

New stable matching problems arising from national matching schemes in Hungary

Péter Biró

Department of Computing Science

University of Glasgow

pbiro@dcs.gla.ac.uk

Joint work with Tamás Fleiner, Rob Irving and David Manlove

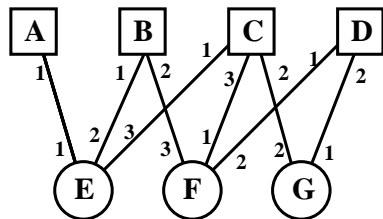
Bristol Algorithm Day
12 May 2009



Stable Marriage problem

Gale, Shapley [1962]:

“College admission and the stability of marriage”

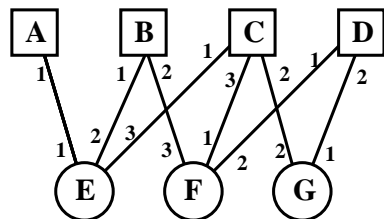


“Each person ranks those of the opposite sex in accordance with his or her preferences for a marriage partner.”

Stable Marriage problem

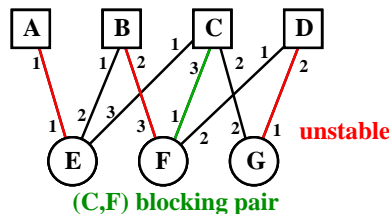
Gale, Shapley [1962]:

“College admission and the stability of marriage”



“Each person ranks those of the opposite sex in accordance with his or her preferences for a marriage partner.”

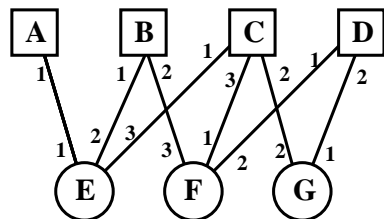
A set of marriages is **stable**, if there is no “**blocking pair**”: a man and a woman who are not married to each other but prefer each other to their actual mates.



Stable Marriage problem

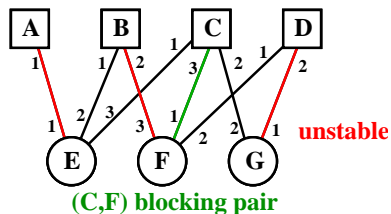
Gale, Shapley [1962]:

“College admission and the stability of marriage”



“Each person ranks those of the opposite sex in accordance with his or her preferences for a marriage partner.”

A set of marriages is **stable**, if there is no “**blocking pair**”: a man and a woman who are not married to each other but prefer each other to their actual mates.

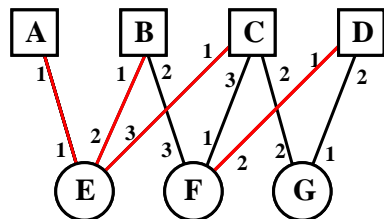


Gale-Shapley (1962): The deferred-acceptance algorithm finds a stable matching in $O(m)$ time.

Stable Marriage problem

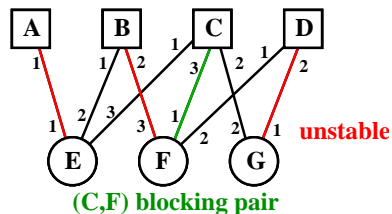
Gale, Shapley [1962]:

“College admission and the stability of marriage”



“Each person ranks those of the opposite sex in accordance with his or her preferences for a marriage partner.”

A set of marriages is **stable**, if there is no “**blocking pair**”: a man and a woman who are not married to each other but prefer each other to their actual mates.

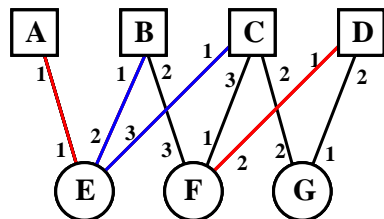


Gale-Shapley (1962): The deferred-acceptance algorithm finds a stable matching in $O(m)$ time.

