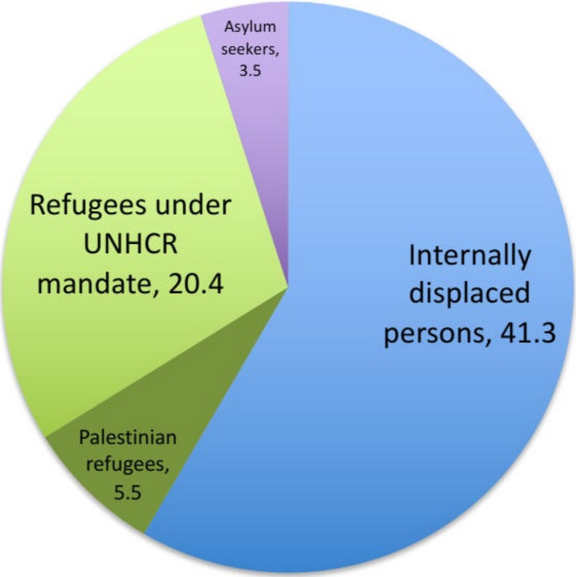
Matching Mechanisms for Refugee Resettlement

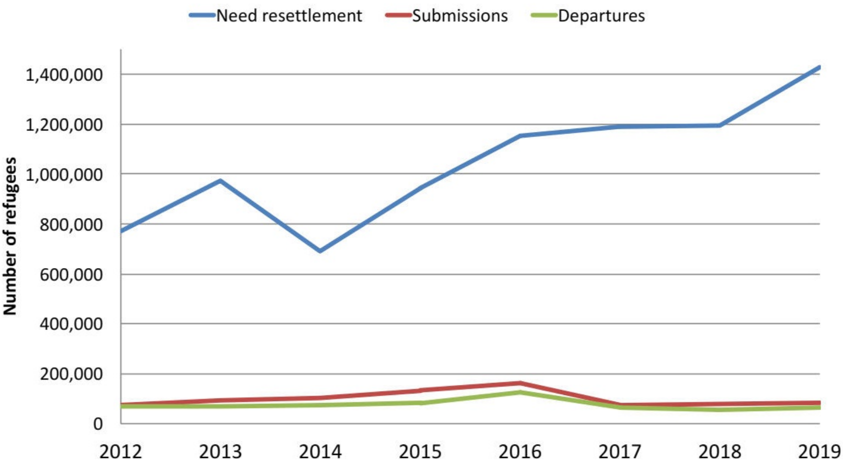
David Delacrétaz	Scott Kominers	Alex Teytelboym
Oxford	Harvard	Oxford

Guest Lecture
AS.180.244 Market Design
AS.180.645 Topics in Economic Theory
Johns Hopkins University
May 2020

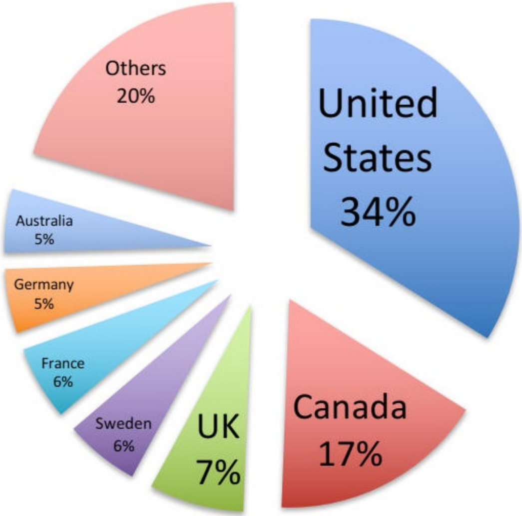
70M Refugees around the World



Resettlement Needs, Submissions, and Departures



Resettlement Destinations (2018)



Resettlement Process - USA

Refugees apply directly or are referred by the UNHCR

- Often while living in refugee camps

Resettlement Process - USA

Refugees apply directly or are referred by the UNHCR

- Often while living in refugee camps

US Refugee Admissions Program (USRAP) selects candidates

Resettlement Process - USA

Refugees apply directly or are referred by the UNHCR

- Often while living in refugee camps

US Refugee Admissions Program (USRAP) selects candidates

- Security checks
- Medical checks
- Cultural orientation

Resettlement Process - USA

Refugees apply directly or are referred by the UNHCR

- Often while living in refugee camps

US Refugee Admissions Program (USRAP) selects candidates

- Security checks
- Medical checks
- Cultural orientation

The case is handed over to one of nine resettlement agencies

Resettlement Process - USA

Refugees apply directly or are referred by the UNHCR

- Often while living in refugee camps

US Refugee Admissions Program (USRAP) selects candidates

- Security checks
- Medical checks
- Cultural orientation

The case is handed over to one of nine resettlement agencies

- One of them is the Hebrew Immigrant Aid Society (HIAS)

The Matching Problem

Each resettlement agency has links to localities

The Matching Problem

Each resettlement agency has links to localities

- Local communities willing to host refugees

The Matching Problem

Each resettlement agency has links to localities

- Local communities willing to host refugees
- Support their assigned refugees for the first year

The Matching Problem

Each resettlement agency has links to localities

- Local communities willing to host refugees
- Support their assigned refugees for the first year
- Spread across the US (most in CA, FL, NY, TX)

The Matching Problem

Each resettlement agency has links to localities

- Local communities willing to host refugees
- Support their assigned refugees for the first year
- Spread across the US (most in CA, FL, NY, TX)

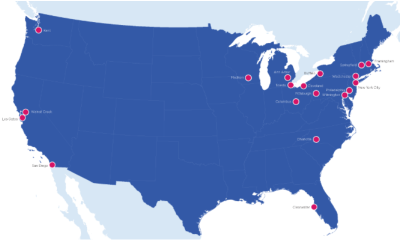


Figure: HIAS' network of localities.

The Matching Problem

Each resettlement agency has links to localities

- Local communities willing to host refugees
- Support their assigned refugees for the first year
- Spread across the US (most in CA, FL, NY, TX)

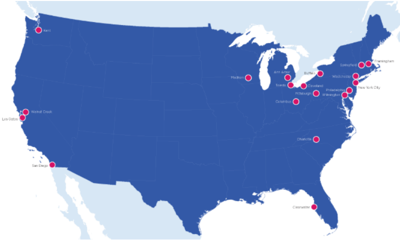


Figure: HIAS' network of localities.

How to best match refugees and localities?

