The Nonnegative Inverse Eigenvalue Problem is NP-hard[1]

Alberto Borobia

Universidad Nacional de Educación a Distancia

Jornadas PEI (Madrid 2017)

[1] Published in LAA 522 127-139 (2017), a joint work with Roberto Canogar

Decision problems

Given $\Lambda \in \mathbb{C}^n$, is Λ the spectrum of a nonnegative matrix?

Decision problems

Given $\Lambda \in \mathbb{C}^n$, is Λ the spectrum of a nonnegative matrix?

Let $\mathcal{C}$ be sufficient conditions for the NIEP. Given $\Lambda \in \mathbb{C}^n$, does Λ satisfy $\mathcal{C}$?

Decision problems

Given $\Lambda \in \mathbb{C}^n$, is Λ the spectrum of a nonnegative matrix?

Let $\mathcal{C}$ be sufficient conditions for the NIEP. Given $\Lambda \in \mathbb{C}^n$, does Λ satisfy $\mathcal{C}$?

In computational complexity theory a **decision problem** *is a problem posed as a yes-no question of the input values*

Decision problems

Given $\Lambda \in \mathbb{C}^n$, is Λ the spectrum of a nonnegative matrix?

Let $\mathcal{C}$ be sufficient conditions for the NIEP. Given $\Lambda \in \mathbb{C}^n$, does Λ satisfy $\mathcal{C}$?

In computational complexity theory a **decision problem** *is a problem posed as a yes-no question of the input values*

Is prime a natural number n? Is hamiltonian a graph G? Is diagonalizable a matrix A?

Decision problems

Given $\Lambda \in \mathbb{C}^n$, is Λ the spectrum of a nonnegative matrix?

Let $\mathcal{C}$ be sufficient conditions for the NIEP. Given $\Lambda \in \mathbb{C}^n$, does Λ satisfy $\mathcal{C}$?

In computational complexity theory a **decision problem** *is a problem posed as a yes-no question of the input values*

Is prime a natural number n? Is hamiltonian a graph G? Is diagonalizable a matrix A?

Most of mathematical problems admit a decision version

Decision problems

Given $\Lambda \in \mathbb{C}^n$, is Λ the spectrum of a nonnegative matrix?

Let $\mathcal{C}$ be sufficient conditions for the NIEP. Given $\Lambda \in \mathbb{C}^n$, does Λ satisfy $\mathcal{C}$?

In computational complexity theory a **decision problem** *is a problem posed as a yes-no question of the input values*

Is prime a natural number n? Is hamiltonian a graph G? Is diagonalizable a matrix A?

Most of mathematical problems admit a decision version

1. Calculate the size of a **maximum matching** of a given graph G

   Decision: Given $G \in \mathcal{G}$ and $k \in \mathbb{N}$, does G have a matching of k edges?

Decision problems

Given $\Lambda \in \mathbb{C}^n$, is Λ the spectrum of a nonnegative matrix?

Let $\mathcal{C}$ be sufficient conditions for the NIEP. Given $\Lambda \in \mathbb{C}^n$, does Λ satisfy $\mathcal{C}$?

In computational complexity theory a **decision problem** *is a problem posed as a yes-no question of the input values*

Is prime a natural number n? Is hamiltonian a graph G? Is diagonalizable a matrix A?

Most of mathematical problems admit a decision version

1. Calculate the size of a **maximum matching** of a given graph G

   Decision: Given $G \in \mathcal{G}$ and $k \in \mathbb{N}$, does G have a matching of k edges?

2. **NIEP**: Characterize the set of spectra of nonnegative matrices

   Decision: Given $\Lambda \in \mathbb{C}^n$, is Λ the spectrum of a nonnegative matrix?

Certificate for a yes-instance of a decision problem

For a decision problem

- A **yes-instance** is an input that has output "yes"
- A **no-instance** is an input that has output "no"
- A **certificate** permits to check yes-instances (how to get it is not the question)

Certificate for a yes-instance of a decision problem

For a decision problem

- A **yes-instance** is an input that has output "yes"
- A **no-instance** is an input that has output "no"
- A **certificate** permits to check yes-instances (how to get it is not the question)

1. $\mathbf{D}_1$: Is eulerian a given graph G?
   - An eulerian circuit of G is a certificate for $\mathbf{D}_1$
   - The list of (even) out degrees of the vertices of G is a certificate for $\mathbf{D}_1$

Certificate for a yes-instance of a decision problem

For a decision problem

- A **yes-instance** is an input that has output "yes"
- A **no-instance** is an input that has output "no"
- A **certificate** permits to check yes-instances (how to get it is not the question)

1. $\mathbf{D}_1$: Is eulerian a given graph G?
   - An eulerian circuit of G is a certificate for $\mathbf{D}_1$
   - The list of (even) out degrees of the vertices of G is a certificate for $\mathbf{D}_1$
2. $\mathbf{D}_2$: Is $(\lambda_1; \lambda_2, \ldots, \lambda_n) \in \mathbb{N} \times (-\mathbb{N})^{n-1}$ the spectrum of a nonnegative matrix?
   The companion matrix of $f(x) = (x - \lambda_1) \cdots (x - \lambda_n)$ is a certificate for $\mathbf{D}_2$

Certificate for a yes-instance of a decision problem

For a decision problem

- A **yes-instance** is an input that has output "yes"
- A **no-instance** is an input that has output "no"
- A **certificate** permits to check yes-instances (how to get it is not the question)

1. $\mathbf{D}_1$: Is eulerian a given graph G?
   - An eulerian circuit of G is a certificate for $\mathbf{D}_1$
   - The list of (even) out degrees of the vertices of G is a certificate for $\mathbf{D}_1$
2. $\mathbf{D}_2$: Is $(\lambda_1; \lambda_2, \ldots, \lambda_n) \in \mathbb{N} \times (-\mathbb{N})^{n-1}$ the spectrum of a nonnegative matrix?
   The companion matrix of $f(x) = (x - \lambda_1) \cdots (x - \lambda_n)$ is a certificate for $\mathbf{D}_2$
3. $\mathbf{D}_3$: Is hamiltonian a given graph G?
   A hamiltonian cycle of G is a certificate for $\mathbf{D}_3$

Certificate for a yes-instance of a decision problem

For a decision problem

- A **yes-instance** is an input that has output "yes"
- A **no-instance** is an input that has output "no"
- A **certificate** permits to check yes-instances (how to get it is not the question)

1. $\mathbf{D}_1$: Is eulerian a given graph G?
   - An eulerian circuit of G is a certificate for $\mathbf{D}_1$
   - The list of (even) out degrees of the vertices of G is a certificate for $\mathbf{D}_1$
2. $\mathbf{D}_2$: Is $(\lambda_1; \lambda_2, \ldots, \lambda_n) \in \mathbb{N} \times (-\mathbb{N})^{n-1}$ the spectrum of a nonnegative matrix?
   The companion matrix of $f(x) = (x - \lambda_1) \cdots (x - \lambda_n)$ is a certificate for $\mathbf{D}_2$
3. $\mathbf{D}_3$: Is hamiltonian a given graph G?
   A hamiltonian cycle of G is a certificate for $\mathbf{D}_3$
4. $\mathbf{D}_4$: Is $(\lambda_1, \ldots, \lambda_n) \in \mathbb{C}^n$ the spectrum of a nonnegative matrix?
   A nonnegative matrix with spectrum $(\lambda_1, \ldots, \lambda_n)$ is a certificate for $\mathbf{D}_4$

Polynomial time reduction

S. Cook (1971): Let $\mathbf{D_1}$ and $\mathbf{D_2}$ decision problems. A **polynomial-time reduction** from $\mathbf{D_1}$ to $\mathbf{D_2}$ is a polynomial-time algorithm that transforms inputs from $\mathbf{D_1}$ to $\mathbf{D_2}$ so that

1. **Yes-instances**: yes-instances for $\mathbf{D_1}$ go to yes-instances for $\mathbf{D_2}$
2. **No-instances**: no-instances for $\mathbf{D_1}$ go to no-instances for $\mathbf{D_2}$

Polynomial time reduction

S. Cook (1971): Let $\mathbf{D_1}$ and $\mathbf{D_2}$ decision problems. A **polynomial-time reduction** from $\mathbf{D_1}$ to $\mathbf{D_2}$ is a polynomial-time algorithm that transforms inputs from $\mathbf{D_1}$ to $\mathbf{D_2}$ so that

1. **Yes-instances**: yes-instances for $\mathbf{D_1}$ go to yes-instances for $\mathbf{D_2}$
2. **No-instances**: no-instances for $\mathbf{D_1}$ go to no-instances for $\mathbf{D_2}$

Example

$\mathbf{D_1}$: Given a graph G without loops, has G a vertex cycle cover?

$\mathbf{D_2}$: Given a square real matrix A, is the permanent of A positive?

Polynomial time reduction

S. Cook (1971): Let $\mathbf{D_1}$ and $\mathbf{D_2}$ decision problems. A **polynomial-time reduction** from $\mathbf{D_1}$ to $\mathbf{D_2}$ is a polynomial-time algorithm that transforms inputs from $\mathbf{D_1}$ to $\mathbf{D_2}$ so that

1. **Yes-instances**: yes-instances for $\mathbf{D_1}$ go to yes-instances for $\mathbf{D_2}$
2. **No-instances**: no-instances for $\mathbf{D_1}$ go to no-instances for $\mathbf{D_2}$

Example

$\mathbf{D_1}$: Given a graph G without loops, has G a vertex cycle cover?

$\mathbf{D_2}$: Given a square real matrix A, is the permanent of A positive?

Let $\mathcal{G}$ the set of graphs without loops and let $\mathcal{M}$ is the set of square real matrices. A polynomial-time reduction from $\mathbf{D_1}$ to $\mathbf{D_2}$ is given by the function

$$\begin{array}{rccc} \phi: & \mathcal{G} & \longrightarrow & \mathcal{M} \\ & G & \mapsto & \phi(G) = M_G \end{array}$$

where $M(G)$ is the $1-0$ matrix whose (i,j) entry is 1 iff (i,j) is an edge of G.

Polynomial time reduction

S. Cook (1971): Let $\mathbf{D_1}$ and $\mathbf{D_2}$ decision problems. A **polynomial-time reduction** from $\mathbf{D_1}$ to $\mathbf{D_2}$ is a polynomial-time algorithm that transforms inputs from $\mathbf{D_1}$ to $\mathbf{D_2}$ so that

1. **Yes-instances**: yes-instances for $\mathbf{D_1}$ go to yes-instances for $\mathbf{D_2}$
2. **No-instances**: no-instances for $\mathbf{D_1}$ go to no-instances for $\mathbf{D_2}$

Example

$\mathbf{D_1}$: Given a graph G without loops, has G a vertex cycle cover?

$\mathbf{D_2}$: Given a square real matrix A, is the permanent of A positive?

Let $\mathcal{G}$ the set of graphs without loops and let $\mathcal{M}$ is the set of square real matrices. A polynomial-time reduction from $\mathbf{D_1}$ to $\mathbf{D_2}$ is given by the function

$$\begin{array}{rccc} \phi: & \mathcal{G} & \longrightarrow & \mathcal{M} \\ & G & \mapsto & \phi(G) = M_G \end{array}$$

where $M(G)$ is the $1-0$ matrix whose (i,j) entry is 1 iff (i,j) is an edge of G.

1. Yes-instances: If G is a yes-instance for $\mathbf{D_1}$ then $\phi(G)$ is a yes-instance for $\mathbf{D_2}$

Polynomial time reduction

S. Cook (1971): Let $\mathbf{D_1}$ and $\mathbf{D_2}$ decision problems. A **polynomial-time reduction** from $\mathbf{D_1}$ to $\mathbf{D_2}$ is a polynomial-time algorithm that transforms inputs from $\mathbf{D_1}$ to $\mathbf{D_2}$ so that

1. **Yes-instances**: yes-instances for $\mathbf{D_1}$ go to yes-instances for $\mathbf{D_2}$
2. **No-instances**: no-instances for $\mathbf{D_1}$ go to no-instances for $\mathbf{D_2}$

Example

$\mathbf{D_1}$: Given a graph G without loops, has G a vertex cycle cover?

$\mathbf{D_2}$: Given a square real matrix A, is the permanent of A positive?

Let $\mathcal{G}$ the set of graphs without loops and let $\mathcal{M}$ is the set of square real matrices. A polynomial-time reduction from $\mathbf{D_1}$ to $\mathbf{D_2}$ is given by the function

$$\begin{array}{rccc} \phi: & \mathcal{G} & \longrightarrow & \mathcal{M} \\ & G & \mapsto & \phi(G) = M_G \end{array}$$

where $M(G)$ is the $1-0$ matrix whose (i,j) entry is 1 iff (i,j) is an edge of G.