Stable Marriage problem

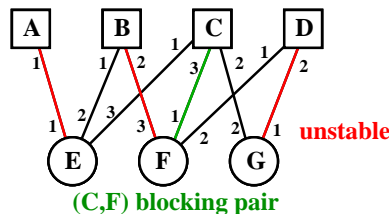
Gale, Shapley [1962]:

“College admission and the stability of marriage”



“Each person ranks those of the opposite sex in accordance with his or her preferences for a marriage partner.”

A set of marriages is **stable**, if there is no “**blocking pair**”: a man and a woman who are not married to each other but prefer each other to their actual mates.

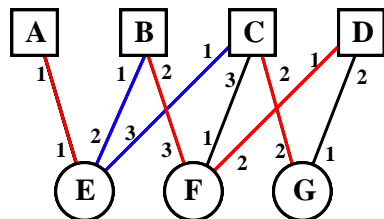


Gale-Shapley (1962): The deferred-acceptance algorithm finds a stable matching in $O(m)$ time.

Stable Marriage problem

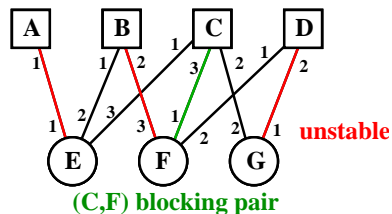
Gale, Shapley [1962]:

“College admission and the stability of marriage”



“Each person ranks those of the opposite sex in accordance with his or her preferences for a marriage partner.”

A set of marriages is **stable**, if there is no “**blocking pair**”: a man and a woman who are not married to each other but prefer each other to their actual mates.

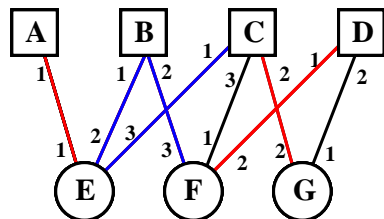


Gale-Shapley (1962): The deferred-acceptance algorithm finds a stable matching in $O(m)$ time.

Stable Marriage problem

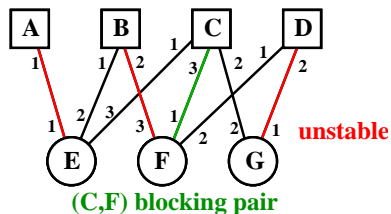
Gale, Shapley [1962]:

“College admission and the stability of marriage”



“Each person ranks those of the opposite sex in accordance with his or her preferences for a marriage partner.”

A set of marriages is **stable**, if there is no “**blocking pair**”: a man and a woman who are not married to each other but prefer each other to their actual mates.

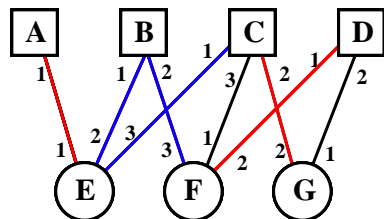


Gale-Shapley (1962): The deferred-acceptance algorithm finds a stable matching in $O(m)$ time.

Stable Marriage problem

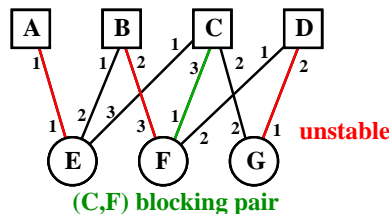
Gale, Shapley [1962]:

“College admission and the stability of marriage”



“Each person ranks those of the opposite sex in accordance with his or her preferences for a marriage partner.”

A set of marriages is **stable**, if there is no “**blocking pair**”: a man and a woman who are not married to each other but prefer each other to their actual mates.



Gale-Shapley (1962): The deferred-acceptance algorithm finds a stable matching in $O(m)$ time. This matching is *man-optimal*.

Stable Marriage problem \longrightarrow College Admissions

Each college C has a *quota* q_C . (*Many-to-one*) *matching*:
no college can admit more applicants than its quota.

