \Delta_E = 1$

Achieving Constant Expected Loss

Theorem: There is an online allocation policy for which the expected sum of efficiency and priority loss is at most:

$$\mathbb{E}[\Delta_P + \Delta_E] \leq \frac{|\Theta|^5 \, (|C|+1)^4}{{p_{min}}^4}$$

Notably, the loss is constant with respect to the instance size (T and q)

Achieving Constant Expected Loss

Theorem: There is an online allocation policy for which the expected sum of efficiency and priority loss is at most:

$$\mathbb{E}[\Delta_P + \Delta_E] \leq \frac{|\Theta|^5 \, (|C|+1)^4}{{p_{min}}^4}$$

Notably, the loss is constant with respect to the instance size (T and q)

Algorithm Idea: Use LP Sensitivity and Compensated Coupling

The Interim LP in Round t

$$P_\delta(N_\theta[t], E_c[t], q_c[t])$$

$N[t] =$ (expected) # of future arrivals of each type θ

$E[t]$ = current eligible agents in each category c

$q[t]$ = remaining quota in each category c

Maximize $\quad \sum_{\theta \in \Theta} \sum_{c \in C} (1 - \delta_{\theta,c}) x_{\theta,c}$

Subject to

$$\sum_{\theta \in \Theta} x_{\theta,c} \leq q_c[t] \qquad \forall\, c \in C$$

$$\sum_{c \in C} x_{\theta,c} \leq N_\theta[t] \qquad \forall\, \theta \in \Theta$$

$$x_{\theta,c} = 0 \qquad \forall\, c \in C, \theta \notin E_c[t]$$

$$x_{\theta,c} \geq 0 \qquad \forall\, \theta \in \Theta, c \in C$$

Our Algorithm

For each Round $t \in [T]$:

- $x^*[t] =$ solution to $P_\delta\left(\underbrace{\left(p_\theta(T-t) + \mathbb{I}(\theta_t = \theta)\right)_{\theta\in\Theta}}_{\text{Expected Number of Future Arrivals of } \theta}, (E_c)_{c\in C}, (q_c)_{c\in C}\right)$

Our Algorithm

For each Round $t \in [T]$:

- $x^*[t] = \text{solution to } \; P_\delta\left(\underbrace{\left(p_\theta(T-t) + \mathbb{I}(\theta_t = \theta)\right)_{\theta \in \Theta}}_{\text{Expected Number of Future Arrivals of } \theta}, (E_c)_{c \in C}, (q_c)_{c \in C} \right)$
- $c_t = \underset{c \in C \cup \{\perp\}}{\text{argmax}} \; x^*_{\theta_t, c}[t]$

Our Algorithm

For each Round $t \in [T]$:

- $x^*[t] = \text{solution to } \; P_\delta\left(\underbrace{\left(p_\theta(T-t) + \mathbb{I}(\theta_t = \theta)\right)_{\theta\in\Theta}}_{\text{Expected Number of Future Arrivals of } \theta}, (E_c)_{c\in C}, (q_c)_{c\in C}\right)$
- $c_t = \underset{c \,\in\, C \,\cup\, \{\perp\}}{\text{argmax}} \; x^*_{\theta_t,c}[t]$

If $c_t \in C$:

- Allocate θ_t to in c_t
- $\underbrace{q_{c_t} \leftarrow q_{c_t}[t] \; - 1}_{\text{Reduce Remaining Quota}}$

Our Algorithm

For each Round $t \in [T]$:

- $x^*[t] = \text{solution to } P_\delta\left(\underbrace{\left(p_\theta(T-t) + \mathbb{I}(\theta_t = \theta)\right)_{\theta\in\Theta}}_{\text{Expected Number of Future Arrivals of } \theta}, (E_c)_{c\in C}, (q_c)_{c\in C}\right)$
- $c_t = \underset{c \in C \cup \{\perp\}}{\text{argmax}}\ x^*_{\theta_t,c}[t]$

If $c_t \in C$:

- Allocate θ_t to in c_t
- $\underbrace{q_{c_t} \leftarrow q_{c_t}[t] - 1}_{\text{Reduce Remaining Quota}}$

If $c_t = \perp$:

- Leave θ_t unallocated
- $\underbrace{E_c \leftarrow E_c \setminus \{\theta \prec_c \theta_t\}\ \ \forall c \in C}_{\text{Prevent Future Allocation to Lower Priority Types}}$

