Allocating with Priorities and Quotas:
Algorithms, Complexity, and Dynamics

Sid Banerjee **Matt Eichhorn** David Kempe

Presented at EC 2023

A Motivating Example: Pandemic Response

Supply-chain constraints place limits on available resources

- Ventilators, Vaccines, Antiviral treatments

Many considerations for who to prioritize

- Healthcare / essential workers
- Individuals with comorbidities
- Residents of high-density housing

What is a *fair* way to allocate care?

Commonly used (1D) priority schemes have issues

The Priority-Respecting Allocation Problem

Agents : $\mathcal{A}$, *unit demand* for resource
indifferent about categories

Categories : $\mathcal{C}$, allocate to agents

Quotas : $q_c \in \mathbb{N}$, $q = \sum_{c \in \mathcal{C}} q_c$

Eligibility : $\mathcal{E}_c \subseteq \mathcal{A}$

Priorities : *Total pre-order* $\succeq_c$ over $\mathcal{E}_c$
Ranks agents in *priority tiers*
$a \succeq_c a' \implies c$ prioritizes a over a'

α (2)	β (1)	γ (1)
a	b	b
b	c , e	a
c	d	
d		
e		

Pathak, Sönmez, Ünver, and Yenmez. *Fair allocation of vaccines, ventilators and antiviral treatments: leaving no ethical value behind in health care rationing*. EC 2021.

Delacrétaz. *Processing reserves simultaneously*. EC 2021.

Aziz and Brandl. *Efficient, fair, and incentive-compatible healthcare rationing*. EC 2021.

Feasible Allocations

Goal: Select an *allocation map* $\varphi : \mathcal{A} \to \mathcal{C} \cup \{\perp\}$

What properties should φ have?

Feasible Allocations

Goal: Select an *allocation map* $\varphi : \mathcal{A} \to \mathcal{C} \cup \{\perp\}$

What properties should φ have?

Quota Respecting [QR]: Categories allocate at most their quotas

$$|\varphi^{-1}(c)| \leq q_c$$

Eligibility Respecting [ER]: Categories allocate only to eligible agents

$$\varphi^{-1}(c) \subseteq \mathcal{E}_c$$

Feasible Allocations

Goal: Select an *allocation map* $\varphi : \mathcal{A} \to \mathcal{C} \cup \{\perp\}$

What properties should φ have?

Quota Respecting [QR]: Categories allocate at most their quotas

$$|\varphi^{-1}(c)| \leq q_c$$

Eligibility Respecting [ER]: Categories allocate only to eligible agents

$$\varphi^{-1}(c) \subseteq \mathcal{E}_c$$

Priority Respecting [PR]: A category allocates to an agent only if all higher-priority agents have been allocated

$$\varphi(a') = c \ \wedge \ a \succeq_c a' \implies \varphi(a) \neq \perp$$

Visualizing an Allocation

α (2)	β (1)	γ (1)
a	b	b
b	c , e	a
c	d	
d		
e		

Visualizing an Allocation

α (2)	β (1)	γ (1)
a	$\boxed{b}$	b
$\boxed{b}$	c , e	$\boxed{a}$
$\boxed{c}$	$\boxed{d}$	
d		
e		

Locating Valid Allocations

Pareto Efficient [PE]: No alternate allocation satisfying [ER], [QR], [PR] allocates to a strict superset of agents

$$\neg\exists\, \psi \;:\; \psi^{-1}(\perp) \subsetneq \varphi^{-1}(\perp)$$

How can we find *valid* ([QR],[ER],[PR],[PE]) allocations?

Locating Valid Allocations

Pareto Efficient [PE]: No alternate allocation satisfying [ER], [QR], [PR] allocates to a strict superset of agents

$$\neg\exists\, \psi \;:\; \psi^{-1}(\perp) \subsetneq \varphi^{-1}(\perp)$$

How can we find *valid* ([QR],[ER],[PR],[PE]) allocations?

Our Work in Three Acts:

1. Efficient algorithm based on LP characterization
2. Problem extensions and complexity results
3. Online allocation with priorities and quotas

Toward an Efficient Algorithm

Decision variables: $\mathrm{x} = \{x_{a,c}\}_{a \in \mathcal{A}, c \in \mathcal{C}}$. $\quad x_{a,c} = \mathbb{I}\big(\varphi(a) = c\big)$,

(P_0)

$$\begin{array}{llll}
\max & \displaystyle\sum_{a \in \mathcal{A}} \sum_{c \in \mathcal{C}} x_{a,c} & & \text{[PE]} \\
\text{s.t.} & \displaystyle\sum_{a \in \mathcal{A}} x_{a,c} \le q_c & \forall\, c \in \mathcal{C} & \text{[QR]} \\
& \displaystyle\sum_{c \in \mathcal{C}} x_{a,c} \le 1 & \forall\, a \in \mathcal{A} & \text{[UD]} \\
& x_{a,c} = 0 & \forall\, a, c \;:\; a \notin \mathcal{E}_c & \text{[ER]} \\
& x_{a,c} \in \{0, 1\} & \forall\, a \in \mathcal{A}, c \in \mathcal{C} &
\end{array}$$

