

MO829  
Tópicos em Teoria da Computação  
Teoria dos Jogos Algorítmica

Rafael C. S. Schouery  
rafael@ic.unicamp.br

Universidade Estadual de Campinas

1º semestre/2017

## Complexidade computacional para TJA

# Recap. de complexidade computacional

# Recap. de complexidade computacional

No fim de 1960

# Recap. de complexidade computacional

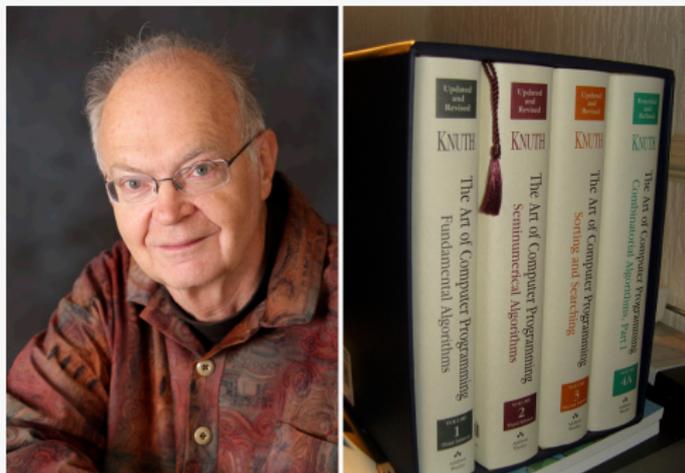
No fim de 1960

- Já era popular a análise formal de algoritmos

# Recap. de complexidade computacional

No fim de 1960

- Já era popular a análise formal de algoritmos
- Popularizado pelo trabalho de **Don Knuth** (The Art of Computer Programming, 68, 69, 73)



## Recap. de complexidade computacional

Mas alguns cientistas estavam intrigados:

## Recap. de complexidade computacional

Mas alguns cientistas estavam intrigados:

- Vários problemas podiam ser resolvidos rapidamente: ordenação, caminho mínimo, etc

## Recap. de complexidade computacional

Mas alguns cientistas estavam intrigados:

- Vários problemas podiam ser resolvidos rapidamente: ordenação, caminho mínimo, etc
- Mas havia outros problemas para os quais não se conhecia algoritmos rápidos

## Recap. de complexidade computacional

Mas alguns cientistas estavam intrigados:

- Vários problemas podiam ser resolvidos rapidamente: ordenação, caminho mínimo, etc
- Mas havia outros problemas para os quais não se conhecia algoritmos rápidos

Algoritmo rápido, ou eficiente:

## Recap. de complexidade computacional

Mas alguns cientistas estavam intrigados:

- Vários problemas podiam ser resolvidos rapidamente: ordenação, caminho mínimo, etc
- Mas havia outros problemas para os quais não se conhecia algoritmos rápidos

Algoritmo rápido, ou eficiente:

- com complexidade de tempo  $O(n^k)$  para alguma constante  $k$  (polinomial)

## Recap. de complexidade computacional

Mas alguns cientistas estavam intrigados:

- Vários problemas podiam ser resolvidos rapidamente: ordenação, caminho mínimo, etc
- Mas havia outros problemas para os quais não se conhecia algoritmos rápidos

Algoritmo rápido, ou eficiente:

- com complexidade de tempo  $O(n^k)$  para alguma constante  $k$  (polinomial)

Um algoritmo  $O(n^{100})$  pode não ser rápido na prática...

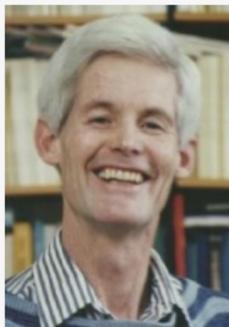
## Recap. de complexidade computacional

Será que poderemos achar algoritmos rápidos para vários problemas práticos que pertencem a uma classe especial chamada **NP**?

## Recap. de complexidade computacional

Será que poderemos achar algoritmos rápidos para vários problemas práticos que pertencem a uma classe especial chamada **NP**?

- **Cook** em 1971 e **Levin** em 1973 (independentemente) nos deram uma pista



## Recap. de complexidade computacional

Cook em 1971 mostrou:

# Recap. de complexidade computacional

Cook em 1971 mostrou:

- Todos os problemas da classe **NP** podem ser reduzidos em tempo **polinomial** para o problema de **Satisfatibilidade Booleana** (SAT)

# Recap. de complexidade computacional

Cook em 1971 mostrou:

- Todos os problemas da classe **NP** podem ser reduzidos em tempo **polinomial** para o problema de **Satisfatibilidade Booleana** (SAT)
- Ou seja, se tivermos um algoritmo polinomial para o **SAT** teremos um algoritmo polinomial para **todos** os problemas em **NP**

# Recap. de complexidade computacional

Cook em 1971 mostrou:

- Todos os problemas da classe **NP** podem ser reduzidos em tempo **polinomial** para o problema de **Satisfatibilidade Booleana** (SAT)
- Ou seja, se tivermos um algoritmo polinomial para o **SAT** teremos um algoritmo polinomial para **todos** os problemas em **NP**
- Este é o primeiro problema **NP-Completo**

## Recap. de complexidade computacional

Em 1972 **Richard Karp** mostrou como reduzir em tempo polinomial o **SAT** para outros 21 problemas importantes

## Recap. de complexidade computacional

Em 1972 **Richard Karp** mostrou como reduzir em tempo polinomial o **SAT** para outros 21 problemas importantes

Até hoje ninguém conseguiu encontrar um algoritmo **polinomial** para qualquer um dos problemas em **NP-Completo**

## Recap. de complexidade computacional

Em 1972 **Richard Karp** mostrou como reduzir em tempo polinomial o **SAT** para outros 21 problemas importantes

Até hoje ninguém conseguiu encontrar um algoritmo **polinomial** para qualquer um dos problemas em **NP-Completo**

Conjectura: **P = NP** ?

## Recap. de complexidade computacional

Muitas vezes problemas **NP-completos** são casos particulares ou podem ser reduzidos facilmente para outros de caráter mais prático, conhecidos como **NP-difíceis**

## Recap. de complexidade computacional

Muitas vezes problemas **NP-completos** são casos particulares ou podem ser reduzidos facilmente para outros de caráter mais prático, conhecidos como **NP-difíceis**

Exemplos:

## Recap. de complexidade computacional

Muitas vezes problemas **NP-completos** são casos particulares ou podem ser reduzidos facilmente para outros de caráter mais prático, conhecidos como **NP-difíceis**

Exemplos:

- Problema do Caixeiro Viajante

## Recap. de complexidade computacional

Muitas vezes problemas **NP-completos** são casos particulares ou podem ser reduzidos facilmente para outros de caráter mais prático, conhecidos como **NP-difíceis**

Exemplos:

- Problema do Caixeiro Viajante
- Escalonamento de Funcionários em Turnos de Trabalho

## Recap. de complexidade computacional

Muitas vezes problemas **NP-completos** são casos particulares ou podem ser reduzidos facilmente para outros de caráter mais prático, conhecidos como **NP-difíceis**

Exemplos:

- Problema do Caixeiro Viajante
- Escalonamento de Funcionários em Turnos de Trabalho
- Escalonamento de Tarefas em Computadores

## Recap. de complexidade computacional

Muitas vezes problemas **NP-completos** são casos particulares ou podem ser reduzidos facilmente para outros de caráter mais prático, conhecidos como **NP-difíceis**

Exemplos:

- Problema do Caixeiro Viajante
- Escalonamento de Funcionários em Turnos de Trabalho
- Escalonamento de Tarefas em Computadores
- Vários e vários outros problemas práticos...