Why the matching matters

Empirical evidence that the initial match matters in for a refugee's long-term economic outcomes: employment, welfare, education

Why the matching matters

Empirical evidence that the initial match matters in for a refugee's long-term economic outcomes: employment, welfare, education

- Denmark and Sweden: Åslund and Rooth (2007); Åslund and Fredriksson (2009); Damm (2009); Åslund et al. (2010, 2011); Damm (2014)

Why the matching matters

Empirical evidence that the initial match matters in for a refugee's long-term economic outcomes: employment, welfare, education

- Denmark and Sweden: Åslund and Rooth (2007); Åslund and Fredriksson (2009); Damm (2009); Åslund et al. (2010, 2011); Damm (2014)
- USA: Feywerda and Gest (2016); Bansak et al. (2018)

Why the matching matters

Empirical evidence that the initial match matters in for a refugee's long-term economic outcomes: employment, welfare, education

- Denmark and Sweden: Åslund and Rooth (2007); Åslund and Fredriksson (2009); Damm (2009); Åslund et al. (2010, 2011); Damm (2014)
- USA: Feywerda and Gest (2016); Bansak et al. (2018)
- Finland: Sarvimäki et al. (2018)

Why the matching is difficut

Families must be kept together

- Match families to localities

Why the matching is difficut

Families must be kept together

- Match families to localities

Each locality can host a maximum number of refugees

Why the matching is difficut

Families must be kept together

- Match families to localities

Each locality can host a maximum number of refugees

- Families have different sizes

Why the matching is difficut

Families must be kept together

- Match families to localities

Each locality can host a maximum number of refugees

- Families have different sizes

Additional capacity constraints

- e.g., # of refugees from a given region, # of school-age children

Why the matching is difficut

Families must be kept together

- Match families to localities

Each locality can host a maximum number of refugees

- Families have different sizes

Additional capacity constraints

- e.g., # of refugees from a given region, # of school-age children
- HIAS has no such constraints (but may in the future)
- Other resettlement agencies might

Why the matching is difficut

Families must be kept together

- Match families to localities

Each locality can host a maximum number of refugees

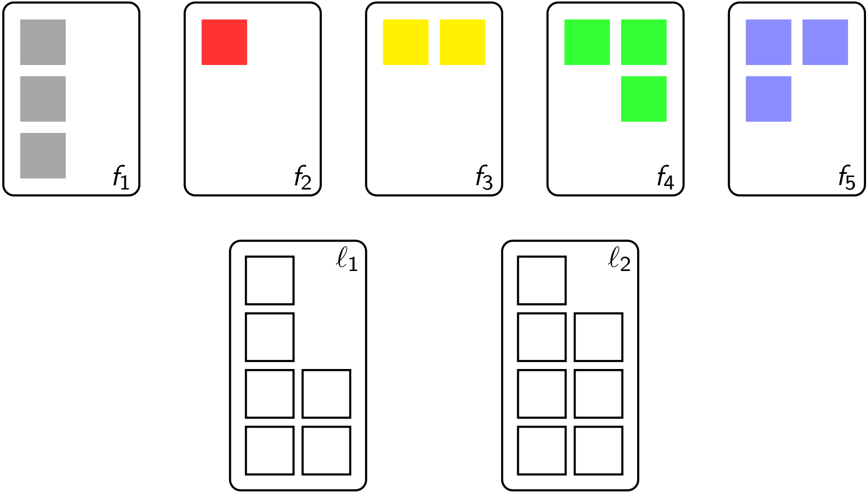
- Families have different sizes

Additional capacity constraints

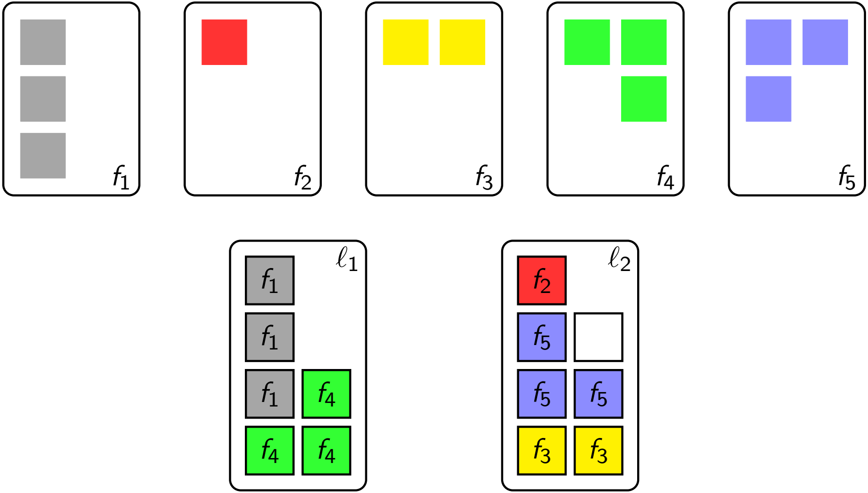
- e.g., # of refugees from a given region, # of school-age children
- HIAS has no such constraints (but may in the future)
- Other resettlement agencies might