(P_0) encodes a bipartite b-matching problem

LP-relaxation is totally unimodular $\Longrightarrow$ integer corner points

(P_0) Doesn't Account for Priorities

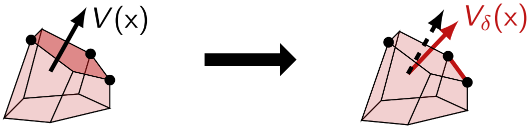
α (2)	β (1)	γ (1)
a	$\boxed{b}$	b
$\boxed{b}$	c , e	$\boxed{a}$
c	$\boxed{d}$	
d		
$\boxed{e}$		

(P_0) Doesn't Account for Priorities

α (2)	β (1)	γ (1)
a	b	b
b	c , e	a
c	d	
d		
e		

Adding Priorities

Idea: Tilt the objective so remaining optima respect priorities

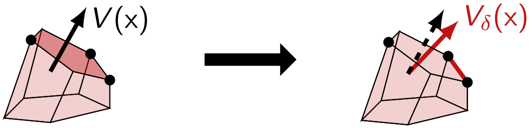


New objective $V_\delta(x) = \sum_{a \in \mathcal{A}} \sum_{c \in \mathcal{C}} \left(1 - \delta_{a,c}\right) x_{a,c} \quad \longrightarrow \quad$ LP (P_δ).

Valid δ should satisfy:

Adding Priorities

Idea: Tilt the objective so remaining optima respect priorities



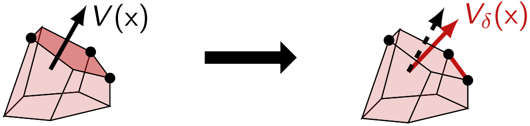
New objective $V_\delta(x) = \sum_{a \in \mathcal{A}} \sum_{c \in \mathcal{C}} \left(1 - \delta_{a,c}\right) x_{a,c} \quad \longrightarrow \quad$ LP (P_δ).

Valid δ should satisfy:

Consistent: Prioritized agents have lower cost $\quad a \succeq_c a' \iff \delta_{a,c} \leq \delta_{a',c}$

Adding Priorities

Idea: Tilt the objective so remaining optima respect priorities



New objective $V_\delta(x) = \sum_{a\in\mathcal{A}} \sum_{c\in\mathcal{C}} \left(1 - \delta_{a,c}\right) x_{a,c} \quad \longrightarrow \quad$ LP (P_δ).

Valid δ should satisfy:

Consistent: Prioritized agents have lower cost $\quad a \succeq_c a' \iff \delta_{a,c} \leq \delta_{a',c}$

Small Effect: Costs don't disincentivize allocation $\quad \sum_{a\in\mathcal{A}} \sum_{c\in\mathcal{C}} \delta_{a,c} \leq \frac{1}{2}$

LP Characterization of Valid Allocations

Theorem

Let x^* *be a solution of* (P_δ) *for any valid* δ. *Then,* x^* *corresponds to an allocation satisfying* $[ER], [QR], [PR], [PE]$.

LP Characterization of Valid Allocations

Theorem

Let x^* be a solution of (P_δ) for any valid δ. Then, x^* corresponds to an allocation satisfying $[ER], [QR], [PR], [PE]$.

Converse Result:

- Valid allocations are corner points on optimal face of (P_0)
- Requiring valid δ restricts angles we can tilt the objective
- Can the allowed angles find all valid allocations?

LP Characterization of Valid Allocations

Theorem

Let x^* be a solution of (P_δ) for any valid δ. Then, x^* corresponds to an allocation satisfying $[ER], [QR], [PR], [PE]$.

Converse Result:

- Valid allocations are corner points on optimal face of (P_0)
- Requiring valid δ restricts angles we can tilt the objective
- Can the allowed angles find all valid allocations?

Theorem (Informal)

We can locate any set of agents who receive units in a valid allocation by solving (P_δ) for some valid δ.

Problem Extensions

- Weighted matching framework is a standard setting
- The restrictions on δ are minimal

How far can we extend our techniques to handle related problems?

Problem Extensions

- Weighted matching framework is a standard setting
- The restrictions on δ are minimal

How far can we extend our techniques to handle related problems?

Case Studies: Demonstrate "computational knife's edge"

- One extension is easy: small modification to our algorithm
- Related extension is NP-Hard

1. Reasoning about a Particular Agent

Can agent a be allocated?

A *serviceable* agent is a recipient in *some* valid allocation.

Decide if agent a is serviceable.

Must agent a be allocated?

A *unanimous* agent is a recipient in *every* valid allocation.

Decide if agent a is unanimous.