Stable Marriage problem \longrightarrow College Admissions

Each college C has a *quota* q_C . (*Many-to-one*) matching:
no college can admit more applicants than its quota.

The **stability** of a matching:

If an applicant a_i is not matched to a college C , then

- either a_i is matched to a better college
- or C has filled its quota with better applicants than a_i

Stable Marriage problem \longrightarrow College Admissions

Each college C has a *quota* q_C . (*Many-to-one*) matching:
no college can admit more applicants than its quota.

The **stability** of a matching:

If an applicant a_i is not matched to a college C , then

- either a_i is matched to a better college
- or C has filled its quota with better applicants than a_i

applicant-proposing algorithm:

- each applicant proposes to her favorite college
- each college C who receives more than q_C proposals rejects all but its q_C favorite applicants...

Stable Marriage problem \longrightarrow College Admissions

Each college C has a *quota* q_C . (*Many-to-one*) matching:
no college can admit more applicants than its quota.

The **stability** of a matching:

If an applicant a_i is not matched to a college C , then

- either a_i is matched to a better college
- or C has filled its quota with better applicants than a_i

applicant-proposing algorithm:

- each applicant proposes to her favorite college
- each college C who receives more than q_C proposals rejects all but its q_C favorite applicants...

college-proposing algorithm:

- each college C proposes to its q_C favorite applicants
- each applicant who receives more than one offer rejects all but her favorite...

Nice properties of this centralised mechanism:

- ▶ **Fair:** If an application was rejected by a college then all places must have been filled there with better applicants.

Nice properties of this centralised mechanism:

- ▶ **Fair:** If an application was rejected by a college then all places must have been filled there with better applicants.
- ▶ **Optimal:** No student could be admitted to a better college in any other fair solution.

Nice properties of this centralised mechanism:

- ▶ **Fair:** If an application was rejected by a college then all places must have been filled there with better applicants.
- ▶ **Optimal:** No student could be admitted to a better college in any other fair solution.
- ▶ **Strategy-proof:** No student can end up in a better college by giving false preferences.

Nice properties of this centralised mechanism:

- ▶ **Fair:** If an application was rejected by a college then all places must have been filled there with better applicants.
- ▶ **Optimal:** No student could be admitted to a better college in any other fair solution.
- ▶ **Strategy-proof:** No student can end up in a better college by giving false preferences.
- ▶ **Fast:** Linear-time (10 sec in Hungary!)

Nice properties of this centralised mechanism:

- ▶ **Fair:** If an application was rejected by a college then all places must have been filled there with better applicants.
- ▶ **Optimal:** No student could be admitted to a better college in any other fair solution.
- ▶ **Strategy-proof:** No student can end up in a better college by giving false preferences.
- ▶ **Fast:** Linear-time (10 sec in Hungary!)

+practical benefits:

no **frustration** for the students (whether to accept an early offer)

no **fluctuation** in the number of students (at popular places)

The Gale–Shapley algorithm in practice

Allocating residents to positions:

- ▶ National Resident Matching Program in the US since 1952!
- ▶ and in many other professions / countries...
e.g., **Scottish Foundation Allocation Scheme since 2000**

The Gale–Shapley algorithm in practice

Allocating residents to positions:

- ▶ National Resident Matching Program in the US since 1952!
- ▶ and in many other professions / countries...
e.g., **Scottish Foundation Allocation Scheme since 2000**

Admission systems in the education:

- ▶ New York high schools since 2004,
Boston high schools since 2005
- ▶ Higher education admissions in Spain (1998)
- ▶ **Higher education admissions in Hungary since 1985**
- ▶ **Secondary school admissions in Hungary since 2000**
(Original Gale–Shapley model and algorithm!)