Matching Market with Multidimensional Constraints

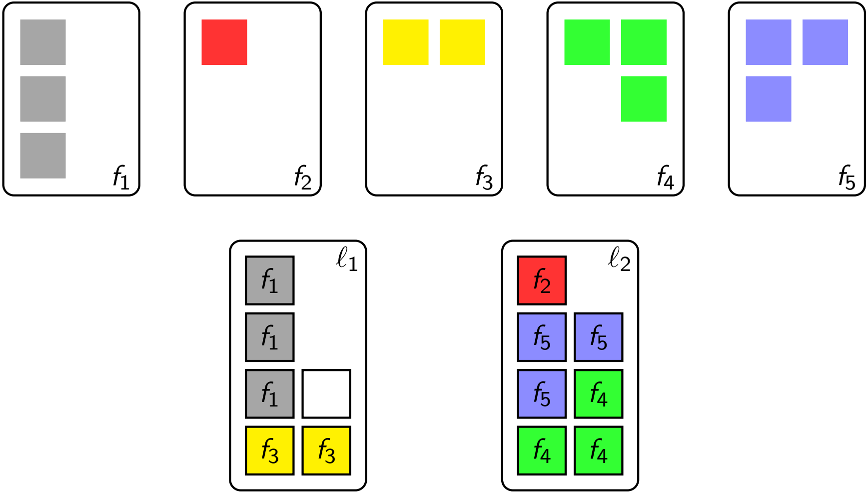
Multidimensional Constraints



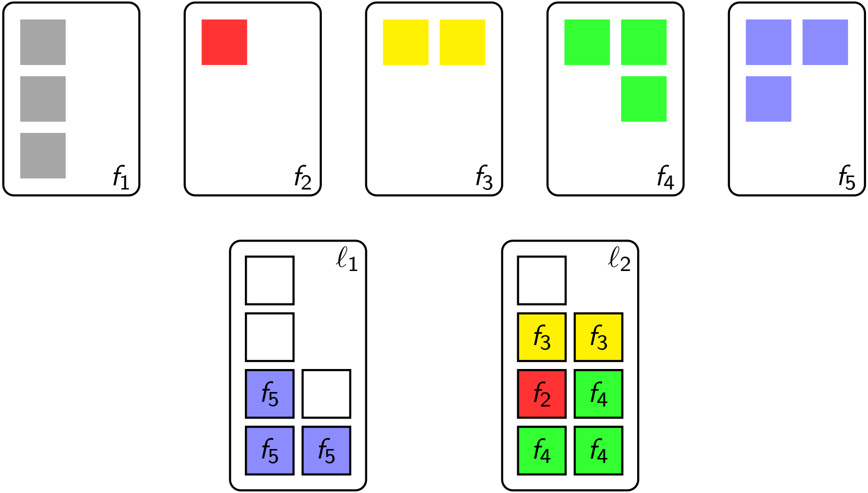
Multidimensional Constraints



Multidimensional Constraints



Multidimensional Constraints



Static vs Dynamic Matching

Capacity constraints are set once a year

- But families arrive stochastically throughout the year

Static vs Dynamic Matching

Capacity constraints are set once a year

- But families arrive stochastically throughout the year

Currently, a matching occurs every fortnight

- Capacity constraints are set proportionally
- Treated as hard constraints

Static vs Dynamic Matching

Capacity constraints are set once a year

- But families arrive stochastically throughout the year

Currently, a matching occurs every fortnight

- Capacity constraints are set proportionally
- Treated as hard constraints

We consider the static problem

- We take capacities as given and treat them as hard constraints
- Dynamic capacity management would constitute a valuable extension

Matching on Observables

1 May 2018: For the first time, a US resettlement agency used a software to algorithmically match refugees

- Trapp, Teytelboym, Martinello, Andersson, and Ahani (2018)

Matching on Observables

1 May 2018: For the first time, a US resettlement agency used a software to algorithmically match refugees

- Trapp, Teytelboym, Martinello, Andersson, and Ahani (2018)
- *Annie^{TM} MOORE* software

Matching on Observables

1 May 2018: For the first time, a US resettlement agency used a software to algorithmically match refugees

- Trapp, Teytelboym, Martinello, Andersson, and Ahani (2018)
- *Annie^{TM} MOORE* software

For each family-locality pair, estimate a probability of employment

- Based on observable characteristics, e.g., age, education, language, etc
- Matching found by solving an integer program

Matching on Observables

1 May 2018: For the first time, a US resettlement agency used a software to algorithmically match refugees

- Trapp, Teytelboym, Martinello, Andersson, and Ahani (2018)
- *Annie™ MOORE* software

For each family-locality pair, estimate a probability of employment

- Based on observable characteristics, e.g., age, education, language, etc
- Matching found by solving an integer program

HIAS uses *Annie™ MOORE*

Matching on Observables

1 May 2018: For the first time, a US resettlement agency used a software to algorithmically match refugees

- Trapp, Teytelboym, Martinello, Andersson, and Ahani (2018)
- *Annie*^{TM} *MOORE* software

For each family-locality pair, estimate a probability of employment

- Based on observable characteristics, e.g., age, education, language, etc
- Matching found by solving an integer program

HIAS uses *Annie*^{TM} *MOORE*

- Other resettlement agencies do the matching by hand

Contribution

We propose to take preferences into account

Contribution

We propose to take preferences into account

- Accounts for unobservable characteristics

Contribution

We propose to take preferences into account

- Accounts for unobservable characteristics
- Involves refugees and localities in the process

Contribution

We propose to take preferences into account

- Accounts for unobservable characteristics
- Involves refugees and localities in the process
- Reduces the risk refugees will move away shortly after resettlement

Contribution

We propose to take preferences into account

- Accounts for unobservable characteristics
- Involves refugees and localities in the process
- Reduces the risk refugees will move away shortly after resettlement

"Many Somali refugees initially settled around the country subsequently migrated to Lewiston, Maine. Lewiston has a weak economy but an established Somali community. Consequently, efforts to resettle these refugees elsewhere in the U.S. were less effective than they could have been. Their preferences should have been taken into account from the start."
— Mark Hetfield (CEO of HIAS) in Roth (2015), "Migrants aren't widgets", Politico

Model

Set of families F, set of localities L

- Families have strict and ordinal preferences over localities

$$\succ_f : \ell_1, \ell_2, \ell_3, \ldots$$

- Each locality strictly ranks families in order of priority

$$\triangleright_\ell : f_1, f_2, f_3, \ldots$$

Model

Set of families F, set of localities L

- Families have strict and ordinal preferences over localities

$$\succ_f : \ell_1, \ell_2, \ell_3, \ldots$$

- Each locality strictly ranks families in order of priority

$$\triangleright_\ell : f_1, f_2, f_3, \ldots$$

Set of services S (# of refugees, school places, medical needs, etc)

- Family f requires $\nu_s^f \in \mathbb{Z}_{\geq 0}$ units of service s
- Locality ℓ can provide $\kappa_s^\ell \in \mathbb{Z}_{\geq 0}$ units of service s

Model

Set of families F, set of localities L

- Families have strict and ordinal preferences over localities

$$\succ_f : \ell_1, \ell_2, \ell_3, \ldots$$

- Each locality strictly ranks families in order of priority

$$\rhd_\ell : f_1, f_2, f_3, \ldots$$

Set of services S (# of refugees, school places, medical needs, etc)

- Family f requires $\nu_s^f \in \mathbb{Z}_{\geq 0}$ units of service s
- Locality ℓ can provide $\kappa_s^\ell \in \mathbb{Z}_{\geq 0}$ units of service s
- "Null" locality $\emptyset \in L$ with infinite capacity: $\kappa_s^\emptyset = +\infty$ for all $s \in S$

Model

Set of families F, set of localities L

- Families have strict and ordinal preferences over localities

$$\succ_f : \ell_1, \ell_2, \ell_3, \dots$$

- Each locality strictly ranks families in order of priority

$$\rhd_\ell : f_1, f_2, f_3, \dots$$

Set of services S (# of refugees, school places, medical needs, etc)

- Family f requires $\nu_s^f \in \mathbb{Z}_{\geq 0}$ units of service s
- Locality ℓ can provide $\kappa_s^\ell \in \mathbb{Z}_{\geq 0}$ units of service s
- "Null" locality $\emptyset \in L$ with infinite capacity: $\kappa_s^\emptyset = +\infty$ for all $s \in S$

A matching μ:

- Assigns every family $f \in F$ to a locality $\mu(f) \in L$ (possibly the null)

Model

Set of families F, set of localities L

- Families have strict and ordinal preferences over localities

$$\succ_f : \ell_1, \ell_2, \ell_3, \dots$$

- Each locality strictly ranks families in order of priority

$$\rhd_\ell : f_1, f_2, f_3, \dots$$

Set of services S (# of refugees, school places, medical needs, etc)