# Representações sucintas de jogos

# Representações sucintas de jogos

Jogo de dois jogadores:

# Representações sucintas de jogos

Jogo de dois jogadores:

- um com  $m$  estratégias, outro com  $n$

# Representações sucintas de jogos

Jogo de dois jogadores:

- um com  $m$  estratégias, outro com  $n$
- $2mn$  números são necessários para representar tal jogo

# Representações sucintas de jogos

Jogo de dois jogadores:

- um com  $m$  estratégias, outro com  $n$
- $2mn$  números são necessários para representar tal jogo

Jogo com  $n$  jogadores, cada um com  $s$  estratégias:

# Representações sucintas de jogos

Jogo de dois jogadores:

- um com  $m$  estratégias, outro com  $n$
- $2mn$  números são necessários para representar tal jogo

Jogo com  $n$  jogadores, cada um com  $s$  estratégias:

- $ns^n$  números são necessários para representar tal jogo

# Representações sucintas de jogos

Jogo de dois jogadores:

- um com  $m$  estratégias, outro com  $n$
- $2mn$  números são necessários para representar tal jogo

Jogo com  $n$  jogadores, cada um com  $s$  estratégias:

- $ns^n$  números são necessários para representar tal jogo

A própria representação de um jogo é **exponencial**

# Representações sucintas de jogos

Jogo de dois jogadores:

- um com  $m$  estratégias, outro com  $n$
- $2mn$  números são necessários para representar tal jogo

Jogo com  $n$  jogadores, cada um com  $s$  estratégias:

- $ns^n$  números são necessários para representar tal jogo

A própria representação de um jogo é **exponencial**

- É fácil criar algoritmos polinomiais em  $ns^n$

# Representações sucintas de jogos

Jogo de dois jogadores:

- um com  $m$  estratégias, outro com  $n$
- $2mn$  números são necessários para representar tal jogo

Jogo com  $n$  jogadores, cada um com  $s$  estratégias:

- $ns^n$  números são necessários para representar tal jogo

A própria representação de um jogo é **exponencial**

- É fácil criar algoritmos polinomiais em  $ns^n$
- Mas ainda podem ser exponenciais em  $n$

# Relembrando o Jogo da Poluição

# Relembrando o Jogo da Poluição

- Conjunto de  $n$  países

# Relembrando o Jogo da Poluição

- Conjunto de  $n$  países
- Precisam decidir se **poluem** ou **não poluem**

# Relembrando o Jogo da Poluição

- Conjunto de  $n$  países
- Precisam decidir se **poluem** ou **não poluem**
- Não poluir custa **3**

# Relembrando o Jogo da Poluição

- Conjunto de  $n$  países
- Precisam decidir se **poluem** ou **não poluem**
- Não poluir custa **3**
- Cada país paga **1** por cada país poluente

# Relembrando o Jogo da Poluição

- Conjunto de  $n$  países
- Precisam decidir se **poluem** ou **não poluem**
- Não poluir custa **3**
- Cada país paga **1** por cada país poluente

Podemos representar esse jogo utilizando  $n2^n$  números

# Relembrando o Jogo da Poluição

- Conjunto de  $n$  países
- Precisam decidir se **poluem** ou **não poluem**
- Não poluir custa **3**
- Cada país paga **1** por cada país poluente

Podemos representar esse jogo utilizando  $n2^n$  números

Ou então, representar de maneira **sucinta**:

# Relembrando o Jogo da Poluição

- Conjunto de  $n$  países
- Precisam decidir se **poluem** ou **não poluem**
- Não poluir custa **3**
- Cada país paga **1** por cada país poluente

Podemos representar esse jogo utilizando  $n2^n$  números

Ou então, representar de maneira **sucinta**:

- O custo de um jogador depende de **quantos** jogadores poluem

# Relembrando o Jogo da Poluição

- Conjunto de  $n$  países
- Precisam decidir se **poluem** ou **não poluem**
- Não poluir custa **3**
- Cada país paga **1** por cada país poluente

Podemos representar esse jogo utilizando  $n2^n$  números

Ou então, representar de maneira **sucinta**:

- O custo de um jogador depende de **quantos** jogadores poluem
- Ao invés de depender de **quais** jogadores poluem

# Relembrando o Jogo da Poluição

- Conjunto de  $n$  países
- Precisam decidir se **poluem** ou **não poluem**
- Não poluir custa **3**
- Cada país paga **1** por cada país poluente

Podemos representar esse jogo utilizando  $n2^n$  números

Ou então, representar de maneira **sucinta**:

- O custo de um jogador depende de **quantos** jogadores poluem
- Ao invés de depender de **quais** jogadores poluem
- Podemos representar esse jogo com  $n + 1$  números

# Jogos com representações sucintas

# Jogos com representações sucintas

- Jogos **esparsos**

# Jogos com representações sucintas

- Jogos **esparcos**
  - ▶ Poucos dos  $n s^n$  números são diferentes de zero

# Jogos com representações sucintas

- Jogos **esparsos**
  - ▶ Poucos dos  $ns^n$  números são diferentes de zero
- Jogos **gráficos**

# Jogos com representações sucintas

- Jogos **esparsos**
  - ▶ Poucos dos  $ns^n$  números são diferentes de zero
- Jogos **gráficos**
  - ▶ A utilidade de um jogador depende apenas de alguns outros jogadores

# Jogos com representações sucintas

- Jogos **esparsos**
  - ▶ Poucos dos  $ns^n$  números são diferentes de zero
- Jogos **gráficos**
  - ▶ A utilidade de um jogador depende apenas de alguns outros jogadores
  - ▶  $ns^{d+1}$  números se  $d$  é grau máximo do grafo que representa o jogo

# Jogos com representações sucintas

- Jogos **esparsos**
  - ▶ Poucos dos  $ns^n$  números são diferentes de zero
- Jogos **gráficos**
  - ▶ A utilidade de um jogador depende apenas de alguns outros jogadores
  - ▶  $ns^{d+1}$  números se  $d$  é grau máximo do grafo que representa o jogo
- Jogos **simétricos**

# Jogos com representações sucintas

- Jogos **esparsos**
  - ▶ Poucos dos  $ns^n$  números são diferentes de zero
- Jogos **gráficos**
  - ▶ A utilidade de um jogador depende apenas de alguns outros jogadores
  - ▶  $ns^{d+1}$  números se  $d$  é grau máximo do grafo que representa o jogo
- Jogos **simétricos**
  - ▶ Todos os jogadores são idênticos

# Jogos com representações sucintas

- Jogos **esparsos**
  - ▶ Poucos dos  $ns^n$  números são diferentes de zero
- Jogos **gráficos**
  - ▶ A utilidade de um jogador depende apenas de alguns outros jogadores
  - ▶  $ns^{d+1}$  números se  $d$  é grau máximo do grafo que representa o jogo
- Jogos **simétricos**
  - ▶ Todos os jogadores são idênticos
  - ▶ A utilidade depende de quantos (ao invés de quais) jogadores jogam cada uma das estratégias

# Jogos com representações sucintas

- Jogos **esparsos**
  - ▶ Poucos dos  $ns^n$  números são diferentes de zero
- Jogos **gráficos**
  - ▶ A utilidade de um jogador depende apenas de alguns outros jogadores
  - ▶  $ns^{d+1}$  números se  $d$  é grau máximo do grafo que representa o jogo
- Jogos **simétricos**
  - ▶ Todos os jogadores são idênticos
  - ▶ A utilidade depende de quantos (ao invés de quais) jogadores jogam cada uma das estratégias
  - ▶ Podemos representar com  $s \binom{n+s-1}{s-1}$  números

# Jogos com representações sucintas

- Jogos **esparsos**
  - ▶ Poucos dos  $ns^n$  números são diferentes de zero
- Jogos **gráficos**
  - ▶ A utilidade de um jogador depende apenas de alguns outros jogadores
  - ▶  $ns^{d+1}$  números se  $d$  é grau máximo do grafo que representa o jogo
- Jogos **simétricos**
  - ▶ Todos os jogadores são idênticos
  - ▶ A utilidade depende de quantos (ao invés de quais) jogadores jogam cada uma das estratégias
  - ▶ Podemos representar com  $s \binom{n+s-1}{s-1}$  números
- Jogos **anônimos**

# Jogos com representações sucintas

- Jogos **esparsos**
  - ▶ Poucos dos  $ns^n$  números são diferentes de zero
- Jogos **gráficos**
  - ▶ A utilidade de um jogador depende apenas de alguns outros jogadores
  - ▶  $ns^{d+1}$  números se  $d$  é grau máximo do grafo que representa o jogo
- Jogos **simétricos**
  - ▶ Todos os jogadores são idênticos
  - ▶ A utilidade depende de quantos (ao invés de quais) jogadores jogam cada uma das estratégias
  - ▶ Podemos representar com  $s \binom{n+s-1}{s-1}$  números
- Jogos **anônimos**
  - ▶ Os jogadores não distinguem os outros jogadores

# Jogos com representações sucintas

- Jogos **esparsos**
  - ▶ Poucos dos  $ns^n$  números são diferentes de zero
- Jogos **gráficos**
  - ▶ A utilidade de um jogador depende apenas de alguns outros jogadores
  - ▶  $ns^{d+1}$  números se  $d$  é grau máximo do grafo que representa o jogo
- Jogos **simétricos**
  - ▶ Todos os jogadores são idênticos
  - ▶ A utilidade depende de quantos (ao invés de quais) jogadores jogam cada uma das estratégias
  - ▶ Podemos representar com  $s \binom{n+s-1}{s-1}$  números
- Jogos **anônimos**
  - ▶ Os jogadores não distinguem os outros jogadores
  - ▶ Podemos representar com  $ns \binom{n+s-1}{s-1}$  números

# Jogos com representações sucintas

- Jogos **esparcos**
  - ▶ Poucos dos  $ns^n$  números são diferentes de zero
- Jogos **gráficos**
  - ▶ A utilidade de um jogador depende apenas de alguns outros jogadores
  - ▶  $ns^{d+1}$  números se  $d$  é grau máximo do grafo que representa o jogo
- Jogos **simétricos**
  - ▶ Todos os jogadores são idênticos
  - ▶ A utilidade depende de quantos (ao invés de quais) jogadores jogam cada uma das estratégias
  - ▶ Podemos representar com  $s \binom{n+s-1}{s-1}$  números
- Jogos **anônimos**
  - ▶ Os jogadores não distinguem os outros jogadores
  - ▶ Podemos representar com  $ns \binom{n+s-1}{s-1}$  números
- Entre outros...

