

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

# Compartilhamento de Custos

# Jogos Cooperativos

Vimos vários jogos não-cooperativos:

# Jogos Cooperativos

Vimos vários jogos não-cooperativos:

- Ou os jogadores estão competindo entre eles

# Jogos Cooperativos

Vimos vários jogos não-cooperativos:

- Ou os jogadores estão competindo entre eles
  - ▶ Ex: leilão, balanceamento de carga, etc...

# Jogos Cooperativos

Vimos vários jogos não-cooperativos:

- Ou os jogadores estão competindo entre eles
  - ▶ Ex: leilão, balanceamento de carga, etc...
- Ou estão agindo de maneira egoísta, sem se preocupar com os outros

# Jogos Cooperativos

Vimos vários jogos não-cooperativos:

- Ou os jogadores estão competindo entre eles
  - ▶ Ex: leilão, balanceamento de carga, etc...
- Ou estão agindo de maneira egoísta, sem se preocupar com os outros
  - ▶ Ex: formação de redes

# Jogos Cooperativos

Vimos vários jogos não-cooperativos:

- Ou os jogadores estão competindo entre eles
  - ▶ Ex: leilão, balanceamento de carga, etc...
- Ou estão agindo de maneira egoísta, sem se preocupar com os outros
  - ▶ Ex: formação de redes

Veremos agora jogos cooperativos



# Jogos Cooperativos

Vimos vários jogos não-cooperativos:

- Ou os jogadores estão competindo entre eles
  - ▶ Ex: leilão, balanceamento de carga, etc...
- Ou estão agindo de maneira egoísta, sem se preocupar com os outros
  - ▶ Ex: formação de redes

Veremos agora jogos cooperativos

- Jogadores colaboram para realizar uma tarefa

# Jogos Cooperativos

Vimos vários jogos não-cooperativos:

- Ou os jogadores estão competindo entre eles
  - ▶ Ex: leilão, balanceamento de carga, etc...
- Ou estão agindo de maneira egoísta, sem se preocupar com os outros
  - ▶ Ex: formação de redes

Veremos agora jogos cooperativos

- Jogadores colaboram para realizar uma tarefa
- Mas ainda consideram seus interesses pessoais

# Jogos Cooperativos

Vimos vários jogos não-cooperativos:

- Ou os jogadores estão competindo entre eles
  - ▶ Ex: leilão, balanceamento de carga, etc...
- Ou estão agindo de maneira egoísta, sem se preocupar com os outros
  - ▶ Ex: formação de redes

Veremos agora jogos cooperativos

- Jogadores colaboram para realizar uma tarefa
- Mas ainda consideram seus interesses pessoais

Já vimos dois exemplos:

# Jogos Cooperativos

Vimos vários jogos não-cooperativos:

- Ou os jogadores estão competindo entre eles
  - ▶ Ex: leilão, balanceamento de carga, etc...
- Ou estão agindo de maneira egoísta, sem se preocupar com os outros
  - ▶ Ex: formação de redes

Veremos agora jogos cooperativos

- Jogadores colaboram para realizar uma tarefa
- Mas ainda consideram seus interesses pessoais

Já vimos dois exemplos:

- Alocação de casas

# Jogos Cooperativos

Vimos vários jogos não-cooperativos:

- Ou os jogadores estão competindo entre eles
  - ▶ Ex: leilão, balanceamento de carga, etc...
- Ou estão agindo de maneira egoísta, sem se preocupar com os outros
  - ▶ Ex: formação de redes

Veremos agora jogos cooperativos

- Jogadores colaboram para realizar uma tarefa
- Mas ainda consideram seus interesses pessoais

Já vimos dois exemplos:

- Alocação de casas
- Emparelhamento estável

# Jogos Cooperativos

Vimos vários jogos não-cooperativos:

- Ou os jogadores estão competindo entre eles
  - ▶ Ex: leilão, balanceamento de carga, etc...
- Ou estão agindo de maneira egoísta, sem se preocupar com os outros
  - ▶ Ex: formação de redes

Veremos agora jogos cooperativos

- Jogadores colaboram para realizar uma tarefa
- Mas ainda consideram seus interesses pessoais

Já vimos dois exemplos:

- Alocação de casas
- Emparelhamento estável

Como manter a cooperação entre os jogadores?

# Interlúdio - Localização de Instalações

No Problema da Localização de Instalações temos:

# Interlúdio - Localização de Instalações

No Problema da Localização de Instalações temos:

- um conjunto  $\mathcal{F}$  de instalações,



## Interlúdio - Localização de Instalações

No **Problema da Localização de Instalações** temos:

- um conjunto  $\mathcal{F}$  de instalações,
- um custo de abertura  $f_i$  para todo  $i \in \mathcal{F}$ ,

# Interlúdio - Localização de Instalações

No **Problema da Localização de Instalações** temos:

- um conjunto  $\mathcal{F}$  de instalações,
- um custo de abertura  $f_i$  para todo  $i \in \mathcal{F}$ ,
- um conjunto  $\mathcal{C}$  de cidades,

# Interlúdio - Localização de Instalações

No **Problema da Localização de Instalações** temos:

- um conjunto  $\mathcal{F}$  de instalações,
- um custo de abertura  $f_i$  para todo  $i \in \mathcal{F}$ ,
- um conjunto  $\mathcal{C}$  de cidades,
- e distâncias  $d_{rs}$  para todo  $r, s \in \mathcal{F} \cup \mathcal{C}$

# Interlúdio - Localização de Instalações

No **Problema da Localização de Instalações** temos:

- um conjunto  $\mathcal{F}$  de instalações,
- um custo de abertura  $f_i$  para todo  $i \in \mathcal{F}$ ,
- um conjunto  $\mathcal{C}$  de cidades,
- e distâncias  $d_{rs}$  para todo  $r, s \in \mathcal{F} \cup \mathcal{C}$ 
  - ▶ satisfazendo  $d_{rs} \leq d_{rt} + d_{ts}$  para todo  $t \in \mathcal{F} \cup \mathcal{C}$

# Interlúdio - Localização de Instalações

No **Problema da Localização de Instalações** temos:

- um conjunto  $\mathcal{F}$  de instalações,
- um custo de abertura  $f_i$  para todo  $i \in \mathcal{F}$ ,
- um conjunto  $\mathcal{C}$  de cidades,
- e distâncias  $d_{rs}$  para todo  $r, s \in \mathcal{F} \cup \mathcal{C}$ 
  - ▶ satisfazendo  $d_{rs} \leq d_{rt} + d_{ts}$  para todo  $t \in \mathcal{F} \cup \mathcal{C}$
  - ▶ “desigualdade triangular”

# Interlúdio - Localização de Instalações

No **Problema da Localização de Instalações** temos:

- um conjunto  $\mathcal{F}$  de instalações,
- um custo de abertura  $f_i$  para todo  $i \in \mathcal{F}$ ,
- um conjunto  $\mathcal{C}$  de cidades,
- e distâncias  $d_{rs}$  para todo  $r, s \in \mathcal{F} \cup \mathcal{C}$ 
  - ▶ satisfazendo  $d_{rs} \leq d_{rt} + d_{ts}$  para todo  $t \in \mathcal{F} \cup \mathcal{C}$
  - ▶ “desigualdade triangular”

Queremos abrir um conjunto  $\mathcal{F}' \subseteq \mathcal{F}$  de instalações para atender todas as cidades, minimizando

# Interlúdio - Localização de Instalações

No **Problema da Localização de Instalações** temos:

- um conjunto  $\mathcal{F}$  de instalações,
- um custo de abertura  $f_i$  para todo  $i \in \mathcal{F}$ ,
- um conjunto  $\mathcal{C}$  de cidades,
- e distâncias  $d_{rs}$  para todo  $r, s \in \mathcal{F} \cup \mathcal{C}$ 
  - ▶ satisfazendo  $d_{rs} \leq d_{rt} + d_{ts}$  para todo  $t \in \mathcal{F} \cup \mathcal{C}$
  - ▶ “desigualdade triangular”

Queremos abrir um conjunto  $\mathcal{F}' \subseteq \mathcal{F}$  de instalações para atender todas as cidades, minimizando

- o custo de abertura  $\sum_{i \in \mathcal{F}'} f_i$

# Interlúdio - Localização de Instalações

No **Problema da Localização de Instalações** temos:

- um conjunto  $\mathcal{F}$  de instalações,
- um custo de abertura  $f_i$  para todo  $i \in \mathcal{F}$ ,
- um conjunto  $\mathcal{C}$  de cidades,
- e distâncias  $d_{rs}$  para todo  $r, s \in \mathcal{F} \cup \mathcal{C}$ 
  - ▶ satisfazendo  $d_{rs} \leq d_{rt} + d_{ts}$  para todo  $t \in \mathcal{F} \cup \mathcal{C}$
  - ▶ “desigualdade triangular”

Queremos abrir um conjunto  $\mathcal{F}' \subseteq \mathcal{F}$  de instalações para atender todas as cidades, minimizando

- o custo de abertura  $\sum_{i \in \mathcal{F}'} f_i$
- mais o custo de conexão  $\sum_{j \in \mathcal{C}} \min_{i \in \mathcal{F}'} d_{ij}$



# Interlúdio - Localização de Instalações

No **Problema da Localização de Instalações** temos:

- um conjunto  $\mathcal{F}$  de instalações,
- um custo de abertura  $f_i$  para todo  $i \in \mathcal{F}$ ,
- um conjunto  $\mathcal{C}$  de cidades,
- e distâncias  $d_{rs}$  para todo  $r, s \in \mathcal{F} \cup \mathcal{C}$ 
  - ▶ satisfazendo  $d_{rs} \leq d_{rt} + d_{ts}$  para todo  $t \in \mathcal{F} \cup \mathcal{C}$
  - ▶ “desigualdade triangular”

Queremos abrir um conjunto  $\mathcal{F}' \subseteq \mathcal{F}$  de instalações para atender todas as cidades, minimizando

- o custo de abertura  $\sum_{i \in \mathcal{F}'} f_i$
- mais o custo de conexão  $\sum_{j \in \mathcal{C}} \min_{i \in \mathcal{F}'} d_{ij}$ 
  - ▶ cada cidade se conecta a instalação aberta mais próxima

# Interlúdio - Localização de Instalações

No **Problema da Localização de Instalações** temos:

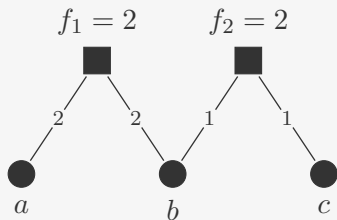
- um conjunto  $\mathcal{F}$  de instalações,
- um custo de abertura  $f_i$  para todo  $i \in \mathcal{F}$ ,
- um conjunto  $\mathcal{C}$  de cidades,
- e distâncias  $d_{rs}$  para todo  $r, s \in \mathcal{F} \cup \mathcal{C}$ 
  - ▶ satisfazendo  $d_{rs} \leq d_{rt} + d_{ts}$  para todo  $t \in \mathcal{F} \cup \mathcal{C}$
  - ▶ “desigualdade triangular”

Queremos abrir um conjunto  $\mathcal{F}' \subseteq \mathcal{F}$  de instalações para atender todas as cidades, minimizando

- o custo de abertura  $\sum_{i \in \mathcal{F}'} f_i$
- mais o custo de conexão  $\sum_{j \in \mathcal{C}} \min_{i \in \mathcal{F}'} d_{ij}$ 
  - ▶ cada cidade se conecta a instalação aberta mais próxima

Esse problema é **NP-difícil**

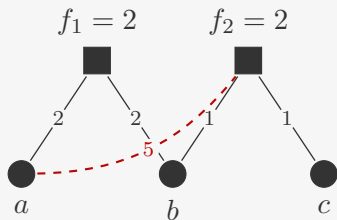
# Exemplo<sup>1</sup>



---

<sup>1</sup>N. Nisan, T. Roughgarden, E. Tardos, V. V. Vazirani, editors. Algorithmic Game Theory. Cambridge University Press, 2007.

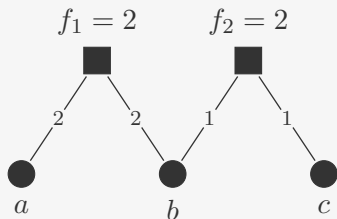
# Exemplo<sup>1</sup>



---

<sup>1</sup>N. Nisan, T. Roughgarden, E. Tardos, V. V. Vazirani, editors. Algorithmic Game Theory. Cambridge University Press, 2007.

# Exemplo<sup>1</sup>



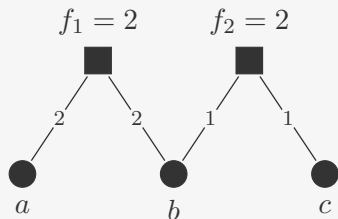
Se abrirmos apenas a instalação 1:

- Custo de abertura: 2
- Custo de conexão:  $2 + 2 + 4 = 8$
- Total: 10

---

<sup>1</sup>N. Nisan, T. Roughgarden, E. Tardos, V. V. Vazirani, editors. Algorithmic Game Theory. Cambridge University Press, 2007.

# Exemplo<sup>1</sup>



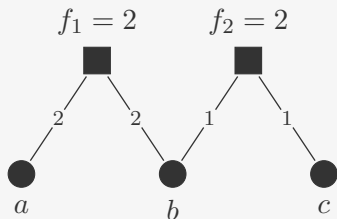
Se abrirmos apenas a instalação 2:

- Custo de abertura: 2
- Custo de conexão:  $5 + 1 + 1 = 7$
- Total: 9

---

<sup>1</sup>N. Nisan, T. Roughgarden, E. Tardos, V. V. Vazirani, editors. Algorithmic Game Theory. Cambridge University Press, 2007.

# Exemplo<sup>1</sup>



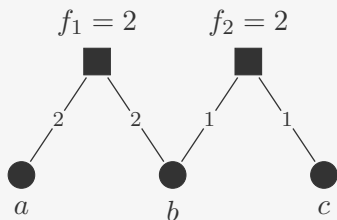
Se abrirmos ambas as instalações:

- Custo de abertura: 4
- Custo de conexão:  $2 + 1 + 1 = 4$
- Total: 8

---

<sup>1</sup>N. Nisan, T. Roughgarden, E. Tardos, V. V. Vazirani, editors. Algorithmic Game Theory. Cambridge University Press, 2007.

# Exemplo<sup>1</sup>



Se abrirmos ambas as instalações:

- Custo de abertura: 4
- Custo de conexão:  $2 + 1 + 1 = 4$
- Total: 8

Como **dividir o custo** entre as cidades de maneira “**justa**”?

---

<sup>1</sup>N. Nisan, T. Roughgarden, E. Tardos, V. V. Vazirani, editors. Algorithmic Game Theory. Cambridge University Press, 2007.



# Jogos cooperativos com utilidades transferíveis

Focaremos em um classe particular de jogos cooperativos:

# Jogos cooperativos com utilidades transferíveis

Focaremos em um classe particular de jogos cooperativos:

- Os jogadores ganham um valor ou tem um custo monetário

# Jogos cooperativos com utilidades transferíveis

Focaremos em um classe particular de jogos cooperativos:

- Os jogadores ganham um valor ou tem um custo monetário
- E portanto pode pagar ou receber dinheiro de outros jogadores

# Jogos cooperativos com utilidades transferíveis

Focaremos em um classe particular de jogos cooperativos:

- Os jogadores ganham um valor ou tem um custo monetário
- E portanto pode pagar ou receber dinheiro de outros jogadores
  - ▶ para incentivar a colaboração

# Jogos cooperativos com utilidades transferíveis

Focaremos em um classe particular de jogos cooperativos:

- Os jogadores ganham um valor ou tem um custo monetário
- E portanto pode pagar ou receber dinheiro de outros jogadores
  - ▶ para incentivar a colaboração
- A utilidade ou custo que eles obtém do jogo é transferível

# Jogos cooperativos com utilidades transferíveis

Focaremos em um classe particular de jogos cooperativos:

- Os jogadores ganham um valor ou tem um custo monetário
- E portanto pode pagar ou receber dinheiro de outros jogadores
  - ▶ para incentivar a colaboração
- A utilidade ou custo que eles obtém do jogo é transferível

Existem também jogos cooperativos com utilidades não-transferíveis

# Jogos cooperativos com utilidades transferíveis

Focaremos em um classe particular de jogos cooperativos:

- Os jogadores ganham um valor ou tem um custo monetário
- E portanto pode pagar ou receber dinheiro de outros jogadores
  - ▶ para incentivar a colaboração
- A utilidade ou custo que eles obtém do jogo é transferível

Existem também jogos cooperativos com utilidades não-transferíveis

- Mas não veremos no curso

# Jogos cooperativos com utilidades transferíveis

Em um jogo cooperativo com utilidades transferíveis temos:



# Jogos cooperativos com utilidades transferíveis

Em um jogo cooperativo com utilidades transferíveis temos:

- Um conjunto  $\mathcal{A}$  de  $n$  jogadores

# Jogos cooperativos com utilidades transferíveis

Em um jogo cooperativo com utilidades transferíveis temos:

- Um conjunto  $\mathcal{A}$  de  $n$  jogadores
  - ▶  $S \subseteq \mathcal{A}$  é chamada de uma **coalizão**

# Jogos cooperativos com utilidades transferíveis

Em um jogo cooperativo com utilidades transferíveis temos:

- Um conjunto  $\mathcal{A}$  de  $n$  jogadores
  - ▶  $S \subseteq \mathcal{A}$  é chamada de uma **coalizão**
  - ▶  $\mathcal{A}$  é chamada de **grande coalizão**

# Jogos cooperativos com utilidades transferíveis

Em um jogo cooperativo com utilidades transferíveis temos:

- Um conjunto  $\mathcal{A}$  de  $n$  jogadores
  - ▶  $S \subseteq \mathcal{A}$  é chamada de uma **coalizão**
  - ▶  $\mathcal{A}$  é chamada de **grande coalizão**
- Um função  $c : 2^{\mathcal{A}} \rightarrow \mathbb{R}$  com  $c(\emptyset) = 0$

# Jogos cooperativos com utilidades transferíveis

Em um jogo cooperativo com utilidades transferíveis temos:

- Um conjunto  $\mathcal{A}$  de  $n$  jogadores
  - ▶  $S \subseteq \mathcal{A}$  é chamada de uma **coalizão**
  - ▶  $\mathcal{A}$  é chamada de **grande coalizão**
- Um função  $c : 2^{\mathcal{A}} \rightarrow \mathbb{R}$  com  $c(\emptyset) = 0$ 
  - ▶  $c(S) \in \mathbb{R}$  é o custo gerado pela coalizão  $S$

# Jogos cooperativos com utilidades transferíveis

Em um jogo cooperativo com utilidades transferíveis temos:

- Um conjunto  $\mathcal{A}$  de  $n$  jogadores
  - ▶  $S \subseteq \mathcal{A}$  é chamada de uma **coalizão**
  - ▶  $\mathcal{A}$  é chamada de **grande coalizão**
- Um função  $c : 2^{\mathcal{A}} \rightarrow \mathbb{R}$  com  $c(\emptyset) = 0$ 
  - ▶  $c(S) \in \mathbb{R}$  é o custo gerado pela coalizão  $S$

O jogo não leva em consideração estratégias ou ações

# Jogos cooperativos com utilidades transferíveis

Em um jogo cooperativo com utilidades transferíveis temos:

- Um conjunto  $\mathcal{A}$  de  $n$  jogadores
  - ▶  $S \subseteq \mathcal{A}$  é chamada de uma **coalizão**
  - ▶  $\mathcal{A}$  é chamada de **grande coalizão**
- Um função  $c : 2^{\mathcal{A}} \rightarrow \mathbb{R}$  com  $c(\emptyset) = 0$ 
  - ▶  $c(S) \in \mathbb{R}$  é o custo gerado pela coalizão  $S$

O jogo não leva em consideração estratégias ou ações

- Considera apenas o custo que uma coalizão tem

# Jogos cooperativos com utilidades transferíveis

Em um jogo cooperativo com utilidades transferíveis temos:

- Um conjunto  $\mathcal{A}$  de  $n$  jogadores
  - ▶  $S \subseteq \mathcal{A}$  é chamada de uma **coalizão**
  - ▶  $\mathcal{A}$  é chamada de **grande coalizão**
- Um função  $c : 2^{\mathcal{A}} \rightarrow \mathbb{R}$  com  $c(\emptyset) = 0$ 
  - ▶  $c(S) \in \mathbb{R}$  é o custo gerado pela coalizão  $S$

O jogo não leva em consideração estratégias ou ações

- Considera apenas o custo que uma coalizão tem
- Não como esse custo é obtido pela coalizão



# Jogo da Localização de Instalações

No Jogo da Localização de Instalações temos:

# Jogo da Localização de Instalações

No Jogo da Localização de Instalações temos:

- um conjunto  $\mathcal{F}$  de instalações,

# Jogo da Localização de Instalações

No Jogo da Localização de Instalações temos:

- um conjunto  $\mathcal{F}$  de instalações,
- um custo de abertura  $f_i$  para todo  $i \in \mathcal{F}$ ,

# Jogo da Localização de Instalações

No Jogo da Localização de Instalações temos:

- um conjunto  $\mathcal{F}$  de instalações,
- um custo de abertura  $f_i$  para todo  $i \in \mathcal{F}$ ,
- um conjunto  $\mathcal{A}$  de cidades,

# Jogo da Localização de Instalações

No Jogo da Localização de Instalações temos:

- um conjunto  $\mathcal{F}$  de instalações,
- um custo de abertura  $f_i$  para todo  $i \in \mathcal{F}$ ,
- um conjunto  $\mathcal{A}$  de cidades,
- e distâncias  $d_{rs}$  para todo  $r, s \in \mathcal{F} \cup \mathcal{A}$

# Jogo da Localização de Instalações

No Jogo da Localização de Instalações temos:

- um conjunto  $\mathcal{F}$  de instalações,
- um custo de abertura  $f_i$  para todo  $i \in \mathcal{F}$ ,
- um conjunto  $\mathcal{A}$  de cidades,
- e distâncias  $d_{rs}$  para todo  $r, s \in \mathcal{F} \cup \mathcal{A}$ 
  - ▶ satisfazendo  $d_{rs} \leq d_{rt} + d_{ts}$  para todo  $t \in \mathcal{F} \cup \mathcal{A}$

# Jogo da Localização de Instalações

No Jogo da Localização de Instalações temos:

- um conjunto  $\mathcal{F}$  de instalações,
- um custo de abertura  $f_i$  para todo  $i \in \mathcal{F}$ ,
- um conjunto  $\mathcal{A}$  de cidades,
- e distâncias  $d_{rs}$  para todo  $r, s \in \mathcal{F} \cup \mathcal{A}$ 
  - ▶ satisfazendo  $d_{rs} \leq d_{rt} + d_{ts}$  para todo  $t \in \mathcal{F} \cup \mathcal{A}$

O custo de uma coalizão  $S \subseteq \mathcal{A}$  é

# Jogo da Localização de Instalações

No Jogo da Localização de Instalações temos:

- um conjunto  $\mathcal{F}$  de instalações,
- um custo de abertura  $f_i$  para todo  $i \in \mathcal{F}$ ,
- um conjunto  $\mathcal{A}$  de cidades,
- e distâncias  $d_{rs}$  para todo  $r, s \in \mathcal{F} \cup \mathcal{A}$ 
  - ▶ satisfazendo  $d_{rs} \leq d_{rt} + d_{ts}$  para todo  $t \in \mathcal{F} \cup \mathcal{A}$

O custo de uma coalizão  $S \subseteq \mathcal{A}$  é

$$c(S) = \min_{\mathcal{F}' \subseteq \mathcal{F}} \left\{ \sum_{i \in \mathcal{F}'} f_i + \sum_{j \in S} \min_{i \in \mathcal{F}'} d_{ij} \right\}$$



# Jogo da Localização de Instalações

No **Jogo da Localização de Instalações** temos:

- um conjunto  $\mathcal{F}$  de instalações,
- um custo de abertura  $f_i$  para todo  $i \in \mathcal{F}$ ,
- um conjunto  $\mathcal{A}$  de cidades,
- e distâncias  $d_{rs}$  para todo  $r, s \in \mathcal{F} \cup \mathcal{A}$ 
  - ▶ satisfazendo  $d_{rs} \leq d_{rt} + d_{ts}$  para todo  $t \in \mathcal{F} \cup \mathcal{A}$

O custo de uma coalizão  $S \subseteq \mathcal{A}$  é

$$c(S) = \min_{\mathcal{F}' \subseteq \mathcal{F}} \left\{ \sum_{i \in \mathcal{F}'} f_i + \sum_{j \in S} \min_{i \in \mathcal{F}'} d_{ij} \right\}$$

Precisa resolver um problema **NP-difícil** para determinar  $c(S)$

# Jogo da Localização de Instalações

No **Jogo da Localização de Instalações** temos:

- um conjunto  $\mathcal{F}$  de instalações,
- um custo de abertura  $f_i$  para todo  $i \in \mathcal{F}$ ,
- um conjunto  $\mathcal{A}$  de cidades,
- e distâncias  $d_{rs}$  para todo  $r, s \in \mathcal{F} \cup \mathcal{A}$ 
  - ▶ satisfazendo  $d_{rs} \leq d_{rt} + d_{ts}$  para todo  $t \in \mathcal{F} \cup \mathcal{A}$

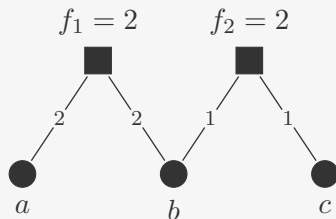
O custo de uma coalizão  $S \subseteq \mathcal{A}$  é

$$c(S) = \min_{\mathcal{F}' \subseteq \mathcal{F}} \left\{ \sum_{i \in \mathcal{F}'} f_i + \sum_{j \in S} \min_{i \in \mathcal{F}'} d_{ij} \right\}$$

Precisa resolver um problema **NP-difícil** para determinar  $c(S)$

- E temos  $2^{|\mathcal{A}|}$  possíveis coalizões...

## Exemplo



Custos das coalizões:

- $c(\{a\}) = 4$ ,  $c(\{b\}) = 3$ ,  $c(\{c\}) = 3$
- $c(\{a, b\}) = 6$ ,  $c(\{b, c\}) = 4$ ,  $c(\{a, c\}) = 7$
- $c(\{a, b, c\}) = 8$

# Núcleo

Queremos dividir o custo  $c(\mathcal{A})$  entre os jogadores

# Núcleo

Queremos dividir o custo  $c(\mathcal{A})$  entre os jogadores

- $\alpha \in \mathbb{R}^{|\mathcal{A}|}$  é uma **alocação de custos**

# Núcleo

Queremos dividir o custo  $c(\mathcal{A})$  entre os jogadores

- $\alpha \in \mathbb{R}^{|\mathcal{A}|}$  é uma **alocação de custos**
  - ▶  $\alpha_j$  é o preço a ser pago pelo jogador  $j \in \mathcal{A}$

# Núcleo

Queremos dividir o custo  $c(\mathcal{A})$  entre os jogadores

- $\alpha \in \mathbb{R}^{|\mathcal{A}|}$  é uma **alocação de custos**
  - ▶  $\alpha_j$  é o preço a ser pago pelo jogador  $j \in \mathcal{A}$
  - ▶  $\alpha$  é uma **precificação**

# Núcleo

Queremos dividir o custo  $c(\mathcal{A})$  entre os jogadores

- $\alpha \in \mathbb{R}^{|\mathcal{A}|}$  é uma **alocação de custos**
  - ▶  $\alpha_j$  é o preço a ser pago pelo jogador  $j \in \mathcal{A}$
  - ▶  $\alpha$  é uma **precificação**

Uma alocação de custos está no **núcleo** se:



# Núcleo

Queremos dividir o custo  $c(\mathcal{A})$  entre os jogadores

- $\alpha \in \mathbb{R}^{|\mathcal{A}|}$  é uma **alocação de custos**
  - ▶  $\alpha_j$  é o preço a ser pago pelo jogador  $j \in \mathcal{A}$
  - ▶  $\alpha$  é uma **precificação**

Uma alocação de custos está no **núcleo** se:

- $\sum_{j \in \mathcal{A}} \alpha_j = c(\mathcal{A})$

# Núcleo

Queremos dividir o custo  $c(\mathcal{A})$  entre os jogadores

- $\alpha \in \mathbb{R}^{|\mathcal{A}|}$  é uma **alocação de custos**
  - ▶  $\alpha_j$  é o preço a ser pago pelo jogador  $j \in \mathcal{A}$
  - ▶  $\alpha$  é uma **precificação**

Uma alocação de custos está no **núcleo** se:

- $\sum_{j \in \mathcal{A}} \alpha_j = c(\mathcal{A})$ 
  - ▶ a soma dos preços é igual ao custo da grande coalizão

# Núcleo

Queremos dividir o custo  $c(\mathcal{A})$  entre os jogadores

- $\alpha \in \mathbb{R}^{|\mathcal{A}|}$  é uma **alocação de custos**
  - ▶  $\alpha_j$  é o preço a ser pago pelo jogador  $j \in \mathcal{A}$
  - ▶  $\alpha$  é uma **precificação**

Uma alocação de custos está no **núcleo** se:

- $\sum_{j \in \mathcal{A}} \alpha_j = c(\mathcal{A})$ 
  - ▶ a soma dos preços é igual ao custo da grande coalizão
  - ▶ a alocação de custos é **orçamento-balanceada**

# Núcleo

Queremos dividir o custo  $c(\mathcal{A})$  entre os jogadores

- $\alpha \in \mathbb{R}^{|\mathcal{A}|}$  é uma **alocação de custos**
  - ▶  $\alpha_j$  é o preço a ser pago pelo jogador  $j \in \mathcal{A}$
  - ▶  $\alpha$  é uma **precificação**

Uma alocação de custos está no **núcleo** se:

- $\sum_{j \in \mathcal{A}} \alpha_j = c(\mathcal{A})$ 
  - ▶ a soma dos preços é igual ao custo da grande coalizão
  - ▶ a alocação de custos é **orçamento-balanceada**
- $\sum_{j \in S} \alpha_j \leq c(S)$  para todo  $S \subseteq \mathcal{A}$

# Núcleo

Queremos dividir o custo  $c(\mathcal{A})$  entre os jogadores

- $\alpha \in \mathbb{R}^{|\mathcal{A}|}$  é uma **alocação de custos**
  - ▶  $\alpha_j$  é o preço a ser pago pelo jogador  $j \in \mathcal{A}$
  - ▶  $\alpha$  é uma **precificação**

Uma alocação de custos está no **núcleo** se:

- $\sum_{j \in \mathcal{A}} \alpha_j = c(\mathcal{A})$ 
  - ▶ a soma dos preços é igual ao custo da grande coalizão
  - ▶ a alocação de custos é **orçamento-balanceada**
- $\sum_{j \in S} \alpha_j \leq c(S)$  para todo  $S \subseteq \mathcal{A}$ 
  - ▶ não existe uma coalizão que deseja desviar

# Núcleo

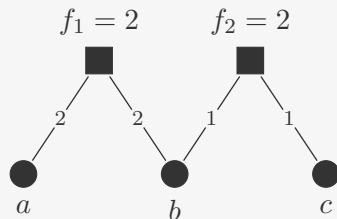
Queremos dividir o custo  $c(\mathcal{A})$  entre os jogadores

- $\alpha \in \mathbb{R}^{|\mathcal{A}|}$  é uma **alocação de custos**
  - ▶  $\alpha_j$  é o preço a ser pago pelo jogador  $j \in \mathcal{A}$
  - ▶  $\alpha$  é uma **precificação**

Uma alocação de custos está no **núcleo** se:

- $\sum_{j \in \mathcal{A}} \alpha_j = c(\mathcal{A})$ 
  - ▶ a soma dos preços é igual ao custo da grande coalizão
  - ▶ a alocação de custos é **orçamento-balanceada**
- $\sum_{j \in S} \alpha_j \leq c(S)$  para todo  $S \subseteq \mathcal{A}$ 
  - ▶ não existe uma coalizão que deseja desviar
  - ▶ a alocação de custos satisfaz a propriedade do núcleo

## Exemplo

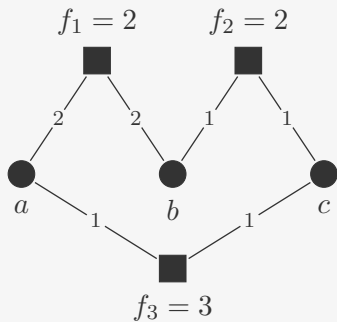


Custos das coalizões:

- $c(\{a\}) = 4$ ,  $c(\{b\}) = 3$ ,  $c(\{c\}) = 3$
- $c(\{a, b\}) = 6$ ,  $c(\{b, c\}) = 4$ ,  $c(\{a, c\}) = 7$
- $c(\{a, b, c\}) = 8$

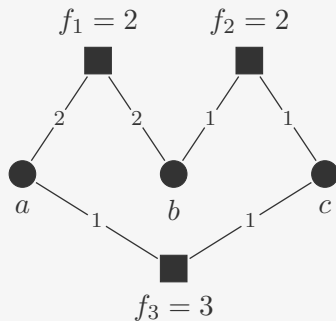
A alocação de custos  $\alpha_a = 4$ ,  $\alpha_b = 2$ ,  $\alpha_c = 2$  está no núcleo

## Exemplo 2



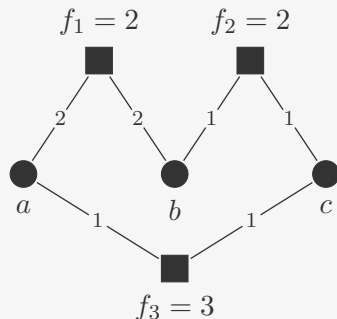


## Exemplo 2



Para qualquer  $\alpha$  no núcleo:

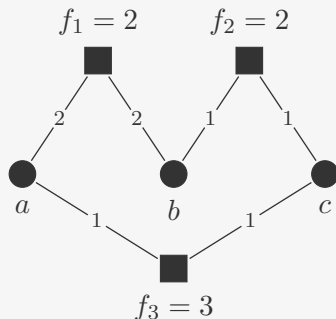
## Exemplo 2



Para qualquer  $\alpha$  no núcleo:

$$\alpha_a + \alpha_b \leq c(\{a, b\}) = 6$$

## Exemplo 2

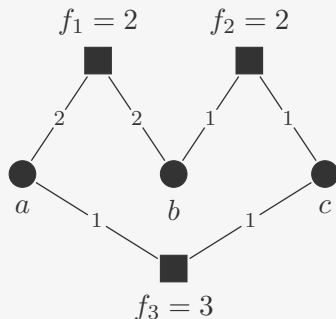


Para qualquer  $\alpha$  no núcleo:

$$\alpha_a + \alpha_b \leq c(\{a, b\}) = 6$$

$$\alpha_b + \alpha_c \leq c(\{b, c\}) = 4$$

## Exemplo 2



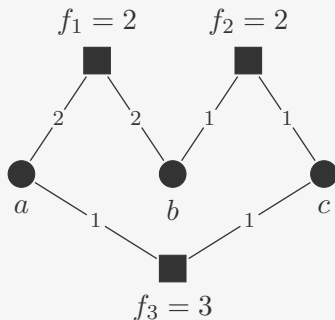
Para qualquer  $\alpha$  no núcleo:

$$\alpha_a + \alpha_b \leq c(\{a, b\}) = 6$$

$$\alpha_b + \alpha_c \leq c(\{b, c\}) = 4$$

$$\alpha_a + \alpha_c \leq c(\{a, c\}) = 5$$

## Exemplo 2



Para qualquer  $\alpha$  no núcleo:

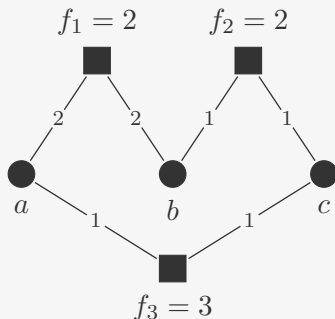
$$\alpha_a + \alpha_b \leq c(\{a, b\}) = 6$$

$$\alpha_b + \alpha_c \leq c(\{b, c\}) = 4$$

$$\alpha_a + \alpha_c \leq c(\{a, c\}) = 5$$

Ou seja,  $\alpha_a + \alpha_b + \alpha_c \leq (6 + 4 + 5)/2$

## Exemplo 2



Para qualquer  $\alpha$  no núcleo:

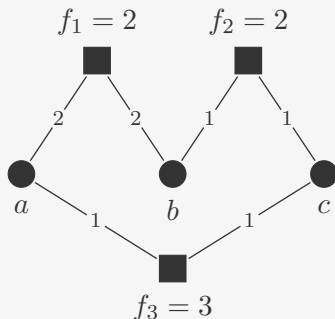
$$\alpha_a + \alpha_b \leq c(\{a, b\}) = 6$$

$$\alpha_b + \alpha_c \leq c(\{b, c\}) = 4$$

$$\alpha_a + \alpha_c \leq c(\{a, c\}) = 5$$

Ou seja,  $\alpha_a + \alpha_b + \alpha_c \leq (6 + 4 + 5)/2 = 7.5$

## Exemplo 2



Para qualquer  $\alpha$  no núcleo:

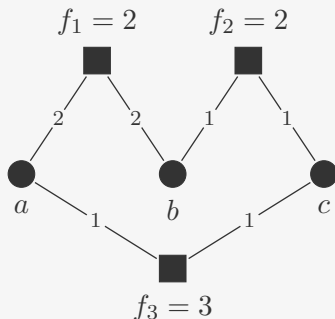
$$\alpha_a + \alpha_b \leq c(\{a, b\}) = 6$$

$$\alpha_b + \alpha_c \leq c(\{b, c\}) = 4$$

$$\alpha_a + \alpha_c \leq c(\{a, c\}) = 5$$

Ou seja,  $\alpha_a + \alpha_b + \alpha_c \leq (6 + 4 + 5)/2 = 7.5 < c(\mathcal{A}) = 8$

## Exemplo 2



Para qualquer  $\alpha$  no núcleo:

$$\alpha_a + \alpha_b \leq c(\{a, b\}) = 6$$

$$\alpha_b + \alpha_c \leq c(\{b, c\}) = 4$$

$$\alpha_a + \alpha_c \leq c(\{a, c\}) = 5$$

Ou seja,  $\alpha_a + \alpha_b + \alpha_c \leq (6 + 4 + 5)/2 = 7.5 < c(\mathcal{A}) = 8$

- O núcleo é vazio!



## O Teorema de Bondareva-Shapley

Um vetor  $\lambda$ , onde  $\lambda_S \geq 0$  para todo  $S \subseteq \mathcal{A}$  é uma **coleção balanceada de pesos** se, para todo  $j \in \mathcal{A}$ ,  $\sum_{S:j \in S} \lambda_S = 1$

## O Teorema de Bondareva-Shapley

Um vetor  $\lambda$ , onde  $\lambda_S \geq 0$  para todo  $S \subseteq \mathcal{A}$  é uma **coleção balanceada de pesos** se, para todo  $j \in \mathcal{A}$ ,  $\sum_{S:j \in S} \lambda_S = 1$

**Teorema:** Um jogo cooperativo com utilidades transferíveis tem um núcleo não-vazio se e somente se, para toda coleção balanceada de pesos  $\lambda$ , temos que  $\sum_{S \subseteq \mathcal{A}} \lambda_S c(S) \geq c(\mathcal{A})$

## O Teorema de Bondareva-Shapley

Um vetor  $\lambda$ , onde  $\lambda_S \geq 0$  para todo  $S \subseteq \mathcal{A}$  é uma **coleção balanceada de pesos** se, para todo  $j \in \mathcal{A}$ ,  $\sum_{S:j \in S} \lambda_S = 1$

**Teorema:** Um jogo cooperativo com utilidades transferíveis tem um núcleo não-vazio se e somente se, para toda coleção balanceada de pesos  $\lambda$ , temos que  $\sum_{S \subseteq \mathcal{A}} \lambda_S c(S) \geq c(\mathcal{A})$

**Prova:** O jogo tem um núcleo não-negativo se e somente se o seguinte programa linear tem valor  $c(\mathcal{A})$

## O Teorema de Bondareva-Shapley

Um vetor  $\lambda$ , onde  $\lambda_S \geq 0$  para todo  $S \subseteq \mathcal{A}$  é uma **coleção balanceada de pesos** se, para todo  $j \in \mathcal{A}$ ,  $\sum_{S:j \in S} \lambda_S = 1$

**Teorema:** Um jogo cooperativo com utilidades transferíveis tem um núcleo não-vazio se e somente se, para toda coleção balanceada de pesos  $\lambda$ , temos que  $\sum_{S \subseteq \mathcal{A}} \lambda_S c(S) \geq c(\mathcal{A})$

**Prova:** O jogo tem um núcleo não-negativo se e somente se o seguinte programa linear tem valor  $c(\mathcal{A})$

$$\max \sum_{j \in \mathcal{A}} \alpha_j$$

$$\text{tal que } \sum_{j \in S} \alpha_j \leq c(S), \quad \forall S \subseteq \mathcal{A}$$

# O Teorema de Bondareva-Shapley - Prova

O dual do programa linear

$$\max \sum_{j \in \mathcal{A}} \alpha_j$$

$$\text{tal que } \sum_{j \in S} \alpha_j \leq c(S), \quad \forall S \subseteq \mathcal{A}$$

# O Teorema de Bondareva-Shapley - Prova

O dual do programa linear

$$\max \sum_{j \in \mathcal{A}} \alpha_j$$

$$\text{tal que } \sum_{j \in S} \alpha_j \leq c(S), \quad \forall S \subseteq \mathcal{A}$$

é

$$\min \sum_{S \subseteq \mathcal{A}} \lambda_S c(S)$$

$$\text{tal que } \sum_{S: j \in S} \lambda_S = 1, \quad \forall j \in \mathcal{A}$$

$$\lambda_S \geq 0, \quad \forall S \subseteq \mathcal{A}$$

# O Teorema de Bondareva-Shapley - Prova

O dual do programa linear

$$\max \sum_{j \in \mathcal{A}} \alpha_j$$

$$\text{tal que } \sum_{j \in S} \alpha_j \leq c(S), \quad \forall S \subseteq \mathcal{A}$$

é

$$\min \sum_{S \subseteq \mathcal{A}} \lambda_S c(S)$$

$$\text{tal que } \sum_{S: j \in S} \lambda_S = 1, \quad \forall j \in \mathcal{A}$$

$$\lambda_S \geq 0, \quad \forall S \subseteq \mathcal{A}$$

Como  $\lambda_{\mathcal{A}} = 1$  e  $\lambda_S = 0$  para todo  $S \subsetneq \mathcal{A}$  é uma solução do dual de valor  $c(\mathcal{A})$

# O Teorema de Bondareva-Shapley - Prova

O dual do programa linear

$$\max \sum_{j \in \mathcal{A}} \alpha_j$$

$$\text{tal que } \sum_{j \in S} \alpha_j \leq c(S), \quad \forall S \subseteq \mathcal{A}$$

é

$$\min \sum_{S \subseteq \mathcal{A}} \lambda_S c(S)$$

$$\text{tal que } \sum_{S: j \in S} \lambda_S = 1, \quad \forall j \in \mathcal{A}$$

