Mind your Ps and Qs

Allocation with Priorities and Quotas

Matthew Eichhorn

Based on joint work with Sid Banerjee and David Kempe
Paper: https://arxiv.org/abs/2204.13019

A Motivating Example: Pandemic Response

Supply-chain constraints place limits on available resources

- Ventilators, Vaccines, Anti-viral treatments

Many considerations for who to prioritize

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

A Motivating Example: Pandemic Response

Supply-chain constraints place limits on available resources

- Ventilators, Vaccines, Anti-viral treatments

Many considerations for who to prioritize

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

What is the "best" way to allocate care?

Commonly used 1-D priority schemes have issues

Formalizing the Reserve Allocation Setting

Agents : $\mathcal{A}, \quad n = |\mathcal{A}|$

- *Unit demand* for the resource
- *Indifferent* about how they are allocated

Categories : $\mathcal{C}, \quad m = |\mathcal{C}|$

Each category $c \in \mathcal{C}$ has:

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

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

Priorities : *Total pre-order* $\succeq_c$ over $\mathcal{E}_c$

- $\succeq_c$ separates agents into ranked *priority tiers*
- $a \succeq_c a' \implies c$ gives priority to a over a'

Visualizing an Instance

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

Feasible Allocations

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

Feasible Allocations

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

What properties should φ have?

Feasible Allocations

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

What properties should φ have?

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

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

Feasible Allocations

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

What properties should φ have?

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

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

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

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

Feasible Allocations

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

What properties should φ have?

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

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

Eligibility Respecting [ER]: Categories only allocate 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

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

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	b	b
b	c , e	a
c	d	
d		
e		

Locating Good Allocations

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

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

Locating Good Allocations

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

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

Is there an efficient algorithm to find allocations with these properties?

Existing Approaches

Pathak et al (2021) [1]: Variant of Deferred Acceptance [2]

- Agents have arbitrary preferences over eligible categories
- Run DA with agents proposing to categories
- [QR], [ER], [PR], not necessarily [PE]

Delacrétaz (2021) [3]: Simultaneous Reserves Algorithm

- "Water-filling" down priority lists determines who gets allocated
- [QR], [ER], [PR], not necessarily [PE]

Aziz and Brandl (2021) [4]: Reverse Rejecting Algorithm

- Iteratively certifies whether a maximal allocation can be found without allocating to a particular agent
- All four properties, but requires $O(n)$ max matching problems

Outline

1. The Reserve Allocation Problem
2. **An IP-Based Allocation Algorithm**
3. Finding Fair Allocations

Toward an IP Formulation

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

Unit Demand: $\sum_{c \in \mathcal{C}} x_{a,c} \leq 1 \qquad \forall\, a \in \mathcal{A}$,

[QR]: $\sum_{a \in \mathcal{A}} x_{a,c} \leq q_c \qquad \forall\, c \in \mathcal{C}$,

[ER]: $x_{a,c} = 0 \qquad \forall\, a, c \,:\, a \notin \mathcal{E}_c$,

[PE]: “Stronger” Condition: Allocate to maximum number of agents subject to above constraints.

Let $V(\mathbf{x}) = \sum_{a \in \mathcal{A}} \sum_{c \in \mathcal{C}} x_{a,c}$ be the *total allocation* of $\mathbf{x}$.

Toward an IP Formulation

(P_0)

$$
\begin{aligned}
\max \quad & V(\mathbf{x}) & \\
\text{s.t.} \quad & \sum_{a \in \mathcal{A}} x_{a,c} \le q_c & \forall\, c \in \mathcal{C} \\
& \sum_{c \in \mathcal{C}} x_{a,c} \le 1 & \forall\, a \in \mathcal{A} \\
& x_{a,c} = 0 & \forall\, a, c \;:\; a \notin \mathcal{E}_c \\
& x_{a,c} \in \{0, 1\} & \forall\, a \in \mathcal{A}, c \in \mathcal{C}
\end{aligned}
$$

(P_0) encodes a bipartite b-matching problem, which we can efficiently solve (Hopcraft-Karp, Hungarian Algorithm, etc.).