- Family f requires $\nu_s^f \in \mathbb{Z}_{\geq 0}$ units of service s
- Locality ℓ can provide $\kappa_s^\ell \in \mathbb{Z}_{\geq 0}$ units of service s
- "Null" locality $\emptyset \in L$ with infinite capacity: $\kappa_s^\emptyset = +\infty$ for all $s \in S$

A matching μ:

- Assigns every family $f \in F$ to a locality $\mu(f) \in L$ (possibly the null)
- Satisfies all multidimensional constraints:

$$\sum_{f \in \mu(\ell)} \nu_s^f \leq \kappa_s^\ell \quad \text{for all} \quad \ell \in L \text{ and } s \in S$$

Plan

Part I: Improve upon an endowment

- Start with a matching (e.g., matching on observables)
- Use refugee preferences to find Pareto improvements
- Mechanism: Multidimensional Top Trading Cycles with Endowment

Part II: Account for refugee preferences and locality priorities

- Priorities come from administrative rules and/or preferences
- Solution concept: Weak Envy-freeness
- Trade-off between efficiency and strategy-proofness

Incorporating Preferences

A natural place to start is the matching based on observables

- Call this the endowment μ^E

Incorporating Preferences

A natural place to start is the matching based on observables

- Call this the endowment μ^E

Can we use preferences to improve upon that endowment?

- We need a mechanism that returns an individually rational matching, i.e., a matching μ that every family weakly prefers to the endowment: $\mu(f) \succeq \mu^E(f)$ for all $f \in F$.

Incorporating Preferences

A natural place to start is the matching based on observables

- Call this the endowment μ^E

Can we use preferences to improve upon that endowment?

- We need a mechanism that returns an individually rational matching, i.e., a matching μ that every family weakly prefers to the endowment: $\mu(f) \succeq \mu^E(f)$ for all $f \in F$.

We use a modified version of Top Trading Cycles

Top Trading Cycles in School Choice

Suppose we are in the *school choice* environment

- $|S| = 1$ and $\nu_s^f = 1$ for all $f \in F$

Top Trading Cycles in School Choice

Suppose we are in the *school choice* environment

- $|S| = 1$ and $\nu_s^f = 1$ for all $f \in F$

Top Trading Cycles mechanism

- Every family points at its most preferred locality
- Every locality point at its highest-priority family
- Every family in a cycle is matched to the family at which it is pointing
- The capacity of a locality that receives a family is reduced by one unit

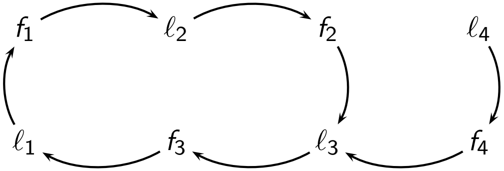
Top Trading Cycles in School Choice

Suppose we are in the *school choice* environment

- $|S| = 1$ and $\nu_s^f = 1$ for all $f \in F$

Top Trading Cycles mechanism

- Every family points at its most preferred locality
- Every locality point at its highest-priority family
- Every family in a cycle is matched to the family at which it is pointing
- The capacity of a locality that receives a family is reduced by one unit



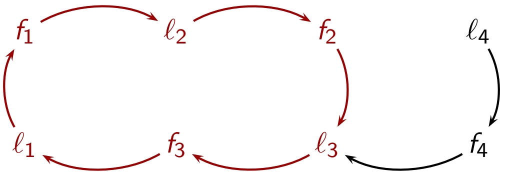
Top Trading Cycles in School Choice

Suppose we are in the *school choice* environment

- $|S| = 1$ and $\nu_s^f = 1$ for all $f \in F$

Top Trading Cycles mechanism

- Every family points at its most preferred locality
- Every locality point at its highest-priority family
- Every family in a cycle is matched to the family at which it is pointing
- The capacity of a locality that receives a family is reduced by one unit



Top Trading Cycles in School Choice

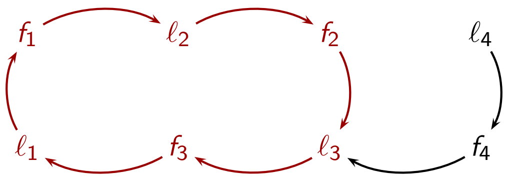
Suppose we are in the *school choice* environment

- $|S| = 1$ and $\nu_s^f = 1$ for all $f \in F$

Top Trading Cycles mechanism

- Every family points at its most preferred locality
- Every locality point at its highest-priority family
- Every family in a cycle is matched to the family at which it is pointing
- The capacity of a locality that receives a family is reduced by one unit

Adding an endowment does not make a difference



Top Trading Cycles in School Choice

Suppose we are in the *school choice* environment

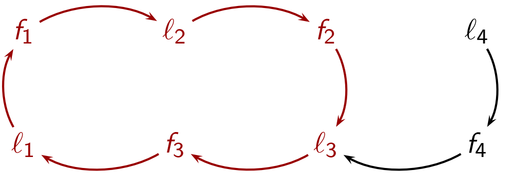
- $|S| = 1$ and $\nu_s^f = 1$ for all $f \in F$

Top Trading Cycles mechanism

- Every family points at its most preferred locality
- Every locality point at its highest-priority family
- Every family in a cycle is matched to the family at which it is pointing
- The capacity of a locality that receives a family is reduced by one unit

Adding an endowment does not make a difference

- A priority can be interpreted as an endowment



Top Trading Cycles in Refugee Resettlement

TTC can easily be adapted to our enviroment

- Multidimensional Top Trading Cycles (MTTC) mechanism
- Only difference: when a family is matched to a locality, the locality's capacity for each service is reduced by the number of units of that service that the family requires

Top Trading Cycles in Refugee Resettlement

TTC can easily be adapted to our enviroment

- Multidimensional Top Trading Cycles (MTTC) mechanism
- Only difference: when a family is matched to a locality, the locality's capacity for each service is reduced by the number of units of that service that the family requires

Proposition

The MTTC mechanism is strategy-proof and Pareto efficient.

Top Trading Cycles in Refugee Resettlement

TTC can easily be adapted to our enviroment

- Multidimensional Top Trading Cycles (MTTC) mechanism
- Only difference: when a family is matched to a locality, the locality's capacity for each service is reduced by the number of units of that service that the family requires

Proposition

The MTTC mechanism is strategy-proof and Pareto efficient.