Higher education admissions in Hungary

Applicants has ranking lists over particular studies they apply for:
study (field, type, state/private financed), faculty, university

Adam's list			
1.	CS, regular, state financed, BME		
2.	Maths, regular, state financed, BME		
3.	CS, regular, privately financed , BME		
4.	Economics, regular, state financed, Corvinus		

Higher education admissions in Hungary

Applicants has ranking lists over particular studies they apply for:
study (field, type, state/private financed), faculty, university

Adam's list		score	
1.	CS, regular, state financed, BME	117	
2.	Maths, regular, state financed, BME	108	
3.	CS, regular, privately financed , BME	117	
4.	Economics, regular, state financed, Corvinus	113	

Applicants receive scores at each place they applied for.

Higher education admissions in Hungary

Applicants has ranking lists over particular studies they apply for:
study (field, type, state/private financed), faculty, university

Adam's list		score	limit
1.	CS, regular, state financed, BME	117	121
2.	Maths, regular, state financed, BME	108	108
3.	CS, regular, privately financed , BME	117	110
4.	Economics, regular, state financed, Corvinus	113	114

Applicants receive scores at each place they applied for.

A centralised program computes the score-limits.

Higher education admissions in Hungary

Applicants has ranking lists over particular studies they apply for:
study (field, type, state/private financed), faculty, university

Adam's list		score	limit
1.	CS, regular, state financed, BME	117	121
2.	Maths, regular, state financed, BME	108	108
3.	CS, regular, privately financed , BME	117	110
4.	Economics, regular, state financed, Corvinus	113	114

Applicants receive scores at each place they applied for.

A centralised program computes the score-limits.

Each applicant is assigned to the **first place** on his list, where his score is **greater than or equal to** the limit.

1st special feature: ties (and stable score-limits)

Two applicants with the same score are in a **tie** at a college. They are both accepted or rejected together! (equal treatment policy...)

1st special feature: ties (and stable score-limits)

Two applicants with the same score are in a **tie** at a college. They are both accepted or rejected together! (equal treatment policy...)

A score-limit l is a mapping $l : \text{Colleges} \rightarrow \mathbb{N}$ (actually ≤ 480).

Every score-limit l produces a particular matching $M(l)$.

1st special feature: ties (and stable score-limits)

Two applicants with the same score are in a **tie** at a college. They are both accepted or rejected together! (equal treatment policy...)

A score-limit l is a mapping $l : \text{Colleges} \rightarrow \mathbb{N}$ (actually ≤ 480).

Every score-limit l produces a particular matching $M(l)$.

A score-limit l is **feasible** if no quota is exceeded in $M(l)$.

1st special feature: ties (and stable score-limits)

Two applicants with the same score are in a **tie** at a college. They are both accepted or rejected together! (equal treatment policy...)

A score-limit l is a mapping $l : \text{Colleges} \rightarrow \mathbb{N}$ (actually ≤ 480).

Every score-limit l produces a particular matching $M(l)$.

A score-limit l is **feasible** if no quota is exceeded in $M(l)$.

A score-limit l is **stable** if it is feasible and no college can decrease its limit without violating its quota (assuming that the others do not change their limits).

1st special feature: ties (and stable score-limits)

Two applicants with the same score are in a **tie** at a college. They are both accepted or rejected together! (equal treatment policy...)

A score-limit l is a mapping $l : \text{Colleges} \rightarrow \mathbb{N}$ (actually ≤ 480).

Every score-limit l produces a particular matching $M(l)$.

A score-limit l is **feasible** if no quota is exceeded in $M(l)$.

A score-limit l is **stable** if it is feasible and no college can decrease its limit without violating its quota (assuming that the others do not change their limits).

Observation: if no two applicants have the same scores at any college then l is stable $\iff M(l)$ is stable in the original sense.

Generalisation of the Gale-Shapley theorem

The Gale-Shapley algorithm can be generalised in an obvious way:
If the admission of the last tie would result a quota-violation then nobody from the last tie may get an offer!

B. (2007): The applicant / college-oriented algorithms produce stable score-limits (denoted by l_A and l_C).

Generalisation of the Gale-Shapley theorem

The Gale-Shapley algorithm can be generalised in an obvious way: If the admission of the last tie would result a quota-violation then nobody from the last tie may get an offer!