1. Yes-instances: If G is a yes-instance for $\mathbf{D_1}$ then $\phi(G)$ is a yes-instance for $\mathbf{D_2}$
2. No-instances: If G is a no-instance for $\mathbf{D_1}$ then $\phi(G)$ is a no-instance for $\mathbf{D_2}$ $\square$

The complexity class of a decision problem **D**

1. **P** (for polynomial time) if **D** can by decidable in polynomial time on all inputs

The complexity class of a decision problem **D**

1. **P** (for polynomial time) if **D** can by decidable in polynomial time on all inputs
   - The maximum matching decision problem

The complexity class of a decision problem **D**

1. **P** (for polynomial time) if **D** can by decidable in polynomial time on all inputs
   - The maximum matching decision problem
   - The decision versions of linear programming

The complexity class of a decision problem **D**

1. **P** (for polynomial time) if **D** can by decidable in polynomial time on all inputs
   - The maximum matching decision problem
   - The decision versions of linear programming
   - Determine if a number is prime

The complexity class of a decision problem **D**

1. **P** (for polynomial time) if **D** can by decidable in polynomial time on all inputs
   - The maximum matching decision problem
   - The decision versions of linear programming
   - Determine if a number is prime
2. **NP** (for nondeterministic polynomial time) if yes-instances of **D** have certificates which are verifiable in polynomial time

The complexity class of a decision problem **D**

1. **P** (for polynomial time) if **D** can by decidable in polynomial time on all inputs
   - The maximum matching decision problem
   - The decision versions of linear programming
   - Determine if a number is prime
2. **NP** (for nondeterministic polynomial time) if yes-instances of **D** have certificates which are verifiable in polynomial time
   - All decision problems in **P**

The complexity class of a decision problem **D**

1. **P** (for polynomial time) if **D** can by decidable in polynomial time on all inputs
   - The maximum matching decision problem
   - The decision versions of linear programming
   - Determine if a number is prime
2. **NP** (for nondeterministic polynomial time) if yes-instances of **D** have certificates which are verifiable in polynomial time
   - All decision problems in **P**
   - Hamiltonian cycle problem

The complexity class of a decision problem **D**

1. **P** (for polynomial time) if **D** can by decidable in polynomial time on all inputs
   - The maximum matching decision problem
   - The decision versions of linear programming
   - Determine if a number is prime
2. **NP** (for nondeterministic polynomial time) if yes-instances of **D** have certificates which are verifiable in polynomial time
   - All decision problems in **P**
   - Hamiltonian cycle problem
   - The clique decision problem

The complexity class of a decision problem **D**

1. **P** (for polynomial time) if **D** can by decidable in polynomial time on all inputs
   - The maximum matching decision problem
   - The decision versions of linear programming
   - Determine if a number is prime
2. **NP** (for nondeterministic polynomial time) if yes-instances of **D** have certificates which are verifiable in polynomial time
   - All decision problems in **P**
   - Hamiltonian cycle problem
   - The clique decision problem
   - The set packing problem

The complexity class of a decision problem **D**

1. **P** (for polynomial time) if **D** can by decidable in polynomial time on all inputs
   - The maximum matching decision problem
   - The decision versions of linear programming
   - Determine if a number is prime
2. **NP** (for nondeterministic polynomial time) if yes-instances of **D** have certificates which are verifiable in polynomial time
   - All decision problems in **P**
   - Hamiltonian cycle problem
   - The clique decision problem
   - The set packing problem
3. **NP-hard** if every **NP** problem can be **reduced** in polynomial time to **D**

The complexity class of a decision problem **D**

1. **P** (for polynomial time) if **D** can by decidable in polynomial time on all inputs
   - The maximum matching decision problem
   - The decision versions of linear programming
   - Determine if a number is prime
2. **NP** (for nondeterministic polynomial time) if yes-instances of **D** have certificates which are verifiable in polynomial time
   - All decision problems in **P**
   - Hamiltonian cycle problem
   - The clique decision problem
   - The set packing problem
3. **NP-hard** if every **NP** problem can be **reduced** in polynomial time to **D**
   - Hamiltonian cycle problem
   - The clique decision problem
   - The set packing problem

The complexity class of a decision problem **D**

1. **P** (for polynomial time) if **D** can by decidable in polynomial time on all inputs
   - The maximum matching decision problem
   - The decision versions of linear programming
   - Determine if a number is prime
2. **NP** (for nondeterministic polynomial time) if yes-instances of **D** have certificates which are verifiable in polynomial time
   - All decision problems in **P**
   - Hamiltonian cycle problem
   - The clique decision problem
   - The set packing problem
3. **NP-hard** if every **NP** problem can be **reduced** in polynomial time to **D**
   - Hamiltonian cycle problem
   - The clique decision problem
   - The set packing problem
   - There are **NP-hard** problems which are not **NP**: the halting problem.

The complexity class of a decision problem **D**

1. **P** (for polynomial time) if **D** can by decidable in polynomial time on all inputs
   - The maximum matching decision problem
   - The decision versions of linear programming
   - Determine if a number is prime
2. **NP** (for nondeterministic polynomial time) if yes-instances of **D** have certificates which are verifiable in polynomial time
   - All decision problems in **P**
   - Hamiltonian cycle problem
   - The clique decision problem
   - The set packing problem
3. **NP-hard** if every **NP** problem can be **reduced** in polynomial time to **D**
   - Hamiltonian cycle problem
   - The clique decision problem
   - The set packing problem
   - There are **NP-hard** problems which are not **NP**: the halting problem.
4. **NP-complete**[a] if **D** is **NP** and **NP-hard**

The complexity class of a decision problem **D**

1. **P** (for polynomial time) if **D** can by decidable in polynomial time on all inputs
   - The maximum matching decision problem
   - The decision versions of linear programming
   - Determine if a number is prime
2. **NP** (for nondeterministic polynomial time) if yes-instances of **D** have certificates which are verifiable in polynomial time
   - All decision problems in **P**
   - Hamiltonian cycle problem
   - The clique decision problem
   - The set packing problem
3. **NP-hard** if every **NP** problem can be **reduced** in polynomial time to **D**
   - Hamiltonian cycle problem
   - The clique decision problem
   - The set packing problem
   - There are **NP-hard** problems which are not **NP**: the halting problem.
4. **NP-complete**[a] if **D** is **NP** and **NP-hard**
   - Hamiltonian cycle problem
   - The clique decision problem
   - The set packing problem

[a] These are some of the 21 **NP-complete** decision problems of Karp's list (1972)

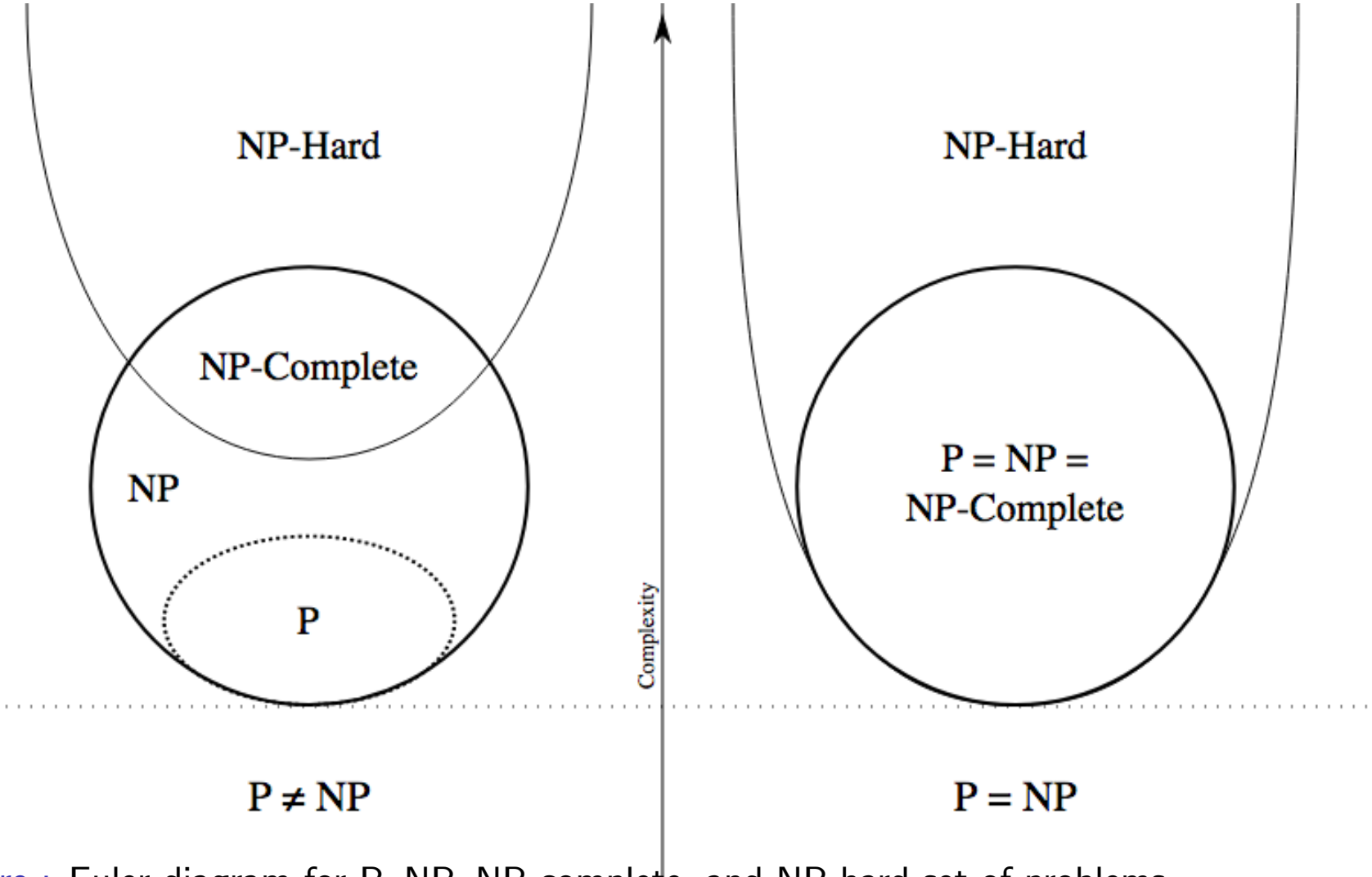


Figure : Euler diagram for P, NP, NP-complete, and NP-hard set of problems. The left side is valid if P≠NP, while the right side is valid if P=NP (author: Behnam Esfahbod)

The $(\Omega, \Pi)-$*Problem*

Given a subset Ω of the set Π, the $(\Omega, \Pi)-$**Problem** is a decision problem where

The $(\Omega, \Pi)-$*Problem*

Given a subset Ω of the set Π, the $(\Omega, \Pi)-$**Problem** is a decision problem where

- Inputs: the elements of Π

The $(\Omega, \Pi)-$*Problem*

Given a subset Ω of the set Π, the $(\Omega, \Pi)-$**Problem** is a decision problem where

- Inputs: the elements of Π
- Yes-instances: the inputs that are elements of Ω

The $(\Omega, \Pi)-$*Problem*

Given a subset Ω of the set Π, the $(\Omega, \Pi)-$**Problem** is a decision problem where

- Inputs: the elements of Π
- Yes-instances: the inputs that are elements of Ω
- No-instances: the inputs that are elements of $\Pi - \Omega$

The $(\Omega, \Pi)-$*Problem*

Given a subset Ω of the set Π, the $(\Omega, \Pi)-$**Problem** is a decision problem where

- Inputs: the elements of Π
- Yes-instances: the inputs that are elements of Ω
- No-instances: the inputs that are elements of $\Pi - \Omega$

The $(\Omega_1, \Pi_1)-$Problem is **polynomial-time reducible** to the $(\Omega_2, \Pi_2)-$Problem if there exists a polynomial time function between the input sets

$$\phi : \Pi_1 \longrightarrow \Pi_2$$

that transforms

The $(\Omega, \Pi)-$*Problem*

Given a subset Ω of the set Π, the $(\Omega, \Pi)-$**Problem** is a decision problem where

- Inputs: the elements of Π
- Yes-instances: the inputs that are elements of Ω
- No-instances: the inputs that are elements of $\Pi - \Omega$

The $(\Omega_1, \Pi_1)-$Problem is **polynomial-time reducible** to the $(\Omega_2, \Pi_2)-$Problem if there exists a polynomial time function between the input sets

$$\phi : \Pi_1 \longrightarrow \Pi_2$$

that transforms

(i) Yes-instances in yes-instances: $\phi(\Omega_1) \subset \Omega_2$

The $(\Omega, \Pi)-$*Problem*

Given a subset Ω of the set Π, the $(\Omega, \Pi)-$**Problem** is a decision problem where

- Inputs: the elements of Π
- Yes-instances: the inputs that are elements of Ω
- No-instances: the inputs that are elements of $\Pi - \Omega$

The $(\Omega_1, \Pi_1)-$Problem is **polynomial-time reducible** to the $(\Omega_2, \Pi_2)-$Problem if there exists a polynomial time function between the input sets

$$\phi : \Pi_1 \longrightarrow \Pi_2$$

that transforms

(i) Yes-instances in yes-instances: $\phi(\Omega_1) \subset \Omega_2$

(ii) No-instances inno-instances: $\phi(\Pi_1 - \Omega_1) \subset \Pi_2 - \Omega_2$

