Matchings and Latin Squares

Michael Simkin
Supervised by: Nati Linial

Institute of Mathematics and Federmann Center for the Study of Rationality, The Hebrew University of Jerusalem, Israel

Rationality 5778

Outline

1. Motivation
   - Stable Matchings
2. High-Dimensional Permutations
   - What is a High-Dimensional Permutation?
   - Latin Squares
   - Case Study: Monotone Subsequences in Latin Squares

Outline

1. Motivation
   - Stable Matchings
2. High-Dimensional Permutations
   - What is a High-Dimensional Permutation?
   - Latin Squares
   - Case Study: Monotone Subsequences in Latin Squares

A Familiar Example - Stable Matchings

Setup:

- There are n residents and n hospitals.
- Each hospital is to be assigned exactly one resident.
- Each resident has a ranking of the hospitals.
- Each hospital has a ranking of the residents.

Problem: We seek a *stable* matching of residents and hospitals.
A matching is stable if there is no resident-hospital pair that would prefer each other over their current assignment.

A Familiar Example - Stable Matchings

Setup:

- There are n residents and n hospitals.
- Each hospital is to be assigned exactly one resident.
- Each resident has a ranking of the hospitals.
- Each hospital has a ranking of the residents.

Problem: We seek a *stable* matching of residents and hospitals.

A matching is stable if there is no resident-hospital pair that would prefer each other over their current assignment.

A Familiar Example - Stable Matchings

Setup:

- There are n residents and n hospitals.
- Each hospital is to be assigned exactly one resident.
- Each resident has a ranking of the hospitals.
- Each hospital has a ranking of the residents.

Problem: We seek a *stable* matching of residents and hospitals.
A matching is stable if there is no resident-hospital pair that would prefer each other over their current assignment.

A Familiar Example - Stable Matchings

Setup:

- There are n residents and n hospitals.
- Each hospital is to be assigned exactly one resident.
- Each resident has a ranking of the hospitals.
- Each hospital has a ranking of the residents.

Problem: We seek a *stable* matching of residents and hospitals.
Solution: The Gale–Shapley algorithm efficiently finds a stable matching (Gale, Shapley, 1962).

A Recipe for Success

Matchings have numerous applications:

- Assigning residents to hospitals.
- Assigning clients to servers on the internet.
- Assigning students to *mechinot*.
- ...

Their success has two ingredients:

1. Binary relations (i.e., *graphs*) are ubiquitous. Matchings arise naturally from graphs.
2. There are many efficient algorithms for analysing graphs.

A Recipe for Success

Matchings have numerous applications:

- Assigning residents to hospitals.
- Assigning clients to servers on the internet.
- Assigning students to *mechinot*.
- ...

Their success has two ingredients:

1. Binary relations (i.e., *graphs*) are ubiquitous. Matchings arise naturally from graphs.
2. There are many efficient algorithms for analysing graphs.

Hospital/Resident/Attending Matching

Setup:

- There are n residents, n *attending physicians*, and n hospitals.
- Each hospital is to be assigned exactly one resident and one attending physician.
- Now the rankings are of *pairs*.

Problem: We seek a *stable* matching of residents and attendings to hospitals.
A matching is stable if there is no (resident, attending, hospital) *triple* that would prefer each other over their current assignment.

Hospital/Resident/Attending Matching

Setup:

- There are n residents, *n attending physicians*, and n hospitals.
- Each hospital is to be assigned exactly one resident and one attending physician.
- Now the rankings are of *pairs*.

Problem: We seek a *stable* matching of residents and attendings to hospitals.

A matching is stable if there is no (resident, attending, hospital) *triple* that would prefer each other over their current assignment.

Hospital/Resident/Attending Matching

Setup:

- There are n residents, *n attending physicians*, and n hospitals.
- Each hospital is to be assigned exactly one resident and one attending physician.
- Now the rankings are of *pairs*.

Problem: We seek a *stable* matching of residents and attendings to hospitals.
A matching is stable if there is no (resident, attending, hospital) *triple* that would prefer each other over their current assignment.

Hospital/Resident/Attending Matching

Setup:

- There are *n* residents, *n attending physicians*, and *n* hospitals.
- Each hospital is to be assigned exactly one resident and one attending physician.
- Now the rankings are of *pairs*.

Problem: We seek a *stable* matching of residents and attendings to hospitals.

Chaos!

- A stable matching need not exist.
- It is computationally difficult to determine if a stable matching exists (Ng, Hirschberg, 1991, Subramaniam 1994).

Hospital/Resident/Attending Matching

Setup:

- There are n residents, n *attending physicians*, and n hospitals.
- Each hospital is to be assigned exactly one resident and one attending physician.
- Now the rankings are of *pairs*.

Problem: We seek a *stable* matching of residents and attendings to hospitals.

Chaos!

- A stable matching need not exist.
- It is computationally difficult to determine if a stable matching exists (Ng, Hirschberg, 1991, Subramaniam 1994).