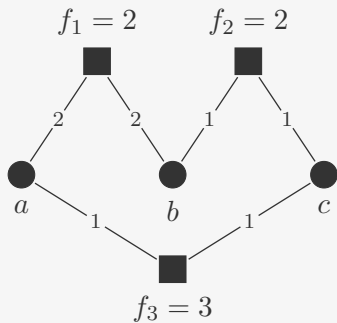
$$\lambda_S \geq 0, \quad \forall S \subseteq \mathcal{A}$$

Como  $\lambda_{\mathcal{A}} = 1$  e  $\lambda_S = 0$  para todo  $S \subsetneq \mathcal{A}$  é uma solução do dual de valor  $c(\mathcal{A})$

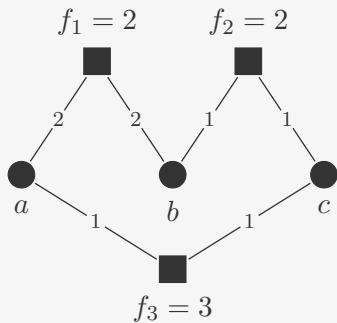
- O valor do primal é igual a  $c(\mathcal{A})$  se e somente, para todo  $\lambda$  dual viável, temos que  $\sum_{S \subseteq \mathcal{A}} \lambda_S c(S) \geq c(\mathcal{A})$



## Exemplo 2

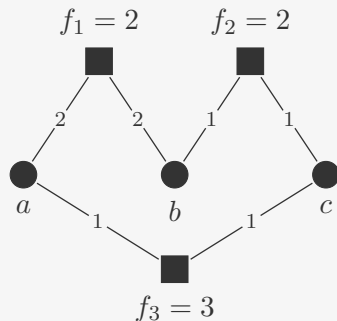


## Exemplo 2



Considere:

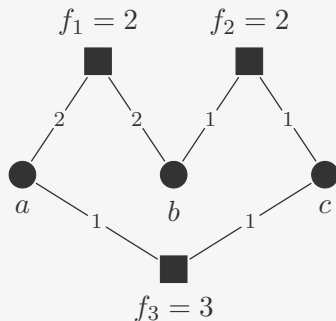
## Exemplo 2



Considere:

- $\lambda_{\{a,b\}} = \lambda_{\{b,c\}} = \lambda_{\{a,c\}} = \frac{1}{2}$

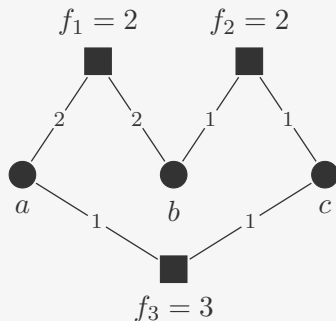
## Exemplo 2



Considere:

- $\lambda_{\{a,b\}} = \lambda_{\{b,c\}} = \lambda_{\{a,c\}} = \frac{1}{2}$
- $\lambda_S = 0$  para as outras coalizões

## Exemplo 2

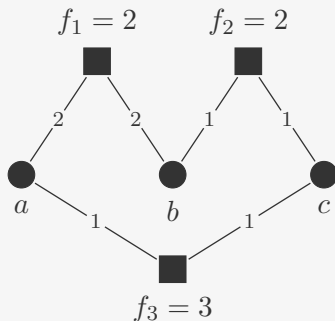


Considere:

- $\lambda_{\{a,b\}} = \lambda_{\{b,c\}} = \lambda_{\{a,c\}} = \frac{1}{2}$
- $\lambda_S = 0$  para as outras coalizões

Lembrando que  $c(\{a, b\}) = 6$ ,  $c(\{b, c\}) = 4$  e  $c(\{a, c\}) = 5$

## Exemplo 2



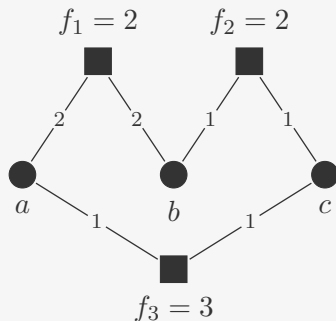
Considere:

- $\lambda_{\{a,b\}} = \lambda_{\{b,c\}} = \lambda_{\{a,c\}} = \frac{1}{2}$
- $\lambda_S = 0$  para as outras coalizões

Lembrando que  $c(\{a, b\}) = 6$ ,  $c(\{b, c\}) = 4$  e  $c(\{a, c\}) = 5$

- $\lambda$  é uma coleção balanceada de pesos e

## Exemplo 2



Considere:

- $\lambda_{\{a,b\}} = \lambda_{\{b,c\}} = \lambda_{\{a,c\}} = \frac{1}{2}$
- $\lambda_S = 0$  para as outras coalizões

Lembrando que  $c(\{a, b\}) = 6$ ,  $c(\{b, c\}) = 4$  e  $c(\{a, c\}) = 5$

- $\lambda$  é uma coleção balanceada de pesos e
- $\sum_{S \subseteq \mathcal{A}} \lambda_S c(S) < c(\mathcal{A})$

## $\gamma$ -Núcleo

Dificuldades em relação ao núcleo:



## $\gamma$ -Núcleo

Dificuldades em relação ao núcleo:

- Ele pode ser vazio... O que fazer então?

## $\gamma$ -Núcleo

Dificuldades em relação ao núcleo:

- Ele pode ser vazio... O que fazer então?
- Calcular  $c(S)$  pode ser **NP**-difícil

## $\gamma$ -Núcleo

Dificuldades em relação ao núcleo:

- Ele pode ser vazio... O que fazer então?
- Calcular  $c(S)$  pode ser NP-difícil
  - ▶ decidir se o núcleo é vazio fica intratável

## $\gamma$ -Núcleo

Dificuldades em relação ao núcleo:

- Ele pode ser vazio... O que fazer então?
- Calcular  $c(S)$  pode ser NP-difícil
  - ▶ decidir se o núcleo é vazio fica intratável

**Solução:** relaxação do conceito de núcleo

## $\gamma$ -Núcleo

Dificuldades em relação ao núcleo:

- Ele pode ser vazio... O que fazer então?
- Calcular  $c(S)$  pode ser NP-difícil
  - ▶ decidir se o núcleo é vazio fica intratável

**Solução:** relaxação do conceito de núcleo

Uma alocação de custos está no  $\gamma$ -núcleo se:

# $\gamma$ -Núcleo

Dificuldades em relação ao núcleo:

- Ele pode ser vazio... O que fazer então?
- Calcular  $c(S)$  pode ser **NP**-difícil
  - ▶ decidir se o núcleo é vazio fica intratável

**Solução:** relaxação do conceito de núcleo

Uma alocação de custos está no  $\gamma$ -núcleo se:

- $\sum_{j \in S} \alpha_j \leq c(S)$  para todo  $S \subseteq \mathcal{A}$

# $\gamma$ -Núcleo

Dificuldades em relação ao núcleo:

- Ele pode ser vazio... O que fazer então?
- Calcular  $c(S)$  pode ser **NP**-difícil
  - ▶ decidir se o núcleo é vazio fica intratável

**Solução:** relaxação do conceito de núcleo

Uma alocação de custos está no  $\gamma$ -núcleo se:

- $\sum_{j \in S} \alpha_j \leq c(S)$  para todo  $S \subseteq \mathcal{A}$ 
  - ▶ a alocação de custos satisfaz a propriedade do núcleo

# $\gamma$ -Núcleo

Dificuldades em relação ao núcleo:

- Ele pode ser vazio... O que fazer então?
- Calcular  $c(S)$  pode ser **NP**-difícil
  - ▶ decidir se o núcleo é vazio fica intratável

**Solução:** relaxação do conceito de núcleo

Uma alocação de custos está no  $\gamma$ -núcleo se:

- $\sum_{j \in S} \alpha_j \leq c(S)$  para todo  $S \subseteq \mathcal{A}$ 
  - ▶ a alocação de custos satisfaz a propriedade do núcleo
- $\gamma c(\mathcal{A}) \leq \sum_{j \in \mathcal{A}} \alpha_j \leq c(\mathcal{A})$



# $\gamma$ -Núcleo

Dificuldades em relação ao núcleo:

- Ele pode ser vazio... O que fazer então?
- Calcular  $c(S)$  pode ser **NP**-difícil
  - ▶ decidir se o núcleo é vazio fica intratável

**Solução:** relaxação do conceito de núcleo

Uma alocação de custos está no  $\gamma$ -núcleo se:

- $\sum_{j \in S} \alpha_j \leq c(S)$  para todo  $S \subseteq \mathcal{A}$ 
  - ▶ a alocação de custos satisfaz a propriedade do núcleo
- $\gamma c(\mathcal{A}) \leq \sum_{j \in \mathcal{A}} \alpha_j \leq c(\mathcal{A})$ 
  - ▶ a alocação de custos é  $\gamma$ -orçamento-balanceada

# $\gamma$ -Núcleo

Dificuldades em relação ao núcleo:

- Ele pode ser vazio... O que fazer então?
- Calcular  $c(S)$  pode ser **NP**-difícil
  - ▶ decidir se o núcleo é vazio fica intratável

**Solução:** relaxação do conceito de núcleo

Uma alocação de custos está no  $\gamma$ -núcleo se:

- $\sum_{j \in S} \alpha_j \leq c(S)$  para todo  $S \subseteq \mathcal{A}$ 
  - ▶ a alocação de custos satisfaz a propriedade do núcleo
- $\gamma c(\mathcal{A}) \leq \sum_{j \in \mathcal{A}} \alpha_j \leq c(\mathcal{A})$ 
  - ▶ a alocação de custos é  $\gamma$ -orçamento-balanceada
  - ▶ o valor pago pelos jogadores paga apenas  $\gamma$  do custo total

# $\gamma$ -Núcleo

Dificuldades em relação ao núcleo:

- Ele pode ser vazio... O que fazer então?
- Calcular  $c(S)$  pode ser **NP**-difícil
  - ▶ decidir se o núcleo é vazio fica intratável

**Solução:** relaxação do conceito de núcleo

Uma alocação de custos está no  $\gamma$ -núcleo se:

- $\sum_{j \in S} \alpha_j \leq c(S)$  para todo  $S \subseteq \mathcal{A}$ 
  - ▶ a alocação de custos satisfaz a propriedade do núcleo
- $\gamma c(\mathcal{A}) \leq \sum_{j \in \mathcal{A}} \alpha_j \leq c(\mathcal{A})$ 
  - ▶ a alocação de custos é  **$\gamma$ -orçamento-balanceada**
  - ▶ o valor pago pelos jogadores paga apenas  $\gamma$  do custo total
    - alguém precisa subsidiar o restante...

## O Teorema de Bondareva-Shapley para $\gamma$ -núcleo

Estamos interessados no maior  $\gamma$  tal que o  $\gamma$ -núcleo é não-vazio

## O Teorema de Bondareva-Shapley para $\gamma$ -núcleo

Estamos interessados no maior  $\gamma$  tal que o  $\gamma$ -núcleo é não-vazio

**Teorema:** Um jogo cooperativo com utilidades transferíveis tem um  $\gamma$ -núcleo não-vazio sse, para toda coleção balanceada de pesos  $\lambda$ , temos que  $\sum_{S \subseteq \mathcal{A}} \lambda_S c(S) \geq \gamma c(\mathcal{A})$

## O Teorema de Bondareva-Shapley para $\gamma$ -núcleo

Estamos interessados no maior  $\gamma$  tal que o  $\gamma$ -núcleo é não-vazio

**Teorema:** Um jogo cooperativo com utilidades transferíveis tem um  $\gamma$ -núcleo não-vazio sse, para toda coleção balanceada de pesos  $\lambda$ , temos que  $\sum_{S \subseteq A} \lambda_S c(S) \geq \gamma c(A)$

A prova é similar a anterior

# Interlúdio - Gap de Integralidade

## Interlúdio - Gap de Integralidade

Se tivermos um programa linear inteiro



## Interlúdio - Gap de Integralidade

Se tivermos um programa linear inteiro

$$\begin{aligned} & \min \sum_{i=1}^n c_i x_i \\ & \text{tal que } \sum_{i=1}^n a_{ij} x_i \leq b_j, \quad \forall j \in \{1, \dots, m\} \\ & \quad \quad \quad x_i \in \mathbb{Z}^+, \quad \forall i \in \{1, \dots, n\} \end{aligned}$$

## Interlúdio - Gap de Integralidade

Se tivermos um programa linear inteiro

$$\min \sum_{i=1}^n c_i x_i$$

$$\text{tal que } \sum_{i=1}^n a_{ij} x_i \leq b_j, \quad \forall j \in \{1, \dots, m\}$$

$$x_i \in \mathbb{Z}^+, \quad \forall i \in \{1, \dots, n\}$$

o **gap de integralidade** é a razão entre

## Interlúdio - Gap de Integralidade

Se tivermos um programa linear inteiro

$$\begin{aligned} \min \quad & \sum_{i=1}^n c_i x_i \\ \text{tal que} \quad & \sum_{i=1}^n a_{ij} x_i \leq b_j, \quad \forall j \in \{1, \dots, m\} \\ & x_i \in \mathbb{Z}^+, \quad \forall i \in \{1, \dots, n\} \end{aligned}$$

o **gap de integralidade** é a razão entre

- o valor de uma solução inteira ótima

# Interlúdio - Gap de Integralidade

Se tivermos um programa linear inteiro

$$\begin{aligned} \min \quad & \sum_{i=1}^n c_i x_i \\ \text{tal que} \quad & \sum_{i=1}^n a_{ij} x_i \leq b_j, \quad \forall j \in \{1, \dots, m\} \\ & x_i \in \mathbb{Z}^+, \quad \forall i \in \{1, \dots, n\} \end{aligned}$$

o **gap de integralidade** é a razão entre

- o valor de uma solução inteira ótima
- o valor de uma solução fracionária ótima

## Interlúdio - Gap de Integralidade

Se tivermos um programa linear inteiro

$$\begin{aligned} \min \quad & \sum_{i=1}^n c_i x_i \\ \text{tal que} \quad & \sum_{i=1}^n a_{ij} x_i \leq b_j, \quad \forall j \in \{1, \dots, m\} \\ & x_i \in \mathbb{Z}^+, \quad \forall i \in \{1, \dots, n\} \end{aligned}$$

o **gap de integralidade** é a razão entre

- o valor de uma solução inteira ótima
- o valor de uma solução fracionária ótima

Ex: Problema da Localização de Instalações

## Interlúdio - Gap de Integralidade

Se tivermos um programa linear inteiro

$$\begin{aligned} \min \quad & \sum_{i=1}^n c_i x_i \\ \text{tal que} \quad & \sum_{i=1}^n a_{ij} x_i \leq b_j, \quad \forall j \in \{1, \dots, m\} \\ & x_i \in \mathbb{Z}^+, \quad \forall i \in \{1, \dots, n\} \end{aligned}$$

o **gap de integralidade** é a razão entre

- o valor de uma solução inteira ótima
- o valor de uma solução fracionária ótima

Ex: Problema da Localização de Instalações

- O gap de integralidade é no máximo **1,463**

## Interlúdio - Gap de Integralidade

Se tivermos um programa linear inteiro

$$\begin{aligned} \min \quad & \sum_{i=1}^n c_i x_i \\ \text{tal que} \quad & \sum_{i=1}^n a_{ij} x_i \leq b_j, \quad \forall j \in \{1, \dots, m\} \\ & x_i \in \mathbb{Z}^+, \quad \forall i \in \{1, \dots, n\} \end{aligned}$$

o **gap de integralidade** é a razão entre

- o valor de uma solução inteira ótima
- o valor de uma solução fracionária ótima

Ex: Problema da Localização de Instalações

- O gap de integralidade é no máximo **1,463**
- i.e., o valor de uma solução ótima inteira é no máximo **1,463** do valor da solução ótima fracionária

# Custos subaditivos

Uma função  $f : 2^U \rightarrow \mathbb{R}$  é subaditiva se



## Custos subaditivos

Uma função  $f : 2^U \rightarrow \mathbb{R}$  é subaditiva se

- $f(S_1 \cup S_2) \leq f(S_1) + f(S_2)$  para todo  $S_1, S_2 \subseteq U$

## Custos subaditivos

Uma função  $f : 2^U \rightarrow \mathbb{R}$  é subaditiva se

- $f(S_1 \cup S_2) \leq f(S_1) + f(S_2)$  para todo  $S_1, S_2 \subseteq U$
- Ex: o custo do jogo de localização de instalações

## Custos subaditivos

Uma função  $f : 2^U \rightarrow \mathbb{R}$  é subaditiva se

- $f(S_1 \cup S_2) \leq f(S_1) + f(S_2)$  para todo  $S_1, S_2 \subseteq U$
- Ex: o custo do jogo de localização de instalações

Se  $c$  é subaditiva, então o valor de uma solução ótima inteira do programa linear

## Custos subaditivos

Uma função  $f : 2^U \rightarrow \mathbb{R}$  é subaditiva se

- $f(S_1 \cup S_2) \leq f(S_1) + f(S_2)$  para todo  $S_1, S_2 \subseteq U$
- Ex: o custo do jogo de localização de instalações

Se  $c$  é subaditiva, então o valor de uma solução ótima inteira do programa linear

$$\begin{aligned} \min \quad & \sum_{S \subseteq \mathcal{A}} \lambda_S c(S) \\ \text{tal que} \quad & \sum_{S: j \in S} \lambda_S = 1, \quad \forall j \in \mathcal{A} \\ & \lambda_S \geq 0, \quad \forall S \subseteq \mathcal{A} \end{aligned}$$

## Custos subaditivos

Uma função  $f : 2^U \rightarrow \mathbb{R}$  é subaditiva se

- $f(S_1 \cup S_2) \leq f(S_1) + f(S_2)$  para todo  $S_1, S_2 \subseteq U$
- Ex: o custo do jogo de localização de instalações

Se  $c$  é subaditiva, então o valor de uma solução ótima inteira do programa linear

$$\begin{aligned} \min \quad & \sum_{S \subseteq \mathcal{A}} \lambda_S c(S) \\ \text{tal que} \quad & \sum_{S: j \in S} \lambda_S = 1, \quad \forall j \in \mathcal{A} \\ & \lambda_S \geq 0, \quad \forall S \subseteq \mathcal{A} \end{aligned}$$

é precisamente  $c(\mathcal{A})$

## Custos subaditivos

Uma função  $f : 2^U \rightarrow \mathbb{R}$  é subaditiva se

- $f(S_1 \cup S_2) \leq f(S_1) + f(S_2)$  para todo  $S_1, S_2 \subseteq U$
- Ex: o custo do jogo de localização de instalações

Se  $c$  é subaditiva, então o valor de uma solução ótima inteira do programa linear

$$\begin{aligned} \min \quad & \sum_{S \subseteq \mathcal{A}} \lambda_S c(S) \\ \text{tal que} \quad & \sum_{S: j \in S} \lambda_S = 1, \quad \forall j \in \mathcal{A} \\ & \lambda_S \geq 0, \quad \forall S \subseteq \mathcal{A} \end{aligned}$$

é precisamente  $c(\mathcal{A})$

Nesse caso, o maior  $\gamma$  tal que o  $\gamma$ -núcleo é não-vazio é o inverso do gap de integralidade

## Custos subaditivos

Uma função  $f : 2^U \rightarrow \mathbb{R}$  é subaditiva se

- $f(S_1 \cup S_2) \leq f(S_1) + f(S_2)$  para todo  $S_1, S_2 \subseteq U$
- Ex: o custo do jogo de localização de instalações

Se  $c$  é subaditiva, então o valor de uma solução ótima inteira do programa linear

$$\begin{aligned} \min \quad & \sum_{S \subseteq \mathcal{A}} \lambda_S c(S) \\ \text{tal que} \quad & \sum_{S: j \in S} \lambda_S = 1, \quad \forall j \in \mathcal{A} \\ & \lambda_S \geq 0, \quad \forall S \subseteq \mathcal{A} \end{aligned}$$

é precisamente  $c(\mathcal{A})$

Nesse caso, o maior  $\gamma$  tal que o  $\gamma$ -núcleo é não-vazio é o inverso do gap de integralidade

- Alguns problemas têm uma formulação equivalente a essa

## Custos subaditivos

Uma função  $f : 2^U \rightarrow \mathbb{R}$  é subaditiva se

- $f(S_1 \cup S_2) \leq f(S_1) + f(S_2)$  para todo  $S_1, S_2 \subseteq U$
- Ex: o custo do jogo de localização de instalações

Se  $c$  é subaditiva, então o valor de uma solução ótima inteira do programa linear

$$\begin{aligned} \min \quad & \sum_{S \subseteq \mathcal{A}} \lambda_S c(S) \\ \text{tal que} \quad & \sum_{S: j \in S} \lambda_S = 1, \quad \forall j \in \mathcal{A} \\ & \lambda_S \geq 0, \quad \forall S \subseteq \mathcal{A} \end{aligned}$$

é precisamente  $c(\mathcal{A})$

Nesse caso, o maior  $\gamma$  tal que o  $\gamma$ -núcleo é não-vazio é o inverso do gap de integralidade

- Alguns problemas têm uma formulação equivalente a essa
- Porém de tamanho polinomial



# Formulação para Localização de Instalações

Variáveis:

# Formulação para Localização de Instalações

Variáveis:

- $x_i$  indica se a instalação  $i \in \mathcal{F}$  é aberta

# Formulação para Localização de Instalações

Variáveis:

- $x_i$  indica se a instalação  $i \in \mathcal{F}$  é aberta
- $y_{ij}$  indica se a cidade  $j \in \mathcal{A}$  se conecta a instalação  $i \in \mathcal{F}$

# Formulação para Localização de Instalações

Variáveis:

- $x_i$  indica se a instalação  $i \in \mathcal{F}$  é aberta
- $y_{ij}$  indica se a cidade  $j \in \mathcal{A}$  se conecta a instalação  $i \in \mathcal{F}$

$$\min \sum_{i \in \mathcal{F}} f_i x_i + \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{A}} d_{ij} y_{ij}$$

$$\text{tal que } \sum_{i \in \mathcal{F}} y_{ij} \geq 1, \quad \forall j \in \mathcal{A}$$

$$x_i \geq y_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$$

$$x_i \in \{0, 1\}, \quad \forall i \in \mathcal{F}$$

$$y_{ij} \in \{0, 1\}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$$

# Formulação para Localização de Instalações

Variáveis:

- $x_i$  indica se a instalação  $i \in \mathcal{F}$  é aberta
- $y_{ij}$  indica se a cidade  $j \in \mathcal{A}$  se conecta a instalação  $i \in \mathcal{F}$

$$\begin{aligned} \min \quad & \sum_{i \in \mathcal{F}} f_i x_i + \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{A}} d_{ij} y_{ij} \\ \text{tal que} \quad & \sum_{i \in \mathcal{F}} y_{ij} \geq 1, \quad \forall j \in \mathcal{A} \\ & x_i \geq y_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \\ & x_i \in \{0, 1\}, \quad \forall i \in \mathcal{F} \\ & y_{ij} \in \{0, 1\}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \end{aligned}$$

Relaxamos as restrições de integralidade de  $x$  e  $y$

# Formulação para Localização de Instalações

$$\begin{aligned} \min & \sum_{i \in \mathcal{F}} f_i x_i + \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{A}} d_{ij} y_{ij} \\ \text{tal que} & \sum_{i \in \mathcal{F}} y_{ij} \geq 1, \quad \forall j \in \mathcal{A} \\ & x_i \geq y_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \\ & x_i \geq 0, \quad \forall i \in \mathcal{F} \\ & y_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \end{aligned}$$

# Formulação para Localização de Instalações

$$\begin{aligned} \min \quad & \sum_{i \in \mathcal{F}} f_i x_i + \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{A}} d_{ij} y_{ij} \\ \text{tal que} \quad & \sum_{i \in \mathcal{F}} y_{ij} \geq 1, \quad \forall j \in \mathcal{A} \\ & x_i \geq y_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \\ & x_i \geq 0, \quad \forall i \in \mathcal{F} \\ & y_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \end{aligned}$$

O dual é

# Formulação para Localização de Instalações

$$\begin{aligned} \min \quad & \sum_{i \in \mathcal{F}} f_i x_i + \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{A}} d_{ij} y_{ij} \\ \text{tal que} \quad & \sum_{i \in \mathcal{F}} y_{ij} \geq 1, \quad \forall j \in \mathcal{A} \\ & x_i \geq y_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \\ & x_i \geq 0, \quad \forall i \in \mathcal{F} \\ & y_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \end{aligned}$$

O dual é

$$\begin{aligned} \max \quad & \sum_{j \in \mathcal{A}} \alpha_j \\ \text{tal que} \quad & \beta_{ij} \geq \alpha_j - d_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \\ & \sum_{j \in \mathcal{A}} \beta_{ij} \leq f_i, \quad \forall i \in \mathcal{F} \\ & \alpha_j \geq 0, \quad \forall j \in \mathcal{A} \\ & \beta_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \end{aligned}$$



# Formulação para Localização de Instalações

$$\begin{aligned} \min \quad & \sum_{i \in \mathcal{F}} f_i x_i + \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{A}} d_{ij} y_{ij} \\ \text{tal que} \quad & \sum_{i \in \mathcal{F}} y_{ij} \geq 1, \quad \forall j \in \mathcal{A} \\ & x_i \geq y_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \\ & x_i \geq 0, \quad \forall i \in \mathcal{F} \\ & y_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \end{aligned}$$

O dual é

$$\begin{aligned} \max \quad & \sum_{j \in \mathcal{A}} \alpha_j \\ \text{tal que} \quad & \beta_{ij} \geq \alpha_j - d_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \\ & \sum_{j \in \mathcal{A}} \beta_{ij} \leq f_i, \quad \forall i \in \mathcal{F} \\ & \alpha_j \geq 0, \quad \forall j \in \mathcal{A} \\ & \beta_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \end{aligned}$$

Vamos mostrar que  $\alpha$  satisfaz a propriedade do núcleo

$\alpha$  satisfaz a propriedade do núcleo

Seja  $S \subseteq \mathcal{A}$

## $\alpha$ satisfaz a propriedade do núcleo

Seja  $S \subseteq \mathcal{A}$

- Considere uma solução ótima com conjunto de cidades  $S$

## $\alpha$ satisfaz a propriedade do núcleo

Seja  $S \subseteq \mathcal{A}$

- Considere uma solução ótima com conjunto de cidades  $S$
- Sejam  $i_1, \dots, i_k$  as instalações abertas

## $\alpha$ satisfaz a propriedade do núcleo

Seja  $S \subseteq \mathcal{A}$

- Considere uma solução ótima com conjunto de cidades  $S$
- Sejam  $i_1, \dots, i_k$  as instalações abertas
- Sejam  $R_1, \dots, R_k$  os conjuntos de clientes conectados a tais instalações

## $\alpha$ satisfaz a propriedade do núcleo

Seja  $S \subseteq \mathcal{A}$

- Considere uma solução ótima com conjunto de cidades  $S$
- Sejam  $i_1, \dots, i_k$  as instalações abertas
- Sejam  $R_1, \dots, R_k$  os conjuntos de clientes conectados a tais instalações

Como, para todo  $t \in \{1, \dots, k\}$

## $\alpha$ satisfaz a propriedade do núcleo

Seja  $S \subseteq \mathcal{A}$

- Considere uma solução ótima com conjunto de cidades  $S$
- Sejam  $i_1, \dots, i_k$  as instalações abertas
- Sejam  $R_1, \dots, R_k$  os conjuntos de clientes conectados a tais instalações

Como, para todo  $t \in \{1, \dots, k\}$

- $\sum_{j \in \mathcal{A}} \beta_{ijt} \leq f_{i_t}$  e

## $\alpha$ satisfaz a propriedade do núcleo

Seja  $S \subseteq \mathcal{A}$

- Considere uma solução ótima com conjunto de cidades  $S$
- Sejam  $i_1, \dots, i_k$  as instalações abertas
- Sejam  $R_1, \dots, R_k$  os conjuntos de clientes conectados a tais instalações

Como, para todo  $t \in \{1, \dots, k\}$

- $\sum_{j \in \mathcal{A}} \beta_{ijt} \leq f_{i_t}$  e
- para todo  $j \in R_t$ ,  $\beta_{ijt} \geq \alpha_j - d_{ijt}$



## $\alpha$ satisfaz a propriedade do núcleo

Seja  $S \subseteq \mathcal{A}$

- Considere uma solução ótima com conjunto de cidades  $S$
- Sejam  $i_1, \dots, i_k$  as instalações abertas
- Sejam  $R_1, \dots, R_k$  os conjuntos de clientes conectados a tais instalações

Como, para todo  $t \in \{1, \dots, k\}$

- $\sum_{j \in \mathcal{A}} \beta_{itj} \leq f_{i_t}$  e
- para todo  $j \in R_t$ ,  $\beta_{itj} \geq \alpha_j - d_{itj}$

Temos

## $\alpha$ satisfaz a propriedade do núcleo

Seja  $S \subseteq \mathcal{A}$

- Considere uma solução ótima com conjunto de cidades  $S$
- Sejam  $i_1, \dots, i_k$  as instalações abertas
- Sejam  $R_1, \dots, R_k$  os conjuntos de clientes conectados a tais instalações

Como, para todo  $t \in \{1, \dots, k\}$

- $\sum_{j \in \mathcal{A}} \beta_{ijt} \leq f_{i_t}$  e
- para todo  $j \in R_t$ ,  $\beta_{ijt} \geq \alpha_j - d_{ijt}$

Temos

$$\sum_{j \in R_t} \alpha_j$$

## $\alpha$ satisfaz a propriedade do núcleo

Seja  $S \subseteq \mathcal{A}$

- Considere uma solução ótima com conjunto de cidades  $S$
- Sejam  $i_1, \dots, i_k$  as instalações abertas
- Sejam  $R_1, \dots, R_k$  os conjuntos de clientes conectados a tais instalações

Como, para todo  $t \in \{1, \dots, k\}$

- $\sum_{j \in \mathcal{A}} \beta_{i_t j} \leq f_{i_t}$  e
- para todo  $j \in R_t$ ,  $\beta_{i_t j} \geq \alpha_j - d_{i_t j}$

Temos

$$\sum_{j \in R_t} \alpha_j \leq \sum_{j \in R_t} \beta_{i_t j} + \sum_{j \in R_t} d_{i_t j}$$

## $\alpha$ satisfaz a propriedade do núcleo

Seja  $S \subseteq \mathcal{A}$

- Considere uma solução ótima com conjunto de cidades  $S$
- Sejam  $i_1, \dots, i_k$  as instalações abertas
- Sejam  $R_1, \dots, R_k$  os conjuntos de clientes conectados a tais instalações

Como, para todo  $t \in \{1, \dots, k\}$

- $\sum_{j \in \mathcal{A}} \beta_{itj} \leq f_{i_t}$  e
- para todo  $j \in R_t$ ,  $\beta_{itj} \geq \alpha_j - d_{itj}$

Temos

$$\sum_{j \in R_t} \alpha_j \leq \sum_{j \in R_t} \beta_{itj} + \sum_{j \in R_t} d_{itj} \leq \sum_{j \in \mathcal{A}} \beta_{itj} + \sum_{j \in R_t} d_{itj}$$

## $\alpha$ satisfaz a propriedade do núcleo

Seja  $S \subseteq \mathcal{A}$

- Considere uma solução ótima com conjunto de cidades  $S$
- Sejam  $i_1, \dots, i_k$  as instalações abertas
- Sejam  $R_1, \dots, R_k$  os conjuntos de clientes conectados a tais instalações

Como, para todo  $t \in \{1, \dots, k\}$

- $\sum_{j \in \mathcal{A}} \beta_{itj} \leq f_{i_t}$  e
- para todo  $j \in R_t$ ,  $\beta_{itj} \geq \alpha_j - d_{itj}$

Temos

$$\sum_{j \in R_t} \alpha_j \leq \sum_{j \in R_t} \beta_{itj} + \sum_{j \in R_t} d_{itj} \leq \sum_{j \in \mathcal{A}} \beta_{itj} + \sum_{j \in R_t} d_{itj} \leq f_{i_t} + \sum_{j \in R_t} d_{itj}$$

## $\alpha$ satisfaz a propriedade do núcleo

Seja  $S \subseteq \mathcal{A}$

- Considere uma solução ótima com conjunto de cidades  $S$
- Sejam  $i_1, \dots, i_k$  as instalações abertas
- Sejam  $R_1, \dots, R_k$  os conjuntos de clientes conectados a tais instalações

Como, para todo  $t \in \{1, \dots, k\}$

- $\sum_{j \in \mathcal{A}} \beta_{itj} \leq f_{i_t}$  e
- para todo  $j \in R_t$ ,  $\beta_{itj} \geq \alpha_j - d_{itj}$

Temos

$$\sum_{j \in R_t} \alpha_j \leq \sum_{j \in R_t} \beta_{itj} + \sum_{j \in R_t} d_{itj} \leq \sum_{j \in \mathcal{A}} \beta_{itj} + \sum_{j \in R_t} d_{itj} \leq f_{i_t} + \sum_{j \in R_t} d_{itj}$$

Somando sobre  $t$ , temos que  $\sum_{j \in S} \alpha_j \leq c(S)$

## Calculando uma alocação de custos

Usando o dual da formulação do Problema da Localização de Instalações

# Calculando uma alocação de custos

Usando o dual da formulação do Problema da Localização de Instalações

$$\begin{aligned} \max \quad & \sum_{j \in \mathcal{A}} \alpha_j \\ \text{tal que} \quad & \beta_{ij} \geq \alpha_j - d_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \\ & \sum_{j \in \mathcal{A}} \beta_{ij} \leq f_i, \quad \forall i \in \mathcal{F} \\ & \alpha_j \geq 0, \quad \forall j \in \mathcal{A} \\ & \beta_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \end{aligned}$$



## Calculando uma alocação de custos

Usando o dual da formulação do Problema da Localização de Instalações

$$\begin{aligned} \max \quad & \sum_{j \in \mathcal{A}} \alpha_j \\ \text{tal que} \quad & \beta_{ij} \geq \alpha_j - d_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \\ & \sum_{j \in \mathcal{A}} \beta_{ij} \leq f_i, \quad \forall i \in \mathcal{F} \\ & \alpha_j \geq 0, \quad \forall j \in \mathcal{A} \\ & \beta_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \end{aligned}$$

podemos calcular uma alocação de custos para o Jogo da Localização de Instalações

## Calculando uma alocação de custos

Usando o dual da formulação do Problema da Localização de Instalações

$$\begin{aligned} \max \quad & \sum_{j \in \mathcal{A}} \alpha_j \\ \text{tal que} \quad & \beta_{ij} \geq \alpha_j - d_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \\ & \sum_{j \in \mathcal{A}} \beta_{ij} \leq f_i, \quad \forall i \in \mathcal{F} \\ & \alpha_j \geq 0, \quad \forall j \in \mathcal{A} \\ & \beta_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \end{aligned}$$

podemos calcular uma alocação de custos para o Jogo da Localização de Instalações

- que está no  $(1/1,463)$ -núcleo ( $1/1,463 \approx 0,683$ )

# Calculando uma alocação de custos

Usando o dual da formulação do Problema da Localização de Instalações

$$\begin{aligned} \max \quad & \sum_{j \in \mathcal{A}} \alpha_j \\ \text{tal que} \quad & \beta_{ij} \geq \alpha_j - d_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \\ & \sum_{j \in \mathcal{A}} \beta_{ij} \leq f_i, \quad \forall i \in \mathcal{F} \\ & \alpha_j \geq 0, \quad \forall j \in \mathcal{A} \\ & \beta_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \end{aligned}$$

podemos calcular uma alocação de custos para o Jogo da Localização de Instalações

- que está no  $(1/1,463)$ -núcleo ( $1/1,463 \approx 0,683$ )
- em tempo polinomial

# Calculando uma alocação de custos

Usando o dual da formulação do Problema da Localização de Instalações

$$\begin{aligned} \max \quad & \sum_{j \in \mathcal{A}} \alpha_j \\ \text{tal que} \quad & \beta_{ij} \geq \alpha_j - d_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \\ & \sum_{j \in \mathcal{A}} \beta_{ij} \leq f_i, \quad \forall i \in \mathcal{F} \\ & \alpha_j \geq 0, \quad \forall j \in \mathcal{A} \\ & \beta_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \end{aligned}$$

podemos calcular uma alocação de custos para o Jogo da Localização de Instalações

- que está no  $(1/1,463)$ -núcleo ( $1/1,463 \approx 0,683$ )
- em tempo polinomial
  - ▶ mesmo sem saber explicitamente os valores de  $c(S)$

# Mecanismo de compartilhamento de custos

Um provedor irá fornecer serviço para os clientes

# Mecanismo de compartilhamento de custos

Um provedor irá fornecer serviço para os clientes

- Ex: instalar instalações e conectar as cidades

# Mecanismo de compartilhamento de custos

Um provedor irá fornecer serviço para os clientes

- Ex: instalar instalações e conectar as cidades
- Atender um conjunto  $S$  de clientes tem um custo  $c(S)$

# Mecanismo de compartilhamento de custos

Um provedor irá fornecer serviço para os clientes

- Ex: instalar instalações e conectar as cidades
- Atender um conjunto  $S$  de clientes tem um custo  $c(S)$
- Para isso, realiza um leilão



# Mecanismo de compartilhamento de custos

Um provedor irá fornecer serviço para os clientes

- Ex: instalar instalações e conectar as cidades
- Atender um conjunto  $S$  de clientes tem um custo  $c(S)$
- Para isso, realiza um leilão
  - ▶ Escolhe o conjunto  $Q$  de clientes servidos

# Mecanismo de compartilhamento de custos

Um provedor irá fornecer serviço para os clientes

- Ex: instalar instalações e conectar as cidades
- Atender um conjunto  $S$  de clientes tem um custo  $c(S)$
- Para isso, realiza um leilão
  - ▶ Escolhe o conjunto  $Q$  de clientes servidos
  - ▶ Escolhe um preço  $p_i$  para cada cliente  $i$

# Mecanismo de compartilhamento de custos

Um provedor irá fornecer serviço para os clientes

- Ex: instalar instalações e conectar as cidades
- Atender um conjunto  $S$  de clientes tem um custo  $c(S)$
- Para isso, realiza um leilão
  - ▶ Escolhe o conjunto  $Q$  de clientes servidos
  - ▶ Escolhe um preço  $p_i$  para cada cliente  $i$

O cliente:

# Mecanismo de compartilhamento de custos

Um provedor irá fornecer serviço para os clientes

- Ex: instalar instalações e conectar as cidades
- Atender um conjunto  $S$  de clientes tem um custo  $c(S)$
- Para isso, realiza um leilão
  - ▶ Escolhe o conjunto  $Q$  de clientes servidos
  - ▶ Escolhe um preço  $p_i$  para cada cliente  $i$

O cliente:

- Dá um lance  $b_i$  para ser servido

# Mecanismo de compartilhamento de custos

Um provedor irá fornecer serviço para os clientes

- Ex: instalar instalações e conectar as cidades
- Atender um conjunto  $S$  de clientes tem um custo  $c(S)$
- Para isso, realiza um leilão
  - ▶ Escolhe o conjunto  $Q$  de clientes servidos
  - ▶ Escolhe um preço  $p_i$  para cada cliente  $i$

O cliente:

- Dá um lance  $b_i$  para ser servido
- Tem utilidade  $v_i q_i - p_i$ , onde  $q_i$  indica se foi servido

# Mecanismo de compartilhamento de custos

Um mecanismo de compartilhamento de custos é uma função que dados o vetor  $b$  de lances define

# Mecanismo de compartilhamento de custos

Um mecanismo de compartilhamento de custos é uma função que dados o vetor  $b$  de lances define

- o conjunto  $Q \subseteq \mathcal{A}$  de clientes servidos

# Mecanismo de compartilhamento de custos

Um mecanismo de compartilhamento de custos é uma função que dados o vetor  $b$  de lances define

- o conjunto  $Q \subseteq \mathcal{A}$  de clientes servidos
- o vetor de pagamentos  $p \in \mathbb{R}^n$



# Mecanismo de compartilhamento de custos

Um mecanismo de compartilhamento de custos é uma função que dados o vetor  $b$  de lances define

- o conjunto  $Q \subseteq \mathcal{A}$  de clientes servidos
- o vetor de pagamentos  $p \in \mathbb{R}^n$

Consideramos as seguintes restrições para o mecanismo:

# Mecanismo de compartilhamento de custos

Um **mecanismo de compartilhamento de custos** é uma função que dados o vetor  $b$  de lances define

- o conjunto  $Q \subseteq \mathcal{A}$  de clientes servidos
- o vetor de pagamentos  $p \in \mathbb{R}^n$

Consideramos as seguintes restrições para o mecanismo:

- **Sem transferências positivas:**  $p_i \geq 0$  para todo  $i \in \mathcal{A}$

# Mecanismo de compartilhamento de custos

Um **mecanismo de compartilhamento de custos** é uma função que dados o vetor  $b$  de lances define

- o conjunto  $Q \subseteq \mathcal{A}$  de clientes servidos
- o vetor de pagamentos  $p \in \mathbb{R}^n$

Consideramos as seguintes restrições para o mecanismo:

- **Sem transferências positivas:**  $p_i \geq 0$  para todo  $i \in \mathcal{A}$
- **Participação voluntária** (ou racionalidade individual):

# Mecanismo de compartilhamento de custos

Um **mecanismo de compartilhamento de custos** é uma função que dados o vetor  $b$  de lances define

- o conjunto  $Q \subseteq \mathcal{A}$  de clientes servidos
- o vetor de pagamentos  $p \in \mathbb{R}^n$

Consideramos as seguintes restrições para o mecanismo:

- **Sem transferências positivas:**  $p_i \geq 0$  para todo  $i \in \mathcal{A}$
- **Participação voluntária** (ou racionalidade individual):
  - ▶  $p_i = 0$  se  $i$  não é servido

# Mecanismo de compartilhamento de custos

Um **mecanismo de compartilhamento de custos** é uma função que dados o vetor  $b$  de lances define

- o conjunto  $Q \subseteq \mathcal{A}$  de clientes servidos
- o vetor de pagamentos  $p \in \mathbb{R}^n$

Consideramos as seguintes restrições para o mecanismo:

- **Sem transferências positivas:**  $p_i \geq 0$  para todo  $i \in \mathcal{A}$
- **Participação voluntária** (ou racionalidade individual):
  - ▶  $p_i = 0$  se  $i$  não é servido
  - ▶  $p_i \leq b_i$  para  $i \in Q$

# Mecanismo de compartilhamento de custos

Um **mecanismo de compartilhamento de custos** é uma função que dados o vetor  $b$  de lances define

- o conjunto  $Q \subseteq \mathcal{A}$  de clientes servidos
- o vetor de pagamentos  $p \in \mathbb{R}^n$

Consideramos as seguintes restrições para o mecanismo:

- **Sem transferências positivas:**  $p_i \geq 0$  para todo  $i \in \mathcal{A}$
- **Participação voluntária** (ou racionalidade individual):
  - ▶  $p_i = 0$  se  $i$  não é servido
  - ▶  $p_i \leq b_i$  para  $i \in Q$
- **Soberania do consumidor:** para todo  $i \in \mathcal{A}$ , existe um lance  $b_i^*$  que faz com que  $i$  seja servido

# Mecanismo de compartilhamento de custos

Um **mecanismo de compartilhamento de custos** é uma função que dados o vetor  $b$  de lances define

- o conjunto  $Q \subseteq \mathcal{A}$  de clientes servidos
- o vetor de pagamentos  $p \in \mathbb{R}^n$