B. (2007): The applicant / college-oriented algorithms produce stable score-limits (denoted by l_A and l_C).

We say that $l \leq l'$ if $l(K) \leq l'(K)$ for each college K .

B. (2007): If l_* is a stable score-limit, then $l_A \leq l_* \leq l_C$.
(so l_A is the best and l_C is the worst for the applicants!)

Generalisation of the Gale-Shapley theorem

The Gale-Shapley algorithm can be generalised in an obvious way: If the admission of the last tie would result a quota-violation then nobody from the last tie may get an offer!

B. (2007): The **applicant** / **college**-oriented algorithms produce stable score-limits (denoted by l_A and l_C).

We say that $l \leq l'$ if $l(K) \leq l'(K)$ for each college K .

B. (2007): If l_* is a stable score-limit, then $l_A \leq l_* \leq l_C$.
(so l_A is the best and l_C is the worst for the applicants!)

Application: the **college-oriented** algorithm was replaced with the **applicant-oriented** version in 2007.

2nd special feature: lower quotas

Each college C has lower quota, $l(C)$ and upper quota, $u(C)$.

A solution is a matching, where each college C has either

- no assignees (“closed college”) or
- at least $l(C)$ and at most $u(C)$ assignees (“open college”).

2nd special feature: lower quotas

Each college C has lower quota, $l(C)$ and upper quota, $u(C)$.

A solution is a matching, where each college C has either

- no assignees (“closed college”) or
- at least $l(C)$ and at most $u(C)$ assignees (“open college”).

A matching is **stable** if there exist no

- “**blocking pair**”, consisting of an open college and an unsatisfied applicant,
- “**blocking coalition**”, consisting of a closed college C and $l(C)$ unsatisfied applicants.

An unsolvable instance, hardness

Studies:	Saxophone	Trompet
lower and upper quotas	$1 \leq \dots \leq 1$	$2 \leq \dots \leq 2$
1st applicant:	Adam	Adam
2nd applicant:	Bill	Bill

Adam's list: Trompet, Saxophone

Bill's list: Saxophone, Trompet

An unsolvable instance, hardness

Studies:	Saxophone	Trompet
lower and upper quotas	$1 \leq \dots \leq 1$	$2 \leq \dots \leq 2$
1st applicant:	Adam	Adam
2nd applicant:	Bill	Bill

Adam's list: Trompet, Saxophone

Bill's list: Saxophone, Trompet

An unsolvable instance, hardness

Studies:	Saxophone	Trompet
lower and upper quotas	$1 \leq \dots \leq 1$	$2 \leq \dots \leq 2$
1st applicant:	Adam	Adam
2nd applicant:	Bill	Bill

Adam's list: Trompet, Saxophone

Bill's list: Saxophone, Trompet

An unsolvable instance, hardness

Studies:	Saxophone	Trompet
lower and upper quotas	$1 \leq \dots \leq 1$	$2 \leq \dots \leq 2$
1st applicant:	Adam	Adam
2nd applicant:	Bill	Bill

Adam's list: Trompet, Saxophone

Bill's list: **Saxophone**, Trompet

An unsolvable instance, hardness

Studies:	Saxophone	Trompet
lower and upper quotas	$1 \leq \dots \leq 1$	$2 \leq \dots \leq 2$
1st applicant:	Adam	Adam
2nd applicant:	Bill	Bill

Adam's list: Trompet, Saxophone

Bill's list: Saxophone, Trompet

An unsolvable instance, hardness

Studies:	Saxophone	Trompet
lower and upper quotas	$1 \leq \dots \leq 1$	$2 \leq \dots \leq 2$
1st applicant:	Adam	Adam
2nd applicant:	Bill	Bill

Adam's list: Trompet, Saxophone

Bill's list: Saxophone, Trompet

An unsolvable instance, hardness

Studies:	Saxophone	Trompet
lower and upper quotas	$1 \leq \dots \leq 1$	$2 \leq \dots \leq 2$
1st applicant:	Adam	Adam
2nd applicant:	Bill	Bill

Adam's list: Trompet, Saxophone

Bill's list: Saxophone, Trompet

An unsolvable instance, hardness

Studies:	Saxophone	Trompet
lower and upper quotas	$1 \leq \dots \leq 1$	$2 \leq \dots \leq 2$
1st applicant:	Adam	Adam
2nd applicant:	Bill	Bill

Adam's list: Trompet, Saxophone

Bill's list: Saxophone, Trompet

B.-Fleiner-Irving-Manlove (2009): Stable matching may not exist, and the related decision problem is NP-complete, even if no upper quota exceeds 3.