Where Should We Go from Here?

- Adding non-binary constraints makes matching difficult.
- Faced with this situation we wonder if there is an interesting theory of high-dimensional matching.

Outline

One-Dimensional Permutations

- A matching between sets of size n can be represented by a *permutation matrix* - an $n \times n$ $(0,1)$-matrix with exactly one 1 in each row and column.
- An order-n (one-dimensional) *permutation* is an ordering of the integers $\{1, 2, \ldots, n\}$.

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \leftrightarrow \begin{pmatrix} 4 & 5 & 1 & 3 & 2 \end{pmatrix}$$

One-Dimensional Permutations

- A matching between sets of size n can be represented by a *permutation matrix* - an $n \times n$ $(0,1)$-matrix with exactly one 1 in each row and column.
- An order-n (one-dimensional) *permutation* is an ordering of the integers $\{1, 2, \ldots, n\}$.

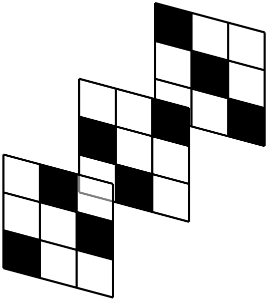
$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \leftrightarrow \begin{pmatrix} 4 & 5 & 1 & 3 & 2 \end{pmatrix}$$

High-Dimensional Permutations

- An order-n permutation is an $n \times n$ $(0,1)$-matrix with exactly one 1 in each row and column.
- An order-n *two-dimensional permutation* is an $n \times n \times n$ $(0,1)$-array with exactly one 1 in each row, column, and *shaft*.

High-Dimensional Permutations

- An order-n permutation is an $n \times n$ $(0, 1)$-matrix with exactly one 1 in each row and column.
- An order-n *two-dimensional permutation* is an $n \times n \times n$ $(0, 1)$-array with exactly one 1 in each row, column, and *shaft*.



Outline

1. Motivation
   - Stable Matchings
2. High-Dimensional Permutations
   - What is a High-Dimensional Permutation?
   - Latin Squares
   - Case Study: Monotone Subsequences in Latin Squares

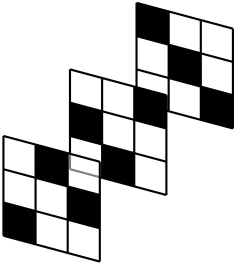
Two-Dimensional Permutations = Latin Squares

- An order-n *two-dimensional permutation* is an $n \times n \times n$ $(0,1)$-array with exactly one 1 in each row, column, and *shaft*.
- An order-n *Latin square* is an $n \times n$ matrix in which each row and column contains all the numbers $\{1, 2, \ldots, n\}$.
- These are naturally equivalent:

$$\longleftrightarrow \quad \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{pmatrix}$$

Two-Dimensional Permutations = Latin Squares

- An order-n *two-dimensional permutation* is an $n \times n \times n$ $(0, 1)$-array with exactly one 1 in each row, column, and *shaft*.
- An order-n *Latin square* is an $n \times n$ matrix in which each row and column contains all the numbers $\{1, 2, \ldots, n\}$.
- These are naturally equivalent:



$$\longleftrightarrow \quad \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{pmatrix}$$

Some Questions

1. How many order-n Latin squares are there?
2. What does a typical Latin square look like?
3. Is there a way to efficiently generate random Latin squares?
4. What is the probability that a random array contains a Latin square?
5. How do properties of (one-dimensional) permutations generalize to Latin squares?

Some Questions

1. How many order-n Latin squares are there?
2. What does a typical Latin square look like?
3. Is there a way to efficiently generate random Latin squares?
4. What is the probability that a random array contains a Latin square?
5. How do properties of (one-dimensional) permutations generalize to Latin squares?

Some Questions

1. How many order-n Latin squares are there?
2. What does a typical Latin square look like?
3. Is there a way to efficiently generate random Latin squares?
4. What is the probability that a random array contains a Latin square?
5. How do properties of (one-dimensional) permutations generalize to Latin squares?

Some Questions

1. How many order-n Latin squares are there?
2. What does a typical Latin square look like?
3. Is there a way to efficiently generate random Latin squares?
4. What is the probability that a random array contains a Latin square?
5. How do properties of (one-dimensional) permutations generalize to Latin squares?

Some Questions

1. How many order-n Latin squares are there?
2. What does a typical Latin square look like?
3. Is there a way to efficiently generate random Latin squares?
4. What is the probability that a random array contains a Latin square?
5. How do properties of (one-dimensional) permutations generalize to Latin squares?

Outline

Monotone Subsequences in Permutations

Definition

A *monotone subsequence* in a permutation is a sequence of 1s that either ascend from left to right or descend from left to right.

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\updownarrow$$

$$\begin{pmatrix} 4 & 5 & 1 & 3 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 5 & 4 & 3 \end{pmatrix}$$

Monotone Subsequences in Permutations

Definition

A *monotone subsequence* in a permutation is a sequence of 1s that either ascend from left to right or descend from left to right.