(P_0) Doesn't Account for Priorities

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

(P_0) Doesn't Account for Priorities

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

To incorporate priorities:

(P_0) Doesn't Account for Priorities

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

To incorporate priorities:

1. Local updates: Valid approach, can require $O(mn)$ steps

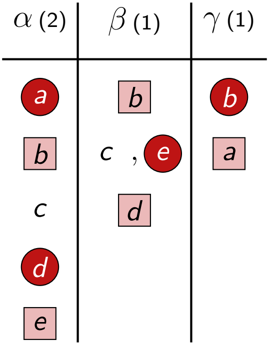
(P_0) Doesn't Account for Priorities

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

To incorporate priorities:

1. Local updates: Valid approach, can require $O(mn)$ steps
2. Add constraints: We lose IP structure (integrality of corner pts)

(P_0) Doesn't Account for Priorities

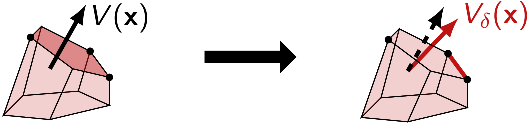


To incorporate priorities:

1. Local updates: Valid approach, can require $O(mn)$ steps
2. Add constraints: We lose IP structure (integrality of corner pts)
3. Change the IP objective

Adding Priorities

Idea: Tilt the objective so remaining optima respect priorities

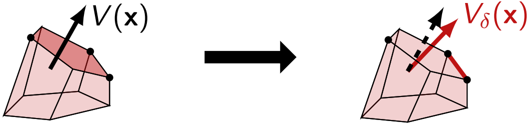


Replace $V(\mathbf{x})$ with $V_\delta(\mathbf{x}) = \sum_{a \in \mathcal{A}} \sum_{c \in \mathcal{C}} \left(1 - \delta_{a,c}\right) x_{a,c}$.

Interpreting $\delta_{a,c}$ as the cost of allocating *a* through *c*, a *valid* δ satisfies:

Adding Priorities

Idea: Tilt the objective so remaining optima respect priorities



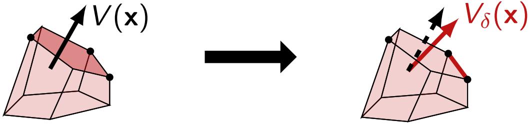
Replace $V(\mathbf{x})$ with $V_\delta(\mathbf{x}) = \sum_{a \in \mathcal{A}} \sum_{c \in \mathcal{C}} \left(1 - \delta_{a,c}\right) x_{a,c}$.

Interpreting $\delta_{a,c}$ as the cost of allocating a through c, a *valid* δ satisfies:

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

Adding Priorities

Idea: Tilt the objective so remaining optima respect priorities



Replace $V(\mathbf{x})$ with $V_\delta(\mathbf{x}) = \sum_{a\in\mathcal{A}} \sum_{c\in\mathcal{C}} \left(1 - \delta_{a,c}\right) x_{a,c}$.

Interpreting $\delta_{a,c}$ as the cost of allocating a through c, a *valid* δ satisfies:

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

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

Choosing the Perturbation Weights

Given any δ, define the IP

(P_δ)

$$
\begin{array}{lll}
\max & V_\delta(\mathbf{x}) & \\
\text{s.t.} & \sum_{a \in \mathcal{A}} x_{a,c} \le q_c & \forall\, c \in \mathcal{C} \\
& \sum_{c \in \mathcal{C}} x_{a,c} \le 1 & \forall\, a \in \mathcal{A} \\
& x_{a,c} = 0 & \forall\, a, c \ :\ a \notin \mathcal{E}_c \\
& x_{a,c} \in \{0, 1\} & \forall\, a \in \mathcal{A}, c \in \mathcal{C}
\end{array}
$$

Choosing the Perturbation Weights

Given any δ, define the IP

(P_δ)

$$
\begin{aligned}
\max \quad & V_\delta(\mathbf{x}) & \\
\text{s.t.} \quad & \sum_{a \in \mathcal{A}} x_{a,c} \leq q_c & \forall\, c \in \mathcal{C} \\
& \sum_{c \in \mathcal{C}} x_{a,c} \leq 1 & \forall\, a \in \mathcal{A} \\
& x_{a,c} = 0 & \forall\, a, c \ :\ a \notin \mathcal{E}_c \\
& x_{a,c} \in \{0, 1\} & \forall\, a \in \mathcal{A}, c \in \mathcal{C}
\end{aligned}
$$

Theorem

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

Proving this Theorem

Theorem

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

Proof Sketch.

[ER],[QR]: Ensured by (P_δ) constraints.

[PR]: δ is Consistent.