The $(\Pi_{PP}, \Pi_{\mathbb{N}})-$Problem or Partition Problem

Consider the set

$$\Pi_{\mathbb{N}} \equiv \left\{(i_1, \ldots, i_n) \in \mathbb{N}^n : n \in \mathbb{N}\right\}$$

and its subset

$$\Pi_{PP} \equiv \{I \in \Pi_{\mathbb{N}} : \exists \text{ a partition } I = J \cup K \text{ such that } \Sigma(J) = \Sigma(K)\}$$

The **Partition Problem** is the $(\Pi_{PP}, \Pi_{\mathbb{N}})-$Problem

The $(\Pi_{PP}, \Pi_{\mathbb{N}})-$Problem or Partition Problem

Consider the set

$$\Pi_{\mathbb{N}} \equiv \left\{(i_1, \ldots, i_n) \in \mathbb{N}^n : n \in \mathbb{N}\right\}$$

and its subset

$$\Pi_{PP} \equiv \{I \in \Pi_{\mathbb{N}} : \exists \text{ a partition } I = J \cup K \text{ such that } \Sigma(J) = \Sigma(K)\}$$

The **Partition Problem** is the $(\Pi_{PP}, \Pi_{\mathbb{N}})-$Problem

(i) $(9, 4, 4, 2, 1) \in \Pi_{PP} \subset \Pi_{\mathbb{N}}$ is a yes-instance of the Partition Problem

$$(9, 4, 4, 2, 1) = (9, 1) \cup (4, 4, 2)$$

The $(\Pi_{PP}, \Pi_{\mathbb{N}})-$Problem or Partition Problem

Consider the set

$$\Pi_{\mathbb{N}} \equiv \left\{(i_1, \ldots, i_n) \in \mathbb{N}^n : n \in \mathbb{N}\right\}$$

and its subset

$$\Pi_{PP} \equiv \{I \in \Pi_{\mathbb{N}} : \exists \text{ a partition } I = J \cup K \text{ such that } \Sigma(J) = \Sigma(K)\}$$

The **Partition Problem** is the $(\Pi_{PP}, \Pi_{\mathbb{N}})-$Problem

(i) $(9, 4, 4, 2, 1) \in \Pi_{PP} \subset \Pi_{\mathbb{N}}$ is a yes-instance of the Partition Problem

$$(9, 4, 4, 2, 1) = (9, 1) \cup (4, 4, 2)$$

(ii) $(8, 6, 4, 1) \in \Pi_{\mathbb{N}} - \Pi_{PP}$ is a no-instance of the Partition Problem

$$8 + 6 + 4 + 1 \quad \text{is odd}$$

The NIEP

Let $\Pi_{\mathbb{C}} = \mathbb{C} \cup \mathbb{C}^2 \cup \mathbb{C}^3 \cup \cdots$

Nonnegative Inverse Eigenvalue Problem (NIEP)

Characterize the **NIEP-set** $\Pi_{NIEP} \equiv \left\{ \Lambda \in \Pi_{\mathbb{C}} : \exists A \geq 0 \text{ with } \sigma(A) = \Lambda \right\}$

The NIEP

Let $\Pi_{\mathbb{C}} = \mathbb{C} \cup \mathbb{C}^2 \cup \mathbb{C}^3 \cup \cdots$

Nonnegative Inverse Eigenvalue Problem (NIEP)

Characterize the **NIEP-set** $\Pi_{NIEP} \equiv \left\{ \Lambda \in \Pi_{\mathbb{C}} : \exists A \geq 0 \text{ with } \sigma(A) = \Lambda \right\}$

The decision version of the NIEP is the $(\Pi_{NIEP}, \Pi_{\mathbb{C}}) - Problem$

The NIEP

Let $\Pi_{\mathbb{C}} = \mathbb{C} \cup \mathbb{C}^2 \cup \mathbb{C}^3 \cup \cdots$

Nonnegative Inverse Eigenvalue Problem (NIEP)

Characterize the **NIEP-set** $\Pi_{NIEP} \equiv \left\{ \Lambda \in \Pi_{\mathbb{C}} : \exists A \geq 0 \text{ with } \sigma(A) = \Lambda \right\}$

The decision version of the NIEP is the $(\Pi_{NIEP}, \Pi_{\mathbb{C}})-$*Problem*

(i) $(6, -3) \in \Pi_{\mathbb{C}}$ is a yes-instance of the $(\Pi_{\text{NIEP}}, \Pi_{\mathbb{C}})-$Problem since $(6, -3) = \sigma(\left[\begin{smallmatrix} 1 & 5 \\ 4 & 2 \end{smallmatrix}\right])$

The NIEP

Let $\Pi_{\mathbb{C}} = \mathbb{C} \cup \mathbb{C}^2 \cup \mathbb{C}^3 \cup \cdots$

Nonnegative Inverse Eigenvalue Problem (NIEP)

Characterize the **NIEP-set** $\Pi_{NIEP} \equiv \left\{ \Lambda \in \Pi_{\mathbb{C}} : \exists A \geq 0 \text{ with } \sigma(A) = \Lambda \right\}$

The decision version of the NIEP is the $(\Pi_{NIEP}, \Pi_{\mathbb{C}})-$*Problem*

(i) $(6, -3) \in \Pi_{\mathbb{C}}$ is a yes-instance of the $(\Pi_{\text{NIEP}}, \Pi_{\mathbb{C}})-$Problem since $(6, -3) = \sigma(\left[\begin{smallmatrix} 1 & 5 \\ 4 & 2 \end{smallmatrix}\right])$

(ii) $(3, -2, -2) \in \Pi_{\mathbb{C}}$ is a no-instance of the $(\Pi_{\text{NIEP}}, \Pi_{\mathbb{C}})-$Problem since the trace of a nonnegative matrix is nonnegative

The decision $(X, \Pi_{\mathbb{C}})-Problem$ $(X-Problem$ *by short*$)$

Each $X \subset \Pi_{NIEP}$ *has associated the decision* $(X, \Pi_{\mathbb{C}})-Problem$

The decision $(X, \Pi_{\mathbb{C}})-$*Problem (*$X-$*Problem by short)*

Each $X \subset \Pi_{NIEP}$ *has associated the decision* $(X, \Pi_{\mathbb{C}})-$*Problem*

Subsets X of Π_{NIEP} which are useful for our interests

The **Suleĭmanova Set**

$$\Pi_{\text{Su}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^n : n \in \mathbb{N};\ \lambda_1 \geq 0 > \lambda_2 \geq \cdots \geq \lambda_n;\ \lambda_1 + \cdots + \lambda_n \geq 0\right\}$$

The decision $(X, \Pi_{\mathbb{C}})-$*Problem (*$X-$*Problem by short)*

Each $X \subset \Pi_{NIEP}$ *has associated the decision* $(X, \Pi_{\mathbb{C}})-$*Problem*

Subsets X of Π_{NIEP} which are useful for our interests

The **Suleĭmanova Set**

$$\Pi_{\text{Su}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^n : n \in \mathbb{N};\ \lambda_1 \geq 0 > \lambda_2 \geq \cdots \geq \lambda_n;\ \lambda_1 + \cdots + \lambda_n \geq 0\right\}$$

The **Suleĭmanova-Perfect Set**

$$\Pi_{\text{SP}} \equiv \left\{\Lambda \in \mathbb{R}^n : n \in \mathbb{N};\ \exists \text{ a partition } \Lambda = \Lambda_1 \cup \cdots \cup \Lambda_k \text{ such that } \Lambda_1, \ldots, \Lambda_k \in \Pi_{\text{Su}}\right\}$$

The decision $(X, \Pi_{\mathbb{C}})-Problem$ $(X-Problem$ *by short)*

Each $X \subset \Pi_{NIEP}$ *has associated the decision* $(X, \Pi_{\mathbb{C}})-Problem$

Subsets X of Π_{NIEP} which are useful for our interests

The **Suleĭmanova Set**

$$\Pi_{\text{Su}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^n : n \in \mathbb{N};\ \lambda_1 \geq 0 > \lambda_2 \geq \cdots \geq \lambda_n;\ \lambda_1 + \cdots + \lambda_n \geq 0\right\}$$

The **Suleĭmanova-Perfect Set**

$$\Pi_{\text{SP}} \equiv \left\{\Lambda \in \mathbb{R}^n : n \in \mathbb{N};\ \exists \text{ a partition } \Lambda = \Lambda_1 \cup \cdots \cup \Lambda_k \text{ such that } \Lambda_1, \ldots, \Lambda_k \in \Pi_{\text{Su}}\right\}$$

The Suleĭmanova decision Problem $(\Pi_{\text{Su}}, \Pi_{\mathbb{C}})$ seems trivial

The decision $(X, \Pi_{\mathbb{C}})-Problem$ $(X-Problem$ *by short*$)$

Each $X \subset \Pi_{NIEP}$ *has associated the decision* $(X, \Pi_{\mathbb{C}})-Problem$

Subsets X of Π_{NIEP} which are useful for our interests

The **Suleĭmanova Set**

$$\Pi_{\text{Su}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^n : n \in \mathbb{N};\ \lambda_1 \geq 0 > \lambda_2 \geq \cdots \geq \lambda_n;\ \lambda_1 + \cdots + \lambda_n \geq 0\right\}$$

The **Suleĭmanova-Perfect Set**

$$\Pi_{\text{SP}} \equiv \left\{\Lambda \in \mathbb{R}^n : n \in \mathbb{N};\ \exists \text{ a partition } \Lambda = \Lambda_1 \cup \cdots \cup \Lambda_k \text{ such that } \Lambda_1, \ldots, \Lambda_k \in \Pi_{\text{Su}}\right\}$$

The Suleĭmanova decision Problem $(\Pi_{\text{Su}}, \Pi_{\mathbb{C}})$ seems trivial

Is trivial the Suleĭmanova-Perfect decision Problem $(\Pi_{\text{SP}}, \Pi_{\mathbb{C}})$?

Main results

Theorem

$(\Pi_{PP}, \Pi_{\mathbb{N}})-$Problem is polynomial-time reducible to $(X, \Pi_{\mathbb{C}})-$Problem for $\Pi_{SP} \subseteq X \subseteq \Pi_{NIEP}$

Theorem

$(\Pi_{PP}, \Pi_{\mathbb{N}})-$Problem is polynomial-time reducible to $(X, \Pi_{\mathbb{C}})-$Problem for $\Pi_{SP} \subseteq X \subseteq \Pi_{NIEP}$

Proof: Define

$$\begin{array}{ccccc} \phi: & \Pi_{\mathbb{N}} & \longrightarrow & \Pi_{\mathbb{C}} \\ & K = (k_1, \ldots, k_n) & \mapsto & \phi(K) = (\frac{\Sigma(K)}{2}, \frac{\Sigma(K)}{2}, -k_1, \ldots, -k_n) \end{array}$$

Theorem

$(\Pi_{PP}, \Pi_{\mathbb{N}})-$Problem is polynomial-time reducible to $(X, \Pi_{\mathbb{C}})-$Problem for $\Pi_{SP} \subseteq X \subseteq \Pi_{NIEP}$

Proof: Define

$$\begin{array}{ccccc} \phi: & \Pi_{\mathbb{N}} & \longrightarrow & \Pi_{\mathbb{C}} \\ & K = (k_1, \ldots, k_n) & \mapsto & \phi(K) = (\frac{\Sigma(K)}{2}, \frac{\Sigma(K)}{2}, -k_1, \ldots, -k_n) \end{array}$$

(i) Yes-instances: if $K \in \Pi_{PP}$ then $\phi(K) \in X$

Theorem

$(\Pi_{PP}, \Pi_{\mathbb{N}})-$Problem is polynomial-time reducible to $(X, \Pi_{\mathbb{C}})-$Problem for $\Pi_{SP} \subseteq X \subseteq \Pi_{NIEP}$

Proof: Define

$$\begin{array}{ccccc} \phi: & \Pi_{\mathbb{N}} & \longrightarrow & \Pi_{\mathbb{C}} \\ & K=(k_1,\ldots,k_n) & \mapsto & \phi(K)=(\frac{\Sigma(K)}{2},\frac{\Sigma(K)}{2},-k_1,\ldots,-k_n) \end{array}$$

(i) Yes-instances: if $K \in \Pi_{PP}$ then $\phi(K) \in X$

- $K \in \Pi_{PP} \Rightarrow K=(\alpha_1,\ldots,\alpha_i) \cup (\beta_1,\ldots,\beta_j)$ with

$$\alpha_1+\ldots+\alpha_i=\beta_1+\ldots+\beta_j=\frac{\Sigma(K)}{2}$$

Theorem

$(\Pi_{PP}, \Pi_{\mathbb{N}})-$Problem is polynomial-time reducible to $(X, \Pi_{\mathbb{C}})-$Problem for $\Pi_{SP} \subseteq X \subseteq \Pi_{NIEP}$