Improving upon an endowment is challenging

- Families of different sizes may not be able to swap with each other
- This problem does not occur in school choice

Pareto-improving Chains

Identifying Pareto-improvements is challenging

- Families with different needs may have to move simultaneously
- A mechanism that finds an individually rational and Pareto-efficient is computationally intractable

Pareto-improving Chains

Identifying Pareto-improvements is challenging

- Families with different needs may have to move simultaneously
- A mechanism that finds an individually rational and Pareto-efficient is computationally intractable

We focus on Pareto-improving chains

- Each family in the chain takes the place of the next one

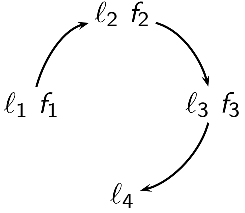
Pareto-improving Chains

Identifying Pareto-improvements is challenging

- Families with different needs may have to move simultaneously
- A mechanism that finds an individually rational and Pareto-efficient is computationally intractable

We focus on Pareto-improving chains

- Each family in the chain takes the place of the next one
- The last family in the chain either
  - moves to a locality where it can be accommodated ("open" chain)



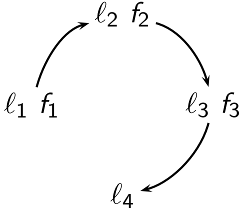
Pareto-improving Chains

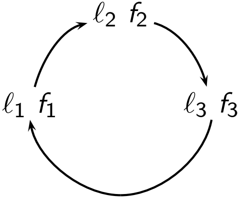
Identifying Pareto-improvements is challenging

- Families with different needs may have to move simultaneously
- A mechanism that finds an individually rational and Pareto-efficient is computationally intractable

We focus on Pareto-improving chains

- Each family in the chain takes the place of the next one
- The last family in the chain either
  - moves to a locality where it can be accommodated ("open" chain) or
  - takes the place of the first one ("closed" chain)





Chain Efficiency

Definition

A matching is chain-efficient if it has no Pareto-improving chains

Chain Efficiency

Definition

A matching is chain-efficient if it has no Pareto-improving chains

Chain efficiency is a relaxation of Pareto efficiency

- There may still be Pareto-improvements that are not chains

Chain Efficiency

Definition

A matching is chain-efficient if it has no Pareto-improving chains

Chain efficiency is a relaxation of Pareto efficiency

- There may still be Pareto-improvements that are not chains

Individually rational and chain-efficient matching:

- Carry out one Pareto-improving chain until none remains

Chain Efficiency

Definition

A matching is chain-efficient if it has no Pareto-improving chains

Chain efficiency is a relaxation of Pareto efficiency

- There may still be Pareto-improvements that are not chains

Individually rational and chain-efficient matching:

- Carry out one Pareto-improving chain until none remains
- Not clear it works in polynomial time

Chain Efficiency

Definition

A matching is chain-efficient if it has no Pareto-improving chains

Chain efficiency is a relaxation of Pareto efficiency

- There may still be Pareto-improvements that are not chains

Individually rational and chain-efficient matching:

- Carry out one Pareto-improving chain until none remains
- Not clear it works in polynomial time

Theorem

There does not exist any individually rational, chain-efficient, and strategy-proof mechanism.

Pareto Improvement

Pareto Improvement

Theorem

When $|S| > 1$, there does not exist any strategy-proof mechanism that Pareto improves upon every chain-inefficient endowment.

Pareto Improvement

Theorem

When $|S| > 1$, there does not exist any strategy-proof mechanism that Pareto improves upon every chain-inefficient endowment.

Theorem

When $|S| = 1$, there exists a strategy-proof mechanism that Pareto improves upon every chain-inefficient endowment.

Pareto Improvement

Theorem

When $|S| > 1$, <u>there does not exist</u> any strategy-proof mechanism that Pareto improves upon every chain-inefficient endowment.

Theorem

When $|S| = 1$, <u>there exists</u> a strategy-proof mechanism that Pareto improves upon every chain-inefficient endowment.

Multidimensional Top-Trading Cycles with Endowment (MTTCE).

- Identifies and carries out chains to improve upon the endowment
- Individually rational and strategy-proof
- Pareto-improves upon any chain-inefficient endowment when $|S| = 1$

Plan

- Part I: Improve upon an endowment
  - Start with a matching (e.g., matching on observables)
  - Use refugee preferences to find Pareto improvements
  - Mechanism: Multidimensional Top Trading Cycles with Endowment

Part II: Account for refugee preferences and locality priorities
  - Priorities come from administrative rules and/or preferences
  - Solution concept: Weak Envy-freeness
  - Trade-off between efficiency and strategy-proofness

Respecting Priorities

Respecting Priorities

Goodwill from localities matters

- They provide services
- Respecting priorities can increase their willingness to participate

Respecting Priorities

Goodwill from localities matters

- They provide services
- Respecting priorities can increase their willingness to participate

How to account for both preferences and priorities?

- Stability is the natural solution concept

Example from McDermid and Manlove (2010)

Preferences

$f_1 : \ell_2, \ell_1, \emptyset$

$f_2 : \ell_1, \ell_2, \emptyset$

$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$

$\ell_2 : f_2, f_1, f_3$

Example from McDermid and Manlove (2010)

Preferences

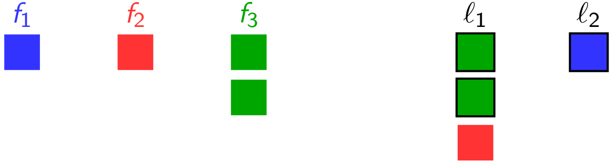
$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

Round 1		
f_1	$\rightarrow$	ℓ_2
f_2	$\rightarrow$	ℓ_1
f_3	$\rightarrow$	ℓ_1

Example from McDermid and Manlove (2010)



Preferences

$f_1 : \ell_2, \ell_1, \emptyset$

$f_2 : \ell_1, \ell_2, \emptyset$

$f_3 : \ell_1, \emptyset$

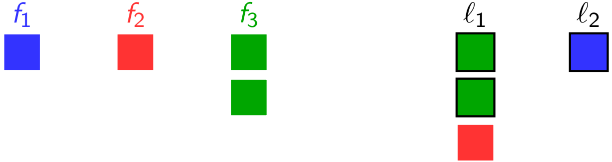
Priorities

$\ell_1 : f_1, f_3, f_2$

$\ell_2 : f_2, f_1, f_3$

Round 1		
f_1	$\rightarrow$	ℓ_2
f_2	$\rightarrow$	ℓ_1
f_3	$\rightarrow$	ℓ_1

Example from McDermid and Manlove (2010)



Preferences

$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

Round 1			
f_1	$\rightarrow$	ℓ_2	✓
f_2	$\rightarrow$	ℓ_1	✗
f_3	$\rightarrow$	ℓ_1	✓

Example from McDermid and Manlove (2010)

Preferences

$f_1 : \ell_2, \ell_1, \emptyset$

$f_2 : \ell_1, \ell_2, \emptyset$

$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$

$\ell_2 : f_2, f_1, f_3$

Round 1				Round 2		
f_1	$\rightarrow$	ℓ_2	✓	f_1	$\rightarrow$	ℓ_2
f_2	$\rightarrow$	ℓ_1	✗	f_2	$\rightarrow$	ℓ_2
f_3	$\rightarrow$	ℓ_1	✓	f_3	$\rightarrow$	ℓ_1