# Estrutura de um equilíbrio misto

## Estrutura de um equilíbrio misto

**Suporte** de um vetor: conjunto dos índices das entradas não nulas

## Estrutura de um equilíbrio misto

**Suporte** de um vetor: conjunto dos índices das entradas não nulas

- Ex:  $(0, -1, 0, 2)$  tem como suporte  $\{2, 4\}$

## Estrutura de um equilíbrio misto

**Suporte** de um vetor: conjunto dos índices das entradas não nulas

- Ex:  $(0, -1, 0, 2)$  tem como suporte  $\{2, 4\}$

**Teorema:** Uma estratégia **mista** é uma **resposta ótima** se e somente todas as estratégias **puras** no seu **suporte** são **respostas ótimas**

## Estrutura de um equilíbrio misto

**Suporte** de um vetor: conjunto dos índices das entradas não nulas

- Ex:  $(0, -1, 0, 2)$  tem como suporte  $\{2, 4\}$

**Teorema:** Uma estratégia **mista** é uma **resposta ótima** se e somente todas as estratégias **puras** no seu **suporte** são **respostas ótimas**

**Prova:**

## Estrutura de um equilíbrio misto

**Suporte** de um vetor: conjunto dos índices das entradas não nulas

- Ex:  $(0, -1, 0, 2)$  tem como suporte  $\{2, 4\}$

**Teorema:** Uma estratégia **mista** é uma **resposta ótima** se e somente todas as estratégias **puras** no seu **suporte** são **respostas ótimas**

**Prova:** Note que

## Estrutura de um equilíbrio misto

**Suporte** de um vetor: conjunto dos índices das entradas não nulas

- Ex:  $(0, -1, 0, 2)$  tem como suporte  $\{2, 4\}$

**Teorema:** Uma estratégia **mista** é uma **resposta ótima** se e somente todas as estratégias **puras** no seu **suporte** são **respostas ótimas**

**Prova:** Note que

$$\mathbb{E}[u_i(\sigma)]$$

## Estrutura de um equilíbrio misto

**Suporte** de um vetor: conjunto dos índices das entradas não nulas

- Ex:  $(0, -1, 0, 2)$  tem como suporte  $\{2, 4\}$

**Teorema:** Uma estratégia **mista** é uma **resposta ótima** se e somente todas as estratégias **puras** no seu **suporte** são **respostas ótimas**

**Prova:** Note que

$$\mathbb{E}[u_i(\sigma)] = \sum_{s \in S} \sigma(s) u_i(s)$$

## Estrutura de um equilíbrio misto

**Suporte** de um vetor: conjunto dos índices das entradas não nulas

- Ex:  $(0, -1, 0, 2)$  tem como suporte  $\{2, 4\}$

**Teorema:** Uma estratégia **mista** é uma **resposta ótima** se e somente todas as estratégias **puras** no seu **suporte** são **respostas ótimas**

**Prova:** Note que

$$\mathbb{E}[u_i(\sigma)] = \sum_{s \in S} \sigma(s) u_i(s) = \sum_{s_i \in S_i} \sum_{s_{-i} \in S_{-i}} \sigma(s_i, s_{-i}) u_i(s_i, s_{-i})$$

## Estrutura de um equilíbrio misto

**Suporte** de um vetor: conjunto dos índices das entradas não nulas

- Ex:  $(0, -1, 0, 2)$  tem como suporte  $\{2, 4\}$

**Teorema:** Uma estratégia **mista** é uma **resposta ótima** se e somente todas as estratégias **puras** no seu **suporte** são **respostas ótimas**

**Prova:** Note que

$$\begin{aligned}\mathbb{E}[u_i(\sigma)] &= \sum_{s \in S} \sigma(s) u_i(s) = \sum_{s_i \in S_i} \sum_{s_{-i} \in S_{-i}} \sigma(s_i, s_{-i}) u_i(s_i, s_{-i}) \\ &= \sum_{s_i \in S_i} \sigma_i(s_i) \sum_{s_{-i} \in S_{-i}} \sigma_{-i}(s_{-i}) u_i(s_i, s_{-i})\end{aligned}$$

## Estrutura de um equilíbrio misto

**Suporte** de um vetor: conjunto dos índices das entradas não nulas

- Ex:  $(0, -1, 0, 2)$  tem como suporte  $\{2, 4\}$

**Teorema:** Uma estratégia **mista** é uma **resposta ótima** se e somente todas as estratégias **puras** no seu **suporte** são **respostas ótimas**

**Prova:** Note que

$$\begin{aligned}\mathbb{E}[u_i(\sigma)] &= \sum_{s \in S} \sigma(s) u_i(s) = \sum_{s_i \in S_i} \sum_{s_{-i} \in S_{-i}} \sigma(s_i, s_{-i}) u_i(s_i, s_{-i}) \\ &= \sum_{s_i \in S_i} \sigma_i(s_i) \sum_{s_{-i} \in S_{-i}} \sigma_{-i}(s_{-i}) u_i(s_i, s_{-i}) \\ &= \sum_{s_i \in S_i} \sigma_i(s_i) \mathbb{E}[u_i(s_i, \sigma_{-i})]\end{aligned}$$

# Suporte e Respostas Ótimas

$$\mathbb{E}[u_i(\sigma)] = \sum_{s_i \in S_i} \sigma(s_i) \mathbb{E}[u_i(s_i, \sigma_{-i})]$$

# Suporte e Respostas Ótimas

$$\mathbb{E}[u_i(\sigma)] = \sum_{s_i \in S_i} \sigma(s_i) \mathbb{E}[u_i(s_i, \sigma_{-i})]$$

Se  $\sigma_i$  é uma resposta ótima para  $\sigma_{-i}$ , então todo  $s_i'$  no suporte de  $\sigma_i$  é uma resposta ótima para  $\sigma_{-i}$ :

# Suporte e Respostas Ótimas

$$\mathbb{E}[u_i(\sigma)] = \sum_{s_i \in S_i} \sigma(s_i) \mathbb{E}[u_i(s_i, \sigma_{-i})]$$

Se  $\sigma_i$  é uma resposta ótima para  $\sigma_{-i}$ , então todo  $s_i'$  no suporte de  $\sigma_i$  é uma resposta ótima para  $\sigma_{-i}$ :

- Caso contrário, poderíamos:

# Suporte e Respostas Ótimas

$$\mathbb{E}[u_i(\sigma)] = \sum_{s_i \in S_i} \sigma(s_i) \mathbb{E}[u_i(s_i, \sigma_{-i})]$$

Se  $\sigma_i$  é uma resposta ótima para  $\sigma_{-i}$ , então todo  $s_{i'}$  no suporte de  $\sigma_i$  é uma resposta ótima para  $\sigma_{-i}$ :

- Caso contrário, poderíamos:
  - ▶ zerar  $\sigma(s_{i'})$  e

# Suporte e Respostas Ótimas

$$\mathbb{E}[u_i(\sigma)] = \sum_{s_i \in S_i} \sigma(s_i) \mathbb{E}[u_i(s_i, \sigma_{-i})]$$

Se  $\sigma_i$  é uma resposta ótima para  $\sigma_{-i}$ , então todo  $s_{i'}$  no suporte de  $\sigma_i$  é uma resposta ótima para  $\sigma_{-i}$ :

- Caso contrário, poderíamos:
  - ▶ zerar  $\sigma(s_{i'})$  e
  - ▶ para uma resposta pura ótima  $s_{i^*}$  para  $\sigma_{-i}$ , poderíamos aumentar  $\sigma(s_{i^*})$  (em  $\sigma(s_{i'})$ )

# Suporte e Respostas Ótimas

$$\mathbb{E}[u_i(\sigma)] = \sum_{s_i \in S_i} \sigma(s_i) \mathbb{E}[u_i(s_i, \sigma_{-i})]$$