Proof: Define

$$\begin{array}{rccc} \phi: & \Pi_{\mathbb{N}} & \longrightarrow & \Pi_{\mathbb{C}} \\ & K = (k_1, \ldots, k_n) & \mapsto & \phi(K) = (\frac{\Sigma(K)}{2}, \frac{\Sigma(K)}{2}, -k_1, \ldots, -k_n) \end{array}$$

(i) Yes-instances: if $K \in \Pi_{PP}$ then $\phi(K) \in X$

- $K \in \Pi_{PP} \Rightarrow K = (\alpha_1, \ldots, \alpha_i) \cup (\beta_1, \ldots, \beta_j)$ with

$$\alpha_1 + \ldots + \alpha_i = \beta_1 + \ldots + \beta_j = \frac{\Sigma(K)}{2}$$

- $(\frac{\Sigma(K)}{2}, -\alpha_1, \ldots, -\alpha_i) \in \Pi_{Su}$ and $(\frac{\Sigma(K)}{2}, -\beta_1, \ldots, -\beta_j) \in \Pi_{Su}$

Theorem

$(\Pi_{PP}, \Pi_{\mathbb{N}})-$Problem is polynomial-time reducible to $(X, \Pi_{\mathbb{C}})-$Problem for $\Pi_{SP} \subseteq X \subseteq \Pi_{NIEP}$

Proof: Define

$$\begin{array}{ccccc} \phi: & \Pi_{\mathbb{N}} & \longrightarrow & \Pi_{\mathbb{C}} \\ & K = (k_1, \ldots, k_n) & \mapsto & \phi(K) = (\frac{\Sigma(K)}{2}, \frac{\Sigma(K)}{2}, -k_1, \ldots, -k_n) \end{array}$$

(i) Yes-instances: if $K \in \Pi_{PP}$ then $\phi(K) \in X$

- $K \in \Pi_{PP} \Rightarrow K = (\alpha_1, \ldots, \alpha_i) \cup (\beta_1, \ldots, \beta_j)$ with

$$\alpha_1 + \ldots + \alpha_i = \beta_1 + \ldots + \beta_j = \frac{\Sigma(K)}{2}$$

- $(\frac{\Sigma(K)}{2}, -\alpha_1, \ldots, -\alpha_i) \in \Pi_{Su}$ and $(\frac{\Sigma(K)}{2}, -\beta_1, \ldots, -\beta_j) \in \Pi_{Su}$
- $\phi(K) = (\frac{\Sigma(K)}{2}, -\alpha_1, \ldots, -\alpha_i) \cup (\frac{\Sigma(K)}{2}, -\beta_1, \ldots, -\beta_j) \in \Pi_{SP}$

Theorem

$(\Pi_{PP}, \Pi_{\mathbb{N}})-$Problem is polynomial-time reducible to $(X, \Pi_{\mathbb{C}})-$Problem for $\Pi_{SP} \subseteq X \subseteq \Pi_{NIEP}$

Proof: Define

$$\begin{array}{ccccc} \phi: & \Pi_{\mathbb{N}} & \longrightarrow & \Pi_{\mathbb{C}} \\ & K = (k_1, \ldots, k_n) & \mapsto & \phi(K) = (\frac{\Sigma(K)}{2}, \frac{\Sigma(K)}{2}, -k_1, \ldots, -k_n) \end{array}$$

(i) Yes-instances: if $K \in \Pi_{PP}$ then $\phi(K) \in X$

- $K \in \Pi_{PP} \Rightarrow K = (\alpha_1, \ldots, \alpha_i) \cup (\beta_1, \ldots, \beta_j)$ with

$$\alpha_1 + \ldots + \alpha_i = \beta_1 + \ldots + \beta_j = \frac{\Sigma(K)}{2}$$

- $(\frac{\Sigma(K)}{2}, -\alpha_1, \ldots, -\alpha_i) \in \Pi_{Su}$ and $(\frac{\Sigma(K)}{2}, -\beta_1, \ldots, -\beta_j) \in \Pi_{Su}$
- $\phi(K) = (\frac{\Sigma(K)}{2}, -\alpha_1, \ldots, -\alpha_i) \cup (\frac{\Sigma(K)}{2}, -\beta_1, \ldots, -\beta_j) \in \Pi_{SP}$
- As $\Pi_{SP} \subset X$ then $\phi(K) \in X$

Main result

(ii) No-instances: if $K \notin \Pi_{PP}$ then $\phi(K) \notin X$

Main result

(ii) No-instances: if $K \notin \Pi_{\text{PP}}$ then $\phi(K) \notin X$

- Suppose $\phi(K) = (\frac{\Sigma(K)}{2}, \frac{\Sigma(K)}{2}, -k_1, \ldots, -k_n) \in X \subseteq \Pi_{\text{NIEP}}$, then it is realizable

(ii) No-instances: if $K \notin \Pi_{PP}$ then $\phi(K) \notin X$

- Suppose $\phi(K) = (\frac{\Sigma(K)}{2}, \frac{\Sigma(K)}{2}, -k_1, \ldots, -k_n) \in X \subseteq \Pi_{NIEP}$, then it is realizable
- By the Perron-Fröbenius Theorem if $\Phi(K)$ is realizable by $A \geq 0$ then A is reducible, that is, there exists a permutation matrix P such that

$$PAP^T = \begin{bmatrix} A_1 & * \\ 0 & A_2 \end{bmatrix}$$

where A_1 and A_2 have spectral radius $\Sigma(K)/2$. So

$$\sigma(A) = \sigma(A_1) \cup \sigma(A_2) = (\frac{\Sigma(K)}{2}, -\gamma_1, \ldots, -\gamma_r) \cup (\frac{\Sigma(K)}{2}, -\lambda_1, \ldots, -\lambda_s) = \phi(K)$$

(ii) No-instances: if $K \notin \Pi_{PP}$ then $\phi(K) \notin X$

- Suppose $\phi(K) = (\frac{\Sigma(K)}{2}, \frac{\Sigma(K)}{2}, -k_1, \ldots, -k_n) \in X \subseteq \Pi_{NIEP}$, then it is realizable
- By the Perron-Fröbenius Theorem if $\Phi(K)$ is realizable by $A \geq 0$ then A is reducible, that is, there exists a permutation matrix P such that

$$PAP^T = \begin{bmatrix} A_1 & * \\ 0 & A_2 \end{bmatrix}$$

where A_1 and A_2 have spectral radius $\Sigma(K)/2$. So

$$\sigma(A) = \sigma(A_1) \cup \sigma(A_2) = (\frac{\Sigma(K)}{2}, -\gamma_1, \ldots, -\gamma_r) \cup (\frac{\Sigma(K)}{2}, -\lambda_1, \ldots, -\lambda_s) = \phi(K)$$

- As $A_1, A_2 \geq 0$ and $\Sigma(\Phi(K)) = 0$ then $\frac{\Sigma(k)}{2} = \gamma_1 + \cdots + \gamma_r = \lambda_1 + \cdots + \lambda_s$

(ii) No-instances: if $K \notin \Pi_{PP}$ then $\phi(K) \notin X$

- Suppose $\phi(K) = (\frac{\Sigma(K)}{2}, \frac{\Sigma(K)}{2}, -k_1, \ldots, -k_n) \in X \subseteq \Pi_{NIEP}$, then it is realizable
- By the Perron-Fröbenius Theorem if $\Phi(K)$ is realizable by $A \geq 0$ then A is reducible, that is, there exists a permutation matrix P such that

$$PAP^T = \begin{bmatrix} A_1 & * \\ 0 & A_2 \end{bmatrix}$$

where A_1 and A_2 have spectral radius $\Sigma(K)/2$. So

$$\sigma(A) = \sigma(A_1) \cup \sigma(A_2) = (\tfrac{\Sigma(K)}{2}, -\gamma_1, \ldots, -\gamma_r) \cup (\tfrac{\Sigma(K)}{2}, -\lambda_1, \ldots, -\lambda_s) = \phi(K)$$

- As $A_1, A_2 \geq 0$ and $\Sigma(\Phi(K)) = 0$ then $\frac{\Sigma(k)}{2} = \gamma_1 + \cdots + \gamma_r = \lambda_1 + \cdots + \lambda_s$
- $K = \Gamma \cup \Lambda = (\gamma_1, \ldots, \gamma_r) \cup (\lambda_1, \ldots, \lambda_s)$ with $\Sigma(\Gamma) = \Sigma(\Lambda) = \frac{\Sigma(K)}{2}$

(ii) No-instances: if $K \notin \Pi_{PP}$ then $\phi(K) \notin X$

- Suppose $\phi(K) = (\frac{\Sigma(K)}{2}, \frac{\Sigma(K)}{2}, -k_1, \ldots, -k_n) \in X \subseteq \Pi_{NIEP}$, then it is realizable
- By the Perron-Fröbenius Theorem if $\Phi(K)$ is realizable by $A \geq 0$ then A is reducible, that is, there exists a permutation matrix P such that

$$PAP^T = \begin{bmatrix} A_1 & * \\ 0 & A_2 \end{bmatrix}$$

where A_1 and A_2 have spectral radius $\Sigma(K)/2$. So

$$\sigma(A) = \sigma(A_1) \cup \sigma(A_2) = (\tfrac{\Sigma(K)}{2}, -\gamma_1, \ldots, -\gamma_r) \cup (\tfrac{\Sigma(K)}{2}, -\lambda_1, \ldots, -\lambda_s) = \phi(K)$$

- As $A_1, A_2 \geq 0$ and $\Sigma(\Phi(K)) = 0$ then $\frac{\Sigma(k)}{2} = \gamma_1 + \cdots + \gamma_r = \lambda_1 + \cdots + \lambda_s$
- $K = \Gamma \cup \Lambda = (\gamma_1, \ldots, \gamma_r) \cup (\lambda_1, \ldots, \lambda_s)$ with $\Sigma(\Gamma) = \Sigma(\Lambda) = \frac{\Sigma(K)}{2}$
- Therefore $K \in \Pi_{PP}$. Contradiction!!

(ii) No-instances: if $K \notin \Pi_{PP}$ then $\phi(K) \notin X$

- Suppose $\phi(K) = (\frac{\Sigma(K)}{2}, \frac{\Sigma(K)}{2}, -k_1, \ldots, -k_n) \in X \subseteq \Pi_{NIEP}$, then it is realizable
- By the Perron-Fröbenius Theorem if $\Phi(K)$ is realizable by $A \geq 0$ then A is reducible, that is, there exists a permutation matrix P such that

$$PAP^T = \begin{bmatrix} A_1 & * \\ 0 & A_2 \end{bmatrix}$$

where A_1 and A_2 have spectral radius $\Sigma(K)/2$. So

$$\sigma(A) = \sigma(A_1) \cup \sigma(A_2) = (\tfrac{\Sigma(K)}{2}, -\gamma_1, \ldots, -\gamma_r) \cup (\tfrac{\Sigma(K)}{2}, -\lambda_1, \ldots, -\lambda_s) = \phi(K)$$

- As $A_1, A_2 \geq 0$ and $\Sigma(\Phi(K)) = 0$ then $\frac{\Sigma(k)}{2} = \gamma_1 + \cdots + \gamma_r = \lambda_1 + \cdots + \lambda_s$
- $K = \Gamma \cup \Lambda = (\gamma_1, \ldots, \gamma_r) \cup (\lambda_1, \ldots, \lambda_s)$ with $\Sigma(\Gamma) = \Sigma(\Lambda) = \frac{\Sigma(K)}{2}$
- Therefore $K \in \Pi_{PP}$. Contradiction!!
- So $\phi(K) \notin \Pi_{NIEP}$

(ii) No-instances: if $K \notin \Pi_{PP}$ then $\phi(K) \notin X$

- Suppose $\phi(K) = (\frac{\Sigma(K)}{2}, \frac{\Sigma(K)}{2}, -k_1, \ldots, -k_n) \in X \subseteq \Pi_{NIEP}$, then it is realizable
- By the Perron-Fröbenius Theorem if $\Phi(K)$ is realizable by $A \geq 0$ then A is reducible, that is, there exists a permutation matrix P such that

$$PAP^T = \begin{bmatrix} A_1 & * \\ 0 & A_2 \end{bmatrix}$$

where A_1 and A_2 have spectral radius $\Sigma(K)/2$. So

$$\sigma(A) = \sigma(A_1) \cup \sigma(A_2) = (\tfrac{\Sigma(K)}{2}, -\gamma_1, \ldots, -\gamma_r) \cup (\tfrac{\Sigma(K)}{2}, -\lambda_1, \ldots, -\lambda_s) = \phi(K)$$

- As $A_1, A_2 \geq 0$ and $\Sigma(\Phi(K)) = 0$ then $\frac{\Sigma(k)}{2} = \gamma_1 + \cdots + \gamma_r = \lambda_1 + \cdots + \lambda_s$
- $K = \Gamma \cup \Lambda = (\gamma_1, \ldots, \gamma_r) \cup (\lambda_1, \ldots, \lambda_s)$ with $\Sigma(\Gamma) = \Sigma(\Lambda) = \frac{\Sigma(K)}{2}$
- Therefore $K \in \Pi_{PP}$. Contradiction!!
- So $\phi(K) \notin \Pi_{NIEP}$
- As $X \subseteq \Pi_{NIEP}$ then $\phi(K) \notin X$ $\square$