Consideramos as seguintes restrições para o mecanismo:

- **Sem transferências positivas:**  $p_i \geq 0$  para todo  $i \in \mathcal{A}$
- **Participação voluntária** (ou racionalidade individual):
  - ▶  $p_i = 0$  se  $i$  não é servido
  - ▶  $p_i \leq b_i$  para  $i \in Q$
- **Soberania do consumidor:** para todo  $i \in \mathcal{A}$ , existe um lance  $b_i^*$  que faz com que  $i$  seja servido
  - ▶ independentemente dos outros jogadores

# Esquema de compartilhamento de custos

Um **esquema de compartilhamento de custos** é uma função  $\xi : \mathcal{A} \times 2^{\mathcal{A}}$  tal que, para todo  $S \subseteq \mathcal{A}$  e todo  $i \notin S$ ,  $\xi(i, S) = 0$



# Esquema de compartilhamento de custos

Um **esquema de compartilhamento de custos** é uma função  $\xi : \mathcal{A} \times 2^{\mathcal{A}}$  tal que, para todo  $S \subseteq \mathcal{A}$  e todo  $i \notin S$ ,  $\xi(i, S) = 0$

- define como compartilhar o custo entre uma coalizão  $S$

# Esquema de compartilhamento de custos

Um **esquema de compartilhamento de custos** é uma função  $\xi : \mathcal{A} \times 2^{\mathcal{A}}$  tal que, para todo  $S \subseteq \mathcal{A}$  e todo  $i \notin S$ ,  $\xi(i, S) = 0$

- define como compartilhar o custo entre uma coalizão  $S$
- quem não está na coalizão não paga

# Esquema de compartilhamento de custos

Um **esquema de compartilhamento de custos** é uma função  $\xi : \mathcal{A} \times 2^{\mathcal{A}}$  tal que, para todo  $S \subseteq \mathcal{A}$  e todo  $i \notin S$ ,  $\xi(i, S) = 0$

- define como compartilhar o custo entre uma coalizão  $S$
- quem não está na coalizão não paga

Um **esquema de compartilhamento de custos** é  $\gamma$ -**orçamento balanceado** se, para todo,  $S \subseteq \mathcal{A}$ ,  $\gamma c(S) \leq \sum_{i \in S} \xi(i, S) \leq c(S)$

# Esquema de compartilhamento de custos

Um **esquema de compartilhamento de custos** é uma função  $\xi : \mathcal{A} \times 2^{\mathcal{A}}$  tal que, para todo  $S \subseteq \mathcal{A}$  e todo  $i \notin S$ ,  $\xi(i, S) = 0$

- define como compartilhar o custo entre uma coalizão  $S$
- quem não está na coalizão não paga

Um **esquema de compartilhamento de custos** é  $\gamma$ -**orçamento balanceado** se, para todo,  $S \subseteq \mathcal{A}$ ,  $\gamma c(S) \leq \sum_{i \in S} \xi(i, S) \leq c(S)$

Um **esquema de compartilhamento de custos** é **monotônico cruzado** se, para todo  $S, T \subseteq \mathcal{A}$  e  $i \in S$ ,  $\xi(i, S) \geq \xi(i, S \cup T)$

# Esquema de compartilhamento de custos

Um **esquema de compartilhamento de custos** é uma função  $\xi : \mathcal{A} \times 2^{\mathcal{A}}$  tal que, para todo  $S \subseteq \mathcal{A}$  e todo  $i \notin S$ ,  $\xi(i, S) = 0$

- define como compartilhar o custo entre uma coalizão  $S$
- quem não está na coalizão não paga

Um **esquema de compartilhamento de custos** é  $\gamma$ -**orçamento balanceado** se, para todo,  $S \subseteq \mathcal{A}$ ,  $\gamma c(S) \leq \sum_{i \in S} \xi(i, S) \leq c(S)$

Um **esquema de compartilhamento de custos** é **monotônico cruzado** se, para todo  $S, T \subseteq \mathcal{A}$  e  $i \in S$ ,  $\xi(i, S) \geq \xi(i, S \cup T)$

- Se o conjunto de jogadores servidos aumenta

# Esquema de compartilhamento de custos

Um **esquema de compartilhamento de custos** é uma função  $\xi : \mathcal{A} \times 2^{\mathcal{A}}$  tal que, para todo  $S \subseteq \mathcal{A}$  e todo  $i \notin S$ ,  $\xi(i, S) = 0$

- define como compartilhar o custo entre uma coalizão  $S$
- quem não está na coalizão não paga

Um **esquema de compartilhamento de custos** é  $\gamma$ -**orçamento balanceado** se, para todo,  $S \subseteq \mathcal{A}$ ,  $\gamma c(S) \leq \sum_{i \in S} \xi(i, S) \leq c(S)$

Um **esquema de compartilhamento de custos** é **monotônico cruzado** se, para todo  $S, T \subseteq \mathcal{A}$  e  $i \in S$ ,  $\xi(i, S) \geq \xi(i, S \cup T)$

- Se o conjunto de jogadores servidos aumenta
- o preço pago por  $i$  não aumenta

## Monotonicidade cruzada

**Proposição:** Seja  $\gamma$  um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então  $\xi(\cdot, \mathcal{A})$  está no  $\gamma$ -núcleo do jogo.

# Monotonicidade cruzada

**Proposição:** Seja  $\gamma$  um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então  $\xi(\cdot, \mathcal{A})$  está no  $\gamma$ -núcleo do jogo.

**Prova:**



# Monotonicidade cruzada

**Proposição:** Seja  $\gamma$  um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então  $\xi(\cdot, \mathcal{A})$  está no  $\gamma$ -núcleo do jogo.

**Prova:** Como  $\xi$  é  $\gamma$ -orçamento balanceado,  $\xi(\cdot, \mathcal{A})$  também é

# Monotonicidade cruzada

**Proposição:** Seja  $\gamma$  um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então  $\xi(\cdot, \mathcal{A})$  está no  $\gamma$ -núcleo do jogo.

**Prova:** Como  $\xi$  é  $\gamma$ -orçamento balanceado,  $\xi(\cdot, \mathcal{A})$  também é

Seja  $S \subseteq \mathcal{A}$

# Monotonicidade cruzada

**Proposição:** Seja  $\gamma$  um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então  $\xi(\cdot, \mathcal{A})$  está no  $\gamma$ -núcleo do jogo.

**Prova:** Como  $\xi$  é  $\gamma$ -orçamento balanceado,  $\xi(\cdot, \mathcal{A})$  também é

Seja  $S \subseteq \mathcal{A}$

- Por monotonicidade,  $\xi(i, \mathcal{A}) \leq \xi(i, S)$  para todo  $i \in S$

# Monotonicidade cruzada

**Proposição:** Seja  $\gamma$  um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então  $\xi(\cdot, \mathcal{A})$  está no  $\gamma$ -núcleo do jogo.

**Prova:** Como  $\xi$  é  $\gamma$ -orçamento balanceado,  $\xi(\cdot, \mathcal{A})$  também é

Seja  $S \subseteq \mathcal{A}$

- Por monotonicidade,  $\xi(i, \mathcal{A}) \leq \xi(i, S)$  para todo  $i \in S$
- Por ser  $\gamma$ -orçamento balanceado,  $\sum_{i \in S} \xi(i, S) \leq c(S)$

# Monotonicidade cruzada

**Proposição:** Seja  $\gamma$  um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então  $\xi(\cdot, \mathcal{A})$  está no  $\gamma$ -núcleo do jogo.

**Prova:** Como  $\xi$  é  $\gamma$ -orçamento balanceado,  $\xi(\cdot, \mathcal{A})$  também é

Seja  $S \subseteq \mathcal{A}$

- Por monotonicidade,  $\xi(i, \mathcal{A}) \leq \xi(i, S)$  para todo  $i \in S$
- Por ser  $\gamma$ -orçamento balanceado,  $\sum_{i \in S} \xi(i, S) \leq c(S)$

Portanto,  $\sum_{i \in S} \xi(i, \mathcal{A}) \leq \sum_{i \in S} \xi(i, S) \leq c(S)$

# Monotonicidade cruzada

**Proposição:** Seja  $\gamma$  um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então  $\xi(\cdot, \mathcal{A})$  está no  $\gamma$ -núcleo do jogo.

**Prova:** Como  $\xi$  é  $\gamma$ -orçamento balanceado,  $\xi(\cdot, \mathcal{A})$  também é

Seja  $S \subseteq \mathcal{A}$

- Por monotonicidade,  $\xi(i, \mathcal{A}) \leq \xi(i, S)$  para todo  $i \in S$
- Por ser  $\gamma$ -orçamento balanceado,  $\sum_{i \in S} \xi(i, S) \leq c(S)$

Portanto,  $\sum_{i \in S} \xi(i, \mathcal{A}) \leq \sum_{i \in S} \xi(i, S) \leq c(S)$

- $\xi(\cdot, \mathcal{A})$  satisfaz a propriedade do núcleo

# Mecanismo $\mathcal{M}_\xi$

# Mecanismo $\mathcal{M}_\xi$

Mecanismo  $\mathcal{M}_\xi$

- 1  $S \leftarrow \mathcal{A}$
- 2 **repita**
- 3      $S \leftarrow \{i \in S : b_i \geq \xi(i, S)\}$
- 4 **até que**  $b_i \geq \xi(i, S)$  para todo  $i \in S$
- 5 **retorne**  $Q = S$  e  $p_i = \xi(i, S)$  para todo  $i \in S$



# Mecanismo $\mathcal{M}_\xi$

Mecanismo  $\mathcal{M}_\xi$

- 1  $S \leftarrow \mathcal{A}$
- 2 **repita**
- 3      $S \leftarrow \{i \in S : b_i \geq \xi(i, S)\}$
- 4 **até que**  $b_i \geq \xi(i, S)$  para todo  $i \in S$
- 5 **retorne**  $Q = S$  e  $p_i = \xi(i, S)$  para todo  $i \in S$

Note que se  $\xi$  é  $\gamma$ -orçamento balanceado então  $\mathcal{M}_\xi$  também é

# Mecanismo $\mathcal{M}_\xi$

Mecanismo  $\mathcal{M}_\xi$

- 1  $S \leftarrow \mathcal{A}$
- 2 **repita**
- 3      $S \leftarrow \{i \in S : b_i \geq \xi(i, S)\}$
- 4 **até que**  $b_i \geq \xi(i, S)$  para todo  $i \in S$
- 5 **retorne**  $Q = S$  e  $p_i = \xi(i, S)$  para todo  $i \in S$

Note que se  $\xi$  é  $\gamma$ -orçamento balanceado então  $\mathcal{M}_\xi$  também é

**Proposição:** Se  $\xi$  é um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então o mecanismo  $\mathcal{M}_\xi$  devolve o único conjunto maximal  $S \subseteq \mathcal{A}$  tal que, para todo  $i \in S$ ,  $b_i \geq \xi(i, S)$ .

# Mecanismo

**Proposição:** Se  $\xi$  é um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então o mecanismo  $\mathcal{M}_\xi$  devolve o único conjunto maximal  $S \subseteq \mathcal{A}$  tal que, para todo  $i \in S$ ,  $b_i \geq \xi(i, S)$ .

# Mecanismo

**Proposição:** Se  $\xi$  é um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então o mecanismo  $\mathcal{M}_\xi$  devolve o único conjunto maximal  $S \subseteq \mathcal{A}$  tal que, para todo  $i \in S$ ,  $b_i \geq \xi(i, S)$ .

**Lema 1:** Se  $\xi$  é um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então existe um único conjunto maximal  $S \subseteq \mathcal{A}$  tal que, para todo  $i \in S$ ,  $b_i \geq \xi(i, S)$ .

# Lema 1

**Lema 1:** Se  $\xi$  é um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então existe um único conjunto maximal  $S \subseteq \mathcal{A}$  tal que, para todo  $i \in S$ ,  $b_i \geq \xi(i, S)$ .

# Lema 1

**Lema 1:** Se  $\xi$  é um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então existe um único conjunto maximal  $S \subseteq \mathcal{A}$  tal que, para todo  $i \in S$ ,  $b_i \geq \xi(i, S)$ .

**Prova:**

# Lema 1

**Lema 1:** Se  $\xi$  é um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então existe um único conjunto maximal  $S \subseteq \mathcal{A}$  tal que, para todo  $i \in S$ ,  $b_i \geq \xi(i, S)$ .

**Prova:** Suponha que existem conjuntos maximais  $S_1$  e  $S_2$  que satisfazem a propriedade

# Lema 1

**Lema 1:** Se  $\xi$  é um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então existe um único conjunto maximal  $S \subseteq \mathcal{A}$  tal que, para todo  $i \in S$ ,  $b_i \geq \xi(i, S)$ .

**Prova:** Suponha que existem conjuntos maximais  $S_1$  e  $S_2$  que satisfazem a propriedade

- $b_i \geq \xi(i, S_1) \geq \xi(i, S_1 \cup S_2)$  para todo  $i \in S_1$



# Lema 1

**Lema 1:** Se  $\xi$  é um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então existe um único conjunto maximal  $S \subseteq \mathcal{A}$  tal que, para todo  $i \in S$ ,  $b_i \geq \xi(i, S)$ .

**Prova:** Suponha que existem conjuntos maximais  $S_1$  e  $S_2$  que satisfazem a propriedade

- $b_i \geq \xi(i, S_1) \geq \xi(i, S_1 \cup S_2)$  para todo  $i \in S_1$
- $b_i \geq \xi(i, S_2) \geq \xi(i, S_1 \cup S_2)$  para todo  $i \in S_2$

# Lema 1

**Lema 1:** Se  $\xi$  é um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então existe um único conjunto maximal  $S \subseteq \mathcal{A}$  tal que, para todo  $i \in S$ ,  $b_i \geq \xi(i, S)$ .

**Prova:** Suponha que existem conjuntos maximais  $S_1$  e  $S_2$  que satisfazem a propriedade

- $b_i \geq \xi(i, S_1) \geq \xi(i, S_1 \cup S_2)$  para todo  $i \in S_1$
- $b_i \geq \xi(i, S_2) \geq \xi(i, S_1 \cup S_2)$  para todo  $i \in S_2$

Portanto,  $S_1 \cup S_2$  também satisfaz a propriedade e obtemos uma contradição.

## Proposição

**Proposição:** Se  $\xi$  é um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então o mecanismo  $\mathcal{M}_\xi$  devolve o único conjunto maximal  $S \subseteq \mathcal{A}$  tal que, para todo  $i \in S$ ,  $b_i \geq \xi(i, S)$ .

# Proposição

**Proposição:** Se  $\xi$  é um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então o mecanismo  $\mathcal{M}_\xi$  devolve o único conjunto maximal  $S \subseteq \mathcal{A}$  tal que, para todo  $i \in S$ ,  $b_i \geq \xi(i, S)$ .

**Prova:**

# Proposição

**Proposição:** Se  $\xi$  é um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então o mecanismo  $\mathcal{M}_\xi$  devolve o único conjunto maximal  $S \subseteq \mathcal{A}$  tal que, para todo  $i \in S$ ,  $b_i \geq \xi(i, S)$ .

**Prova:** Seja  $S^*$  o único conjunto maximal satisfazendo  $b_i \geq \xi(i, S^*)$  para todo  $i \in S^*$ .

# Proposição

**Proposição:** Se  $\xi$  é um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então o mecanismo  $\mathcal{M}_\xi$  devolve o único conjunto maximal  $S \subseteq \mathcal{A}$  tal que, para todo  $i \in S$ ,  $b_i \geq \xi(i, S)$ .

**Prova:** Seja  $S^*$  o único conjunto maximal satisfazendo  $b_i \geq \xi(i, S^*)$  para todo  $i \in S^*$ .

Mostraremos, por contradição, que  $\mathcal{M}_\xi$  nunca elimina  $i \in S^*$

# Proposição

**Proposição:** Se  $\xi$  é um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então o mecanismo  $\mathcal{M}_\xi$  devolve o único conjunto maximal  $S \subseteq \mathcal{A}$  tal que, para todo  $i \in S$ ,  $b_i \geq \xi(i, S)$ .

**Prova:** Seja  $S^*$  o único conjunto maximal satisfazendo  $b_i \geq \xi(i, S^*)$  para todo  $i \in S^*$ .

Mostraremos, por contradição, que  $\mathcal{M}_\xi$  nunca elimina  $i \in S^*$

- Considere a primeira vez que algum  $i \in S^*$  é eliminado

# Proposição

**Proposição:** Se  $\xi$  é um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então o mecanismo  $\mathcal{M}_\xi$  devolve o único conjunto maximal  $S \subseteq \mathcal{A}$  tal que, para todo  $i \in S$ ,  $b_i \geq \xi(i, S)$ .

**Prova:** Seja  $S^*$  o único conjunto maximal satisfazendo  $b_i \geq \xi(i, S^*)$  para todo  $i \in S^*$ .

Mostraremos, por contradição, que  $\mathcal{M}_\xi$  nunca elimina  $i \in S^*$

- Considere a primeira vez que algum  $i \in S^*$  é eliminado
- Digamos que ele foi eliminado de  $S$



# Proposição

**Proposição:** Se  $\xi$  é um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então o mecanismo  $\mathcal{M}_\xi$  devolve o único conjunto maximal  $S \subseteq \mathcal{A}$  tal que, para todo  $i \in S$ ,  $b_i \geq \xi(i, S)$ .

**Prova:** Seja  $S^*$  o único conjunto maximal satisfazendo  $b_i \geq \xi(i, S^*)$  para todo  $i \in S^*$ .

Mostraremos, por contradição, que  $\mathcal{M}_\xi$  nunca elimina  $i \in S^*$

- Considere a primeira vez que algum  $i \in S^*$  é eliminado
- Digamos que ele foi eliminado de  $S$
- Então  $b_i < \xi(i, S)$

# Proposição

**Proposição:** Se  $\xi$  é um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então o mecanismo  $\mathcal{M}_\xi$  devolve o único conjunto maximal  $S \subseteq \mathcal{A}$  tal que, para todo  $i \in S$ ,  $b_i \geq \xi(i, S)$ .

**Prova:** Seja  $S^*$  o único conjunto maximal satisfazendo  $b_i \geq \xi(i, S^*)$  para todo  $i \in S^*$ .

Mostraremos, por contradição, que  $\mathcal{M}_\xi$  nunca elimina  $i \in S^*$

- Considere a primeira vez que algum  $i \in S^*$  é eliminado
- Digamos que ele foi eliminado de  $S$
- Então  $b_i < \xi(i, S)$
- Mas,  $S^* \subseteq S$  (foi a primeira eliminação)

# Proposição

**Proposição:** Se  $\xi$  é um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então o mecanismo  $\mathcal{M}_\xi$  devolve o único conjunto maximal  $S \subseteq \mathcal{A}$  tal que, para todo  $i \in S$ ,  $b_i \geq \xi(i, S)$ .

**Prova:** Seja  $S^*$  o único conjunto maximal satisfazendo  $b_i \geq \xi(i, S^*)$  para todo  $i \in S^*$ .

Mostraremos, por contradição, que  $\mathcal{M}_\xi$  nunca elimina  $i \in S^*$

- Considere a primeira vez que algum  $i \in S^*$  é eliminado
- Digamos que ele foi eliminado de  $S$
- Então  $b_i < \xi(i, S)$
- Mas,  $S^* \subseteq S$  (foi a primeira eliminação)
- Concluimos que  $b_i < \xi(i, S) \leq \xi(i, S^*)$

# Proposição

**Proposição:** Se  $\xi$  é um esquema de compartilhamento de custos monotônico cruzado  $\gamma$ -orçamento balanceado. Então o mecanismo  $\mathcal{M}_\xi$  devolve o único conjunto maximal  $S \subseteq \mathcal{A}$  tal que, para todo  $i \in S$ ,  $b_i \geq \xi(i, S)$ .