An unsolvable instance, hardness

Studies:	Saxophone	Trompet
lower and upper quotas	$1 \leq \dots \leq 1$	$2 \leq \dots \leq 2$
1st applicant:	Adam	Adam
2nd applicant:	Bill	Bill

Adam's list: Trompet, Saxophone

Bill's list: Saxophone, Trompet

B.-Fleiner-Irving-Manlove (2009): Stable matching may not exist, and the related decision problem is NP-complete, even if no upper quota exceeds 3.

Open: if no lower quota exceeds 2.

An unsolvable instance, hardness

Studies:	Saxophone	Trompet
lower and upper quotas	$1 \leq \dots \leq 1$	$2 \leq \dots \leq 2$
1st applicant:	Adam	Adam
2nd applicant:	Bill	Bill

Adam's list: Trompet, Saxophone

Bill's list: Saxophone, Trompet

B.-Fleiner-Irving-Manlove (2009): Stable matching may not exist, and the related decision problem is NP-complete, even if no upper quota exceeds 3.

Open: if no lower quota exceeds 2.

A heuristic is used in the application.

3rd special feature: common quotas

Some set of colleges may have a **common quota**.
(this quota cannot be exceeded in a matching...)

3rd special feature: common quotas

Some set of colleges may have a **common quota**.
(this quota cannot be exceeded in a matching...)

The **stability** of a matching:

If an applicant a_i is not matched to a college C , then

- either a_i is matched to a better college
- or C has filled its quota with better applicants than a_i
- or there is a set of colleges S such that $C \in S$ and S filled its quota with better applicants.

Common quotas in the application

Studies:	p. CS_{BME}	s. CS_{BME}	...	s. CS_{GD}	...
c. quotas:		CS national quota: ≤ 3000			
quotas:	≤ 50	≤ 450	...	≤ 400	...
2004:	49 (78p)	474 (113p)	...	336 (74p)	...
2005:	51 (90p)	423 (126p)	...	369 (77p)	...
2006:	41 (80p)	443 (125p)	...	321 (78p)	...
2007:	51 (100p)	478 (120p)	...	246 (79p)	...

Common quotas in the application

Studies:	p. CS_{BME}	s. CS_{BME}	...	s. CS_{GD}	...
c. quotas:		CS national quota: ≤ 3000			
quotas:	≤ 50	≤ 450	...	≤ 400	...
2004:	49 (78p)	474 (113p)	...	336 (74p)	...
2005:	51 (90p)	423 (126p)	...	369 (77p)	...
2006:	41 (80p)	443 (125p)	...	321 (78p)	...
2007:	51 (100p)	478 (120p)	...	246 (79p)	...

Studies:	p. CS_{BME}	s. CS_{BME}	...	s. CS_{GD}	...
c. quotas:		CS national quota: ≤ 3000			
c. quotas:	faculty quota: ≤ 500		...	≤ 400	...
2008:	8 (365p)	492 (366p)	...	165 (160p)	...

Nested set systems: nice properties

A set system \mathcal{C} is **nested** if, for every pair of sets in \mathcal{C} such that $S \cap S' \neq \emptyset$, either $S \subseteq S'$ or $S \supseteq S'$ holds.

Nested set systems: nice properties

A set system \mathcal{C} is **nested** if, for every pair of sets in \mathcal{C} such that $S \cap S' \neq \emptyset$, either $S \subseteq S'$ or $S \supseteq S'$ holds.