Se  $\sigma_i$  é uma resposta ótima para  $\sigma_{-i}$ , então todo  $s_{i'}$  no suporte de  $\sigma_i$  é uma resposta ótima para  $\sigma_{-i}$ :

- Caso contrário, poderíamos:
  - ▶ zerar  $\sigma(s_{i'})$  e
  - ▶ para uma resposta pura ótima  $s_{i^*}$  para  $\sigma_{-i}$ , poderíamos aumentar  $\sigma(s_{i^*})$  (em  $\sigma(s_{i'})$ )
  - ▶ Com isso, aumentaríamos a utilidade

# Suporte e Respostas Ótimas

$$\mathbb{E}[u_i(\sigma)] = \sum_{s_i \in S_i} \sigma(s_i) \mathbb{E}[u_i(s_i, \sigma_{-i})]$$

# Suporte e Respostas Ótimas

$$\mathbb{E}[u_i(\sigma)] = \sum_{s_i \in S_i} \sigma(s_i) \mathbb{E}[u_i(s_i, \sigma_{-i})]$$

Se todo  $s_i'$  no suporte de  $\sigma_i$  é uma resposta ótima para  $\sigma_{-i}$ ,  
então  $\sigma_i$  é uma resposta ótima para  $\sigma_{-i}$

# Suporte e Respostas Ótimas

$$\mathbb{E}[u_i(\sigma)] = \sum_{s_i \in S_i} \sigma(s_i) \mathbb{E}[u_i(s_i, \sigma_{-i})]$$

Se todo  $s_i'$  no suporte de  $\sigma_i$  é uma resposta ótima para  $\sigma_{-i}$ , então  $\sigma_i$  é uma resposta ótima para  $\sigma_{-i}$

- Se  $s_i'$  e  $s_i^*$  são ambas respostas ótimas para  $\sigma_{-i}$ , então  $\mathbb{E}[u_i(s_i', \sigma_{-i})] = \mathbb{E}[u_i(s_i^*, \sigma_{-i})]$

# Suporte e Respostas Ótimas

$$\mathbb{E}[u_i(\sigma)] = \sum_{s_i \in S_i} \sigma(s_i) \mathbb{E}[u_i(s_i, \sigma_{-i})]$$

Se todo  $s_{i'}$  no suporte de  $\sigma_i$  é uma resposta ótima para  $\sigma_{-i}$ , então  $\sigma_i$  é uma resposta ótima para  $\sigma_{-i}$

- Se  $s_{i'}$  e  $s_{i^*}$  são ambas respostas ótimas para  $\sigma_{-i}$ , então  $\mathbb{E}[u_i(s_{i'}, \sigma_{-i})] = \mathbb{E}[u_i(s_{i^*}, \sigma_{-i})]$
- Todo termo não nulo de  $\sum_{s_i \in S_i} \sigma(s_i) \mathbb{E}[u_i(s_i, \sigma_{-i})]$  tem o mesmo valor e é igual a  $\sigma(s_i) \mathbb{E}[u_i(s_{i'}, \sigma_{-i})]$

# Suporte e Respostas Ótimas

$$\mathbb{E}[u_i(\sigma)] = \sum_{s_i \in S_i} \sigma(s_i) \mathbb{E}[u_i(s_i, \sigma_{-i})]$$

Se todo  $s_{i'}$  no suporte de  $\sigma_i$  é uma resposta ótima para  $\sigma_{-i}$ , então  $\sigma_i$  é uma resposta ótima para  $\sigma_{-i}$

- Se  $s_{i'}$  e  $s_{i^*}$  são ambas respostas ótimas para  $\sigma_{-i}$ , então  $\mathbb{E}[u_i(s_{i'}, \sigma_{-i})] = \mathbb{E}[u_i(s_{i^*}, \sigma_{-i})]$
- Todo termo não nulo de  $\sum_{s_i \in S_i} \sigma(s_i) \mathbb{E}[u_i(s_i, \sigma_{-i})]$  tem o mesmo valor e é igual a  $\sigma(s_i) \mathbb{E}[u_i(s_{i'}, \sigma_{-i})]$
- Assim,  $\sum_{s_i \in S_i} \sigma(s_i) \mathbb{E}[u_i(s_i, \sigma_{-i})] = \mathbb{E}[u_i(s_{i'}, \sigma_{-i})]$

# Jogos de soma zero

# Jogos de soma zero

Em um jogo de **soma zero** com  $n$  jogadores, para qualquer perfil  $s \in S$  temos que

$$\sum_{i=1}^n u_i(s) = 0$$

# Jogos de soma zero

Em um jogo de **soma zero** com  $n$  jogadores, para qualquer perfil  $s \in S$  temos que

$$\sum_{i=1}^n u_i(s) = 0$$

Trata-se de um conceito muito comum em economia

# Jogos de soma zero

Em um jogo de **soma zero** com  $n$  jogadores, para qualquer perfil  $s \in S$  temos que

$$\sum_{i=1}^n u_i(s) = 0$$

Trata-se de um conceito muito comum em economia

Exemplos: Par-ou-Ímpar, Pedra-Papel-Tesoura, Xadrez, Poker, etc...

# Jogos de soma zero com dois jogadores

## Jogos de soma zero com dois jogadores

Basta dar **uma** matriz, pois o ganho do jogador **1** é o custo do jogador **2** e vice-versa

# Jogos de soma zero com dois jogadores

Basta dar **uma** matriz, pois o ganho do jogador **1** é o custo do jogador **2** e vice-versa

**Problema:** Dada uma matriz  $A_{m \times n}$ , encontrar um equilíbrio misto de Nash

# Jogos de soma zero com dois jogadores

Basta dar **uma** matriz, pois o ganho do jogador **1** é o custo do jogador **2** e vice-versa

**Problema:** Dada uma matriz  $A_{m \times n}$ , encontrar um equilíbrio misto de Nash

- $m$ : número de estratégias do jogador **1**

# Jogos de soma zero com dois jogadores

Basta dar **uma** matriz, pois o ganho do jogador **1** é o custo do jogador **2** e vice-versa

**Problema:** Dada uma matriz  $A_{m \times n}$ , encontrar um equilíbrio misto de Nash

- $m$ : número de estratégias do jogador **1**
- $n$ : número de estratégias do jogador **2**

# Jogos de soma zero com dois jogadores

Basta dar **uma** matriz, pois o ganho do jogador **1** é o custo do jogador **2** e vice-versa

**Problema:** Dada uma matriz  $A_{m \times n}$ , encontrar um equilíbrio misto de Nash

- $m$ : número de estratégias do jogador **1**
- $n$ : número de estratégias do jogador **2**

Pelo Teorema de Nash, **sempre** existe um equilíbrio

# Jogos de soma zero com dois jogadores

Basta dar **uma** matriz, pois o ganho do jogador **1** é o custo do jogador **2** e vice-versa

**Problema:** Dada uma matriz  $A_{m \times n}$ , encontrar um equilíbrio misto de Nash

- $m$ : número de estratégias do jogador **1**
- $n$ : número de estratégias do jogador **2**

Pelo Teorema de Nash, **sempre** existe um equilíbrio

Como podemos resolver esse problema?

## Jogos de soma zero com dois jogadores

**Problema:** Dada uma matriz  $A_{m \times n}$ , encontrar um equilíbrio misto de Nash.

# Jogos de soma zero com dois jogadores

**Problema:** Dada uma matriz  $A_{m \times n}$ , encontrar um equilíbrio misto de Nash.

O que é um **equilíbrio**?

# Jogos de soma zero com dois jogadores

**Problema:** Dada uma matriz  $A_{m \times n}$ , encontrar um equilíbrio misto de Nash.

O que é um **equilíbrio**?

São duas distribuições de probabilidade, ou seja, vetores  $p$ , com  $m$  entradas, e  $q$ , com  $n$  entradas,

# Jogos de soma zero com dois jogadores

**Problema:** Dada uma matriz  $A_{m \times n}$ , encontrar um equilíbrio misto de Nash.

O que é um **equilíbrio**?

São duas distribuições de probabilidade, ou seja, vetores  $p$ , com  $m$  entradas, e  $q$ , com  $n$  entradas,

- $\sum_{i=1}^m p_i = 1$  e  $p_i \geq 0$  para  $i = 1, \dots, m$

# Jogos de soma zero com dois jogadores

**Problema:** Dada uma matriz  $A_{m \times n}$ , encontrar um equilíbrio misto de Nash.

O que é um **equilíbrio**?

São duas distribuições de probabilidade, ou seja, vetores  $p$ , com  $m$  entradas, e  $q$ , com  $n$  entradas,

- $\sum_{i=1}^m p_i = 1$  e  $p_i \geq 0$  para  $i = 1, \dots, m$
- $\sum_{j=1}^n q_j = 1$  e  $q_j \geq 0$  para  $j = 1, \dots, n$

# Jogos de soma zero com dois jogadores

**Problema:** Dada uma matriz  $A_{m \times n}$ , encontrar um equilíbrio misto de Nash.

O que é um **equilíbrio**?