Main result

Theorem

The $(X, \Pi_{\mathbb{C}})-$Problem for $\Pi_{SP} \subseteq X \subseteq \Pi_{NIEP}$ is **NP-hard**

Main result

Theorem

The $(X, \Pi_{\mathbb{C}})-$Problem for $\Pi_{SP} \subseteq X \subseteq \Pi_{NIEP}$ is **NP-hard**

Proof:

(i) The Partition Problem is **NP-hard** (Karp 1972). So all **NP** problems are polynomial-time reducible to the Partition Problem

Main result

Theorem

The $(X, \Pi_{\mathbb{C}})-$Problem for $\Pi_{SP} \subseteq X \subseteq \Pi_{NIEP}$ is **NP-hard**

Proof:

(i) The Partition Problem is **NP-hard** (Karp 1972). So all **NP** problems are polynomial-time reducible to the Partition Problem

(ii) We have seen that the Partition Problem is polynomial-time reducible to the $(X, \Pi_{\mathbb{C}})-$Problem

Theorem

The $(X, \Pi_{\mathbb{C}})-$Problem for $\Pi_{SP} \subseteq X \subseteq \Pi_{NIEP}$ is **NP-hard**

Proof:

(i) The Partition Problem is **NP-hard** (Karp 1972). So all **NP** problems are polynomial-time reducible to the Partition Problem

(ii) We have seen that the Partition Problem is polynomial-time reducible to the $(X, \Pi_{\mathbb{C}})-$Problem

(iii) By (i), (ii) and the transitivity of the polynomial-time reduction relation, all **NP** problems are polynomial-time reducible to the $(X, \Pi_{\mathbb{C}})-$Problem. $\square$

Main result

Theorem

The $(X, \Pi_{\mathbb{C}})-$Problem for $\Pi_{SP} \subseteq X \subseteq \Pi_{NIEP}$ is **NP-hard**

Proof:

(i) The Partition Problem is **NP-hard** (Karp 1972). So all **NP** problems are polynomial-time reducible to the Partition Problem

(ii) We have seen that the Partition Problem is polynomial-time reducible to the $(X, \Pi_{\mathbb{C}})-$Problem

(iii) By (i), (ii) and the transitivity of the polynomial-time reduction relation, all **NP** problems are polynomial-time reducible to the $(X, \Pi_{\mathbb{C}})-$Problem. $\square$

Main result

The $(\Pi_{NIEP}, \Pi_{\mathbb{C}})$ and the $(\Pi_{RNIEP}, \Pi_{\mathbb{C}})-$Problems are **NP-hard**

Implications

If an algorithm solves the $(\Pi_{SP}, \Pi_{\mathbb{C}})-$*Problem in polynomial time, then* **P=NP**

Implications

If an algorithm solves the $(\Pi_{SP}, \Pi_{\mathbb{C}})-$*Problem in polynomial time, then* **P=NP**

Proof: Let **D** be a decision problem. If **D** is **P** then is **NP**. Let prove that if **D** is **NP** then is **P**

Implications

If an algorithm solves the $(\Pi_{SP}, \Pi_{\mathbb{C}})-$*Problem in polynomial time, then* **P=NP**

Proof: Let **D** be a decision problem. If **D** is **P** then is **NP**. Let prove that if **D** is **NP** then is **P**

(i) The $(\Pi_{SP}, \Pi_{\mathbb{C}})-$Problem is **NP-hard**, so any instance of **D** is transformed in polynomial time to an instance of $(\Pi_{SP}, \Pi_{\mathbb{C}})$

Implications

If an algorithm solves the $(\Pi_{SP}, \Pi_{\mathbb{C}})-$*Problem in polynomial time, then* **P=NP**

Proof: Let **D** be a decision problem. If **D** is **P** then is **NP**. Let prove that if **D** is **NP** then is **P**

(i) The $(\Pi_{SP}, \Pi_{\mathbb{C}})-$Problem is **NP-hard**, so any instance of **D** is transformed in polynomial time to an instance of $(\Pi_{SP}, \Pi_{\mathbb{C}})$

(ii) If the $(\Pi_{SP}, \Pi_{\mathbb{C}})-$Problem is **P** then its yes-no instances are answered in polynomial time

Implications

If an algorithm solves the $(\Pi_{SP}, \Pi_{\mathbb{C}})-$*Problem in polynomial time, then* **P=NP**

Proof: Let **D** be a decision problem. If **D** is **P** then is **NP**. Let prove that if **D** is **NP** then is **P**

(i) The $(\Pi_{SP}, \Pi_{\mathbb{C}})-$Problem is **NP-hard**, so any instance of **D** is transformed in polynomial time to an instance of $(\Pi_{SP}, \Pi_{\mathbb{C}})$

(ii) If the $(\Pi_{SP}, \Pi_{\mathbb{C}})-$Problem is **P** then its yes-no instances are answered in polynomial time

(iii) From (i)-(ii) the yes-no instances of **D** are answered in polynomial time. So **D** is **P** □

Implications

If an algorithm solves the $(\Pi_{SP}, \Pi_{\mathbb{C}})-$*Problem in polynomial time, then* **P=NP**

Proof: Let **D** be a decision problem. If **D** is **P** then is **NP**. Let prove that if **D** is **NP** then is **P**

(i) The $(\Pi_{SP}, \Pi_{\mathbb{C}})-$Problem is **NP-hard**, so any instance of **D** is transformed in polynomial time to an instance of $(\Pi_{SP}, \Pi_{\mathbb{C}})$

(ii) If the $(\Pi_{SP}, \Pi_{\mathbb{C}})-$Problem is **P** then its yes-no instances are answered in polynomial time

(iii) From (i)-(ii) the yes-no instances of **D** are answered in polynomial time. So **D** is **P** □

To find a polynomial time algorithm for the $(\Pi_{SP}, \Pi_{\mathbb{C}})-$Problem is quite unlikely, since **P=NP** or **P≠NP** is the main unsolved problem in computer science

Implications

If an algorithm solves the $(\Pi_{SP}, \Pi_{\mathbb{C}})-$*Problem in polynomial time, then* **P=NP**

Proof: Let **D** be a decision problem. If **D** is **P** then is **NP**. Let prove that if **D** is **NP** then is **P**

(i) The $(\Pi_{SP}, \Pi_{\mathbb{C}})-$Problem is **NP-hard**, so any instance of **D** is transformed in polynomial time to an instance of $(\Pi_{SP}, \Pi_{\mathbb{C}})$

(ii) If the $(\Pi_{SP}, \Pi_{\mathbb{C}})-$Problem is **P** then its yes-no instances are answered in polynomial time

(iii) From (i)-(ii) the yes-no instances of **D** are answered in polynomial time. So **D** is **P** □

To find a polynomial time algorithm for the $(\Pi_{SP}, \Pi_{\mathbb{C}})-$Problem is quite unlikely, since **P=NP** or **P≠NP** is the main unsolved problem in computer science

For $\Pi_{SP} \subseteq X \subseteq \Pi_{NIEP}$ the $(X, \Pi_{\mathbb{C}})-$Problem is **NP-hard**. Same implication

We will restrict to **reals** (and later on to **rationals**)

$$\mathbb{R}_{\downarrow} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^n : n \in \mathbb{N};\ \lambda_1 \geq \cdots \geq \lambda_n\right\}$$

and its subset

$$\Pi_{\mathbb{R}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}_{\downarrow} : \lambda_1 + \cdots + \lambda_n \geq 0 \ \text{ and } \ \lambda_1 \geq |\lambda_n|\right\}$$

The RNIEP

We will restrict to **reals** (and later on to **rationals**)

$$\mathbb{R}_{\downarrow} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^n : n \in \mathbb{N};\ \lambda_1 \geq \cdots \geq \lambda_n\right\}$$

and its subset

$$\Pi_{\mathbb{R}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}_{\downarrow} : \lambda_1 + \cdots + \lambda_n \geq 0 \ \text{ and } \ \lambda_1 \geq |\lambda_n|\right\}$$

Real Nonnegative Inverse Eigenvalue Problem: Characterize

$$\Pi_{\text{RNIEP}} \equiv \left\{\Lambda \in \Pi_{\mathbb{R}} : \exists A \geq 0 \text{ with } \sigma(A) = \Lambda\right\}$$

The RNIEP

We will restrict to **reals** (and later on to **rationals**)

$$\mathbb{R}_{\downarrow} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^n : n \in \mathbb{N};\ \lambda_1 \geq \cdots \geq \lambda_n\right\}$$

and its subset

$$\Pi_{\mathbb{R}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}_{\downarrow} : \lambda_1 + \cdots + \lambda_n \geq 0 \ \text{ and } \ \lambda_1 \geq |\lambda_n|\right\}$$

Real Nonnegative Inverse Eigenvalue Problem: Characterize

$$\Pi_{\text{RNIEP}} \equiv \left\{\Lambda \in \Pi_{\mathbb{R}} : \exists A \geq 0 \text{ with } \sigma(A) = \Lambda\right\}$$

Which $(X, \Pi_{\mathbb{R}})-$Problems for $X \subseteq \Pi_{\text{RNIEP}}$ are **P** or **NP**?

Subsets of Π_{RNIEP}

The RNIEP is intractable. Some known subsets of Π_{RNIEP}

Subsets of Π_{RNIEP}

The RNIEP is intractable. Some known subsets of Π_{RNIEP}

1. Defined by linear inequalities

Subsets of Π_{RNIEP}

The RNIEP is intractable. Some known subsets of Π_{RNIEP}

1. Defined by linear inequalities
2. Its elements admit a special partition

Subsets of Π_{RNIEP}

The RNIEP is intractable. Some known subsets of Π_{RNIEP}

1. Defined by linear inequalities
2. Its elements admit a special partition
3. Its elements are defined recursively

Subsets of Π_{RNIEP}

The RNIEP is intractable. Some known subsets of Π_{RNIEP}

1. Defined by linear inequalities
2. Its elements admit a special partition
3. Its elements are defined recursively
4. Its elements require the existence of an auxiliary nonnegative matrix

Some notation for $\mathbb{R}_{\downarrow}$

Let $\Lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{R}_{\downarrow}$ and $\Gamma = (\gamma_1, \ldots, \gamma_m) \in \mathbb{R}_{\downarrow}$

Some notation for $\mathbb{R}_{\downarrow}$

Let $\Lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{R}_{\downarrow}$ and $\Gamma = (\gamma_1, \ldots, \gamma_m) \in \mathbb{R}_{\downarrow}$

- $\rho(\Lambda) = \max\{|\lambda_1|, \ldots, |\lambda_n|\}$ and $\Sigma(\Lambda) = \lambda_1 + \cdots + \lambda_n$

Some notation for $\mathbb{R}_{\downarrow}$

Let $\Lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{R}_{\downarrow}$ and $\Gamma = (\gamma_1, \ldots, \gamma_m) \in \mathbb{R}_{\downarrow}$

- $\rho(\Lambda) = \max\{|\lambda_1|, \ldots, |\lambda_n|\}$ and $\Sigma(\Lambda) = \lambda_1 + \cdots + \lambda_n$
- Λ^+ contains all nonnegative values of Λ

Some notation for $\mathbb{R}_{\downarrow}$

Let $\Lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{R}_{\downarrow}$ and $\Gamma = (\gamma_1, \ldots, \gamma_m) \in \mathbb{R}_{\downarrow}$

- $\rho(\Lambda) = \max\{|\lambda_1|, \ldots, |\lambda_n|\}$ and $\Sigma(\Lambda) = \lambda_1 + \cdots + \lambda_n$
- Λ^+ contains all nonnegative values of Λ
- Λ^- contains all negative values of Λ

Some notation for $\mathbb{R}_{\downarrow}$

Let $\Lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{R}_{\downarrow}$ and $\Gamma = (\gamma_1, \ldots, \gamma_m) \in \mathbb{R}_{\downarrow}$

- $\rho(\Lambda) = \max\{|\lambda_1|, \ldots, |\lambda_n|\}$ and $\Sigma(\Lambda) = \lambda_1 + \cdots + \lambda_n$
- Λ^+ contains all nonnegative values of Λ
- Λ^- contains all negative values of Λ
- $\Lambda \cup \Gamma$ contains 'all' elements of lists Λ and Γ

Some notation for $\mathbb{R}_{\downarrow}$

Let $\Lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{R}_{\downarrow}$ and $\Gamma = (\gamma_1, \ldots, \gamma_m) \in \mathbb{R}_{\downarrow}$