**Prova:** Seja  $S^*$  o único conjunto maximal satisfazendo  $b_i \geq \xi(i, S^*)$  para todo  $i \in S^*$ .

Mostraremos, por contradição, que  $\mathcal{M}_\xi$  nunca elimina  $i \in S^*$

- Considere a primeira vez que algum  $i \in S^*$  é eliminado
- Digamos que ele foi eliminado de  $S$
- Então  $b_i < \xi(i, S)$
- Mas,  $S^* \subseteq S$  (foi a primeira eliminação)
- Concluimos que  $b_i < \xi(i, S) \leq \xi(i, S^*)$ 
  - ▶ obtemos uma contradição com a definição de  $S^*$

## Mecanismo à prova de estratégia de grupo

Um mecanismo  $\mathcal{M}$  é à prova de estratégia de grupo se,

## Mecanismo à prova de estratégia de grupo

Um mecanismo  $\mathcal{M}$  é à prova de estratégia de grupo se,

- quando qualquer coalizão  $S$  mente e  $i \in S$  melhora

## Mecanismo à prova de estratégia de grupo

Um mecanismo  $\mathcal{M}$  é à prova de estratégia de grupo se,

- quando qualquer coalizão  $S$  mente e  $i \in S$  melhora
- então existe  $j \in S$  que piora

## Mecanismo à prova de estratégia de grupo

Um mecanismo  $\mathcal{M}$  é à prova de estratégia de grupo se,

- quando qualquer coalizão  $S$  mente e  $i \in S$  melhora
- então existe  $j \in S$  que piora

Formalmente, seja  $S \subseteq \mathcal{A}$



## Mecanismo à prova de estratégia de grupo

Um mecanismo  $\mathcal{M}$  é à prova de estratégia de grupo se,

- quando qualquer coalizão  $S$  mente e  $i \in S$  melhora
- então existe  $j \in S$  que piora

Formalmente, seja  $S \subseteq \mathcal{A}$

- Sejam  $v$  e  $b$  dois vetores de lances

## Mecanismo à prova de estratégia de grupo

Um mecanismo  $\mathcal{M}$  é à prova de estratégia de grupo se,

- quando qualquer coalizão  $S$  mente e  $i \in S$  melhora
- então existe  $j \in S$  que piora

Formalmente, seja  $S \subseteq \mathcal{A}$

- Sejam  $v$  e  $b$  dois vetores de lances
  - ▶  $v$  é quando todos falam a verdade e  $b$  é quando  $S$  mente

## Mecanismo à prova de estratégia de grupo

Um mecanismo  $\mathcal{M}$  é à prova de estratégia de grupo se,

- quando qualquer coalizão  $S$  mente e  $i \in S$  melhora
- então existe  $j \in S$  que piora

Formalmente, seja  $S \subseteq \mathcal{A}$

- Sejam  $v$  e  $b$  dois vetores de lances
  - ▶  $v$  é quando todos falam a verdade e  $b$  é quando  $S$  mente
  - ▶  $b_i = v_i$  se  $i \notin S$

## Mecanismo à prova de estratégia de grupo

Um mecanismo  $\mathcal{M}$  é à prova de estratégia de grupo se,

- quando qualquer coalizão  $S$  mente e  $i \in S$  melhora
- então existe  $j \in S$  que piora

Formalmente, seja  $S \subseteq \mathcal{A}$

- Sejam  $v$  e  $b$  dois vetores de lances
  - ▶  $v$  é quando todos falam a verdade e  $b$  é quando  $S$  mente
  - ▶  $b_i = v_i$  se  $i \notin S$
- Seja  $(Q, p)$  o resultado de  $\mathcal{M}$  para  $v$

## Mecanismo à prova de estratégia de grupo

Um mecanismo  $\mathcal{M}$  é à prova de estratégia de grupo se,

- quando qualquer coalizão  $S$  mente e  $i \in S$  melhora
- então existe  $j \in S$  que piora

Formalmente, seja  $S \subseteq \mathcal{A}$

- Sejam  $v$  e  $b$  dois vetores de lances
  - ▶  $v$  é quando todos falam a verdade e  $b$  é quando  $S$  mente
  - ▶  $b_i = v_i$  se  $i \notin S$
- Seja  $(Q, p)$  o resultado de  $\mathcal{M}$  para  $v$
- Seja  $(Q', p')$  o resultado de  $\mathcal{M}$  para  $b$

## Mecanismo à prova de estratégia de grupo

Um mecanismo  $\mathcal{M}$  é à prova de estratégia de grupo se,

- quando qualquer coalizão  $S$  mente e  $i \in S$  melhora
- então existe  $j \in S$  que piora

Formalmente, seja  $S \subseteq \mathcal{A}$

- Sejam  $v$  e  $b$  dois vetores de lances
  - ▶  $v$  é quando todos falam a verdade e  $b$  é quando  $S$  mente
  - ▶  $b_i = v_i$  se  $i \notin S$
- Seja  $(Q, p)$  o resultado de  $\mathcal{M}$  para  $v$
- Seja  $(Q', p')$  o resultado de  $\mathcal{M}$  para  $b$

Se o mecanismo é à prova de estratégia de grupos, temos que

## Mecanismo à prova de estratégia de grupo

Um mecanismo  $\mathcal{M}$  é à prova de estratégia de grupo se,

- quando qualquer coalizão  $S$  mente e  $i \in S$  melhora
- então existe  $j \in S$  que piora

Formalmente, seja  $S \subseteq \mathcal{A}$

- Sejam  $v$  e  $b$  dois vetores de lances
  - ▶  $v$  é quando todos falam a verdade e  $b$  é quando  $S$  mente
  - ▶  $b_i = v_i$  se  $i \notin S$
- Seja  $(Q, p)$  o resultado de  $\mathcal{M}$  para  $v$
- Seja  $(Q', p')$  o resultado de  $\mathcal{M}$  para  $b$

Se o mecanismo é à prova de estratégia de grupos, temos que

- Se  $v_i q'_i - p'_i \geq v_i q_i - p_i$  para todo  $i \in S$

## Mecanismo à prova de estratégia de grupo

Um mecanismo  $\mathcal{M}$  é à prova de estratégia de grupo se,

- quando qualquer coalizão  $S$  mente e  $i \in S$  melhora
- então existe  $j \in S$  que piora

Formalmente, seja  $S \subseteq \mathcal{A}$

- Sejam  $v$  e  $b$  dois vetores de lances
  - ▶  $v$  é quando todos falam a verdade e  $b$  é quando  $S$  mente
  - ▶  $b_i = v_i$  se  $i \notin S$
- Seja  $(Q, p)$  o resultado de  $\mathcal{M}$  para  $v$
- Seja  $(Q', p')$  o resultado de  $\mathcal{M}$  para  $b$

Se o mecanismo é à prova de estratégia de grupos, temos que

- Se  $v_i q'_i - p'_i \geq v_i q_i - p_i$  para todo  $i \in S$ 
  - ▶ então  $v_i q'_i - p'_i = v_i q_i - p_i$  para todo  $i \in S$



## Mecanismo à prova de estratégia de grupo

Um mecanismo  $\mathcal{M}$  é à prova de estratégia de grupo se,

- quando qualquer coalizão  $S$  mente e  $i \in S$  melhora
- então existe  $j \in S$  que piora

Formalmente, seja  $S \subseteq \mathcal{A}$

- Sejam  $v$  e  $b$  dois vetores de lances
  - ▶  $v$  é quando todos falam a verdade e  $b$  é quando  $S$  mente
  - ▶  $b_i = v_i$  se  $i \notin S$
- Seja  $(Q, p)$  o resultado de  $\mathcal{M}$  para  $v$
- Seja  $(Q', p')$  o resultado de  $\mathcal{M}$  para  $b$

Se o mecanismo é à prova de estratégia de grupos, temos que

- Se  $v_i q'_i - p'_i \geq v_i q_i - p_i$  para todo  $i \in S$ 
  - ▶ então  $v_i q'_i - p'_i = v_i q_i - p_i$  para todo  $i \in S$
- Analogamente, se existe  $i \in S$  tal que  $v_i q'_i - p'_i > v_i q_i - p_i$

## Mecanismo à prova de estratégia de grupo

Um mecanismo  $\mathcal{M}$  é à prova de estratégia de grupo se,

- quando qualquer coalizão  $S$  mente e  $i \in S$  melhora
- então existe  $j \in S$  que piora

Formalmente, seja  $S \subseteq \mathcal{A}$

- Sejam  $v$  e  $b$  dois vetores de lances
  - ▶  $v$  é quando todos falam a verdade e  $b$  é quando  $S$  mente
  - ▶  $b_i = v_i$  se  $i \notin S$
- Seja  $(Q, p)$  o resultado de  $\mathcal{M}$  para  $v$
- Seja  $(Q', p')$  o resultado de  $\mathcal{M}$  para  $b$

Se o mecanismo é à prova de estratégia de grupos, temos que

- Se  $v_i q'_i - p'_i \geq v_i q_i - p_i$  para todo  $i \in S$ 
  - ▶ então  $v_i q'_i - p'_i = v_i q_i - p_i$  para todo  $i \in S$
- Analogamente, se existe  $i \in S$  tal que  $v_i q'_i - p'_i > v_i q_i - p_i$ 
  - ▶ então existe  $j \in S$  tal que  $v_j q'_j - p'_j < v_j q_j - p_j$

# $\mathcal{M}_\xi$ é à prova de estratégia de grupo

Teorema:  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

# $\mathcal{M}_\xi$ é à prova de estratégia de grupo

Teorema:  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

Prova:

## $\mathcal{M}_\xi$ é à prova de estratégia de grupo

**Teorema:**  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

**Prova:** Suponha que existe uma coalizão  $T$  que se beneficia ao mentir

## $\mathcal{M}_\xi$ é à prova de estratégia de grupo

**Teorema:**  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

**Prova:** Suponha que existe uma coalizão  $T$  que se beneficia ao mentir

- $T^+$  são os que dão lances maiores do que  $v_i$  e

## $\mathcal{M}_\xi$ é à prova de estratégia de grupo

**Teorema:**  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

**Prova:** Suponha que existe uma coalizão  $T$  que se beneficia ao mentir

- $T^+$  são os que dão lances maiores do que  $v_i$  e
- $T^- = T \setminus T^+$

## $\mathcal{M}_\xi$ é à prova de estratégia de grupo

**Teorema:**  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

**Prova:** Suponha que existe uma coalizão  $T$  que se beneficia ao mentir

- $T^+$  são os que dão lances maiores do que  $v_i$  e
- $T^- = T \setminus T^+$

Primeiro, vamos mostrar que, s.p.g.,  $T^+ = \emptyset$



## $\mathcal{M}_\xi$ é à prova de estratégia de grupo

**Teorema:**  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

**Prova:** Suponha que existe uma coalizão  $T$  que se beneficia ao mentir

- $T^+$  são os que dão lances maiores do que  $v_i$  e
- $T^- = T \setminus T^+$

Primeiro, vamos mostrar que, s.p.g.,  $T^+ = \emptyset$

- reduza os lances de  $i \in T^+$  de  $b_i$  para  $u_i$  um-a-um

## $\mathcal{M}_\xi$ é à prova de estratégia de grupo

**Teorema:**  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

**Prova:** Suponha que existe uma coalizão  $T$  que se beneficia ao mentir

- $T^+$  são os que dão lances maiores do que  $v_i$  e
- $T^- = T \setminus T^+$

Primeiro, vamos mostrar que, s.p.g.,  $T^+ = \emptyset$

- reduza os lances de  $i \in T^+$  de  $b_i$  para  $u_i$  um-a-um
  - ▶ até que o resultado  $(Q', p')$  do leilão mude para  $(Q'', p'')$

## $\mathcal{M}_\xi$ é à prova de estratégia de grupo

**Teorema:**  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

**Prova:** Suponha que existe uma coalizão  $T$  que se beneficia ao mentir

- $T^+$  são os que dão lances maiores do que  $v_i$  e
- $T^- = T \setminus T^+$

Primeiro, vamos mostrar que, s.p.g.,  $T^+ = \emptyset$

- reduza os lances de  $i \in T^+$  de  $b_i$  para  $u_i$  um-a-um
  - ▶ até que o resultado  $(Q', p')$  do leilão mude para  $(Q'', p'')$
- note que  $i \in Q'$ , pois caso contrário o resultado não muda

## $\mathcal{M}_\xi$ é à prova de estratégia de grupo

**Teorema:**  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

**Prova:** Suponha que existe uma coalizão  $T$  que se beneficia ao mentir

- $T^+$  são os que dão lances maiores do que  $v_i$  e
- $T^- = T \setminus T^+$

Primeiro, vamos mostrar que, s.p.g.,  $T^+ = \emptyset$

- reduza os lances de  $i \in T^+$  de  $b_i$  para  $u_i$  um-a-um
  - ▶ até que o resultado  $(Q', p')$  do leilão mude para  $(Q'', p'')$
- note que  $i \in Q'$ , pois caso contrário o resultado não muda
- e  $i \notin Q''$ , caso contrário,  $p'_i = p''_i$

## $\mathcal{M}_\xi$ é à prova de estratégia de grupo

**Teorema:**  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

**Prova:** Suponha que existe uma coalizão  $T$  que se beneficia ao mentir

- $T^+$  são os que dão lances maiores do que  $v_i$  e
- $T^- = T \setminus T^+$

Primeiro, vamos mostrar que, s.p.g.,  $T^+ = \emptyset$

- reduza os lances de  $i \in T^+$  de  $b_i$  para  $u_i$  um-a-um
  - ▶ até que o resultado  $(Q', p')$  do leilão mude para  $(Q'', p'')$
- note que  $i \in Q'$ , pois caso contrário o resultado não muda
- e  $i \notin Q''$ , caso contrário,  $p'_i = p''_i$
- concluímos que  $\xi(i, Q') > v_i$

## $\mathcal{M}_\xi$ é à prova de estratégia de grupo

**Teorema:**  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

**Prova:** Suponha que existe uma coalizão  $T$  que se beneficia ao mentir

- $T^+$  são os que dão lances maiores do que  $v_i$  e
- $T^- = T \setminus T^+$

Primeiro, vamos mostrar que, s.p.g.,  $T^+ = \emptyset$

- reduza os lances de  $i \in T^+$  de  $b_i$  para  $u_i$  um-a-um
  - ▶ até que o resultado  $(Q', p')$  do leilão mude para  $(Q'', p'')$
- note que  $i \in Q'$ , pois caso contrário o resultado não muda
- e  $i \notin Q''$ , caso contrário,  $p'_i = p''_i$
- concluímos que  $\xi(i, Q') > v_i$ 
  - ▶ então  $v_i q'_i - p'_i = v_i - \xi(i, Q') < 0$  e  $i$  não se beneficia

# $\mathcal{M}_\xi$ é à prova de estratégia de grupo

Teorema:  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

# $\mathcal{M}_\xi$ é à prova de estratégia de grupo

Teorema:  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

Prova:



## $\mathcal{M}_\xi$ é à prova de estratégia de grupo

**Teorema:**  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

**Prova:** Suponha que existe uma coalizão  $T$  que se beneficia ao mentir

## $\mathcal{M}_\xi$ é à prova de estratégia de grupo

**Teorema:**  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

**Prova:** Suponha que existe uma coalizão  $T$  que se beneficia ao mentir

- $T^+$  são os que dão lances maiores do que  $v_i$  e

## $\mathcal{M}_\xi$ é à prova de estratégia de grupo

**Teorema:**  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

**Prova:** Suponha que existe uma coalizão  $T$  que se beneficia ao mentir

- $T^+$  são os que dão lances maiores do que  $v_i$  e
- $T^- = T \setminus T^+$

## $\mathcal{M}_\xi$ é à prova de estratégia de grupo

**Teorema:**  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

**Prova:** Suponha que existe uma coalizão  $T$  que se beneficia ao mentir

- $T^+$  são os que dão lances maiores do que  $v_i$  e
- $T^- = T \setminus T^+$
- $(Q, p)$  resultado para  $v$

## $\mathcal{M}_\xi$ é à prova de estratégia de grupo

**Teorema:**  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

**Prova:** Suponha que existe uma coalizão  $T$  que se beneficia ao mentir

- $T^+$  são os que dão lances maiores do que  $v_i$  e
- $T^- = T \setminus T^+$
- $(Q, p)$  resultado para  $v$
- $(Q', p')$  resultado para  $b$

## $\mathcal{M}_\xi$ é à prova de estratégia de grupo

**Teorema:**  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

**Prova:** Suponha que existe uma coalizão  $T$  que se beneficia ao mentir

- $T^+$  são os que dão lances maiores do que  $v_i$  e
- $T^- = T \setminus T^+$
- $(Q, p)$  resultado para  $v$
- $(Q', p')$  resultado para  $b$
- Como s.p.g.  $T^+ = \emptyset$ , temos que  $Q' \subseteq Q$

## $\mathcal{M}_\xi$ é à prova de estratégia de grupo

**Teorema:**  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

**Prova:** Suponha que existe uma coalizão  $T$  que se beneficia ao mentir

- $T^+$  são os que dão lances maiores do que  $v_i$  e
- $T^- = T \setminus T^+$
- $(Q, p)$  resultado para  $v$
- $(Q', p')$  resultado para  $b$
- Como s.p.g.  $T^+ = \emptyset$ , temos que  $Q' \subseteq Q$ 
  - ▶  $\mathcal{M}_\epsilon$  devolve  $Q$  único maximal tal que  $b_i \geq \xi(i, Q)$

## $\mathcal{M}_\xi$ é à prova de estratégia de grupo

**Teorema:**  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

**Prova:** Suponha que existe uma coalizão  $T$  que se beneficia ao mentir

- $T^+$  são os que dão lances maiores do que  $v_i$  e
- $T^- = T \setminus T^+$
- $(Q, p)$  resultado para  $v$
- $(Q', p')$  resultado para  $b$
- Como s.p.g.  $T^+ = \emptyset$ , temos que  $Q' \subseteq Q$ 
  - ▶  $\mathcal{M}_\epsilon$  devolve  $Q$  único maximal tal que  $b_i \geq \xi(i, Q)$

$v_i$  é uma resposta ótima para  $i \in T^-$ ?



## $\mathcal{M}_\xi$ é à prova de estratégia de grupo

**Teorema:**  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

**Prova:** Suponha que existe uma coalizão  $T$  que se beneficia ao mentir

- $T^+$  são os que dão lances maiores do que  $v_i$  e
- $T^- = T \setminus T^+$
- $(Q, p)$  resultado para  $v$
- $(Q', p')$  resultado para  $b$
- Como s.p.g.  $T^+ = \emptyset$ , temos que  $Q' \subseteq Q$ 
  - ▶  $\mathcal{M}_\epsilon$  devolve  $Q$  único maximal tal que  $b_i \geq \xi(i, Q)$

$v_i$  é uma resposta ótima para  $i \in T^-$ ?

- Se  $i \in Q'$ , então reportar  $v_i$  não aumenta o preço a ser pago, já que  $\xi$  é monotônico cruzado.

## $\mathcal{M}_\xi$ é à prova de estratégia de grupo

**Teorema:**  $\mathcal{M}_\xi$  é à prova de estratégia de grupo

**Prova:** Suponha que existe uma coalizão  $T$  que se beneficia ao mentir

- $T^+$  são os que dão lances maiores do que  $v_i$  e
- $T^- = T \setminus T^+$
- $(Q, p)$  resultado para  $v$
- $(Q', p')$  resultado para  $b$
- Como s.p.g.  $T^+ = \emptyset$ , temos que  $Q' \subseteq Q$ 
  - ▶  $\mathcal{M}_\epsilon$  devolve  $Q$  único maximal tal que  $b_i \geq \xi(i, Q)$

$v_i$  é uma resposta ótima para  $i \in T^-$ ?

- Se  $i \in Q'$ , então reportar  $v_i$  não aumenta o preço a ser pago, já que  $\xi$  é monotônico cruzado.
- Se  $i \notin Q'$ , sua utilidade é **zero** e reportar  $v_i$  não pode piorar a utilidade de  $i$

# Compartilhamento de custos e mecanismo

Se tivermos um esquema de compartilhamento de custos  $\xi$

# Compartilhamento de custos e mecanismo

Se tivermos um esquema de compartilhamento de custos  $\xi$

- monotônico cruzado

# Compartilhamento de custos e mecanismo

Se tivermos um esquema de compartilhamento de custos  $\xi$

- monotônico cruzado
- $\gamma$ -orçamento balanceado

# Compartilhamento de custos e mecanismo

Se tivermos um esquema de compartilhamento de custos  $\xi$

- monotônico cruzado
- $\gamma$ -orçamento balanceado

Então, o mecanismo  $\mathcal{M}_\xi$  é

# Compartilhamento de custos e mecanismo

Se tivermos um esquema de compartilhamento de custos  $\xi$

- monotônico cruzado
- $\gamma$ -orçamento balanceado

Então, o mecanismo  $\mathcal{M}_\xi$  é

- $\gamma$ -orçamento balanceado

# Compartilhamento de custos e mecanismo

Se tivermos um esquema de compartilhamento de custos  $\xi$

- monotônico cruzado
- $\gamma$ -orçamento balanceado

Então, o mecanismo  $\mathcal{M}_\xi$  é

- $\gamma$ -orçamento balanceado
- à prova de estratégia de grupos



# Compartilhamento de custos e mecanismo

Se tivermos um esquema de compartilhamento de custos  $\xi$

- monotônico cruzado
- $\gamma$ -orçamento balanceado

Então, o mecanismo  $\mathcal{M}_\xi$  é

- $\gamma$ -orçamento balanceado
- à prova de estratégia de grupos

Basta então encontrar um esquema de compartilhamento de custos  $\xi$  com tais propriedades

# Compartilhamento de custos e mecanismo

Se tivermos um esquema de compartilhamento de custos  $\xi$

- monotônico cruzado
- $\gamma$ -orçamento balanceado

Então, o mecanismo  $\mathcal{M}_\xi$  é

- $\gamma$ -orçamento balanceado
- à prova de estratégia de grupos

Basta então encontrar um esquema de compartilhamento de custos  $\xi$  com tais propriedades

- com  $\gamma$  que conseguirmos

## Voltando a Localização de Instalações

Vamos ver como obter um esquema de compartilhamento de custos monotônico cruzado  $1/3$ -orçamento balanceado para o Jogo da Localização de Instalações

## Voltando a Localização de Instalações

Vamos ver como obter um esquema de compartilhamento de custos monotônico cruzado  $1/3$ -orçamento balanceado para o Jogo da Localização de Instalações

Vamos usar um algoritmo primal-dual para esse jogo

## Voltando a Localização de Instalações

Vamos ver como obter um esquema de compartilhamento de custos monotônico cruzado  $1/3$ -orçamento balanceado para o Jogo da Localização de Instalações

Vamos usar um algoritmo primal-dual para esse jogo

- dado  $S \subseteq \mathcal{A}$

## Voltando a Localização de Instalações

Vamos ver como obter um esquema de compartilhamento de custos monotônico cruzado  $1/3$ -orçamento balanceado para o Jogo da Localização de Instalações

Vamos usar um algoritmo primal-dual para esse jogo

- dado  $S \subseteq \mathcal{A}$
- encontrar uma solução viável (primal) para  $S$

## Voltando a Localização de Instalações

Vamos ver como obter um esquema de compartilhamento de custos monotônico cruzado  $1/3$ -orçamento balanceado para o Jogo da Localização de Instalações

Vamos usar um algoritmo primal-dual para esse jogo

- dado  $S \subseteq \mathcal{A}$
- encontrar uma solução viável (primal) para  $S$
- e uma alocação de custos (dual) para  $S$

## Voltando a Localização de Instalações

Vamos ver como obter um esquema de compartilhamento de custos monotônico cruzado  $1/3$ -orçamento balanceado para o Jogo da Localização de Instalações