[PE]: Small Effect of δ and integrality:

$$V(\hat{\mathbf{x}}) \geq V(\mathbf{x}^*) \geq V_\delta(\mathbf{x}^*) \geq V_\delta(\hat{\mathbf{x}}) = V(\hat{\mathbf{x}}) - \sum_{a,c} \delta_{a,c} \geq V(\hat{\mathbf{x}}) - \tfrac{1}{2}.$$

so $V(\hat{\mathbf{x}}) = V(\mathbf{x}^*)$ for a solution $\hat{\mathbf{x}}$ to (P_0). □

Outline

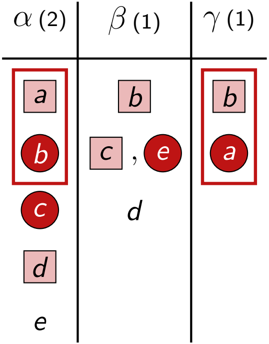
Realizability of Good Allocations

Previous theorem shows that any valid δ can locate a good allocation.

Converse result:

Theorem (Informal)

All 'meaningful' good allocations are solutions to (P_δ) for some valid δ.



The restrictions we've placed on δ are minimal.

We can carefully choose δ to find allocations with additional properties.

Example: Have perturbations on different orders of magnitude in different categories to enforce a "priority over categories"

Fair Allocations

Auditability: What info must categories reveal to assure agents that the allocation φ is fair?

Fair Allocations

Auditability: What info must categories reveal to assure agents that the allocation φ is fair?

- For each eligible agent $a \in \mathcal{E}_c$, let $r_c(a)$ be their priority tier in c ($1 =$ highest priority, $2 =$ next priority tier, etc.)
- In each $c \in \mathcal{C}$, $\tau_c \in \mathbb{N}$ is a *cutoff* if:

1. All agents allocated in c sit at or above the cutoff

$$\varphi(a) = c \implies r_c(a) \leq \tau_c,$$

2. All un-allocated agents sit at or below the cutoff

$$\varphi(a) = \varnothing \implies r_c(a) \geq \tau_c.$$

Notable Cutoff Tiers

- **Inner Cutoff:** For each $c \in \mathcal{C}$,

$$\underline{\tau_c} = \max_{a \in \varphi^{-1}(c)} \{r_c(a)\}.$$

α (3)	β (2)	γ (2)
a_1	a_5	a_6
a_2	a_3 , a_8	a_3
a_3	a_4	a_1
a_4	a_0 , a_9	a_8
a_5	a_1	a_7
a_6		a_0
a_7		
a_8		

Notable Cutoff Tiers

- **Inner Cutoff:** For each $c \in \mathcal{C}$,

$$\underline{\tau_c} = \max_{a \in \varphi^{-1}(c)} \{r_c(a)\}.$$

- **Outer Cutoff:** For each $c \in \mathcal{C}$,

$$\overline{\tau_c} = \min_{a \in \varphi^{-1}(\varnothing)} \{r_c(a)\}.$$

α (3)	β (2)	γ (2)
a_1	a_5	a_6
a_2	a_3 , a_8	a_3
a_3	a_4	a_1
a_4	a_0 , a_9	a_8
a_5	a_1	a_7
a_6		a_0
a_7		
a_8		

Notable Cutoff Tiers

- **Inner Cutoff:** For each $c \in \mathcal{C}$,

$$\underline{\tau_c} = \max_{a \in \varphi^{-1}(c)} \{r_c(a)\}.$$

- **Outer Cutoff:** For each $c \in \mathcal{C}$,

$$\overline{\tau_c} = \min_{a \in \varphi^{-1}(\varnothing)} \{r_c(a)\}.$$

- Every $\{\tau_c\}$ has $\underline{\tau_c} \leq \tau_c \leq \overline{\tau_c}$ for each $c \in \mathcal{C}$.

α (3)	β (2)	γ (2)
a_1	a_5	a_6
a_2	a_3 , a_8	a_3
a_3	a_4	a_1
a_4	a_0 , a_9	a_8
a_5	a_1	a_7
a_6		a_0
a_7		
a_8		

Promoting High Priority Allocations

Allocations should go to agents ranked highly in a category.

1. Minimize sum of allocated ranks: $\sum_{a,c} x_{a,c} \cdot r_c(a)$
   - Choose δ to grow *arithmetically* in rank: $\delta_{a,c} = \frac{r_c(a)}{2n^2 m}$

$$V_\delta(\mathbf{x}) = V(\mathbf{x}) - \frac{1}{2n^2 m} \sum_{a,c} x_{a,c} \cdot r_c(a)$$

2. Minimize max inner cutoff: $\max_c \left\{ \underline{\tau_c} \right\} = \max_{a,c} \left\{ x_{a,c} \cdot r_c(a) \right\}$
   - Choose δ to grow *geometrically* in rank: $\delta_{a,c} = \frac{1}{2nm} \cdot \left(\frac{1}{n+1} \right)^{n - r_c(a)}$

** In subsequent work, we show maximizing the min outer cutoff is computationally hard and consider other extensions (agent utility)

Conclusion

- Reserve Allocation is a reasonable modeling framework for assignment problems with "competing" objectives
- Can locate good allocations via an IP with perturbed objective
  - More efficient than existing approaches
  - Provides flexibility to consider secondary objectives such as fairness

Open Questions:

Natural desiderata that locate a unique (fractional) allocation?