São duas distribuições de probabilidade, ou seja, vetores  $p$ , com  $m$  entradas, e  $q$ , com  $n$  entradas,

- $\sum_{i=1}^m p_i = 1$  e  $p_i \geq 0$  para  $i = 1, \dots, m$
- $\sum_{j=1}^n q_j = 1$  e  $q_j \geq 0$  para  $j = 1, \dots, n$

tais que um é uma resposta ótima para o outro...

# Jogos de soma zero com dois jogadores

**Problema:** Dada uma matriz  $A_{m \times n}$ , encontrar um equilíbrio misto de Nash.

O que é um **equilíbrio**?

São duas distribuições de probabilidade, ou seja, vetores  $p$ , com  $m$  entradas, e  $q$ , com  $n$  entradas,

- $\sum_{i=1}^m p_i = 1$  e  $p_i \geq 0$  para  $i = 1, \dots, m$
- $\sum_{j=1}^n q_j = 1$  e  $q_j \geq 0$  para  $j = 1, \dots, n$

tais que um é uma resposta ótima para o outro...

Se o jogador **1** usa a estratégia (mista)  $p$  e o **2** usa a  $q$ , qual é o valor esperado que o jogador **1** ganha?

## Jogos de soma zero com dois jogadores

Considere que  $A_{m \times n}$  é a matriz de utilidade do jogador 1

Se o jogador 1 usa a estratégia (mista)  $p$  e o 2 usa a  $q$ , qual é o valor esperado que o jogador 1 ganha?

## Jogos de soma zero com dois jogadores

Considere que  $A_{m \times n}$  é a matriz de utilidade do jogador 1

Se o jogador 1 usa a estratégia (mista)  $p$  e o 2 usa a  $q$ , qual é o valor esperado que o jogador 1 ganha?

$$v := \sum_{i=1}^m \sum_{j=1}^n a_{ij} p_i q_j$$

## Jogos de soma zero com dois jogadores

Considere que  $A_{m \times n}$  é a matriz de utilidade do jogador 1

Se o jogador 1 usa a estratégia (mista)  $p$  e o 2 usa a  $q$ , qual é o valor esperado que o jogador 1 ganha?

$$v := \sum_{i=1}^m \sum_{j=1}^n a_{ij} p_i q_j = \sum_{i=1}^m p_i \sum_{j=1}^n a_{ij} q_j$$

## Jogos de soma zero com dois jogadores

Considere que  $A_{m \times n}$  é a matriz de utilidade do jogador 1

Se o jogador 1 usa a estratégia (mista)  $p$  e o 2 usa a  $q$ , qual é o valor esperado que o jogador 1 ganha?

$$v := \sum_{i=1}^m \sum_{j=1}^n a_{ij} p_i q_j = \sum_{i=1}^m p_i \sum_{j=1}^n a_{ij} q_j = \sum_{j=1}^n q_j \sum_{i=1}^m a_{ij} p_i$$

## Jogos de soma zero com dois jogadores

Considere que  $A_{m \times n}$  é a matriz de utilidade do jogador 1

Se o jogador 1 usa a estratégia (mista)  $p$  e o 2 usa a  $q$ , qual é o valor esperado que o jogador 1 ganha?

$$v := \sum_{i=1}^m \sum_{j=1}^n a_{ij} p_i q_j = \sum_{i=1}^m p_i \sum_{j=1}^n a_{ij} q_j = \sum_{j=1}^n q_j \sum_{i=1}^m a_{ij} p_i$$

Vimos que, se  $p$  é uma resposta ótima para  $q$ , então para toda estratégia  $i'$  no suporte de  $p$  temos que  $\sum_{j=1}^n a_{i'j} q_j = v$

# Jogos de soma zero com dois jogadores

Considere que  $A_{m \times n}$  é a matriz de utilidade do jogador 1

Se o jogador 1 usa a estratégia (mista)  $p$  e o 2 usa a  $q$ , qual é o valor esperado que o jogador 1 ganha?

$$v := \sum_{i=1}^m \sum_{j=1}^n a_{ij} p_i q_j = \sum_{i=1}^m p_i \sum_{j=1}^n a_{ij} q_j = \sum_{j=1}^n q_j \sum_{i=1}^m a_{ij} p_i$$

Vimos que, se  $p$  é uma resposta ótima para  $q$ , então para toda estratégia  $i'$  no suporte de  $p$  temos que  $\sum_{j=1}^n a_{i'j} q_j = v$

E se  $q$  é uma resposta ótima para  $p$ , então para toda estratégia pura  $j'$  no suporte de  $q$  temos que  $\sum_{i=1}^m (-a_{ij'}) p_i = -v$

## Jogos de soma zero com dois jogadores

Assim, o jogador 1 quer encontrar  $p$  tal que

## Jogos de soma zero com dois jogadores

Assim, o jogador 1 quer encontrar  $p$  tal que

- $p$  é uma estratégia mista (distribuição de probabilidades)

## Jogos de soma zero com dois jogadores

Assim, o jogador 1 quer encontrar  $p$  tal que

- $p$  é uma estratégia mista (distribuição de probabilidades)
- sendo que o jogador 2 escolhe uma resposta ótima

# Jogos de soma zero com dois jogadores

Assim, o jogador 1 quer encontrar  $p$  tal que

- $p$  é uma estratégia mista (distribuição de probabilidades)
- sendo que o jogador 2 escolhe uma resposta ótima
  - ▶ aleatoriza entre  $j$  tal que  $\sum_{i=1}^m (-a_{ij})p_i$  é máximo

# Jogos de soma zero com dois jogadores

Assim, o jogador 1 quer encontrar  $p$  tal que

- $p$  é uma estratégia mista (distribuição de probabilidades)
- sendo que o jogador 2 escolhe uma resposta ótima
  - ▶ aleatoriza entre  $j$  tal que  $\sum_{i=1}^m (-a_{ij'})p_i$  é máximo
  - ▶  $j$  está no suporte sse  $\sum_{i=1}^m a_{ij'}p_i$  é mínimo

# Jogos de soma zero com dois jogadores

Assim, o jogador 1 quer encontrar  $p$  tal que

- $p$  é uma estratégia mista (distribuição de probabilidades)
- sendo que o jogador 2 escolhe uma resposta ótima
  - ▶ aleatoriza entre  $j$  tal que  $\sum_{i=1}^m (-a_{ij'})p_i$  é máximo
  - ▶  $j$  está no suporte sse  $\sum_{i=1}^m a_{ij'}p_i$  é mínimo
- Sendo que o jogo é de soma zero

# Jogos de soma zero com dois jogadores

Assim, o jogador 1 quer encontrar  $p$  tal que

- $p$  é uma estratégia mista (distribuição de probabilidades)
- sendo que o jogador 2 escolhe uma resposta ótima
  - ▶ aleatoriza entre  $j$  tal que  $\sum_{i=1}^m (-a_{ij'})p_i$  é máximo
  - ▶  $j$  está no suporte sse  $\sum_{i=1}^m a_{ij'}p_i$  é mínimo
- Sendo que o jogo é de soma zero
  - ▶ Basta minimizar o ganho do outro para maximizar o seu

# Jogos de soma zero com dois jogadores

Assim, o jogador 1 quer encontrar  $p$  tal que

- $p$  é uma estratégia mista (distribuição de probabilidades)
- sendo que o jogador 2 escolhe uma resposta ótima
  - ▶ aleatoriza entre  $j$  tal que  $\sum_{i=1}^m (-a_{ij'})p_i$  é máximo
  - ▶  $j$  está no suporte sse  $\sum_{i=1}^m a_{ij'}p_i$  é mínimo
- Sendo que o jogo é de soma zero
  - ▶ Basta minimizar o ganho do outro para maximizar o seu

maximize  $v$

# Jogos de soma zero com dois jogadores

Assim, o jogador 1 quer encontrar  $p$  tal que

- $p$  é uma estratégia mista (distribuição de probabilidades)
- sendo que o jogador 2 escolhe uma resposta ótima
  - ▶ aleatoriza entre  $j$  tal que  $\sum_{i=1}^m (-a_{ij'})p_i$  é máximo
  - ▶  $j$  está no suporte sse  $\sum_{i=1}^m a_{ij'}p_i$  é mínimo
- Sendo que o jogo é de soma zero
  - ▶ Basta minimizar o ganho do outro para maximizar o seu