Vamos usar um algoritmo primal-dual para esse jogo

- dado  $S \subseteq \mathcal{A}$
- encontrar uma solução viável (primal) para  $S$
- e uma alocação de custos (dual) para  $S$

Faremos também um pós-processamento para abrir menos instalações



## Voltando a Localização de Instalações

Vamos ver como obter um esquema de compartilhamento de custos monotônico cruzado  $1/3$ -orçamento balanceado para o Jogo da Localização de Instalações

Vamos usar um algoritmo primal-dual para esse jogo

- dado  $S \subseteq \mathcal{A}$
- encontrar uma solução viável (primal) para  $S$
- e uma alocação de custos (dual) para  $S$

Faremos também um pós-processamento para abrir menos instalações

- Assim, o custo pago será  $1/3$  do valor da solução

# Relação entre Primal e Dual

Primal:  $\min \sum_{i \in \mathcal{F}} f_i x_i + \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{A}} d_{ij} y_{ij}$   
tal que  $\sum_{i \in \mathcal{F}} y_{ij} \geq 1, \quad \forall j \in \mathcal{A}$   
 $x_i \geq y_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$   
 $x_i \in \{0, 1\}, \quad \forall i \in \mathcal{F}$   
 $y_{ij} \in \{0, 1\}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$

Dual:  $\max \sum_{j \in \mathcal{A}} \alpha_j$   
tal que  $\beta_{ij} \geq \alpha_j - d_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$   
 $\sum_{j \in \mathcal{A}} \beta_{ij} \leq f_i, \quad \forall i \in \mathcal{F}$   
 $\alpha_j \geq 0, \quad \forall j \in \mathcal{A}$   
 $\beta_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$

# Relação entre Primal e Dual

Primal:  $\min \sum_{i \in \mathcal{F}} f_i x_i + \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{A}} d_{ij} y_{ij}$   
tal que  $\sum_{i \in \mathcal{F}} y_{ij} \geq 1, \quad \forall j \in \mathcal{A}$   
 $x_i \geq y_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$   
 $x_i \in \{0, 1\}, \quad \forall i \in \mathcal{F}$   
 $y_{ij} \in \{0, 1\}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$

Dual:  $\max \sum_{j \in \mathcal{A}} \alpha_j$   
tal que  $\beta_{ij} \geq \alpha_j - d_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$   
 $\sum_{j \in \mathcal{A}} \beta_{ij} \leq f_i, \quad \forall i \in \mathcal{F}$   
 $\alpha_j \geq 0, \quad \forall j \in \mathcal{A}$   
 $\beta_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$

Folgas complementares:

# Relação entre Primal e Dual

Primal:  $\min \sum_{i \in \mathcal{F}} f_i x_i + \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{A}} d_{ij} y_{ij}$   
tal que  $\sum_{i \in \mathcal{F}} y_{ij} \geq 1, \quad \forall j \in \mathcal{A}$   
 $x_i \geq y_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$   
 $x_i \in \{0, 1\}, \quad \forall i \in \mathcal{F}$   
 $y_{ij} \in \{0, 1\}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$

Dual:  $\max \sum_{j \in \mathcal{A}} \alpha_j$   
tal que  $\beta_{ij} \geq \alpha_j - d_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$   
 $\sum_{j \in \mathcal{A}} \beta_{ij} \leq f_i, \quad \forall i \in \mathcal{F}$   
 $\alpha_j \geq 0, \quad \forall j \in \mathcal{A}$   
 $\beta_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$

Folgas complementares:

- Quando  $x_i > 0$ , temos que  $\sum_{j \in \mathcal{A}} \beta_{ij} = f_i$

# Relação entre Primal e Dual

Primal:  $\min \sum_{i \in \mathcal{F}} f_i x_i + \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{A}} d_{ij} y_{ij}$   
tal que  $\sum_{i \in \mathcal{F}} y_{ij} \geq 1, \quad \forall j \in \mathcal{A}$   
 $x_i \geq y_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$   
 $x_i \in \{0, 1\}, \quad \forall i \in \mathcal{F}$   
 $y_{ij} \in \{0, 1\}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$

Dual:  $\max \sum_{j \in \mathcal{A}} \alpha_j$   
tal que  $\beta_{ij} \geq \alpha_j - d_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$   
 $\sum_{j \in \mathcal{A}} \beta_{ij} \leq f_i, \quad \forall i \in \mathcal{F}$   
 $\alpha_j \geq 0, \quad \forall j \in \mathcal{A}$   
 $\beta_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$

Folgas complementares:

- Quando  $x_i > 0$ , temos que  $\sum_{j \in \mathcal{A}} \beta_{ij} = f_i$
- Quando  $y_{ij} > 0$ , temos que  $\beta_{ij} = \alpha_j - d_{ij}$

# Relação entre Primal e Dual

$$\begin{aligned} \text{Dual:} \quad & \max \sum_{j \in \mathcal{A}} \alpha_j \\ \text{tal que} \quad & \beta_{ij} \geq \alpha_j - d_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \\ & \sum_{j \in \mathcal{A}} \beta_{ij} \leq f_i, \quad \forall i \in \mathcal{F} \\ & \alpha_j \geq 0, \quad \forall j \in \mathcal{A} \\ & \beta_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \end{aligned}$$

Folgas complementares:

# Relação entre Primal e Dual

$$\begin{aligned} \text{Dual:} \quad & \max \sum_{j \in \mathcal{A}} \alpha_j \\ \text{tal que} \quad & \beta_{ij} \geq \alpha_j - d_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \\ & \sum_{j \in \mathcal{A}} \beta_{ij} \leq f_i, \quad \forall i \in \mathcal{F} \\ & \alpha_j \geq 0, \quad \forall j \in \mathcal{A} \\ & \beta_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \end{aligned}$$

Folgas complementares:

- Quando  $x_i > 0$ , temos que  $\sum_{j \in \mathcal{A}} \beta_{ij} = f_i$

# Relação entre Primal e Dual

Dual:  $\max \sum_{j \in \mathcal{A}} \alpha_j$   
tal que  $\beta_{ij} \geq \alpha_j - d_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$   
 $\sum_{j \in \mathcal{A}} \beta_{ij} \leq f_i, \quad \forall i \in \mathcal{F}$   
 $\alpha_j \geq 0, \quad \forall j \in \mathcal{A}$   
 $\beta_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$

Folgas complementares:

- Quando  $x_i > 0$ , temos que  $\sum_{j \in \mathcal{A}} \beta_{ij} = f_i$
- Quando  $y_{ij} > 0$ , temos que  $\beta_{ij} = \alpha_j - d_{ij}$



# Relação entre Primal e Dual

$$\begin{aligned} \text{Dual:} \quad & \max \sum_{j \in \mathcal{A}} \alpha_j \\ \text{tal que} \quad & \beta_{ij} \geq \alpha_j - d_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \\ & \sum_{j \in \mathcal{A}} \beta_{ij} \leq f_i, \quad \forall i \in \mathcal{F} \\ & \alpha_j \geq 0, \quad \forall j \in \mathcal{A} \\ & \beta_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \end{aligned}$$

Folgas complementares:

- Quando  $x_i > 0$ , temos que  $\sum_{j \in \mathcal{A}} \beta_{ij} = f_i$
- Quando  $y_{ij} > 0$ , temos que  $\beta_{ij} = \alpha_j - d_{ij}$

S.p.g.,  $\beta_{ij} = \max\{0, \alpha_j - d_{ij}\}$

# Relação entre Primal e Dual

Dual:  $\max \sum_{j \in \mathcal{A}} \alpha_j$   
tal que  $\beta_{ij} \geq \alpha_j - d_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$   
 $\sum_{j \in \mathcal{A}} \beta_{ij} \leq f_i, \quad \forall i \in \mathcal{F}$   
 $\alpha_j \geq 0, \quad \forall j \in \mathcal{A}$   
 $\beta_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F}$

Folgas complementares:

- Quando  $x_i > 0$ , temos que  $\sum_{j \in \mathcal{A}} \beta_{ij} = f_i$
- Quando  $y_{ij} > 0$ , temos que  $\beta_{ij} = \alpha_j - d_{ij}$

S.p.g.,  $\beta_{ij} = \max\{0, \alpha_j - d_{ij}\}$

- $\alpha_j$  é quanto o jogador  $j$  paga no total

# Relação entre Primal e Dual

$$\begin{aligned} \text{Dual:} \quad & \max \sum_{j \in \mathcal{A}} \alpha_j \\ \text{tal que} \quad & \beta_{ij} \geq \alpha_j - d_{ij}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \\ & \sum_{j \in \mathcal{A}} \beta_{ij} \leq f_i, \quad \forall i \in \mathcal{F} \\ & \alpha_j \geq 0, \quad \forall j \in \mathcal{A} \\ & \beta_{ij} \geq 0, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{F} \end{aligned}$$

Folgas complementares:

- Quando  $x_i > 0$ , temos que  $\sum_{j \in \mathcal{A}} \beta_{ij} = f_i$
- Quando  $y_{ij} > 0$ , temos que  $\beta_{ij} = \alpha_j - d_{ij}$

S.p.g.,  $\beta_{ij} = \max\{0, \alpha_j - d_{ij}\}$

- $\alpha_j$  é quanto o jogador  $j$  paga no total
- $\beta_{ij}$  é quanto ele paga para ajudar a abrir a instalação  $i$

# Ideia do Algoritmo

Folgas complementares:

# Ideia do Algoritmo

Folgas complementares:

- Quando  $x_i > 0$ , temos que  $\sum_{j \in \mathcal{A}} \beta_{ij} = f_i$

# Ideia do Algoritmo

Folgas complementares:

- Quando  $x_i > 0$ , temos que  $\sum_{j \in \mathcal{A}} \beta_{ij} = f_i$
- Quando  $y_{ij} > 0$ , temos que  $\beta_{ij} = \alpha_j - d_{ij}$

# Ideia do Algoritmo

Folgas complementares:

- Quando  $x_i > 0$ , temos que  $\sum_{j \in \mathcal{A}} \beta_{ij} = f_i$
- Quando  $y_{ij} > 0$ , temos que  $\beta_{ij} = \alpha_j - d_{ij}$

Ideia do algoritmo:

# Ideia do Algoritmo

Folgas complementares:

- Quando  $x_i > 0$ , temos que  $\sum_{j \in \mathcal{A}} \beta_{ij} = f_i$
- Quando  $y_{ij} > 0$ , temos que  $\beta_{ij} = \alpha_j - d_{ij}$

Ideia do algoritmo:

1.  $\alpha_j = 0$  para todos os jogadores



# Ideia do Algoritmo

Folgas complementares:

- Quando  $x_i > 0$ , temos que  $\sum_{j \in \mathcal{A}} \beta_{ij} = f_i$
- Quando  $y_{ij} > 0$ , temos que  $\beta_{ij} = \alpha_j - d_{ij}$

Ideia do algoritmo:

1.  $\alpha_j = 0$  para todos os jogadores
2. Aumentamos os  $\alpha_j$  do jogadores não conectados na mesma velocidade até que

# Ideia do Algoritmo

Folgas complementares:

- Quando  $x_i > 0$ , temos que  $\sum_{j \in \mathcal{A}} \beta_{ij} = f_i$
- Quando  $y_{ij} > 0$ , temos que  $\beta_{ij} = \alpha_j - d_{ij}$

Ideia do algoritmo:

1.  $\alpha_j = 0$  para todos os jogadores
2. Aumentamos os  $\alpha_j$  do jogadores não conectados na mesma velocidade até que
  - ▶ ou  $\sum_{j \in \mathcal{A}} \max\{0, \alpha_j - d_{ij}\} = f_i$

# Ideia do Algoritmo

Folgas complementares:

- Quando  $x_i > 0$ , temos que  $\sum_{j \in \mathcal{A}} \beta_{ij} = f_i$
- Quando  $y_{ij} > 0$ , temos que  $\beta_{ij} = \alpha_j - d_{ij}$

Ideia do algoritmo:

1.  $\alpha_j = 0$  para todos os jogadores
2. Aumentamos os  $\alpha_j$  do jogadores não conectados na mesma velocidade até que
  - ▶ ou  $\sum_{j \in \mathcal{A}} \max\{0, \alpha_j - d_{ij}\} = f_i$ 
    - abrimos a  $i$  e conectamos todos os que contribuíram a  $i$

# Ideia do Algoritmo

Folgas complementares:

- Quando  $x_i > 0$ , temos que  $\sum_{j \in \mathcal{A}} \beta_{ij} = f_i$
- Quando  $y_{ij} > 0$ , temos que  $\beta_{ij} = \alpha_j - d_{ij}$

Ideia do algoritmo:

1.  $\alpha_j = 0$  para todos os jogadores
2. Aumentamos os  $\alpha_j$  do jogadores não conectados na mesma velocidade até que
  - ▶ ou  $\sum_{j \in \mathcal{A}} \max\{0, \alpha_j - d_{ij}\} = f_i$ 
    - abrimos a  $i$  e conectamos todos os que contribuíram a  $i$
  - ▶ ou  $\alpha_j = d_{ij}$  e  $i$  já está aberta

# Ideia do Algoritmo

Folgas complementares:

- Quando  $x_i > 0$ , temos que  $\sum_{j \in \mathcal{A}} \beta_{ij} = f_i$
- Quando  $y_{ij} > 0$ , temos que  $\beta_{ij} = \alpha_j - d_{ij}$

Ideia do algoritmo:

1.  $\alpha_j = 0$  para todos os jogadores
2. Aumentamos os  $\alpha_j$  do jogadores não conectados na mesma velocidade até que
  - ▶ ou  $\sum_{j \in \mathcal{A}} \max\{0, \alpha_j - d_{ij}\} = f_i$ 
    - abrimos a  $i$  e conectamos todos os que contribuíram a  $i$
  - ▶ ou  $\alpha_j = d_{ij}$  e  $i$  já está aberta
    - conectamos  $j$  a  $i$

# Ideia do Algoritmo

Folgas complementares:

- Quando  $x_i > 0$ , temos que  $\sum_{j \in \mathcal{A}} \beta_{ij} = f_i$
- Quando  $y_{ij} > 0$ , temos que  $\beta_{ij} = \alpha_j - d_{ij}$

Ideia do algoritmo:

1.  $\alpha_j = 0$  para todos os jogadores
2. Aumentamos os  $\alpha_j$  do jogadores não conectados na mesma velocidade até que
  - ▶ ou  $\sum_{j \in \mathcal{A}} \max\{0, \alpha_j - d_{ij}\} = f_i$ 
    - abrimos a  $i$  e conectamos todos os que contribuíram a  $i$
  - ▶ ou  $\alpha_j = d_{ij}$  e  $i$  já está aberta
    - conectamos  $j$  a  $i$
3. Se houver jogadores não conectados volte para 2

# Ideia do Algoritmo

Folgas complementares:

- Quando  $x_i > 0$ , temos que  $\sum_{j \in \mathcal{A}} \beta_{ij} = f_i$
- Quando  $y_{ij} > 0$ , temos que  $\beta_{ij} = \alpha_j - d_{ij}$

Ideia do algoritmo:

1.  $\alpha_j = 0$  para todos os jogadores
2. Aumentamos os  $\alpha_j$  do jogadores não conectados na mesma velocidade até que
  - ▶ ou  $\sum_{j \in \mathcal{A}} \max\{0, \alpha_j - d_{ij}\} = f_i$ 
    - abrimos a  $i$  e conectamos todos os que contribuíram a  $i$
  - ▶ ou  $\alpha_j = d_{ij}$  e  $i$  já está aberta
    - conectamos  $j$  a  $i$
3. Se houver jogadores não conectados volte para 2

O algoritmo tem também custos fantasmas  $\alpha_j$ 's

# Ideia do Algoritmo

Folgas complementares:

- Quando  $x_i > 0$ , temos que  $\sum_{j \in \mathcal{A}} \beta_{ij} = f_i$
- Quando  $y_{ij} > 0$ , temos que  $\beta_{ij} = \alpha_j - d_{ij}$

Ideia do algoritmo:

1.  $\alpha_j = 0$  para todos os jogadores
2. Aumentamos os  $\alpha_j$  do jogadores não conectados na mesma velocidade até que
  - ▶ ou  $\sum_{j \in \mathcal{A}} \max\{0, \alpha_j - d_{ij}\} = f_i$ 
    - abrimos a  $i$  e conectamos todos os que contribuíram a  $i$
  - ▶ ou  $\alpha_j = d_{ij}$  e  $i$  já está aberta
    - conectamos  $j$  a  $i$
3. Se houver jogadores não conectados volte para 2

O algoritmo tem também custos fantasmas  $\alpha_j$ 's

- ajudarão a pagar por instalações



# Ideia do Algoritmo

Folgas complementares:

- Quando  $x_i > 0$ , temos que  $\sum_{j \in \mathcal{A}} \beta_{ij} = f_i$
- Quando  $y_{ij} > 0$ , temos que  $\beta_{ij} = \alpha_j - d_{ij}$

Ideia do algoritmo:

1.  $\alpha_j = 0$  para todos os jogadores
2. Aumentamos os  $\alpha_j$  do jogadores não conectados na mesma velocidade até que
  - ▶ ou  $\sum_{j \in \mathcal{A}} \max\{0, \alpha_j - d_{ij}\} = f_i$ 
    - abrimos a  $i$  e conectamos todos os que contribuíram a  $i$
  - ▶ ou  $\alpha_j = d_{ij}$  e  $i$  já está aberta
    - conectamos  $j$  a  $i$
3. Se houver jogadores não conectados volte para 2

O algoritmo tem também custos fantasmas  $\alpha_j$ 's

- ajudarão a pagar por instalações
- sem que o jogador tenha que pagar de fato

# Algoritmo

# Algoritmo

FLCostShare( $S$ )

- 1  $\alpha_j \leftarrow 0$  para todo  $j \in S$
- 2  $\alpha'_j \leftarrow 0$  para todo  $j \in S$
- 3  $F \leftarrow \emptyset$
- 4 **enquanto**  $S \setminus F \neq \emptyset$
- 5     Aumente  $\alpha_j$ 's para  $j \in T \setminus F$  e  $\alpha'_j$ 's para  $j \in T$  igualmente até que
- 6     ou para uma instalação não aberta  $i$ ,  $\sum_{j \in T} \max(0, \alpha'_j - d_{ij}) = f_i$
- 7     abra a instalação  $i$
- 8     adicione todo jogador  $j$  com contribuição positiva para  $i$  a  $F$
- 9     ou para uma instalação aberta e um jogador  $j$ ,  $\alpha_j = d_{ij}$
- 10    adicione  $j$  a  $F$

# Algoritmo

FLCostShare( $S$ )

- 1  $\alpha_j \leftarrow 0$  para todo  $j \in S$
- 2  $\alpha'_j \leftarrow 0$  para todo  $j \in S$
- 3  $F \leftarrow \emptyset$
- 4 **enquanto**  $S \setminus F \neq \emptyset$
- 5     Aumente  $\alpha_j$ 's para  $j \in T \setminus F$  e  $\alpha'_j$ 's para  $j \in T$  igualmente até que
- 6     ou para uma instalação não aberta  $i$ ,  $\sum_{j \in T} \max(0, \alpha'_j - d_{ij}) = f_i$
- 7     abra a instalação  $i$
- 8     adicione todo jogador  $j$  com contribuição positiva para  $i$  a  $F$
- 9     ou para uma instalação aberta e um jogador  $j$ ,  $\alpha_j = d_{ij}$
- 10    adicione  $j$  a  $F$

Pós-processamento:

# Algoritmo

FLCostShare( $S$ )

- 1  $\alpha_j \leftarrow 0$  para todo  $j \in S$
- 2  $\alpha'_j \leftarrow 0$  para todo  $j \in S$
- 3  $F \leftarrow \emptyset$
- 4 **enquanto**  $S \setminus F \neq \emptyset$
- 5     Aumente  $\alpha_j$ 's para  $j \in T \setminus F$  e  $\alpha'_j$ 's para  $j \in T$  igualmente até que
- 6     ou para uma instalação não aberta  $i$ ,  $\sum_{j \in T} \max(0, \alpha'_j - d_{ij}) = f_i$
- 7     abra a instalação  $i$
- 8     adicione todo jogador  $j$  com contribuição positiva para  $i$  a  $F$
- 9     ou para uma instalação aberta e um jogador  $j$ ,  $\alpha_j = d_{ij}$
- 10    adicione  $j$  a  $F$

Pós-processamento:

- Seja  $t_i$  o “tempo” no qual a instalação  $i$  é aberta

# Algoritmo

FLCostShare( $S$ )

- 1  $\alpha_j \leftarrow 0$  para todo  $j \in S$
- 2  $\alpha'_j \leftarrow 0$  para todo  $j \in S$
- 3  $F \leftarrow \emptyset$
- 4 **enquanto**  $S \setminus F \neq \emptyset$
- 5     Aumente  $\alpha_j$ 's para  $j \in T \setminus F$  e  $\alpha'_j$ 's para  $j \in T$  igualmente até que
- 6     ou para uma instalação não aberta  $i$ ,  $\sum_{j \in T} \max(0, \alpha'_j - d_{ij}) = f_i$
- 7     abra a instalação  $i$
- 8     adicione todo jogador  $j$  com contribuição positiva para  $i$  a  $F$
- 9     ou para uma instalação aberta e um jogador  $j$ ,  $\alpha_j = d_{ij}$
- 10    adicione  $j$  a  $F$

Pós-processamento:

- Seja  $t_i$  o “tempo” no qual a instalação  $i$  é aberta
- Percorra as instalações abertas na ordem crescente de  $t_i$

# Algoritmo

FLCostShare( $S$ )