Can this allocation be computed efficiently?

- Our perturbation technique seems useful in other related problems

Thank You!

References

- P. A. Pathak, T. Sönmez, M. U. Ünver, and M. B. Yenmez, "Fair allocation of vaccines, ventilators and antiviral treatments: leaving no ethical value behind in health care rationing," in *Proceedings of the 22nd ACM Conference on Economics and Computation*, pp. 785–786, 2021.
- D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962.
- D. Delacrétaz, "Processing reserves simultaneously," in *Proceedings of the 22nd ACM Conference on Economics and Computation*, pp. 345–346, 2021.
- H. Aziz and F. Brandl, "Efficient, fair, and incentive-compatible healthcare rationing," in *Proceedings of the 22nd ACM Conference on Economics and Computation*, pp. 103–104, 2021.

Additional Slides

Category-Stable Allocations

Category-Stable [CS]: No category can organize an agreeable trade that allows it to allocate to a higher priority agent

There is no cycle $a_1, \ldots, a_j = a_0 \in \mathcal{A},\ c_1, \ldots, c_j = c_0 \in \mathcal{C}$ where:

- $\varphi(a_i) = c_i \qquad \forall\, 0 \leq i < j,$
- $a_{i+1} \succeq_{c_i} a_i \qquad \forall\, 0 \leq i < j,$
- At least one priority is strict

Category-Stable Allocations

Category-Stable [CS]: No category can organize an agreeable trade that allows it to allocate to a higher priority agent

There is no cycle $a_1, \ldots, a_j = a_0 \in \mathcal{A}$, $c_1, \ldots, c_j = c_0 \in \mathcal{C}$ where:

- $\varphi(a_i) = c_i \qquad \forall\, 0 \leq i < j,$
- $a_{i+1} \succeq_{c_i} a_i \qquad \forall\, 0 \leq i < j,$
- At least one priority is strict

Theorem

$\mathbf{x}$ *is a good, [CS] allocation* $\iff$ $\mathbf{x}$ *is a solution to* (P_δ) *for some valid* δ.

Key Proof Ingredient: Serial Dictatorship Allocations

1. Choose a (multiset) ordering of categories' units
2. Ask categories to draft their favorite remaining agent
3. Accept the allocation if it's maximal

Key Proof Ingredient: Serial Dictatorship Allocations

1. Choose a (multiset) ordering of categories' units
2. Ask categories to draft their favorite remaining agent
3. Accept the allocation if it's maximal

$$(\alpha, \gamma, \alpha, \beta)$$

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

Key Proof Ingredient: Serial Dictatorship Allocations

1. Choose a (multiset) ordering of categories' units
2. Ask categories to draft their favorite remaining agent
3. Accept the allocation if it's maximal

$$(\alpha, \gamma, \alpha, \beta)$$

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

Key Proof Ingredient: Serial Dictatorship Allocations

1. Choose a (multiset) ordering of categories' units
2. Ask categories to draft their favorite remaining agent
3. Accept the allocation if it's maximal

$$(\alpha, \gamma, \alpha, \beta)$$

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

Key Proof Ingredient: Serial Dictatorship Allocations

1. Choose a (multiset) ordering of categories' units
2. Ask categories to draft their favorite remaining agent
3. Accept the allocation if it's maximal

$$(\alpha, \gamma, \alpha, \beta)$$

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

Key Proof Ingredient: Serial Dictatorship Allocations

1. Choose a (multiset) ordering of categories' units
2. Ask categories to draft their favorite remaining agent
3. Accept the allocation if it's maximal

$$(\alpha, \gamma, \alpha, \beta)$$

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

Key Proof Ingredient: Serial Dictatorship Allocations

1. Choose a (multiset) ordering of categories' units
2. Ask categories to draft their favorite remaining agent
3. Accept the allocation if it's maximal

$(\alpha, \gamma, \alpha, \beta)$

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

$(\alpha, \alpha, \gamma, \beta)$

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

Key Proof Ingredient: Serial Dictatorship Allocations

1. Choose a (multiset) ordering of categories' units
2. Ask categories to draft their favorite remaining agent
3. Accept the allocation if it's maximal

$(\alpha, \gamma, \alpha, \beta)$

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

$(\alpha, \alpha, \gamma, \beta)$

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

Key Proof Ingredient: Serial Dictatorship Allocations

1. Choose a (multiset) ordering of categories' units
2. Ask categories to draft their favorite remaining agent
3. Accept the allocation if it's maximal

$(\alpha, \gamma, \alpha, \beta)$

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

$(\alpha, \alpha, \gamma, \beta)$

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

Promoting Many Allocations in each Category

Choose agents to allocate that are ranked highly in many categories.