B.-Fleiner-Irving-Manlove (2009): For nested set systems, stable matching always exists and it can be obtained by generalised Gale-Shapley type algorithms. Moreover, the **applicant** / **college**-oriented versions produce the **best** / **worst** possible stable matchings for the applicants.

Nested set systems: nice properties

A set system \mathcal{C} is **nested** if, for every pair of sets in \mathcal{C} such that $S \cap S' \neq \emptyset$, either $S \subseteq S'$ or $S \supseteq S'$ holds.

B.-Fleiner-Irving-Manlove (2009): For nested set systems, stable matching always exists and it can be obtained by generalised Gale-Shapley type algorithms. Moreover, the **applicant** / **college**-oriented versions produce the **best** / **worst** possible stable matchings for the applicants.

This was the case in the application until 2007!

Non-nested set systems: unsolvable and hard

Studies:	p. Sax_A	s. Sax_A	s. Sax_B	Trompet
common quotas:		Sax n. quota: ≤ 1		≤ 1
common quotas:	faculty quota: ≤ 1			
1st applicant:			Cliff	Adam
2nd applicant:		Bill		Cliff
3rd applicant:	Adam			

Adam's list: p. Sax_A, Trompet

Bill's list: s. Sax_A

Cliff's list: Trompet, s. Sax_B

Non-nested set systems: unsolvable and hard

Studies:	p. Sax_A	s. Sax_A	s. Sax_B	Trompet
common quotas:		Sax n. quota: ≤ 1		≤ 1
common quotas:	faculty quota: ≤ 1			
1st applicant:			Cliff	Adam
2nd applicant:		Bill		Cliff
3rd applicant:	Adam			

Adam's list: p. Sax_A, Trompet

Bill's list: s. Sax_A

Cliff's list: Trompet, s. Sax_B

Adam cannot be unmatched, otherwise he blocks with Trompet...

Non-nested set systems: unsolvable and hard

Studies:	p. Sax_A	s. Sax_A	s. Sax_B	Trompet
common quotas:		Sax n. quota: ≤ 1		≤ 1
common quotas:	faculty quota: ≤ 1			
1st applicant:			Cliff	Adam
2nd applicant:		Bill		Cliff
3rd applicant:	Adam			

Adam's list: p. Sax_A, Trompet

Bill's list: s. Sax_A

Cliff's list: Trompet, s. Sax_B

Suppose that Adam is matched to p. Sax_A,

Non-nested set systems: unsolvable and hard

Studies:	p. Sax_A	s. Sax_A	s. Sax_B	Trompet
common quotas:		Sax n. quota: ≤ 1		≤ 1
common quotas:	faculty quota: ≤ 1			
1st applicant:			Cliff	Adam
2nd applicant:		Bill		Cliff
3rd applicant:	Adam			

Adam's list: p. Sax_A, Trompet

Bill's list: s. Sax_A

Cliff's list: Trompet, s. Sax_B

Suppose that Adam is matched to p. Sax_A, then Cliff must be matched to Trompet and Bill **blocks** with s. Sax_A

Non-nested set systems: unsolvable and hard

Studies:	p. Sax_A	s. Sax_A	s. Sax_B	Trompet
common quotas:		Sax n. quota: ≤ 1		≤ 1
common quotas:	faculty quota: ≤ 1			
1st applicant:			Cliff	Adam
2nd applicant:		Bill		Cliff
3rd applicant:	Adam			

Adam's list: p. Sax_A, Trompet

Bill's list: s. Sax_A

Cliff's list: Trompet, s. Sax_B

Suppose that Adam is matched to Trompet...

Non-nested set systems: unsolvable and hard

Studies:	p. Sax_A	s. Sax_A	s. Sax_B	Trompet
common quotas:		Sax n. quota: ≤ 1		≤ 1
common quotas:	faculty quota: ≤ 1			
1st applicant:			Cliff	Adam
2nd applicant:		Bill		Cliff
3rd applicant:	Adam			

Adam's list: p. Sax_A, **Trompet**

Bill's list: s. Sax_A

Cliff's list: Trompet, **s. Sax_B**

Suppose that Adam is matched to Trompet...