maximize  $v$

# Jogos de soma zero com dois jogadores

Assim, o jogador 1 quer encontrar  $p$  tal que

- $p$  é uma estratégia mista (distribuição de probabilidades)
- sendo que o jogador 2 escolhe uma resposta ótima
  - ▶ aleatoriza entre  $j$  tal que  $\sum_{i=1}^m (-a_{ij'})p_i$  é máximo
  - ▶  $j$  está no suporte sse  $\sum_{i=1}^m a_{ij'}p_i$  é mínimo
- Sendo que o jogo é de soma zero
  - ▶ Basta minimizar o ganho do outro para maximizar o seu

maximize  $v$

sujeito a 
$$\sum_{i=1}^m p_i = 1$$

# Jogos de soma zero com dois jogadores

Assim, o jogador 1 quer encontrar  $p$  tal que

- $p$  é uma estratégia mista (distribuição de probabilidades)
- sendo que o jogador 2 escolhe uma resposta ótima
  - ▶ aleatoriza entre  $j$  tal que  $\sum_{i=1}^m (-a_{ij'})p_i$  é máximo
  - ▶  $j$  está no suporte sse  $\sum_{i=1}^m a_{ij'}p_i$  é mínimo
- Sendo que o jogo é de soma zero
  - ▶ Basta minimizar o ganho do outro para maximizar o seu

maximize  $v$

sujeito a  $\sum_{i=1}^m p_i = 1$

$$\sum_{i=1}^m a_{ij}p_i \geq v \quad \text{para } j = 1, \dots, n$$

# Jogos de soma zero com dois jogadores

Assim, o jogador 1 quer encontrar  $p$  tal que

- $p$  é uma estratégia mista (distribuição de probabilidades)
- sendo que o jogador 2 escolhe uma resposta ótima
  - ▶ aleatoriza entre  $j$  tal que  $\sum_{i=1}^m (-a_{ij'})p_i$  é máximo
  - ▶  $j$  está no suporte sse  $\sum_{i=1}^m a_{ij'}p_i$  é mínimo
- Sendo que o jogo é de soma zero
  - ▶ Basta minimizar o ganho do outro para maximizar o seu

maximize  $v$

sujeito a  $\sum_{i=1}^m p_i = 1$

$$\sum_{i=1}^m a_{ij}p_i \geq v \quad \text{para } j = 1, \dots, n$$

$$p_i \geq 0 \quad \text{para } i = 1, \dots, m$$

## Jogos de soma zero com dois jogadores

E o Jogador 2 quer encontrar  $q$  que

# Jogos de soma zero com dois jogadores

E o Jogador 2 quer encontrar  $q$  que

minimize  $w$

## Jogos de soma zero com dois jogadores

E o Jogador 2 quer encontrar  $q$  que

minimize  $w$

sujeito a 
$$\sum_{j=1}^n q_j = 1$$

## Jogos de soma zero com dois jogadores

E o Jogador 2 quer encontrar  $q$  que

minimize  $w$

sujeito a  $\sum_{j=1}^n q_j = 1$

$$\sum_{j=1}^n a_{ij} q_j \leq w \quad \text{para } i = 1, \dots, m$$

## Jogos de soma zero com dois jogadores

E o Jogador 2 quer encontrar  $q$  que

minimize  $w$

sujeito a  $\sum_{j=1}^n q_j = 1$

$$\sum_{j=1}^n a_{ij} q_j \leq w \quad \text{para } i = 1, \dots, m$$

$$q_j \geq 0 \quad \text{para } j = 1, \dots, n$$

## Jogos de soma zero com dois jogadores

E o Jogador 2 quer encontrar  $q$  que

minimize  $w$

sujeito a  $\sum_{j=1}^n q_j = 1$

$$\sum_{j=1}^n a_{ij} q_j \leq w \quad \text{para } i = 1, \dots, m$$

$$q_j \geq 0 \quad \text{para } j = 1, \dots, n$$

Estes são programas lineares, e um é o dual do outro!

# Forma padrão dos LPs

# Forma padrão dos LPs

Dados:

# Forma padrão dos LPs

Dados:

- $A$ : matriz do  $\mathbb{Q}^{n \times m}$

# Forma padrão dos LPs

Dados:

- $A$ : matriz do  $\mathbb{Q}^{n \times m}$
- $c$ : vetor do  $\mathbb{Q}^n$

# Forma padrão dos LPs

Dados:

- $A$ : matriz do  $\mathbb{Q}^{n \times m}$
- $c$ : vetor do  $\mathbb{Q}^n$
- $b$ : vetor do  $\mathbb{Q}^m$

# Forma padrão dos LPs

Dados:

- $A$ : matriz do  $\mathbb{Q}^{n \times m}$
- $c$ : vetor do  $\mathbb{Q}^n$
- $b$ : vetor do  $\mathbb{Q}^m$

Programa **primal**:

# Forma padrão dos LPs

Dados:

- $A$ : matriz do  $\mathbb{Q}^{n \times m}$
- $c$ : vetor do  $\mathbb{Q}^n$
- $b$ : vetor do  $\mathbb{Q}^m$

Programa **primal**:

$$\text{minimize} \quad \sum_{i=1}^n c_i x_i$$

# Forma padrão dos LPs

Dados:

- $A$ : matriz do  $\mathbb{Q}^{n \times m}$
- $c$ : vetor do  $\mathbb{Q}^n$
- $b$ : vetor do  $\mathbb{Q}^m$

Programa **primal**:

$$\text{minimize} \quad \sum_{i=1}^n c_i x_i$$

$$\text{sujeito a} \quad \sum_{i=1}^n a_{ij} x_i \geq b_j \quad \text{para } j = 1, \dots, m$$

# Forma padrão dos LPs

Dados:

- $A$ : matriz do  $\mathbb{Q}^{n \times m}$
- $c$ : vetor do  $\mathbb{Q}^n$
- $b$ : vetor do  $\mathbb{Q}^m$

Programa **primal**:

$$\text{minimize} \quad \sum_{i=1}^n c_i x_i$$

$$\text{sujeito a} \quad \sum_{i=1}^n a_{ij} x_i \geq b_j \quad \text{para } j = 1, \dots, m$$

$$x_i \geq 0 \quad \text{para } i = 1, \dots, n$$

# Forma padrão dos LPs

Dados:

- $A$ : matriz do  $\mathbb{Q}^{n \times m}$
- $c$ : vetor do  $\mathbb{Q}^n$
- $b$ : vetor do  $\mathbb{Q}^m$

# Forma padrão dos LPs

Dados:

- $A$ : matriz do  $\mathbb{Q}^{n \times m}$
- $c$ : vetor do  $\mathbb{Q}^n$
- $b$ : vetor do  $\mathbb{Q}^m$

Programa **dual**:

# Forma padrão dos LPs

Dados:

- $A$ : matriz do  $\mathbb{Q}^{n \times m}$
- $c$ : vetor do  $\mathbb{Q}^n$
- $b$ : vetor do  $\mathbb{Q}^m$

Programa **dual**:

$$\text{maximize} \quad \sum_{j=1}^m b_j y_j$$

# Forma padrão dos LPs

Dados:

- $A$ : matriz do  $\mathbb{Q}^{n \times m}$
- $c$ : vetor do  $\mathbb{Q}^n$
- $b$ : vetor do  $\mathbb{Q}^m$

Programa **dual**:

$$\text{maximize} \quad \sum_{j=1}^m b_j y_j$$

$$\text{sujeito a} \quad \sum_{j=1}^m a_{ij} y_j \leq c_i \quad \text{para } i = 1, \dots, n$$

# Forma padrão dos LPs

Dados:

- $A$ : matriz do  $\mathbb{Q}^{n \times m}$
- $c$ : vetor do  $\mathbb{Q}^n$
- $b$ : vetor do  $\mathbb{Q}^m$

Programa **dual**:

$$\text{maximize} \quad \sum_{j=1}^m b_j y_j$$

$$\text{sujeito a} \quad \sum_{j=1}^m a_{ij} y_j \leq c_i \quad \text{para } i = 1, \dots, n$$
$$y_j \geq 0 \quad \text{para } j = 1, \dots, m$$