1. Maximize the minimum outer cutoff: $\min_{a,c} \left\{ \left(1 - \sum_{c' \in \mathcal{C}} x_{a,c'}\right) \cdot r_c(a) \right\}$.

This can be done efficiently if and only if we can efficiently solve the following decision problem.

FillTier $(\mathcal{I}, k)$

Given an allocation instance $\mathcal{I} = (\mathcal{A}, \mathcal{C}, \{\succeq_c\})$, and $k \in \mathbb{N}$, is there a good allocation that gives to all agents within the top k tiers in some category?

Theorem

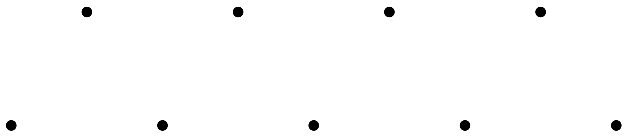
FillTier *is NP-hard.*

Exact Cover by 3-Sets (X3C)

Input: Ground set $E = \{e_1, e_2, \ldots, e_{3n}\}$.

Collection of subsets $\mathcal{S} = \{S_1, \ldots, S_m\}$, each $|S_i| = 3$.

Decide: Is there a collection of subsets $\{S_{i_1}, \ldots, S_{i_n}\}$ such that $E = \bigcup_{j=1}^{n} S_{i_j}$?



Lemma

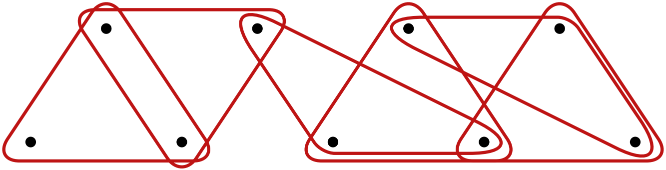
X3C *is NP-Complete.*

Exact Cover by 3-Sets (X3C)

Input: Ground set $E = \{e_1, e_2, \ldots, e_{3n}\}$.

Collection of subsets $\mathcal{S} = \{S_1, \ldots, S_m\}$, each $|S_i| = 3$.

Decide: Is there a collection of subsets $\{S_{i_1}, \ldots, S_{i_n}\}$ such that $E = \bigcup_{j=1}^{n} S_{i_j}$?



Lemma

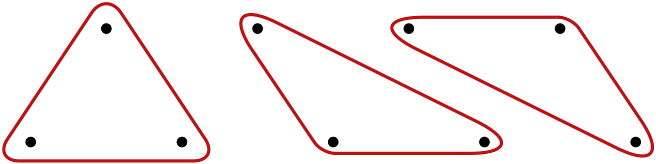
X3C *is NP-Complete.*

Exact Cover by 3-Sets (X3C)

Input: Ground set $E = \{e_1, e_2, \ldots, e_{3n}\}$.

Collection of subsets $\mathcal{S} = \{S_1, \ldots, S_m\}$, each $|S_i| = 3$.

Decide: Is there a collection of subsets $\{S_{i_1}, \ldots, S_{i_n}\}$ such that $E = \bigcup_{j=1}^{n} S_{i_j}$?



Lemma

X3C *is NP-Complete.*

The Reduction

X3C Input: $E = \{e_1, e_2, \ldots, e_{3n}\}$, $\mathcal{S} = \{S_1, \ldots, S_m\}$d

$S_j = \{e_{i_{j,1}}, e_{i_{j,2}}, e_{i_{j,3}}\}$

Allocation Instance: $\mathcal{A} = E \cup \mathcal{S} \cup \{f_1, \ldots, f_{4(m-n)}\}$, $\quad k = 1$

set categories			element categories		
α_1 (4)	$\cdots$	α_m (4)	β_1 (0)	$\cdots$	β_{3n} (0)
f_1		f_1	e_1		e_{3n}
$\vdots$		$\vdots$			
$f_{4(m-n)}$		$f_{4(m-n)}$			
S_1		S_m			
$e_{i_{1,1}}$		$e_{i_{m,1}}$			
$e_{i_{1,2}}$		$e_{i_{m,2}}$			
$e_{i_{1,3}}$		$e_{i_{m,3}}$			