- $\rho(\Lambda) = \max\{|\lambda_1|, \ldots, |\lambda_n|\}$ and $\Sigma(\Lambda) = \lambda_1 + \cdots + \lambda_n$
- Λ^+ contains all nonnegative values of Λ
- Λ^- contains all negative values of Λ
- $\Lambda \cup \Gamma$ contains 'all' elements of lists Λ and Γ
- If Λ' is a sublist of Λ then $\Lambda - \Lambda'$ are the elements of Λ which are not in Λ'

Some notation for $\mathbb{R}_{\downarrow}$

Let $\Lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{R}_{\downarrow}$ and $\Gamma = (\gamma_1, \ldots, \gamma_m) \in \mathbb{R}_{\downarrow}$

- $\rho(\Lambda) = \max\{|\lambda_1|, \ldots, |\lambda_n|\}$ and $\Sigma(\Lambda) = \lambda_1 + \cdots + \lambda_n$
- Λ^+ contains all nonnegative values of Λ
- Λ^- contains all negative values of Λ
- $\Lambda \cup \Gamma$ contains 'all' elements of lists Λ and Γ
- If Λ' is a sublist of Λ then $\Lambda - \Lambda'$ are the elements of Λ which are not in Λ'
- $\Lambda = \Lambda^+ \cup \Lambda^-$, $\Lambda^- = \Lambda - \Lambda^+$, and $\Lambda^+ = \Lambda - \Lambda^-$

Some notation for $\mathbb{R}_{\downarrow}$

Let $\Lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{R}_{\downarrow}$ and $\Gamma = (\gamma_1, \ldots, \gamma_m) \in \mathbb{R}_{\downarrow}$

- $\rho(\Lambda) = \max\{|\lambda_1|, \ldots, |\lambda_n|\}$ and $\Sigma(\Lambda) = \lambda_1 + \cdots + \lambda_n$
- Λ^+ contains all nonnegative values of Λ
- Λ^- contains all negative values of Λ
- $\Lambda \cup \Gamma$ contains 'all' elements of lists Λ and Γ
- If Λ' is a sublist of Λ then $\Lambda - \Lambda'$ are the elements of Λ which are not in Λ'
- $\Lambda = \Lambda^+ \cup \Lambda^-$, $\Lambda^- = \Lambda - \Lambda^+$, and $\Lambda^+ = \Lambda - \Lambda^-$

Example

Let $\Lambda = (8, 7, 3, 0, -1, -1, -6)$ and $\Gamma = (15, 3, 2, -1, -4)$

Some notation for $\mathbb{R}_{\downarrow}$

Let $\Lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{R}_{\downarrow}$ and $\Gamma = (\gamma_1, \ldots, \gamma_m) \in \mathbb{R}_{\downarrow}$

- $\rho(\Lambda) = \max\{|\lambda_1|, \ldots, |\lambda_n|\}$ and $\Sigma(\Lambda) = \lambda_1 + \cdots + \lambda_n$
- Λ^+ contains all nonnegative values of Λ
- Λ^- contains all negative values of Λ
- $\Lambda \cup \Gamma$ contains 'all' elements of lists Λ and Γ
- If Λ' is a sublist of Λ then $\Lambda - \Lambda'$ are the elements of Λ which are not in Λ'
- $\Lambda = \Lambda^+ \cup \Lambda^-$, $\Lambda^- = \Lambda - \Lambda^+$, and $\Lambda^+ = \Lambda - \Lambda^-$

Example

Let $\Lambda = (8, 7, 3, 0, -1, -1, -6)$ and $\Gamma = (15, 3, 2, -1, -4)$

1. $\rho(\Lambda) = 8$ and $\Sigma(\Lambda) = 10$

Some notation for $\mathbb{R}_{\downarrow}$

Let $\Lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{R}_{\downarrow}$ and $\Gamma = (\gamma_1, \ldots, \gamma_m) \in \mathbb{R}_{\downarrow}$

- $\rho(\Lambda) = \max\{|\lambda_1|, \ldots, |\lambda_n|\}$ and $\Sigma(\Lambda) = \lambda_1 + \cdots + \lambda_n$
- Λ^+ contains all nonnegative values of Λ
- Λ^- contains all negative values of Λ
- $\Lambda \cup \Gamma$ contains 'all' elements of lists Λ and Γ
- If Λ' is a sublist of Λ then $\Lambda - \Lambda'$ are the elements of Λ which are not in Λ'
- $\Lambda = \Lambda^+ \cup \Lambda^-$, $\Lambda^- = \Lambda - \Lambda^+$, and $\Lambda^+ = \Lambda - \Lambda^-$

Example

Let $\Lambda = (8, 7, 3, 0, -1, -1, -6)$ and $\Gamma = (15, 3, 2, -1, -4)$

1. $\rho(\Lambda) = 8$ and $\Sigma(\Lambda) = 10$
2. $\Lambda^+ = (8, 7, 3, 0)$ and $\Lambda^- = (-1, -1, -6)$

Some notation for $\mathbb{R}_{\downarrow}$

Let $\Lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{R}_{\downarrow}$ and $\Gamma = (\gamma_1, \ldots, \gamma_m) \in \mathbb{R}_{\downarrow}$

- $\rho(\Lambda) = \max\{|\lambda_1|, \ldots, |\lambda_n|\}$ and $\Sigma(\Lambda) = \lambda_1 + \cdots + \lambda_n$
- Λ^+ contains all nonnegative values of Λ
- Λ^- contains all negative values of Λ
- $\Lambda \cup \Gamma$ contains 'all' elements of lists Λ and Γ
- If Λ' is a sublist of Λ then $\Lambda - \Lambda'$ are the elements of Λ which are not in Λ'
- $\Lambda = \Lambda^+ \cup \Lambda^-$, $\Lambda^- = \Lambda - \Lambda^+$, and $\Lambda^+ = \Lambda - \Lambda^-$

Example

Let $\Lambda = (8, 7, 3, 0, -1, -1, -6)$ and $\Gamma = (15, 3, 2, -1, -4)$

1. $\rho(\Lambda) = 8$ and $\Sigma(\Lambda) = 10$
2. $\Lambda^+ = (8, 7, 3, 0)$ and $\Lambda^- = (-1, -1, -6)$
3. $\Lambda \cup \Gamma = (15, 8, 7, 3, 3, 2, 0, -1, -1, -1, -4, -6)$

Some notation for $\mathbb{R}_{\downarrow}$

Let $\Lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{R}_{\downarrow}$ and $\Gamma = (\gamma_1, \ldots, \gamma_m) \in \mathbb{R}_{\downarrow}$

- $\rho(\Lambda) = \max\{|\lambda_1|, \ldots, |\lambda_n|\}$ and $\Sigma(\Lambda) = \lambda_1 + \cdots + \lambda_n$
- Λ^+ contains all nonnegative values of Λ
- Λ^- contains all negative values of Λ
- $\Lambda \cup \Gamma$ contains 'all' elements of lists Λ and Γ
- If Λ' is a sublist of Λ then $\Lambda - \Lambda'$ are the elements of Λ which are not in Λ'
- $\Lambda = \Lambda^+ \cup \Lambda^-$, $\Lambda^- = \Lambda - \Lambda^+$, and $\Lambda^+ = \Lambda - \Lambda^-$

Example

Let $\Lambda = (8, 7, 3, 0, -1, -1, -6)$ and $\Gamma = (15, 3, 2, -1, -4)$

1. $\rho(\Lambda) = 8$ and $\Sigma(\Lambda) = 10$
2. $\Lambda^+ = (8, 7, 3, 0)$ and $\Lambda^- = (-1, -1, -6)$
3. $\Lambda \cup \Gamma = (15, 8, 7, 3, 3, 2, 0, -1, -1, -1, -4, -6)$
4. If $\Lambda' = (8, -1, -6)$ then $\Lambda - \Lambda' = (7, 3, 0, -1)$

Subsets of Π_{RNIEP}: 1. Defined by linear inequalities

- *Suleĭmanova Set*

$$\Pi_{\text{Su}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \Pi_{\mathbb{R}} : \lambda_1 \geq 0 > \lambda_2 \geq \cdots \geq \lambda_n\right\}$$

Subsets of Π_{RNIEP}: 1. Defined by linear inequalities

- *Suleĭmanova Set*

$$\Pi_{\text{Su}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \Pi_{\mathbb{R}} : \lambda_1 \geq 0 > \lambda_2 \geq \cdots \geq \lambda_n\right\}$$

- *Salzmann Set*

$$\Pi_{\text{Sa}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \Pi_{\mathbb{R}} : \ \frac{\lambda_1 + \ldots + \lambda_n}{n} \geq \frac{\lambda_i + \lambda_{n-i+1}}{2} \ \text{ for } i = 2, \ldots, \lfloor \frac{n+1}{2} \rfloor\right\}$$

Subsets of Π_{RNIEP}: 1. Defined by linear inequalities

- *Suleĭmanova Set*

$$\Pi_{\text{Su}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \Pi_{\mathbb{R}} : \lambda_1 \geq 0 > \lambda_2 \geq \cdots \geq \lambda_n\right\}$$

- *Salzmann Set*

$$\Pi_{\text{Sa}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \Pi_{\mathbb{R}} : \ \frac{\lambda_1 + \ldots + \lambda_n}{n} \geq \frac{\lambda_i + \lambda_{n-i+1}}{2} \ \text{ for } i = 2, \ldots, \lfloor \frac{n+1}{2} \rfloor \right\}$$

- *Fiedler Set*

$$\Pi_{\text{Fi}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \Pi_{\mathbb{R}} : \ \lambda_1 + \ldots + \lambda_n \geq \frac{\sum_{i=2,\ldots,n-1} |\lambda_i + \lambda_{n-i+1}| - 2(\lambda_1 + \lambda_n)}{2} \right\}$$

2. Its elements admit a special partition

- *Suleĭmanova-Perfect Set*

$$\Pi_{SP} \equiv \left\{ \Lambda \in \Pi_{\mathbb{R}} : \exists \text{ a partition } \Lambda = \Lambda_1 \cup \cdots \cup \Lambda_k \text{ such that } \Lambda_1, \ldots, \Lambda_k \in \Pi_{Su} \right\}$$

3. Its elements are defined recursively

Basic tools to construct new subsets from known subsets of Π_{RNIEP}

3. Its elements are defined recursively

Basic tools to construct new subsets from known subsets of Π_{RNIEP}

1. $(0) \in \Pi_{\text{RNIEP}}$

3. Its elements are defined recursively

Basic tools to construct new subsets from known subsets of Π_{RNIEP}

1. $(0) \in \Pi_{\text{RNIEP}}$
2. $\Lambda_1, \ldots, \Lambda_k \in \Pi_{\text{RNIEP}} \Rightarrow \Lambda_1 \cup \cdots \cup \Lambda_k \in \Pi_{\text{RNIEP}}$

3. Its elements are defined recursively

Basic tools to construct new subsets from known subsets of Π_{RNIEP}

1. $(0) \in \Pi_{\text{RNIEP}}$
2. $\Lambda_1, \ldots, \Lambda_k \in \Pi_{\text{RNIEP}} \Rightarrow \Lambda_1 \cup \cdots \cup \Lambda_k \in \Pi_{\text{RNIEP}}$
3. If $(\lambda_1, \lambda_2, \ldots, \lambda_n) \in \Pi_{\text{RNIEP}} \Rightarrow (\lambda_1 + \epsilon, \lambda_2, \ldots, \lambda_n) \in \Pi_{\text{RNIEP}}\ \forall \epsilon > 0$

3. Its elements are defined recursively

Basic tools to construct new subsets from known subsets of Π_{RNIEP}

1. $(0) \in \Pi_{\text{RNIEP}}$
2. $\Lambda_1, \ldots, \Lambda_k \in \Pi_{\text{RNIEP}} \Rightarrow \Lambda_1 \cup \cdots \cup \Lambda_k \in \Pi_{\text{RNIEP}}$
3. If $(\lambda_1, \lambda_2, \ldots, \lambda_n) \in \Pi_{\text{RNIEP}} \Rightarrow (\lambda_1 + \epsilon, \lambda_2, \ldots, \lambda_n) \in \Pi_{\text{RNIEP}}\ \forall \epsilon > 0$
4. If $(\lambda_1, \ldots, \lambda_i, \ldots, \lambda_n) \in \Pi_{\text{RNIEP}} \Rightarrow (\lambda_1 + \epsilon, \ldots, \lambda_i \pm \epsilon, \ldots, \lambda_n) \in \Pi_{\text{RNIEP}}\ \forall \epsilon > 0$

3. Its elements are defined recursively

Basic tools to construct new subsets from known subsets of Π_{RNIEP}

1. $(0) \in \Pi_{\text{RNIEP}}$
2. $\Lambda_1, \ldots, \Lambda_k \in \Pi_{\text{RNIEP}} \Rightarrow \Lambda_1 \cup \cdots \cup \Lambda_k \in \Pi_{\text{RNIEP}}$
3. If $(\lambda_1, \lambda_2, \ldots, \lambda_n) \in \Pi_{\text{RNIEP}} \Rightarrow (\lambda_1 + \epsilon, \lambda_2, \ldots, \lambda_n) \in \Pi_{\text{RNIEP}}\ \forall \epsilon > 0$
4. If $(\lambda_1, \ldots, \lambda_i, \ldots, \lambda_n) \in \Pi_{\text{RNIEP}} \Rightarrow (\lambda_1 + \epsilon, \ldots, \lambda_i \pm \epsilon, \ldots, \lambda_n) \in \Pi_{\text{RNIEP}}\ \forall \epsilon > 0$