Example from McDermid and Manlove (2010)

Preferences

$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

Round 1				Round 2		
f_1	$\rightarrow$	ℓ_2	✓	f_1	$\rightarrow$	ℓ_2
f_2	$\rightarrow$	ℓ_1	✗	f_2	$\rightarrow$	ℓ_2
f_3	$\rightarrow$	ℓ_1	✓	f_3	$\rightarrow$	ℓ_1

Example from McDermid and Manlove (2010)

Preferences

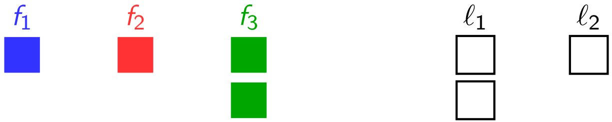
$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

Round 1				Round 2			
f_1	$\rightarrow$	ℓ_2	✓	f_1	$\rightarrow$	ℓ_2	✗
f_2	$\rightarrow$	ℓ_1	✗	f_2	$\rightarrow$	ℓ_2	✓
f_3	$\rightarrow$	ℓ_1	✓	f_3	$\rightarrow$	ℓ_1	✓

Example from McDermid and Manlove (2010)



Preferences

$f_1 : \ell_2, \ell_1, \emptyset$

$f_2 : \ell_1, \ell_2, \emptyset$

$f_3 : \ell_1, \emptyset$

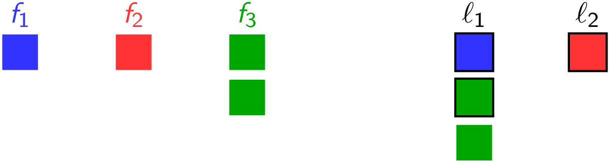
Priorities

$\ell_1 : f_1, f_3, f_2$

$\ell_2 : f_2, f_1, f_3$

Round 1				Round 2				Round 3		
f_1	$\rightarrow$	ℓ_2	✓	f_1	$\rightarrow$	ℓ_2	✗	f_1	$\rightarrow$	ℓ_1
f_2	$\rightarrow$	ℓ_1	✗	f_2	$\rightarrow$	ℓ_2	✓	f_2	$\rightarrow$	ℓ_2
f_3	$\rightarrow$	ℓ_1	✓	f_3	$\rightarrow$	ℓ_1	✓	f_3	$\rightarrow$	ℓ_1

Example from McDermid and Manlove (2010)



Preferences

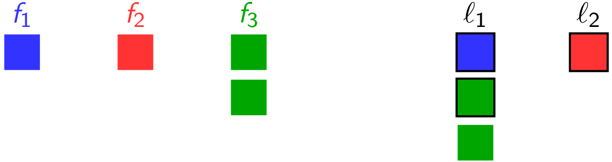
$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

Round 1				Round 2				Round 3		
f_1	$\to$	ℓ_2	✓	f_1	$\to$	ℓ_2	✗	f_1	$\to$	ℓ_1
f_2	$\to$	ℓ_1	✗	f_2	$\to$	ℓ_2	✓	f_2	$\to$	ℓ_2
f_3	$\to$	ℓ_1	✓	f_3	$\to$	ℓ_1	✓	f_3	$\to$	ℓ_1

Example from McDermid and Manlove (2010)



Preferences

$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

Round 1				Round 2				Round 3			
f_1	$\rightarrow$	ℓ_2	✓	f_1	$\rightarrow$	ℓ_2	✗	f_1	$\rightarrow$	ℓ_1	✓
f_2	$\rightarrow$	ℓ_1	✗	f_2	$\rightarrow$	ℓ_2	✓	f_2	$\rightarrow$	ℓ_2	✓
f_3	$\rightarrow$	ℓ_1	✓	f_3	$\rightarrow$	ℓ_1	✓	f_3	$\rightarrow$	ℓ_1	✗

Example from McDermid and Manlove (2010)

Preferences

$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

Round 1				Round 2				Round 3				Round 4		
f_1	$\rightarrow$	ℓ_2	✓	f_1	$\rightarrow$	ℓ_2	✗	f_1	$\rightarrow$	ℓ_1	✓	f_1	$\rightarrow$	ℓ_1
f_2	$\rightarrow$	ℓ_1	✗	f_2	$\rightarrow$	ℓ_2	✓	f_2	$\rightarrow$	ℓ_2	✓	f_2	$\rightarrow$	ℓ_2
f_3	$\rightarrow$	ℓ_1	✓	f_3	$\rightarrow$	ℓ_1	✓	f_3	$\rightarrow$	ℓ_1	✗	f_3	$\rightarrow$	$\emptyset$

Example from McDermid and Manlove (2010)

Preferences

$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

Round 1				Round 2				Round 3				Round 4			
f_1	$\rightarrow$	ℓ_2	✓	f_1	$\rightarrow$	ℓ_2	✗	f_1	$\rightarrow$	ℓ_1	✓	f_1	$\rightarrow$	ℓ_1	
f_2	$\rightarrow$	ℓ_1	✗	f_2	$\rightarrow$	ℓ_2	✓	f_2	$\rightarrow$	ℓ_2	✓	f_2	$\rightarrow$	ℓ_2	
f_3	$\rightarrow$	ℓ_1	✓	f_3	$\rightarrow$	ℓ_1	✓	f_3	$\rightarrow$	ℓ_1	✗	f_3	$\rightarrow$	$\emptyset$	

Example from McDermid and Manlove (2010)

Preferences

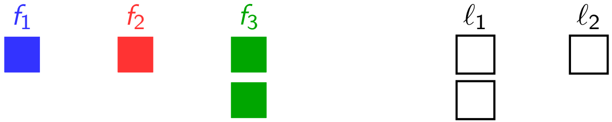
$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

Round 1				Round 2				Round 3				Round 4			
f_1	$\rightarrow$	ℓ_2	✓	f_1	$\rightarrow$	ℓ_2	✗	f_1	$\rightarrow$	ℓ_1	✓	f_1	$\rightarrow$	ℓ_1	✓
f_2	$\rightarrow$	ℓ_1	✗	f_2	$\rightarrow$	ℓ_2	✓	f_2	$\rightarrow$	ℓ_2	✓	f_2	$\rightarrow$	ℓ_2	✓
f_3	$\rightarrow$	ℓ_1	✓	f_3	$\rightarrow$	ℓ_1	✓	f_3	$\rightarrow$	ℓ_1	✗	f_3	$\rightarrow$	$\emptyset$	✓

Example from McDermid and Manlove (2010)



Preferences

$f_1 : \ell_2, \ell_1, \emptyset$

$f_2 : \ell_1, \ell_2, \emptyset$

$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$

$\ell_2 : f_2, f_1, f_3$

Example from McDermid and Manlove (2010)