Cliff must be matched to s. Sax_B, otherwise they block

Non-nested set systems: unsolvable and hard

Studies:	p. Sax_A	s. Sax_A	s. Sax_B	Trompet
common quotas:		Sax n. quota: ≤ 1		≤ 1
common quotas:	faculty quota: ≤ 1			
1st applicant:			Cliff	Adam
2nd applicant:		Bill		Cliff
3rd applicant:	Adam			

Adam's list: p. Sax_A, Trompet

Bill's list: s. Sax_A

Cliff's list: Trompet, s. Sax_B

Suppose that Adam is matched to Trompet...

Cliff must be matched to s. Sax_B, otherwise they block

So Bill cannot be matched to s. Sax_A,

Non-nested set systems: unsolvable and hard

Studies:	p. Sax_A	s. Sax_A	s. Sax_B	Trompet
common quotas:		Sax n. quota: ≤ 1		≤ 1
common quotas:	faculty quota: ≤ 1			
1st applicant:			Cliff	Adam
2nd applicant:		Bill		Cliff
3rd applicant:	Adam			

Adam's list: p. Sax_A, Trompet

Bill's list: s. Sax_A

Cliff's list: Trompet, s. Sax_B

Suppose that Adam is matched to Trompet...

Cliff must be matched to s. Sax_B, otherwise they block

So Bill cannot be matched to s. Sax_A,

Thus, Adam **blocks** with p. Sax_A.

Non-nested set systems: unsolvable and hard

Studies:	p. Sax_A	s. Sax_A	s. Sax_B	Trompet
common quotas:		Sax n. quota: ≤ 1		≤ 1
common quotas:	faculty quota: ≤ 1			
1st applicant:			Cliff	Adam
2nd applicant:		Bill		Cliff
3rd applicant:	Adam			

Adam's list: p. Sax_A, Trompet

Bill's list: s. Sax_A

Cliff's list: Trompet, s. Sax_B

B.-Fleiner-Irving-Manlove (2009): Stable matching may not exist, and the related decision problem is NP-complete, even if every quota is 1 and no college is included more than 2 sets.

Non-nested set systems: unsolvable and hard

Studies:	p. Sax_A	s. Sax_A	s. Sax_B	Trompet
common quotas:		Sax n. quota: ≤ 1		≤ 1
common quotas:	faculty quota: ≤ 1			
1st applicant:			Cliff	Adam
2nd applicant:		Bill		Cliff
3rd applicant:	Adam			

Adam's list: p. Sax_A, Trompet

Bill's list: s. Sax_A

Cliff's list: Trompet, s. Sax_B

B.-Fleiner-Irving-Manlove (2009): Stable matching may not exist, and the related decision problem is NP-complete, even if every quota is 1 and no college is included more than 2 sets.

This is the case in the application since 2007!

Future research

- ▶ **unsolved theoretical problems**
(e.g., lower quotas ≤ 2)

Future research

- ▶ **unsolved theoretical problems**
(e.g., lower quotas ≤ 2)
- ▶ **practical experiments**
(e.g., compare college-oriented and student-oriented solutions)

Future research

- ▶ **unsolved theoretical problems**
(e.g., lower quotas ≤ 2)
- ▶ **practical experiments**
(e.g., compare college-oriented and student-oriented solutions)
- ▶ **other countries? benefits of having a centralised scheme?**
(e.g., Spain vs Poland, or Hungary vs Slovakia)

Future research

- ▶ **unsolved theoretical problems**
(e.g., lower quotas ≤ 2)
- ▶ **practical experiments**
(e.g., compare college-oriented and student-oriented solutions)
- ▶ **other countries? benefits of having a centralised scheme?**
(e.g., Spain vs Poland, or Hungary vs Slovakia)

See more at www.optimalmatching.com!