- *Borobia-Moro-Soto Set* is composed by lists obtained with basic tools

3. Its elements are defined recursively

Basic tools to construct new subsets from known subsets of Π_{RNIEP}

1. $(0) \in \Pi_{\text{RNIEP}}$
2. $\Lambda_1, \ldots, \Lambda_k \in \Pi_{\text{RNIEP}} \Rightarrow \Lambda_1 \cup \cdots \cup \Lambda_k \in \Pi_{\text{RNIEP}}$
3. If $(\lambda_1, \lambda_2, \ldots, \lambda_n) \in \Pi_{\text{RNIEP}} \Rightarrow (\lambda_1 + \epsilon, \lambda_2, \ldots, \lambda_n) \in \Pi_{\text{RNIEP}}\ \forall \epsilon > 0$
4. If $(\lambda_1, \ldots, \lambda_i, \ldots, \lambda_n) \in \Pi_{\text{RNIEP}} \Rightarrow (\lambda_1 + \epsilon, \ldots, \lambda_i \pm \epsilon, \ldots, \lambda_n) \in \Pi_{\text{RNIEP}}\ \forall \epsilon > 0$

- *Borobia-Moro-Soto Set* is composed by lists obtained with basic tools
- Equal to[a]: (i) Soules Set; (ii) Soto Set; (iii) Smigoc and Ellard Set

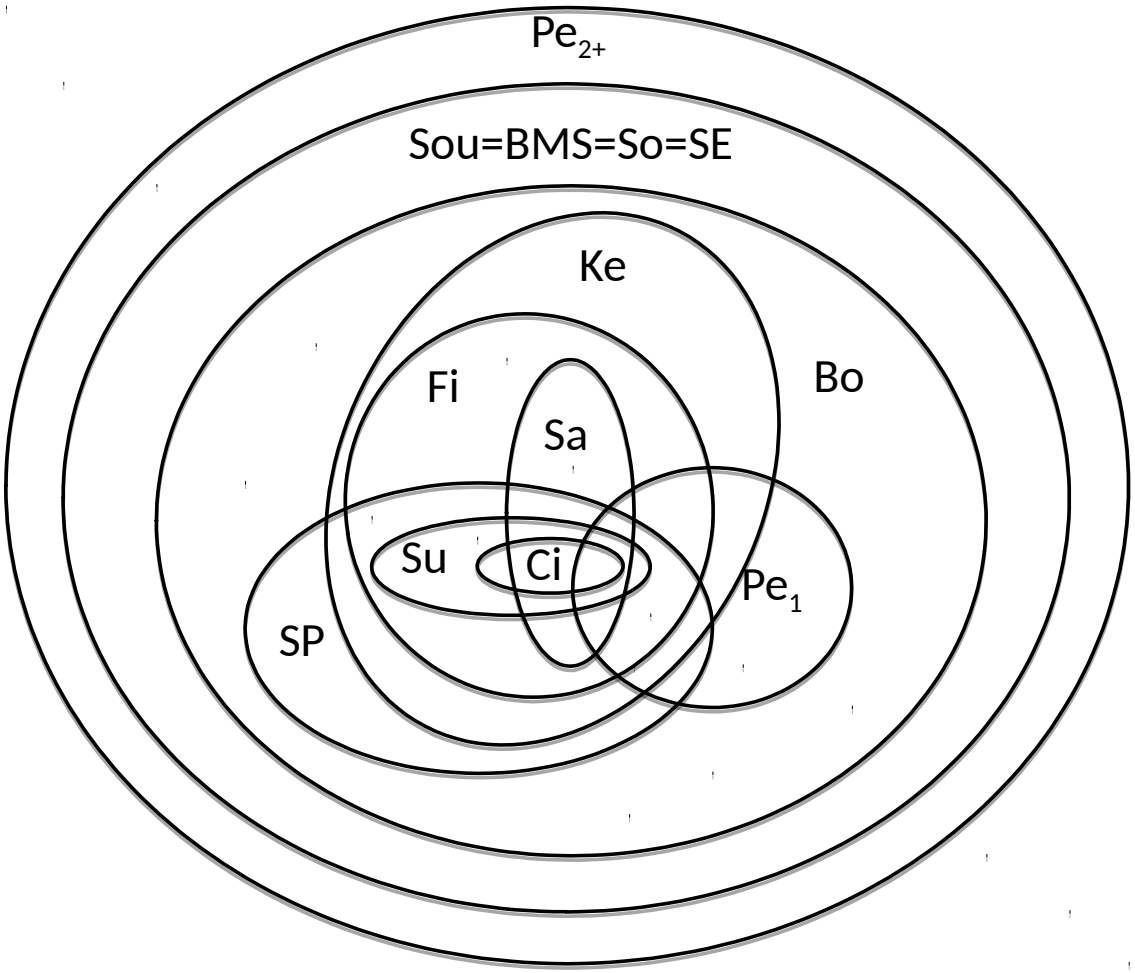
[a] Smigoc-Ellard [LAA2016]

- *Perfect-2^+ Set*

$$\Pi_{Pe_2^+} \equiv \Big\{ \Lambda \in \Pi_{\mathbb{R}} : \Lambda = \Big((\Lambda_1 - \rho(\Lambda_1)) \cup \alpha_1 \Big) \cup \cdots \cup \Big((\Lambda_k - \rho(\Lambda_k)) \cup \alpha_k \Big)$$

$$\text{where } \Lambda_1, \ldots, \Lambda_k \in \Pi_{Su};\ \alpha_1, \ldots, \alpha_k \geq 0;\ \text{and}$$

$$\exists A = \begin{bmatrix} \rho(\Lambda_1) & & * \\ & \ddots & \\ * & & \rho(\Lambda_k) \end{bmatrix} \geq 0 \quad \text{with } \sigma(A) = \{\alpha_1, \ldots, \alpha_k\} \Big\}$$



Suleimanova = Su
Suleimanova-Perfect = SP
Perfect 1 = Pe_1
Perfect 2^+ = Pe_{2+}
Ciarlet = Ci
Kellogg = Ke
Salzman = Sa
Fiedler = Fi
Borobia = Bo
Soules = Sou
Boro-Moro-Soto = BMS
Soto = So
Smigoc-Ellard = SE

Figure : $\Pi_{SP} \subset \Pi_{Bo} \subset \Pi_{Sou} = \Pi_{BMS} = \Pi_{So} = \Pi_{SE} \subset \Pi_{Pe_{2+}} \subset \Pi_{RNIEP}$

(*) Marijuán-Pisonero-Soto [LAA2007], Marijuán-Pisonero [ENDM2014], Smigoc-Ellard [LAA2016]

Recall: Which $(X, \Pi_{\mathbb{R}})-$Problems for $X \subseteq \Pi_{\text{RNIEP}}$ are **P** or **NP**?

Recall: Which $(X, \Pi_{\mathbb{R}})-$Problems for $X \subseteq \Pi_{\text{RNIEP}}$ are **P** or **NP**?

Complexity theory is inadequate to treat problems for real and complex numbers.

Recall: Which $(X, \Pi_{\mathbb{R}})-$Problems for $X \subseteq \Pi_{\text{RNIEP}}$ are **P** or **NP**?

Complexity theory is inadequate to treat problems for real and complex numbers.
To determine if a decision problem belongs to the class **P** or to the seemingly broader class **NP** implies typically that we deal with discrete problems, with integers or rationals, with graphs...

QNIEP

Recall: Which $(X, \Pi_{\mathbb{R}})-$Problems for $X \subseteq \Pi_{\text{RNIEP}}$ are **P** or **NP**?

Complexity theory is inadequate to treat problems for real and complex numbers.
To determine if a decision problem belongs to the class **P** or to the seemingly broader class **NP** implies typically that we deal with discrete problems, with integers or rationals, with graphs...

To *discretize* the NIEP and RNIEP we will consider the QNIEP

QNIEP

Recall: Which $(X, \Pi_{\mathbb{R}})-$Problems for $X \subseteq \Pi_{\text{RNIEP}}$ are **P** or **NP**?

Complexity theory is inadequate to treat problems for real and complex numbers.
To determine if a decision problem belongs to the class **P** or to the seemingly broader class **NP** implies typically that we deal with discrete problems, with integers or rationals, with graphs...

To *discretize* the NIEP and RNIEP we will consider the QNIEP

- $\Pi_{\mathbb{Q}}$ is composed by the rational elements of $\Pi_{\mathbb{R}}$

QNIEP

Recall: Which $(X, \Pi_{\mathbb{R}})-$Problems for $X \subseteq \Pi_{\text{RNIEP}}$ are **P** or **NP**?

Complexity theory is inadequate to treat problems for real and complex numbers.
To determine if a decision problem belongs to the class **P** or to the seemingly broader class **NP** implies typically that we deal with discrete problems, with integers or rationals, with graphs...

To *discretize* the NIEP and RNIEP we will consider the QNIEP

- $\Pi_{\mathbb{Q}}$ is composed by the rational elements of $\Pi_{\mathbb{R}}$
- Π_{QNIEP} is composed by the rational elements of Π_{RNIEP}

QNIEP

Recall: Which $(X, \Pi_{\mathbb{R}})-$Problems for $X \subseteq \Pi_{\text{RNIEP}}$ are **P** or **NP**?

Complexity theory is inadequate to treat problems for real and complex numbers.
To determine if a decision problem belongs to the class **P** or to the seemingly broader class **NP** implies typically that we deal with discrete problems, with integers or rationals, with graphs...

To *discretize* the NIEP and RNIEP we will consider the QNIEP

- $\Pi_{\mathbb{Q}}$ is composed by the rational elements of $\Pi_{\mathbb{R}}$
- Π_{QNIEP} is composed by the rational elements of Π_{RNIEP}
- For a criterion C the set Π_{QC} is composed by the rational elements of Π_{C}

QNIEP

Recall: Which $(X, \Pi_{\mathbb{R}})-$Problems for $X \subseteq \Pi_{\text{RNIEP}}$ are **P** or **NP**?

Complexity theory is inadequate to treat problems for real and complex numbers.
To determine if a decision problem belongs to the class **P** or to the seemingly broader class **NP** implies typically that we deal with discrete problems, with integers or rationals, with graphs...

To *discretize* the NIEP and RNIEP we will consider the QNIEP

- $\Pi_{\mathbb{Q}}$ is composed by the rational elements of $\Pi_{\mathbb{R}}$
- Π_{QNIEP} is composed by the rational elements of Π_{RNIEP}
- For a criterion C the set Π_{QC} is composed by the rational elements of Π_{C}
- Which $(X, \Pi_{\mathbb{Q}})-$Problems for $X \subseteq \Pi_{\text{QNIEP}}$ are in the class **P** or **NP**?

Subsets of Π_{RNIEP}: 1. Defined by linear inequalities

- *Suleĭmanova Set*

$$\Pi_{\text{QSu}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \Pi_{\mathbb{Q}} : \lambda_1 \geq 0 > \lambda_2 \geq \cdots \geq \lambda_n\right\}$$

- *Salzmann Set*

$$\Pi_{\text{QSa}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \Pi_{\mathbb{Q}} : \ \frac{\lambda_1 + \ldots + \lambda_n}{n} \geq \frac{\lambda_i + \lambda_{n-i+1}}{2} \ \text{ for } i = 2, \ldots, \lfloor \frac{n+1}{2} \rfloor \right\}$$

- *Fiedler Set*

$$\Pi_{\text{QFi}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \Pi_{\mathbb{Q}} : \ \lambda_1 + \ldots + \lambda_n \geq \frac{\sum_{i=2,\ldots,n-1} |\lambda_i + \lambda_{n-i+1}| - 2(\lambda_1 + \lambda_n)}{2} \right\}$$

Subsets of Π_{RNIEP}: 1. Defined by linear inequalities

- *Suleĭmanova Set*

$$\Pi_{\text{QSu}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \Pi_{\mathbb{Q}} : \lambda_1 \geq 0 > \lambda_2 \geq \cdots \geq \lambda_n\right\}$$

- *Salzmann Set*

$$\Pi_{\text{QSa}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \Pi_{\mathbb{Q}} : \ \frac{\lambda_1 + \ldots + \lambda_n}{n} \geq \frac{\lambda_i + \lambda_{n-i+1}}{2} \ \text{ for } i = 2, \ldots, \lfloor \frac{n+1}{2} \rfloor \right\}$$

- *Fiedler Set*

$$\Pi_{\text{QFi}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \Pi_{\mathbb{Q}} : \ \lambda_1 + \ldots + \lambda_n \geq \frac{\sum_{i=2,\ldots,n-1} |\lambda_i + \lambda_{n-i+1}| - 2(\lambda_1 + \lambda_n)}{2} \right\}$$