Preferences

$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

If f_3 is matched to ℓ_1

Example from McDermid and Manlove (2010)

Preferences

$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

If f_3 is matched to ℓ_1

Example from McDermid and Manlove (2010)

Preferences

$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

If f_3 is matched to ℓ_1

- (f_2, ℓ_2) is a blocking pair unless f_2 is matched to ℓ_2

Example from McDermid and Manlove (2010)

Preferences

$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

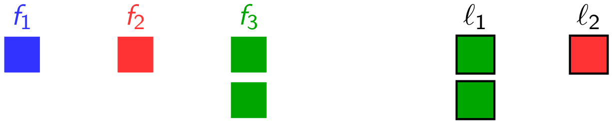
Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

If f_3 is matched to ℓ_1

- (f_2, ℓ_2) is a blocking pair unless f_2 is matched to ℓ_2

Example from McDermid and Manlove (2010)



Preferences

$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

If f_3 is matched to ℓ_1

- (f_2, ℓ_2) is a blocking pair unless f_2 is matched to ℓ_2
- But then f_1 remains unmatched and (f_1, ℓ_1) is a blocking pair

Example from McDermid and Manlove (2010)

Preferences

$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

If f_3 is matched to ℓ_1

- (f_2, ℓ_2) is a blocking pair unless f_2 is matched to ℓ_2
- But then f_1 remains unmatched and (f_1, ℓ_1) is a blocking pair
- Therefore, f_3 is not matched to ℓ_1 in any stable matching

Example from McDermid and Manlove (2010)

Preferences

$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

If f_3 is *not* matched to ℓ_1

Example from McDermid and Manlove (2010)

Preferences

$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

If f_3 is *not* matched to ℓ_1

- (f_2, ℓ_1) is a blocking pair unless f_2 is matched to ℓ_1

Example from McDermid and Manlove (2010)

Preferences

$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

If f_3 is *not* matched to ℓ_1

- (f_2, ℓ_1) is a blocking pair unless f_2 is matched to ℓ_1

Example from McDermid and Manlove (2010)

Preferences

$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

If f_3 is *not* matched to ℓ_1

- (f_2, ℓ_1) is a blocking pair unless f_2 is matched to ℓ_1
- (f_1, ℓ_2) is a blocking pair unless f_1 is matched to ℓ_2

Example from McDermid and Manlove (2010)

Preferences

$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

If f_3 is *not* matched to ℓ_1

- (f_2, ℓ_1) is a blocking pair unless f_2 is matched to ℓ_1
- (f_1, ℓ_2) is a blocking pair unless f_1 is matched to ℓ_2

Example from McDermid and Manlove (2010)

Preferences

$f_1 : \ell_2, \ell_1, \emptyset$
$f_2 : \ell_1, \ell_2, \emptyset$
$f_3 : \ell_1, \emptyset$

Priorities

$\ell_1 : f_1, f_3, f_2$
$\ell_2 : f_2, f_1, f_3$

If f_3 is *not* matched to ℓ_1

- (f_2, ℓ_1) is a blocking pair unless f_2 is matched to ℓ_1
- (f_1, ℓ_2) is a blocking pair unless f_1 is matched to ℓ_2
- But then (f_3, ℓ_1) is a blocking pair

Respecting Priorities

Respecting Priorities

Goodwill from localities matters

- They provide services
- Respecting priorities can increase their willingness to participate

How to account for both preferences and priorities?

- Stability is the natural solution concept

Respecting Priorities

Goodwill from localities matters

- They provide services
- Respecting priorities can increase their willingness to participate

How to account for both preferences and priorities?

- Stability is the natural solution concept

Stable matchings are not guaranteed to exist

Respecting Priorities

Goodwill from localities matters

- They provide services
- Respecting priorities can increase their willingness to participate

How to account for both preferences and priorities?

- Stability is the natural solution concept

Stable matchings are not guaranteed to exist

Underusing capacities may be tolerable in refugee resettlement

- We propose a solution concept that respects priorities but may underuse some capacity

(Weak) Envy-freeness

(Weak) Envy-freeness

Given a matching μ, family g envies family f if g prefers f's locality to its own and has a higher priority for it:

$$\mu(f) \succ_g \mu(g) \quad \text{and} \quad g \rhd_{\mu(f)} f$$

(Weak) Envy-freeness

Given a matching μ, family g envies family f if g prefers f's locality to its own and has a higher priority for it:

$$\mu(f) \succ_g \mu(g) \quad \text{and} \quad g \rhd_{\mu(f)} f$$

A matching is envy-free if no family envies another family

- Envy-freeness respects priorities but may underuse capacity

(Weak) Envy-freeness

Given a matching μ, family g envies family f if g prefers f's locality to its own and has a higher priority for it:

$$\mu(f) \succ_g \mu(g) \quad \text{and} \quad g \rhd_{\mu(f)} f$$

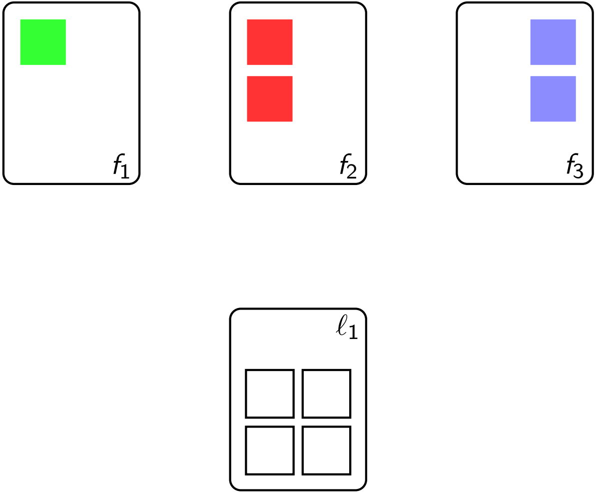
A matching is envy-free if no family envies another family

- Envy-freeness respects priorities but may underuse capacity

Weak envy-freeness is a relaxation

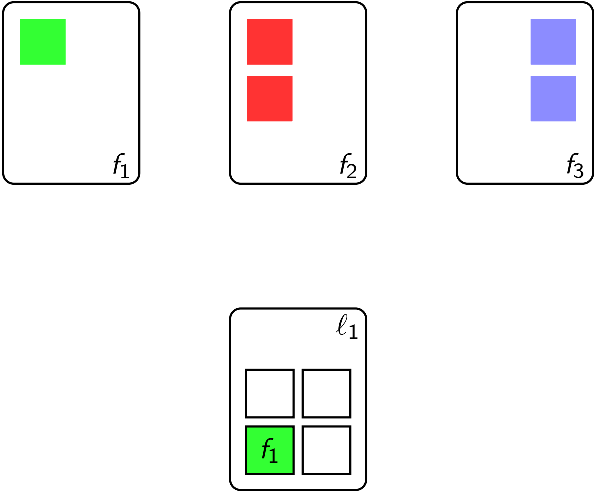
- A family f can be envied if it "fits" even when all families that envy f are matched to $\mu(f)$