1. Reasoning about a Particular Agent

Can agent *a* be allocated?

A *serviceable* agent is a recipient in *some* valid allocation.

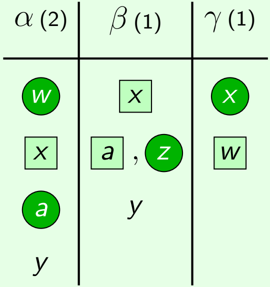
Deciding whether an agent *a* is serviceable is NP-Hard.

Proof Idea: Reduction from X3C.

Must agent *a* be allocated?

Remove *a* and all lower-ranked agents from instance.

Check if max matching size decreases



1. Reasoning about a Particular Agent

Can agent *a* be allocated?

A *serviceable* agent is a recipient in *some* valid allocation.

Deciding whether an agent *a* is serviceable is NP-Hard.

Proof Idea: Reduction from X3C.

Must agent *a* be allocated?

Remove *a* and all lower-ranked agents from instance.

Check if max matching size decreases

α (2)	β (1)	γ (1)
w	x	x
x	a , z	w
a	y	
y		

1. Reasoning about a Particular Agent

Can agent *a* be allocated?

A *serviceable* agent is a recipient in *some* valid allocation.

Deciding whether an agent *a* is serviceable is NP-Hard.

Proof Idea: Reduction from X3C.

Must agent *a* be allocated?

Remove *a* and all lower-ranked agents from instance.

Check if max matching size decreases

α (2)	β (1)	γ (1)
w	*x*	*x*
x	*z*	*w*

1. Reasoning about a Particular Agent

Can agent *a* be allocated?

A *serviceable* agent is a recipient in *some* valid allocation.

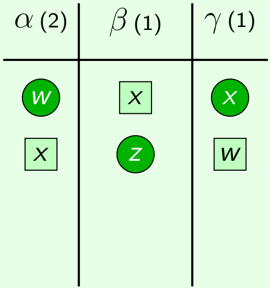
Deciding whether an agent *a* is serviceable is NP-Hard.

Proof Idea: Reduction from X3C.

Must agent *a* be allocated?

Remove *a* and all lower-ranked agents from instance.

Check if max matching size decreases



2. Incorporating Agent Utility

Agent a has a utility function $u_a : \mathcal{C} \to (0, 1]$ encoding preference for certain categories. ($u_a(\perp) = 0$.)

Utility *Pareto-Efficient* Allocation

Select allocation that disincentivizes agents from trying to swap units

Utility *Maximizing* Allocation

Select allocation that maximizes the sum of agent utilities

2. Incorporating Agent Utility

Agent a has a utility function $u_a : \mathcal{C} \to (0, 1]$ encoding preference for certain categories. ($u_a(\perp) = 0$.)

Utility *Pareto-Efficient* Allocation

Run our algorithm twice

First run uses arbitrary δ to determine recipient set.

Second run removes unallocated agents and sets δ according to agent utilities.

Utility *Maximizing* Allocation

NP-Hard via a reduction from serviceable problem.

Proof Idea: One agent has high utility in all categories, others have low utility.

*Hardness reduction can be generalized to other optimization objectives (e.g. Nash Social Welfare)

Online Priority-Respecting Allocation

Setup:

- Instead of agent set $\mathcal{A}$, there is a finite set of agent *types* Θ
- Categories specify eligibility and priorities over types
- T arriving agents
- Agents' types θ_t drawn i.i.d. from *known* distribution $(p_\theta)_{\theta \in \Theta}$

Online Priority-Respecting Allocation

Setup:

- Instead of agent set $\mathcal{A}$, there is a finite set of agent *types* Θ
- Categories specify eligibility and priorities over types
- T arriving agents
- Agents' types θ_t drawn i.i.d. from *known* distribution $(p_\theta)_{\theta \in \Theta}$

Theorem

If we insist on no priority violations, there are instances that incur $\Omega(T)$ loss in efficiency with high probability.

A Multi-Objective Approach

Rather than enforcing a [PR] constraint, we'll treat minimizing priority violations as a second objective

A Multi-Objective Approach

Rather than enforcing a [PR] constraint, we'll treat minimizing priority violations as a second objective

Theorem

Given any online priority-respecting allocation instance, there is an algorithm that ensures that

$$\mathbb{E}\left[\textit{efficiency loss} + \#\ \textit{priority violations}\right] \leq \frac{|\Theta|^5(|\mathcal{C}|+1)^4}{p_{\min}^4}.$$

- Constant with respect to the instance size (T and q)
- Depends only on instance "complexity"
- The algorithm fundamentally relies on our LP characterization

Conclusion

- In economics and CS, we typically model fairness as an objective function to optimize
- Categories provide an instrument to encode "competing" objectives in a transparent way
- Can locate good allocations via a weighted matching LP
  - More efficient than existing approaches
  - Provides flexibility for many problem extensions
- Perturbation technique seems useful in other related problems

Thank You!