# Primeiro LP em forma padrão

# Primeiro LP em forma padrão

Jogador 1 quer encontrar  $p$  que

maximize  $v$

sujeito a  $\sum_{i=1}^m p_i = 1$

$$\sum_{i=1}^m a_{ij} p_i \geq v \quad \text{para } j = 1, \dots, n$$

$$p_i \geq 0 \quad \text{para } i = 1, \dots, m$$

## Primeiro LP em forma padrão

minimize  $v^- - v^+$

sujeito a  $-\sum_{i=1}^m p_i \geq -1$

$$\sum_{i=1}^m p_i \geq 1$$

$$-v^+ + v^- + \sum_{i=1}^m a_{ij} p_i \geq 0 \quad \text{para } j = 1, \dots, n$$

$$p_i \geq 0 \quad \text{para } i = 1, \dots, m$$

$$v^+ \geq 0$$

$$v^- \geq 0$$

## Segundo LP em forma padrão

## Segundo LP em forma padrão

Jogador 2 quer encontrar  $q$  que

minimize  $w$

sujeito a  $\sum_{j=1}^m q_j = 1$

$\sum_{j=1}^m a_{ij} q_j \leq w$  para  $i = 1, \dots, m$

$q_j \geq 0$  para  $j = 1, \dots, n$

## Segundo LP em forma padrão

maximize  $w^- - w^+$

sujeito a 
$$-\sum_{j=1}^m q_j \geq -1$$

$$\sum_{j=1}^m q_j \geq 1$$

$$-w^+ + w^- + \sum_{j=1}^m a_{ij} q_j \leq 0 \quad \text{para } j = 1, \dots, n$$

$$q_j \geq 0 \quad \text{para } j = 1, \dots, n$$

$$w^+ \geq 0$$

$$w^- \geq 0$$

# Resultados de programação linear

# Resultados de programação linear

Seja  $P$  um programa linear e  $D$  o seu programa dual

# Resultados de programação linear

Seja  $P$  um programa linear e  $D$  o seu programa dual

Seja  $x^*$  solução ótima de  $P$  e seja  $y^*$  solução ótima de  $D$ ,  
então:

# Resultados de programação linear

Seja  $P$  um programa linear e  $D$  o seu programa dual

Seja  $x^*$  solução ótima de  $P$  e seja  $y^*$  solução ótima de  $D$ , então:

- Se  $x_i^* > 0$ , então  $\sum_{j=1}^m a_{ij}y_j^* = c_i$

# Resultados de programação linear

Seja  $P$  um programa linear e  $D$  o seu programa dual

Seja  $x^*$  solução ótima de  $P$  e seja  $y^*$  solução ótima de  $D$ , então:

- Se  $x_i^* > 0$ , então  $\sum_{j=1}^m a_{ij}y_j^* = c_i$
- Se  $y_j^* > 0$ , então  $\sum_{i=1}^n a_{ij}x_i^* = b_j$

# Resultados de programação linear

Seja  $P$  um programa linear e  $D$  o seu programa dual

Seja  $x^*$  solução ótima de  $P$  e seja  $y^*$  solução ótima de  $D$ , então:

- Se  $x_i^* > 0$ , então  $\sum_{j=1}^m a_{ij}y_j^* = c_i$
- Se  $y_j^* > 0$ , então  $\sum_{i=1}^n a_{ij}x_i^* = b_j$
- $\sum_{i=1}^n c_i x_i^* = \sum_{j=1}^m b_j y_j^*$

# Conclusão

Se

# Conclusão

Se

- $(p^*, v^*)$  é solução ótima do programa linear do jogador 1

# Conclusão

Se

- $(p^*, v^*)$  é solução ótima do programa linear do jogador 1
- $(q^*, w^*)$  é solução ótima do programa linear do jogador 2

# Conclusão

Se

- $(p^*, v^*)$  é solução ótima do programa linear do jogador 1
- $(q^*, w^*)$  é solução ótima do programa linear do jogador 2

então:

# Conclusão

Se

- $(p^*, v^*)$  é solução ótima do programa linear do jogador 1
- $(q^*, w^*)$  é solução ótima do programa linear do jogador 2

então:

- Se  $q_j^* > 0$  então  $\sum_{i=1}^m a_{ij} p_i^* = v^*$

# Conclusão

Se

- $(p^*, v^*)$  é solução ótima do programa linear do jogador 1
- $(q^*, w^*)$  é solução ótima do programa linear do jogador 2

então:

- Se  $q_j^* > 0$  então  $\sum_{i=1}^m a_{ij}p_i^* = v^*$ 
  - ▶ Outras estratégias tais que  $\sum_{i=1}^m a_{ij}p_i^* > v^*$  tem  $q_j^* = 0$

# Conclusão

Se

- $(p^*, v^*)$  é solução ótima do programa linear do jogador 1
- $(q^*, w^*)$  é solução ótima do programa linear do jogador 2

então:

- Se  $q_j^* > 0$  então  $\sum_{i=1}^m a_{ij} p_i^* = v^*$ 
  - ▶ Outras estratégias tais que  $\sum_{i=1}^m a_{ij} p_i^* > v^*$  tem  $q_j^* = 0$
- Se  $p_i^* > 0$  então  $\sum_{j=1}^n a_{ij} q_j^* = w^*$

# Conclusão

Se

- $(p^*, v^*)$  é solução ótima do programa linear do jogador 1
- $(q^*, w^*)$  é solução ótima do programa linear do jogador 2

então:

- Se  $q_j^* > 0$  então  $\sum_{i=1}^m a_{ij} p_i^* = v^*$ 
  - ▶ Outras estratégias tais que  $\sum_{i=1}^m a_{ij} p_i^* > v^*$  tem  $q_j^* = 0$
- Se  $p_i^* > 0$  então  $\sum_{j=1}^n a_{ij} q_j^* = w^*$ 
  - ▶ Outras estratégias tais que  $\sum_{j=1}^n a_{ij} q_j^* < w^*$  tem  $p_i^* = 0$

# Conclusão

Se

- $(p^*, v^*)$  é solução ótima do programa linear do jogador 1
- $(q^*, w^*)$  é solução ótima do programa linear do jogador 2

então:

- Se  $q_j^* > 0$  então  $\sum_{i=1}^m a_{ij} p_i^* = v^*$ 
  - ▶ Outras estratégias tais que  $\sum_{i=1}^m a_{ij} p_i^* > v^*$  tem  $q_j^* = 0$
- Se  $p_i^* > 0$  então  $\sum_{j=1}^n a_{ij} q_j^* = w^*$ 
  - ▶ Outras estratégias tais que  $\sum_{j=1}^n a_{ij} q_j^* < w^*$  tem  $p_i^* = 0$
- $v^* = w^*$  (a utilidade do jogador 1 é o custo do jogador 2)

# Conclusão

Se

- $(p^*, v^*)$  é solução ótima do programa linear do jogador 1
- $(q^*, w^*)$  é solução ótima do programa linear do jogador 2

então:

- Se  $q_j^* > 0$  então  $\sum_{i=1}^m a_{ij} p_i^* = v^*$ 
  - ▶ Outras estratégias tais que  $\sum_{i=1}^m a_{ij} p_i^* > v^*$  tem  $q_j^* = 0$
- Se  $p_i^* > 0$  então  $\sum_{j=1}^n a_{ij} q_j^* = w^*$ 
  - ▶ Outras estratégias tais que  $\sum_{j=1}^n a_{ij} q_j^* < w^*$  tem  $p_i^* = 0$
- $v^* = w^*$  (a utilidade do jogador 1 é o custo do jogador 2)

Tal par de soluções é um **equilíbrio misto** já que ambos os jogadores não podem melhorar

# Jogos de soma zero com dois jogadores

**Problema:** Dada uma matriz  $A_{m \times n}$  encontrar um equilíbrio misto

# Jogos de soma zero com dois jogadores

**Problema:** Dada uma matriz  $A_{m \times n}$  encontrar um equilíbrio misto

**Conclusão:** Este problema pode ser resolvido em tempo polinomial (usando programação linear)

# Jogos de soma zero com dois jogadores

**Problema:** Dada uma matriz  $A_{m \times n}$  encontrar um equilíbrio misto

**Conclusão:** Este problema pode ser resolvido em tempo polinomial (usando programação linear)

E para jogos mais gerais?

# Encontrado equilibrios mistos

## Encontrado equilíbrios mistos

O **Teorema de Nash** garante a existência de um equilíbrio em qualquer jogo finito

# Encontrado equilíbrios mistos

O **Teorema de Nash** garante a existência de um equilíbrio em qualquer jogo finito

- Mas como encontrar um tal equilíbrio?

# Encontrado equilíbrios mistos

O **Teorema de Nash** garante a existência de um equilíbrio em qualquer jogo finito

- Mas como encontrar um tal equilíbrio?

**Kamal Jain:**

“If your laptop cannot find it, neither can the market.”

# Encontrado equilíbrios mistos

O **Teorema de Nash** garante a existência de um equilíbrio em qualquer jogo finito

- Mas como encontrar um tal equilíbrio?

**Kamal Jain:**

“If your laptop cannot find it, neither can the market.”

**Problema:** Dado um jogo em forma padrão, encontrar um equilíbrio de Nash

# Encontrado equilíbrios mistos

O **Teorema de Nash** garante a existência de um equilíbrio em qualquer jogo finito

- Mas como encontrar um tal equilíbrio?