Weak Envy-freeness



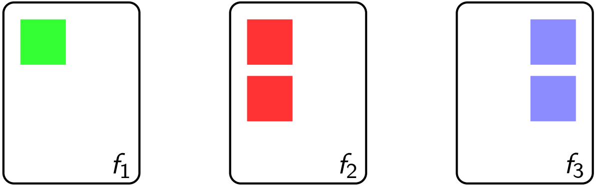
Priority: $f_1 \triangleright f_2 \triangleright f_3$

Weak Envy-freeness

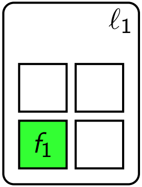


Priority: $f_1 \rhd f_2 \rhd f_3$

Weak Envy-freeness

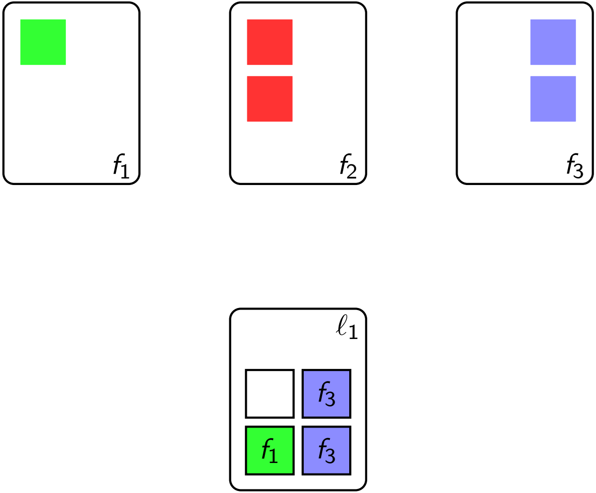


Envy-free and weakly envy-free



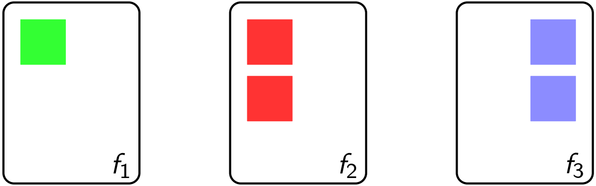
Priority: $f_1 \rhd f_2 \rhd f_3$

Weak Envy-freeness

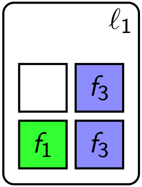


Priority: $f_1 \rhd f_2 \rhd f_3$

Weak Envy-freeness

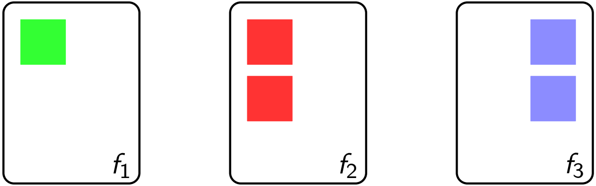


Not envy-free

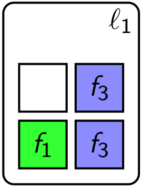


Priority: $f_1 \rhd f_2 \rhd f_3$

Weak Envy-freeness

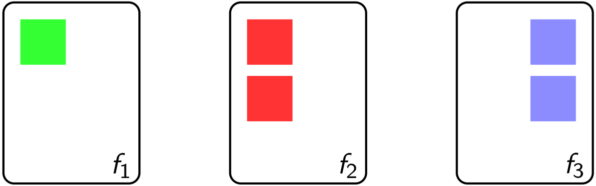


Not envy-free but weakly envy-free

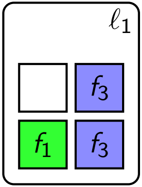


Priority: $f_1 \triangleright f_2 \triangleright f_3$

Weak Envy-freeness

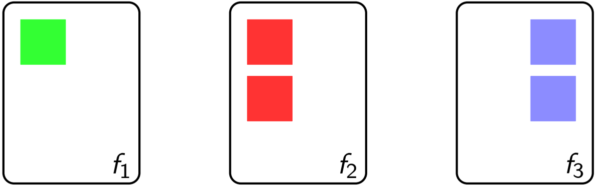


f_2 envies f_3 but f_3 fits even when f_2 is there

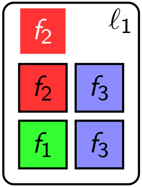


Priority: $f_1 \triangleright f_2 \triangleright f_3$

Weak Envy-freeness



f_2 envies f_3 but f_3 fits even when f_2 is there



Priority: $f_1 \triangleright f_2 \triangleright f_3$

Cascading Multidimensional DA (CMDA)

Round $k \geq 1$

Cascading Multidimensional DA (CMDA)

Round $k \geq 1$

- Every family proposes to the locality it prefers among those that have not rejected it

Cascading Multidimensional DA (CMDA)

Round $k \geq 1$

- Every family proposes to the locality it prefers among those that have not rejected it
- Every locality accepts a family if this does not violate weak envy-freeness and rejects the family otherwise

Cascading Multidimensional DA (CMDA)

Round $k \geq 1$

- Every family proposes to the locality it prefers among those that have not rejected it
- Every locality accepts a family if this does not violate weak envy-freeness and rejects the family otherwise
- If every family is accepted, end; otherwise, go to Round $k+1$

Cascading Multidimensional DA (CMDA)

Round $k \geq 1$

- Every family proposes to the locality it prefers among those that have not rejected it
- Every locality accepts a family if this does not violate weak envy-freeness and rejects the family otherwise
- If every family is accepted, end; otherwise, go to Round $k + 1$

Theorem

There exists a unique family-optimal weakly envy-free matching and the CMDA algorithm finds it.

Strategy Proofness

The CMDA mechanism is not strategy-proof

- Due to the different sizes, a family may gain by not proposing to localities that reject it

Strategy Proofness

The CMDA mechanism is not strategy-proof

- Due to the different sizes, a family may gain by not proposing to localities that reject it

Solution: make the acceptance rule of localities harsher

- Threshold Multidimensional Deferred Acceptance (TMDA)
- Weakly envy-free and strategy-proof but not family-optimal

Conclusion

Refugee resettlement is a matching problem

- Optimally matching families and localities has long-term consequences
- Multidimensional constraints make it a complex matching problem
- Matching over observables has been done in practice

Solutions to account for preferences

- Using preferences has the potential to further improve the outcome
- Improvement over an endowment

Solutions to account for preferences and priorities

- Localities goodwill is important
- Solution concept: weak envy-freeness

Applications are just starting

- Applications will in turn inform theory