In all cases the $(\Pi_{\text{QX}}, \Pi_{\mathbb{Q}})$-Problem belongs to the class **P**

Subsets of Π_{RNIEP}: 1. Defined by linear inequalities

- *Suleĭmanova Set*

$$\Pi_{\text{QSu}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \Pi_{\mathbb{Q}} : \lambda_1 \geq 0 > \lambda_2 \geq \cdots \geq \lambda_n\right\}$$

- *Salzmann Set*

$$\Pi_{\text{QSa}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \Pi_{\mathbb{Q}} : \ \frac{\lambda_1 + \ldots + \lambda_n}{n} \geq \frac{\lambda_i + \lambda_{n-i+1}}{2} \ \text{ for } i = 2, \ldots, \lfloor \frac{n+1}{2} \rfloor \right\}$$

- *Fiedler Set*

$$\Pi_{\text{QFi}} \equiv \left\{(\lambda_1, \ldots, \lambda_n) \in \Pi_{\mathbb{Q}} : \ \lambda_1 + \ldots + \lambda_n \geq \frac{\sum_{i=2,\ldots,n-1} |\lambda_i + \lambda_{n-i+1}| - 2(\lambda_1 + \lambda_n)}{2} \right\}$$

In all cases the $(\Pi_{\text{QX}}, \Pi_{\mathbb{Q}})$-Problem belongs to the class **P**

Let $\Lambda = (\lambda_1, \ldots, \lambda_n) \in \Pi_{\mathbb{Q}}$. Observe that Π_{QX} is defined by a collection of at most n inequalities. Therefore, the overall process to check that $\Lambda \in \Pi_{\text{QX}}$ or not will employ at most quadratic time with respect to the size of the input Λ

2. Its elements admit a special partition

- *Suleĭmanova-Perfect Set*

$$\Pi_{\text{QSP}} \equiv \left\{ \Lambda \in \Pi_{\mathbb{Q}} : \exists \text{ a partition } \Lambda = \Lambda_1 \cup \cdots \cup \Lambda_k \text{ such that } \Lambda_1, \ldots, \Lambda_k \in \Pi_{\text{QSu}} \right\}$$

2. Its elements admit a special partition

- *Suleĭmanova-Perfect Set*

$$\Pi_{\text{QSP}} \equiv \left\{ \Lambda \in \Pi_{\mathbb{Q}} : \exists \text{ a partition } \Lambda = \Lambda_1 \cup \cdots \cup \Lambda_k \text{ such that } \Lambda_1, \ldots, \Lambda_k \in \Pi_{\text{QSu}} \right\}$$

The $(\Pi_{\text{QSP}}, \Pi_{\mathbb{Q}})$-Problem belongs to the class **NP** (so it is **NP-complete)**

2. Its elements admit a special partition

- *Suleĭmanova-Perfect Set*

$$\Pi_{\text{QSP}} \equiv \left\{ \Lambda \in \Pi_{\mathbb{Q}} : \exists \text{ a partition } \Lambda = \Lambda_1 \cup \cdots \cup \Lambda_k \text{ such that } \Lambda_1, \ldots, \Lambda_k \in \Pi_{\text{QSu}} \right\}$$

The $(\Pi_{\text{QSP}}, \Pi_{\mathbb{Q}})$-Problem belongs to the class **NP** (so it is **NP-complete)**

If $\Lambda = (\lambda_1, \ldots, \lambda_n) \in \Pi_{\text{QSP}}$ is a yes-instance, take as certificate for Λ any partition $\Lambda = \Lambda_1 \cup \cdots \cup \Lambda_k$ such that $\Lambda_1, \ldots, \Lambda_k \in \Pi_{\text{QSu}}$. Checking that $\Lambda_1, \ldots, \Lambda_k \in \Pi_{\text{QSu}}$ can be done in linear time

3. Defined recursively

Basic tools to construct new subsets from known subsets of Π_{QNIEP}

1. $(0) \in \Pi_{\text{QNIEP}}$
2. $\Lambda_1, \ldots, \Lambda_k \in \Pi_{\text{QNIEP}} \Rightarrow \Lambda_1 \cup \cdots \cup \Lambda_k \in \Pi_{\text{QNIEP}}$
3. If $(\lambda_1, \lambda_2, \ldots, \lambda_n) \in \Pi_{\text{QNIEP}} \Rightarrow (\lambda_1 + \epsilon, \lambda_2, \ldots, \lambda_n) \in \Pi_{\text{QNIEP}}\ \forall \epsilon > 0$
4. If $(\lambda_1, \ldots, \lambda_i, \ldots, \lambda_n) \in \Pi_{\text{QNIEP}} \Rightarrow (\lambda_1 + \epsilon, \ldots, \lambda_i \pm \epsilon, \ldots, \lambda_n) \in \Pi_{\text{QNIEP}}\ \forall \epsilon > 0$

3. Defined recursively

Basic tools to construct new subsets from known subsets of Π_{QNIEP}

1. $(0) \in \Pi_{\text{QNIEP}}$
2. $\Lambda_1, \ldots, \Lambda_k \in \Pi_{\text{QNIEP}} \Rightarrow \Lambda_1 \cup \cdots \cup \Lambda_k \in \Pi_{\text{QNIEP}}$
3. If $(\lambda_1, \lambda_2, \ldots, \lambda_n) \in \Pi_{\text{QNIEP}} \Rightarrow (\lambda_1 + \epsilon, \lambda_2, \ldots, \lambda_n) \in \Pi_{\text{QNIEP}}\ \forall \epsilon > 0$
4. If $(\lambda_1, \ldots, \lambda_i, \ldots, \lambda_n) \in \Pi_{\text{QNIEP}} \Rightarrow (\lambda_1 + \epsilon, \ldots, \lambda_i \pm \epsilon, \ldots, \lambda_n) \in \Pi_{\text{QNIEP}}\ \forall \epsilon > 0$

- *Borobia-Moro-Soto Set* is composed by lists obtained with the basic tools

3. Defined recursively

Basic tools to construct new subsets from known subsets of Π_{QNIEP}

1. $(0) \in \Pi_{\text{QNIEP}}$
2. $\Lambda_1, \ldots, \Lambda_k \in \Pi_{\text{QNIEP}} \Rightarrow \Lambda_1 \cup \cdots \cup \Lambda_k \in \Pi_{\text{QNIEP}}$
3. If $(\lambda_1, \lambda_2, \ldots, \lambda_n) \in \Pi_{\text{QNIEP}} \Rightarrow (\lambda_1 + \epsilon, \lambda_2, \ldots, \lambda_n) \in \Pi_{\text{QNIEP}}\ \forall \epsilon > 0$
4. If $(\lambda_1, \ldots, \lambda_i, \ldots, \lambda_n) \in \Pi_{\text{QNIEP}} \Rightarrow (\lambda_1 + \epsilon, \ldots, \lambda_i \pm \epsilon, \ldots, \lambda_n) \in \Pi_{\text{QNIEP}}\ \forall \epsilon > 0$

- *Borobia-Moro-Soto Set* is composed by lists obtained with the basic tools

The $(\Pi_{\text{QBMS}}, \Pi_{\mathbb{Q}})$-Problem belongs to the class **NP** (so it is **NP-complete)**

3. Defined recursively

Basic tools to construct new subsets from known subsets of Π_{QNIEP}

1. $(0) \in \Pi_{\text{QNIEP}}$
2. $\Lambda_1, \ldots, \Lambda_k \in \Pi_{\text{QNIEP}} \Rightarrow \Lambda_1 \cup \cdots \cup \Lambda_k \in \Pi_{\text{QNIEP}}$
3. If $(\lambda_1, \lambda_2, \ldots, \lambda_n) \in \Pi_{\text{QNIEP}} \Rightarrow (\lambda_1 + \epsilon, \lambda_2, \ldots, \lambda_n) \in \Pi_{\text{QNIEP}}\ \forall \epsilon > 0$
4. If $(\lambda_1, \ldots, \lambda_i, \ldots, \lambda_n) \in \Pi_{\text{QNIEP}} \Rightarrow (\lambda_1 + \epsilon, \ldots, \lambda_i \pm \epsilon, \ldots, \lambda_n) \in \Pi_{\text{QNIEP}}\ \forall \epsilon > 0$

- *Borobia-Moro-Soto Set* is composed by lists obtained with the basic tools

The $(\Pi_{\text{QBMS}}, \Pi_{\mathbb{Q}})$-Problem belongs to the class **NP** (so it is **NP-complete)**

If $\Lambda = (\lambda_1, \ldots, \lambda_n) \in \Pi_{\text{QBMS}}$ is a yes-instance, take as certificate for Λ any sequence of the allowed moves that transform the n trivially realizable lists $(0), (0), \ldots, (0)$ into the list $(\lambda_1, \ldots, \lambda_n)$. Checking that the moves perform this transformation can be done in polynomial time

- *Perfect-2^+ Set*

$$\Pi_{\mathrm{QPe}_2^+} \equiv \Big\{ \Lambda \in \Pi_{\mathbb{Q}} : \Lambda = \Big((\Lambda_1 - \rho(\Lambda_1)) \cup \alpha_1 \Big) \cup \cdots \cup \Big((\Lambda_k - \rho(\Lambda_k)) \cup \alpha_k \Big)$$

where $\Lambda_1, \ldots, \Lambda_k \in \Pi_{\mathrm{QSu}}$; $\alpha_1, \ldots, \alpha_k \geq 0$ rationals; and

$$\exists A = \begin{bmatrix} \rho(\Lambda_1) & & * \\ & \ddots & \\ * & & \rho(\Lambda_k) \end{bmatrix} \geq 0 \quad \text{with } \sigma(A) = \{\alpha_1, \ldots, \alpha_k\} \Big\}$$

4. Require the existence of an auxiliary nonnegative matrix

- *Perfect-2^+ Set*

$$\Pi_{\mathrm{QPe}_2^+} \equiv \left\{ \Lambda \in \Pi_{\mathbb{Q}} : \Lambda = \Big((\Lambda_1 - \rho(\Lambda_1)) \cup \alpha_1\Big) \cup \cdots \cup \Big((\Lambda_k - \rho(\Lambda_k)) \cup \alpha_k\Big) \right.$$

$$\text{where } \Lambda_1, \ldots, \Lambda_k \in \Pi_{\mathrm{QSu}};\ \alpha_1, \ldots, \alpha_k \geq 0 \text{ rationals; and}$$

$$\left. \exists A = \begin{bmatrix} \rho(\Lambda_1) & & * \\ & \ddots & \\ * & & \rho(\Lambda_k) \end{bmatrix} \geq 0 \quad \text{with } \sigma(A) = \{\alpha_1, \ldots, \alpha_k\} \right\}$$

We do not know if the $(\Pi_{\mathrm{QPe}_2^+}, \Pi_{\mathbb{Q}})$-Problem belongs to the class **NP**, since a certificate of membership in **NP** includes apparently a nonnegative matrix with prescribed diagonal and spectrum. We can not assure that it is a rational matrix and so we can not assure that it can be check that it has the prescribed spectrum in polynomial time

Open questions

We do not know if the $(\Pi_{\text{QNIEP}}, \Pi_{\mathbb{Q}})$-Problem belongs to the class **NP**, since a certificate of membership in **NP** includes a nonnegative matrix with prescribed spectrum. We can not assure that it is a rational matrix and so we can not assure that it can be check that it has the prescribed spectrum in polynomial time

Open questions

We do not know if the $(\Pi_{\text{QNIEP}}, \Pi_{\mathbb{Q}})$-Problem belongs to the class **NP**, since a certificate of membership in **NP** includes a nonnegative matrix with prescribed spectrum. We can not assure that it is a rational matrix and so we can not assure that it can be check that it has the prescribed spectrum in polynomial time

We finish with the following open questions:

Open questions

We do not know if the $(\Pi_{\text{QNIEP}}, \Pi_{\mathbb{Q}})$-Problem belongs to the class **NP**, since a certificate of membership in **NP** includes a nonnegative matrix with prescribed spectrum. We can not assure that it is a rational matrix and so we can not assure that it can be check that it has the prescribed spectrum in polynomial time

We finish with the following open questions:

> Let A be a square nonnegative matrix whose eigenvalues are rational numbers. Does there always exist a rational nonnegative matrix B with the same spectrum than A?

Open questions

We do not know if the $(\Pi_{\text{QNIEP}}, \Pi_{\mathbb{Q}})$-Problem belongs to the class **NP**, since a certificate of membership in **NP** includes a nonnegative matrix with prescribed spectrum. We can not assure that it is a rational matrix and so we can not assure that it can be check that it has the prescribed spectrum in polynomial time

We finish with the following open questions:

Let A be a square nonnegative matrix whose eigenvalues are rational numbers. Does there always exist a rational nonnegative matrix B with the same spectrum than A?

Let A be a square nonnegative matrix whose characteristic polynomial has rational coefficients. Does there always exist a rational nonnegative matrix B with the same characteristic polynomial than A?