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \color{red}{1} \\ 0 & 0 & 0 & \color{red}{1} & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & \color{red}{1} & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} \color{red}{1} & 0 & 0 & 0 & 0 \\ 0 & \color{red}{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \color{red}{1} & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\updownarrow$$

$$\begin{pmatrix} 4 & \color{red}{5} & 1 & \color{red}{3} & \color{red}{2} \end{pmatrix}, \begin{pmatrix} \color{red}{1} & \color{red}{2} & 5 & \color{red}{4} & 3 \end{pmatrix}$$

The Erdős–Szekeres Theorem

Theorem (Erdős, Szekeres, 1935)

Every order-n permutation contains a monotone subsequence of length at least $\sqrt{n}$, and this is tight.

Proof.

"By example", on board. . . □

The Erdős–Szekeres Theorem

Theorem (Erdős, Szekeres, 1935)

Every order-n permutation contains a monotone subsequence of length at least $\sqrt{n}$, and this is tight.

Proof.

“By example”, on board. . . □

How Do We Generalize a Theorem about Permutations?

- We must first generalize the notion of “monotone subsequence” to higher dimensions.
- Here is a one-dimensional monotone subsequence:

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \leftrightarrow \begin{pmatrix} 4 & 5 & 1 & 3 & 2 \end{pmatrix}$$

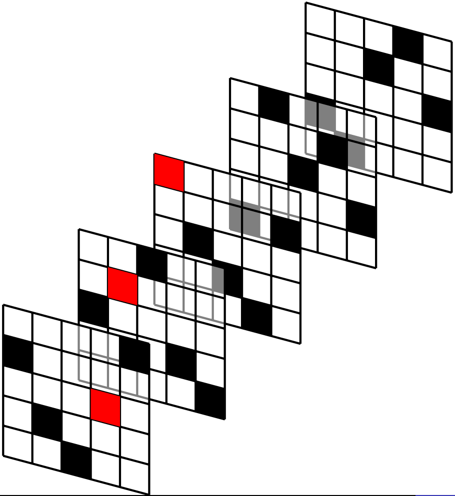
How Do We Generalize a Theorem about Permutations?

- We must first generalize the notion of “monotone subsequence” to higher dimensions.
- Here is a one-dimensional monotone subsequence:

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \color{red}{1} \\ 0 & 0 & 0 & \color{red}{1} & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & \color{red}{1} & 0 & 0 & 0 \end{pmatrix} \leftrightarrow \begin{pmatrix} 4 & \color{red}{5} & 1 & \color{red}{3} & \color{red}{2} \end{pmatrix}$$

Monotone Subsequences in Latin Squares

- This suggests the following:



$$\begin{pmatrix} 4 & 5 & 2 & 1 & 3 \\ 5 & 1 & 3 & 2 & 4 \\ \color{red}{1} & 3 & 4 & 5 & 2 \\ 3 & \color{red}{2} & 1 & 4 & 5 \\ 2 & 4 & 5 & \color{red}{3} & 1 \end{pmatrix}$$

Monotone Subsequences in Latin Squares

Definition (One dimension)

A *monotone subsequence* in a permutation is a sequence of 1s that either ascend from left to right or descend from left to right.

Definition (Two dimensions)

A *monotone subsequence* in a Latin square is a sequence of 1s whose positions are monotone in all three coordinates.

The Erdős–Szekeres Theorem for Latin Squares

Theorem (Erdős, Szekeres, 1935)

Every order-n permutation contains a monotone subsequence of length at least $\sqrt{n}$, and this is tight.

Theorem (Linial, S., 2017)

Every order-n Latin square contains a monotone subsequence of length at least $\frac{1}{3}\sqrt{n}$, and this is tight up to the multiplicative constant.

What About Typical Permutations?

Theorem (Logan, Shepp, 1977, Vershik, Kerov, 1977)

In almost every order-n permutation the longest monotone subsequence is of length $\approx 2\sqrt{n}$.

Theorem (Linial, S., 2017)

In almost every order-n Latin square the longest monotone subsequence is of length $\Theta\left(n^{2/3}\right)$.

What About Typical Permutations?

Theorem (Logan, Shepp, 1977, Vershik, Kerov, 1977)

In almost every order-n permutation the longest monotone subsequence is of length $\approx 2\sqrt{n}$.

Theorem (Linial, S., 2017)

In almost every order-n Latin square the longest monotone subsequence is of length $\Theta\left(n^{2/3}\right)$.

What About Typical Permutations?

Theorem (Logan, Shepp, 1977, Vershik, Kerov, 1977)

In almost every order-n permutation the longest monotone subsequence is of length $\approx 2\sqrt{n}$.

Theorem (Linial, S., 2017)

In almost every order-n Latin square the longest monotone subsequence is of length $\Theta\left(n^{2/3}\right)$.