**Kamal Jain:**

“If your laptop cannot find it, neither can the market.”

**Problema:** Dado um jogo em forma padrão, encontrar um equilíbrio de Nash

- Podemos resolver esse problema eficientemente?

# Encontrado equilíbrios mistos

O **Teorema de Nash** garante a existência de um equilíbrio em qualquer jogo finito

- Mas como encontrar um tal equilíbrio?

**Kamal Jain:**

“If your laptop cannot find it, neither can the market.”

**Problema:** Dado um jogo em forma padrão, encontrar um equilíbrio de Nash

- Podemos resolver esse problema eficientemente?
- Qual é a sua complexidade?

# Encontrado equilíbrios mistos

O **Teorema de Nash** garante a existência de um equilíbrio em qualquer jogo finito

- Mas como encontrar um tal equilíbrio?

**Kamal Jain:**

“If your laptop cannot find it, neither can the market.”

**Problema:** Dado um jogo em forma padrão, encontrar um equilíbrio de Nash

- Podemos resolver esse problema eficientemente?
- Qual é a sua complexidade?
- A versão de decisão (existe equilíbrio de Nash?) é trivial...

# Discussão

Nash descreveu um jogo de Poker com três jogadores, com utilidades **inteiras**, e único equilíbrio envolvendo números **irracionais**

# Discussão

Nash descreveu um jogo de Poker com três jogadores, com utilidades **inteiras**, e único equilíbrio envolvendo números **irracionais**

Porém, podemos resolver NASH encontrando o **suporte** certo das estratégias mistas de cada jogador

# Discussão

Nash descreveu um jogo de Poker com três jogadores, com utilidades **inteiras**, e único equilíbrio envolvendo números **irracionais**

Porém, podemos resolver NASH encontrando o **suporte** certo das estratégias mistas de cada jogador

Dados os suportes, é possível utilizar um **sistema de equações polinomiais** para encontrar o equilíbrio

# A complexidade de encontrar um equilíbrio

O problema de encontrar um equilíbrio misto é

**PPAD**-Completo:

# A complexidade de encontrar um equilíbrio

O problema de encontrar um equilíbrio misto é

**PPAD**-Completo:

- A classe **PPAD** é composta por problemas que a existência de uma solução é **garantida**

# A complexidade de encontrar um equilíbrio

O problema de encontrar um equilíbrio misto é

**PPAD**-Completo:

- A classe **PPAD** é composta por problemas que a existência de uma solução é **garantida**
  - ▶ Porém o espaço de busca é **exponencial** (apesar de bem estruturado)

# A complexidade de encontrar um equilíbrio

O problema de encontrar um equilíbrio misto é

**PPAD**-Completo:

- A classe **PPAD** é composta por problemas que a existência de uma solução é **garantida**
  - ▶ Porém o espaço de busca é **exponencial** (apesar de bem estruturado)
- A classe **PPAD** é um subconjunto da classe **NP**

# A complexidade de encontrar um equilíbrio

O problema de encontrar um equilíbrio misto é

**PPAD**-Completo:

- A classe **PPAD** é composta por problemas que a existência de uma solução é **garantida**
  - ▶ Porém o espaço de busca é **exponencial** (apesar de bem estruturado)
- A classe **PPAD** é um subconjunto da classe **NP**
  - ▶ Ou seja, se  $P = NP$  então  $PPAD = P$

# A complexidade de encontrar um equilíbrio

O problema de encontrar um equilíbrio misto é

**PPAD**-Completo:

- A classe **PPAD** é composta por problemas que a existência de uma solução é **garantida**
  - ▶ Porém o espaço de busca é **exponencial** (apesar de bem estruturado)
- A classe **PPAD** é um subconjunto da classe **NP**
  - ▶ Ou seja, se  $P = NP$  então  $PPAD = P$
  - ▶ Porém, pode ser que  $PPAD = P$  e  $P \neq NP$

# A complexidade de encontrar um equilíbrio

O problema de encontrar um equilíbrio misto é

**PPAD**-Completo:

- A classe **PPAD** é composta por problemas que a existência de uma solução é **garantida**
  - ▶ Porém o espaço de busca é **exponencial** (apesar de bem estruturado)
- A classe **PPAD** é um subconjunto da classe **NP**
  - ▶ Ou seja, se  $P = NP$  então  $PPAD = P$
  - ▶ Porém, pode ser que  $PPAD = P$  e  $P \neq NP$
- Apesar de ser um conceito “**mais fraco**”, vários problemas interessantes estão em **PPAD**

# A complexidade de encontrar um equilíbrio

O problema de encontrar um equilíbrio misto é

**PPAD**-Completo:

- A classe **PPAD** é composta por problemas que a existência de uma solução é **garantida**
  - ▶ Porém o espaço de busca é **exponencial** (apesar de bem estruturado)
- A classe **PPAD** é um subconjunto da classe **NP**
  - ▶ Ou seja, se  $P = NP$  então  $PPAD = P$
  - ▶ Porém, pode ser que  $PPAD = P$  e  $P \neq NP$
- Apesar de ser um conceito “**mais fraco**”, vários problemas interessantes estão em **PPAD**
  - ▶ Encontrar ponto fixo de Brouwer,

# A complexidade de encontrar um equilíbrio

O problema de encontrar um equilíbrio misto é

**PPAD**-Completo:

- A classe **PPAD** é composta por problemas que a existência de uma solução é **garantida**
  - ▶ Porém o espaço de busca é **exponencial** (apesar de bem estruturado)
- A classe **PPAD** é um subconjunto da classe **NP**
  - ▶ Ou seja, se  $P = NP$  então  $PPAD = P$
  - ▶ Porém, pode ser que  $PPAD = P$  e  $P \neq NP$
- Apesar de ser um conceito “**mais fraco**”, vários problemas interessantes estão em **PPAD**
  - ▶ Encontrar ponto fixo de Brouwer,
  - ▶ Achar equilíbrio de Arrow-Debreu em mercados, etc...

Os seguintes problemas são NP-completos

## Os seguintes problemas são NP-completos

Dado um jogo de duas pessoas na forma matricial, decidir se este jogo tem:

## Os seguintes problemas são NP-completos

Dado um jogo de duas pessoas na forma matricial, decidir se este jogo tem:

- pelo menos **dois** equilíbrios de Nash

## Os seguintes problemas são NP-completos

Dado um jogo de duas pessoas na forma matricial, decidir se este jogo tem:

- pelo menos **dois** equilíbrios de Nash
- dado  $k$ , um equilíbrio de Nash para o jogador **1** com utilidade pelo menos  $k$

## Os seguintes problemas são NP-completos

Dado um jogo de duas pessoas na forma matricial, decidir se este jogo tem:

- pelo menos **dois** equilíbrios de Nash
- dado  $k$ , um equilíbrio de Nash para o jogador **1** com utilidade pelo menos  $k$
- dado  $k$ , um equilíbrio de Nash onde a soma das utilidades dos jogadores é pelo menos  $k$

## Os seguintes problemas são NP-completos

Dado um jogo de duas pessoas na forma matricial, decidir se este jogo tem:

- pelo menos **dois** equilíbrios de Nash
- dado  $k$ , um equilíbrio de Nash para o jogador **1** com utilidade pelo menos  $k$
- dado  $k$ , um equilíbrio de Nash onde a soma das utilidades dos jogadores é pelo menos  $k$
- dado  $k$ , um equilíbrio de Nash com pelo menos  $k$  estratégias no seu suporte

## Os seguintes problemas são NP-completos

Dado um jogo de duas pessoas na forma matricial, decidir se este jogo tem:

- pelo menos **dois** equilíbrios de Nash
- dado  $k$ , um equilíbrio de Nash para o jogador **1** com utilidade pelo menos  $k$
- dado  $k$ , um equilíbrio de Nash onde a soma das utilidades dos jogadores é pelo menos  $k$
- dado  $k$ , um equilíbrio de Nash com pelo menos  $k$  estratégias no seu suporte
- dado  $s$ , um equilíbrio de Nash com  $s$  no suporte

# Os seguintes problemas são NP-completos

Dado um jogo de duas pessoas na forma matricial, decidir se este jogo tem:

- pelo menos **dois** equilíbrios de Nash
- dado  $k$ , um equilíbrio de Nash para o jogador **1** com utilidade pelo menos  $k$
- dado  $k$ , um equilíbrio de Nash onde a soma das utilidades dos jogadores é pelo menos  $k$
- dado  $k$ , um equilíbrio de Nash com pelo menos  $k$  estratégias no seu suporte
- dado  $s$ , um equilíbrio de Nash com  $s$  no suporte
- dado  $s$ , um equilíbrio de Nash sem  $s$  no suporte