Compensating for Mistakes

$x^{OPT}[t]$ = hindsight optimal allocation in remaining rounds given past actions

$$\Delta[t] = \text{loss incurred from the decision in round } t$$

Compensating for Mistakes

$x^{OPT}[t]$ = hindsight optimal allocation in remaining rounds given past actions

$$\Delta[t] = \text{loss incurred from the decision in round } t$$

Algorithm incurs loss only when chosen action was never taken by $x^{OPT}[t]$

Compensating for Mistakes

$x^{OPT}[t]$ = hindsight optimal allocation in remaining rounds given past actions

$$\Delta[t] = \text{loss incurred from the decision in round } t$$

Algorithm incurs loss only when chosen action was never taken by $x^{OPT}[t]$

If we chose to allocate:

- $\Delta_E[t] \leq 1$
- $\Delta_P[t] \leq T - t$

Compensating for Mistakes

$x^{OPT}[t]$ = hindsight optimal allocation in remaining rounds given past actions

$$\Delta[t] = \text{loss incurred from the decision in round } t$$

Algorithm incurs loss only when chosen action was never taken by $x^{OPT}[t]$

If we chose to allocate:

- $\Delta_E[t] \le 1$
- $\Delta_P[t] \le T - t$

If we chose not to allocate:

- $\Delta_E[t] \le T - t + 1$
- $\Delta_P[t] \le 1$

Compensating for Mistakes

$x^{OPT}[t]$ = hindsight optimal allocation in remaining rounds given past actions

$$\Delta[t] = \text{loss incurred from the decision in round } t$$

Algorithm incurs loss only when chosen action was never taken by $x^{OPT}[t]$

If we chose to allocate:

- $\Delta_E[t] \leq 1$
- $\Delta_P[t] \leq T - t$

If we chose not to allocate:

- $\Delta_E[t] \leq T - t + 1$
- $\Delta_P[t] \leq 1$

In all, $\Delta[t] = (T - t + 2) \cdot \Pr(\text{chosen action never taken})$

LP Sensitivity

Algorithm takes most common action when expected number of each type arrives

Algorithm incurs loss only when this action is never taken

LP Sensitivity

Algorithm takes most common action when expected number of each type arrives

Algorithm incurs loss only when this action is never taken

For this to occur, our LP solution $x^*[t]$ must vary a lot from $x^{OPT}[t]$

Solutions to our LP are 1-Lipschitz in RHS

➢ Actual number of arrivals must vary widely from expected number of arrivals

LP Sensitivity

Algorithm takes most common action when expected number of each type arrives

Algorithm incurs loss only when this action is never taken

For this to occur, our LP solution $x^*[t]$ must vary a lot from $x^{OPT}[t]$

Solutions to our LP are 1-Lipschitz in RHS

- Actual number of arrivals must vary widely from expected number of arrivals

$$\mathbb{E}[\Delta[t]] \leq (T - t + 2) \cdot 2\,|\Theta| \exp\left(\frac{2{p_{min}}^2(T-t)}{|\Theta|^2(|C|+1)^2}\right)$$

LP Sensitivity

Algorithm takes most common action when expected number of each type arrives

Algorithm incurs loss only when this action is never taken

For this to occur, our LP solution $x^*[t]$ must vary a lot from $x^{OPT}[t]$

Solutions to our LP are 1-Lipschitz in RHS

- Actual number of arrivals must vary widely from expected number of arrivals

$$\mathbb{E}[\Delta[t]] \leq (T - t + 2) \cdot 2\,|\Theta| \exp\left(\frac{2{p_{min}}^2(T-t)}{|\Theta|^2(|C|+1)^2}\right)$$

$$\mathbb{E}[\Delta_P + \Delta_E] = \sum_{t\in[T]} \mathbb{E}[\Delta[t]] \leq \frac{|\Theta|^5\,(|C|+1)^4}{{p_{min}}^4}$$

Improving this Bound

While our bound is constant, it depends on $|\Theta|$, $|C|$, and p_{min}

Improving this Bound

While our bound is constant, it depends on $|\Theta|$, $|C|$, and p_{min}

To remove dependence on $|C|$, slightly modify problem

- Decide online whether to allocate an agent
- After all arrivals, decide which category each allocation comes from

Improving this Bound

While our bound is constant, it depends on $|\Theta|, |C|$, and p_{min}

To remove dependence on $|C|$, slightly modify problem

- Decide online whether to allocate an agent
- After all arrivals, decide which category each allocation comes from

Improved bound: $\mathbb{E}[\Delta_P + \Delta_E] \leq \frac{12|\Theta|^5}{{p_{min}}^4}$

Improving this Bound

While our bound is constant, it depends on $|\Theta|$, $|C|$, and p_{min}

To remove dependence on $|C|$, slightly modify problem

- Decide online whether to allocate an agent
- After all arrivals, decide which category each allocation comes from

Improved bound: $\mathbb{E}[\Delta_P + \Delta_E] \leq \frac{12|\Theta|^5}{{p_{min}}^4}$

Ongoing work to remove dependence on other parameters

Thanks For Listening!