- 1  $\alpha_j \leftarrow 0$  para todo  $j \in S$
- 2  $\alpha'_j \leftarrow 0$  para todo  $j \in S$
- 3  $F \leftarrow \emptyset$
- 4 **enquanto**  $S \setminus F \neq \emptyset$
- 5     Aumente  $\alpha_j$ 's para  $j \in T \setminus F$  e  $\alpha'_j$ 's para  $j \in T$  igualmente até que
- 6     ou para uma instalação não aberta  $i$ ,  $\sum_{j \in T} \max(0, \alpha'_j - d_{ij}) = f_i$
- 7     abra a instalação  $i$
- 8     adicione todo jogador  $j$  com contribuição positiva para  $i$  a  $F$
- 9     ou para uma instalação aberta e um jogador  $j$ ,  $\alpha_j = d_{ij}$
- 10    adicione  $j$  a  $F$

Pós-processamento:

- Seja  $t_i$  o “tempo” no qual a instalação  $i$  é aberta
- Percorra as instalações abertas na ordem crescente de  $t_i$ 
  - ▶ se existir uma instalação aberta com distância  $\leq 2t_i$

# Algoritmo

FLCostShare( $S$ )

- 1  $\alpha_j \leftarrow 0$  para todo  $j \in S$
- 2  $\alpha'_j \leftarrow 0$  para todo  $j \in S$
- 3  $F \leftarrow \emptyset$
- 4 **enquanto**  $S \setminus F \neq \emptyset$
- 5     Aumente  $\alpha_j$ 's para  $j \in T \setminus F$  e  $\alpha'_j$ 's para  $j \in T$  igualmente até que
- 6     ou para uma instalação não aberta  $i$ ,  $\sum_{j \in T} \max(0, \alpha'_j - d_{ij}) = f_i$
- 7     abra a instalação  $i$
- 8     adicione todo jogador  $j$  com contribuição positiva para  $i$  a  $F$
- 9     ou para uma instalação aberta e um jogador  $j$ ,  $\alpha_j = d_{ij}$
- 10    adicione  $j$  a  $F$

Pós-processamento:

- Seja  $t_i$  o “tempo” no qual a instalação  $i$  é aberta
- Percorra as instalações abertas na ordem crescente de  $t_i$ 
  - ▶ se existir uma instalação aberta com distância  $\leq 2t_i$
  - ▶ fechamos  $i$  e caso contrário, mantemos  $i$  aberta



# Algoritmo

FLCostShare( $S$ )

- 1  $\alpha_j \leftarrow 0$  para todo  $j \in S$
- 2  $\alpha'_j \leftarrow 0$  para todo  $j \in S$
- 3  $F \leftarrow \emptyset$
- 4 **enquanto**  $S \setminus F \neq \emptyset$
- 5     Aumente  $\alpha_j$ 's para  $j \in T \setminus F$  e  $\alpha'_j$ 's para  $j \in T$  igualmente até que
- 6     ou para uma instalação não aberta  $i$ ,  $\sum_{j \in T} \max(0, \alpha'_j - d_{ij}) = f_i$
- 7     abra a instalação  $i$
- 8     adicione todo jogador  $j$  com contribuição positiva para  $i$  a  $F$
- 9     ou para uma instalação aberta e um jogador  $j$ ,  $\alpha_j = d_{ij}$
- 10    adicione  $j$  a  $F$

Pós-processamento:

- Seja  $t_i$  o “tempo” no qual a instalação  $i$  é aberta
- Percorra as instalações abertas na ordem crescente de  $t_i$ 
  - ▶ se existir uma instalação aberta com distância  $\leq 2t_i$
  - ▶ fechamos  $i$  e caso contrário, mantemos  $i$  aberta

Seja  $\mathcal{F}'$  as instalações que continuaram abertas

# Algoritmo

FLCostShare( $S$ )

- 1  $\alpha_j \leftarrow 0$  para todo  $j \in S$
- 2  $\alpha'_j \leftarrow 0$  para todo  $j \in S$
- 3  $F \leftarrow \emptyset$
- 4 **enquanto**  $S \setminus F \neq \emptyset$
- 5     Aumente  $\alpha_j$ 's para  $j \in T \setminus F$  e  $\alpha'_j$ 's para  $j \in T$  igualmente até que
- 6     ou para uma instalação não aberta  $i$ ,  $\sum_{j \in T} \max(0, \alpha'_j - d_{ij}) = f_i$
- 7     abra a instalação  $i$
- 8     adicione todo jogador  $j$  com contribuição positiva para  $i$  a  $F$
- 9     ou para uma instalação aberta e um jogador  $j$ ,  $\alpha_j = d_{ij}$
- 10    adicione  $j$  a  $F$

Pós-processamento:

- Seja  $t_i$  o “tempo” no qual a instalação  $i$  é aberta
- Percorra as instalações abertas na ordem crescente de  $t_i$ 
  - ▶ se existir uma instalação aberta com distância  $\leq 2t_i$
  - ▶ fechamos  $i$  e caso contrário, mantemos  $i$  aberta

Seja  $\mathcal{F}'$  as instalações que continuaram abertas

- Conecte  $j \in S$  na instalação de  $\mathcal{F}'$  mais próxima de  $j$

## 1/3-orçamento balanceado

$S_i$ : conjunto de jogadores a distância  $t_i$  da instalação  $i \in \mathcal{F}'$

## 1/3-orçamento balanceado

$S_i$ : conjunto de jogadores a distância  $t_i$  da instalação  $i \in \mathcal{F}'$

- se  $j \in S_i \cap S_{i'}$ , então  $d_{ii'} \leq t_i + t_{i'} \leq 2 \max\{t_i, t_{i'}\}$

## 1/3-orçamento balanceado

$S_i$ : conjunto de jogadores a distância  $t_i$  da instalação  $i \in \mathcal{F}'$

- se  $j \in S_i \cap S_{i'}$ , então  $d_{ii'} \leq t_i + t_{i'} \leq 2 \max\{t_i, t_{i'}\}$
- ou  $i$  ou  $i'$  teria sido fechada, ou seja,  $S_i \cap S_{i'} = \emptyset$

## 1/3-orçamento balanceado

$S_i$ : conjunto de jogadores a distância  $t_i$  da instalação  $i \in \mathcal{F}'$

- se  $j \in S_i \cap S_{i'}$ , então  $d_{ii'} \leq t_i + t_{i'} \leq 2 \max\{t_i, t_{i'}\}$
- ou  $i$  ou  $i'$  teria sido fechada, ou seja,  $S_i \cap S_{i'} = \emptyset$

Vamos mostrar que:

## 1/3-orçamento balanceado

$S_i$ : conjunto de jogadores a distância  $t_i$  da instalação  $i \in \mathcal{F}'$

- se  $j \in S_i \cap S_{i'}$ , então  $d_{ii'} \leq t_i + t_{i'} \leq 2 \max\{t_i, t_{i'}\}$
- ou  $i$  ou  $i'$  teria sido fechada, ou seja,  $S_i \cap S_{i'} = \emptyset$

Vamos mostrar que:

1. todo  $j \notin \bigcup_{i \in \mathcal{F}'} S_i$  é capaz de pagar  $(\min_{i \in \mathcal{F}'} d_{ij})/3$

## 1/3-orçamento balanceado

$S_i$ : conjunto de jogadores a distância  $t_i$  da instalação  $i \in \mathcal{F}'$

- se  $j \in S_i \cap S_{i'}$ , então  $d_{ii'} \leq t_i + t_{i'} \leq 2 \max\{t_i, t_{i'}\}$
- ou  $i$  ou  $i'$  teria sido fechada, ou seja,  $S_i \cap S_{i'} = \emptyset$

Vamos mostrar que:

1. todo  $j \notin \bigcup_{i \in \mathcal{F}'} S_i$  é capaz de pagar  $(\min_{i \in \mathcal{F}'} d_{ij})/3$
2.  $S_i$  é capaz de pagar  $(f_i + \sum_{j \in S_i} d_{ij})/3$



## 1/3-orçamento balanceado

$S_i$ : conjunto de jogadores a distância  $t_i$  da instalação  $i \in \mathcal{F}'$

- se  $j \in S_i \cap S_{i'}$ , então  $d_{ii'} \leq t_i + t_{i'} \leq 2 \max\{t_i, t_{i'}\}$
- ou  $i$  ou  $i'$  teria sido fechada, ou seja,  $S_i \cap S_{i'} = \emptyset$

Vamos mostrar que:

1. todo  $j \notin \bigcup_{i \in \mathcal{F}'} S_i$  é capaz de pagar  $(\min_{i \in \mathcal{F}'} d_{ij})/3$
2.  $S_i$  é capaz de pagar  $(f_i + \sum_{j \in S_i} d_{ij})/3$

I.e.,  $\sum_{j \in S} \alpha_j$  é maior ou igual a  $1/3$  do custo da solução

## 1/3-orçamento balanceado

$S_i$ : conjunto de jogadores a distância  $t_i$  da instalação  $i \in \mathcal{F}'$

- se  $j \in S_i \cap S_{i'}$ , então  $d_{ii'} \leq t_i + t_{i'} \leq 2 \max\{t_i, t_{i'}\}$
- ou  $i$  ou  $i'$  teria sido fechada, ou seja,  $S_i \cap S_{i'} = \emptyset$

Vamos mostrar que:

1. todo  $j \notin \bigcup_{i \in \mathcal{F}'}$  é capaz de pagar  $(\min_{i \in \mathcal{F}'} d_{ij})/3$
2.  $S_i$  é capaz de pagar  $(f_i + \sum_{j \in S_i} d_{ij})/3$

I.e.,  $\sum_{j \in S} \alpha_j$  é maior ou igual a  $1/3$  do custo da solução

- e, portanto, é maior ou igual a  $1/3$  de  $c(S)$

## 1/3-orçamento balanceado

$S_i$ : conjunto de jogadores a distância  $t_i$  da instalação  $i \in \mathcal{F}'$

- se  $j \in S_i \cap S_{i'}$ , então  $d_{ii'} \leq t_i + t_{i'} \leq 2 \max\{t_i, t_{i'}\}$
- ou  $i$  ou  $i'$  teria sido fechada, ou seja,  $S_i \cap S_{i'} = \emptyset$

Vamos mostrar que:

1. todo  $j \notin \bigcup_{i \in \mathcal{F}'} S_i$  é capaz de pagar  $(\min_{i \in \mathcal{F}'} d_{ij})/3$
2.  $S_i$  é capaz de pagar  $(f_i + \sum_{j \in S_i} d_{ij})/3$

I.e.,  $\sum_{j \in S} \alpha_j$  é maior ou igual a  $1/3$  do custo da solução

- e, portanto, é maior ou igual a  $1/3$  de  $c(S)$
- ou seja,  $\alpha$  é  $1/3$ -orçamento balanceado

## 1/3-orçamento balanceado

$S_i$ : conjunto de jogadores a distância  $t_i$  da instalação  $i \in \mathcal{F}'$

- se  $j \in S_i \cap S_{i'}$ , então  $d_{ii'} \leq t_i + t_{i'} \leq 2 \max\{t_i, t_{i'}\}$
- ou  $i$  ou  $i'$  teria sido fechada, ou seja,  $S_i \cap S_{i'} = \emptyset$

Vamos mostrar que:

1. todo  $j \notin \bigcup_{i \in \mathcal{F}'} S_i$  é capaz de pagar  $(\min_{i \in \mathcal{F}'} d_{ij})/3$
2.  $S_i$  é capaz de pagar  $(f_i + \sum_{j \in S_i} d_{ij})/3$

I.e.,  $\sum_{j \in S} \alpha_j$  é maior ou igual a  $1/3$  do custo da solução

- e, portanto, é maior ou igual a  $1/3$  de  $c(S)$
- ou seja,  $\alpha$  é  $1/3$ -orçamento balanceado

A primeira parte é simples:

## 1/3-orçamento balanceado

$S_i$ : conjunto de jogadores a distância  $t_i$  da instalação  $i \in \mathcal{F}'$

- se  $j \in S_i \cap S_{i'}$ , então  $d_{ii'} \leq t_i + t_{i'} \leq 2 \max\{t_i, t_{i'}\}$
- ou  $i$  ou  $i'$  teria sido fechada, ou seja,  $S_i \cap S_{i'} = \emptyset$

Vamos mostrar que:

1. todo  $j \notin \bigcup_{i \in \mathcal{F}'} S_i$  é capaz de pagar  $(\min_{i \in \mathcal{F}'} d_{ij})/3$
2.  $S_i$  é capaz de pagar  $(f_i + \sum_{j \in S_i} d_{ij})/3$

I.e.,  $\sum_{j \in S} \alpha_j$  é maior ou igual a  $1/3$  do custo da solução

- e, portanto, é maior ou igual a  $1/3$  de  $c(S)$
- ou seja,  $\alpha$  é  $1/3$ -orçamento balanceado

A primeira parte é simples:

- seja  $j \notin \bigcup_{i \in \mathcal{F}'} S_i$  e  $i$  tal que  $t_i = \alpha_j$

## 1/3-orçamento balanceado

$S_i$ : conjunto de jogadores a distância  $t_i$  da instalação  $i \in \mathcal{F}'$

- se  $j \in S_i \cap S_{i'}$ , então  $d_{ii'} \leq t_i + t_{i'} \leq 2 \max\{t_i, t_{i'}\}$
- ou  $i$  ou  $i'$  teria sido fechada, ou seja,  $S_i \cap S_{i'} = \emptyset$

Vamos mostrar que:

1. todo  $j \notin \bigcup_{i \in \mathcal{F}'} S_i$  é capaz de pagar  $(\min_{i \in \mathcal{F}'} d_{ij})/3$
2.  $S_i$  é capaz de pagar  $(f_i + \sum_{j \in S_i} d_{ij})/3$

I.e.,  $\sum_{j \in S} \alpha_j$  é maior ou igual a  $1/3$  do custo da solução

- e, portanto, é maior ou igual a  $1/3$  de  $c(S)$
- ou seja,  $\alpha$  é  $1/3$ -orçamento balanceado

A primeira parte é simples:

- seja  $j \notin \bigcup_{i \in \mathcal{F}'} S_i$  e  $i$  tal que  $t_i = \alpha_j$
- existe  $i' \in \mathcal{F}'$  a distância no máximo  $2t_i$  de  $i$

## 1/3-orçamento balanceado

$S_i$ : conjunto de jogadores a distância  $t_i$  da instalação  $i \in \mathcal{F}'$

- se  $j \in S_i \cap S_{i'}$ , então  $d_{ii'} \leq t_i + t_{i'} \leq 2 \max\{t_i, t_{i'}\}$
- ou  $i$  ou  $i'$  teria sido fechada, ou seja,  $S_i \cap S_{i'} = \emptyset$

Vamos mostrar que:

1. todo  $j \notin \bigcup_{i \in \mathcal{F}'} S_i$  é capaz de pagar  $(\min_{i \in \mathcal{F}'} d_{ij})/3$
2.  $S_i$  é capaz de pagar  $(f_i + \sum_{j \in S_i} d_{ij})/3$

I.e.,  $\sum_{j \in S} \alpha_j$  é maior ou igual a  $1/3$  do custo da solução

- e, portanto, é maior ou igual a  $1/3$  de  $c(S)$
- ou seja,  $\alpha$  é  $1/3$ -orçamento balanceado

A primeira parte é simples:

- seja  $j \notin \bigcup_{i \in \mathcal{F}'} S_i$  e  $i$  tal que  $t_i = \alpha_j$
- existe  $i' \in \mathcal{F}'$  a distância no máximo  $2t_i$  de  $i$
- $d_{i'j} \leq d_{ij} + 2t_i \leq 3\alpha_j$

## Segunda parte

Note que



## Segunda parte

Note que

- No momento de abertura de  $i$ ,  $\alpha'_j = t_i$  para todo  $j$  em  $S$

## Segunda parte

Note que

- No momento de abertura de  $i$ ,  $\alpha'_j = t_i$  para todo  $j$  em  $S$
- Para  $j \notin S_i$ ,  $\alpha'_j < d_{ij}$

## Segunda parte

Note que

- No momento de abertura de  $i$ ,  $\alpha'_j = t_i$  para todo  $j$  em  $S$
- Para  $j \notin S_i$ ,  $\alpha'_j < d_{ij}$ 
  - ▶  $j$  não contribui para  $i$

## Segunda parte

Note que

- No momento de abertura de  $i$ ,  $\alpha'_j = t_i$  para todo  $j$  em  $S$
- Para  $j \notin S_i$ ,  $\alpha'_j < d_{ij}$ 
  - ▶  $j$  não contribui para  $i$
- Ou seja,  $f_i = \sum_{j \in S} \max(0, \alpha'_j - d_{ij}) = \sum_{j \in S_i} (t_i - d_{ij})$

## Segunda parte

Note que

- No momento de abertura de  $i$ ,  $\alpha'_j = t_i$  para todo  $j$  em  $S$
- Para  $j \notin S_i$ ,  $\alpha'_j < d_{ij}$ 
  - ▶  $j$  não contribui para  $i$
- Ou seja,  $f_i = \sum_{j \in S} \max(0, \alpha'_j - d_{ij}) = \sum_{j \in S_i} (t_i - d_{ij})$

Suponha que existe  $j \in S_i$  tal que  $\alpha_j < t_i/3$

## Segunda parte

Note que

- No momento de abertura de  $i$ ,  $\alpha'_j = t_i$  para todo  $j$  em  $S$
- Para  $j \notin S_i$ ,  $\alpha'_j < d_{ij}$ 
  - ▶  $j$  não contribui para  $i$
- Ou seja,  $f_i = \sum_{j \in S} \max(0, \alpha'_j - d_{ij}) = \sum_{j \in S_i} (t_i - d_{ij})$

Suponha que existe  $j \in S_i$  tal que  $\alpha_j < t_i/3$

- Seja  $i_1$  a instalação que congelou  $\alpha_j$

## Segunda parte

Note que

- No momento de abertura de  $i$ ,  $\alpha'_j = t_i$  para todo  $j$  em  $S$
- Para  $j \notin S_i$ ,  $\alpha'_j < d_{ij}$ 
  - ▶  $j$  não contribui para  $i$
- Ou seja,  $f_i = \sum_{j \in S} \max(0, \alpha'_j - d_{ij}) = \sum_{j \in S_i} (t_i - d_{ij})$

Suponha que existe  $j \in S_i$  tal que  $\alpha_j < t_i/3$

- Seja  $i_1$  a instalação que congelou  $\alpha_j$
- Seja  $i_2 \in \mathcal{F}'$  a distância no máximo  $2t_{i_1}$  de  $i_1$

## Segunda parte

Note que

- No momento de abertura de  $i$ ,  $\alpha'_j = t_i$  para todo  $j$  em  $S$
- Para  $j \notin S_i$ ,  $\alpha'_j < d_{ij}$ 
  - ▶  $j$  não contribui para  $i$
- Ou seja,  $f_i = \sum_{j \in S} \max(0, \alpha'_j - d_{ij}) = \sum_{j \in S_i} (t_i - d_{ij})$

Suponha que existe  $j \in S_i$  tal que  $\alpha_j < t_i/3$

- Seja  $i_1$  a instalação que congelou  $\alpha_j$
- Seja  $i_2 \in \mathcal{F}'$  a distância no máximo  $2t_{i_1}$  de  $i_1$ 
  - ▶ ou fechamos  $i_1$  por causa de  $i_2$  ou  $i_1 = i_2$



## Segunda parte

Note que

- No momento de abertura de  $i$ ,  $\alpha'_j = t_i$  para todo  $j$  em  $S$
- Para  $j \notin S_i$ ,  $\alpha'_j < d_{ij}$ 
  - ▶  $j$  não contribui para  $i$
- Ou seja,  $f_i = \sum_{j \in S} \max(0, \alpha'_j - d_{ij}) = \sum_{j \in S_i} (t_i - d_{ij})$

Suponha que existe  $j \in S_i$  tal que  $\alpha_j < t_i/3$

- Seja  $i_1$  a instalação que congelou  $\alpha_j$
- Seja  $i_2 \in \mathcal{F}'$  a distância no máximo  $2t_{i_1}$  de  $i_1$ 
  - ▶ ou fechamos  $i_1$  por causa de  $i_2$  ou  $i_1 = i_2$
- $d_{ii_2} \leq d_{ij} + d_{i_1j} + d_{i_1i_2} \leq t_i + 3\alpha_j < 2t_i$

## Segunda parte

Note que

- No momento de abertura de  $i$ ,  $\alpha'_j = t_i$  para todo  $j$  em  $S$
- Para  $j \notin S_i$ ,  $\alpha'_j < d_{ij}$ 
  - ▶  $j$  não contribui para  $i$
- Ou seja,  $f_i = \sum_{j \in S} \max(0, \alpha'_j - d_{ij}) = \sum_{j \in S_i} (t_i - d_{ij})$

Suponha que existe  $j \in S_i$  tal que  $\alpha_j < t_i/3$

- Seja  $i_1$  a instalação que congelou  $\alpha_j$
- Seja  $i_2 \in \mathcal{F}'$  a distância no máximo  $2t_{i_1}$  de  $i_1$ 
  - ▶ ou fechamos  $i_1$  por causa de  $i_2$  ou  $i_1 = i_2$
- $d_{ii_2} \leq d_{ij} + d_{i_1j} + d_{i_1i_2} \leq t_i + 3\alpha_j < 2t_i$
- então  $i$  e  $i_2$  não podem ambas estarem abertas...

## Segunda parte

Note que

- No momento de abertura de  $i$ ,  $\alpha'_j = t_i$  para todo  $j$  em  $S$
- Para  $j \notin S_i$ ,  $\alpha'_j < d_{ij}$ 
  - ▶  $j$  não contribui para  $i$
- Ou seja,  $f_i = \sum_{j \in S} \max(0, \alpha'_j - d_{ij}) = \sum_{j \in S_i} (t_i - d_{ij})$

Suponha que existe  $j \in S_i$  tal que  $\alpha_j < t_i/3$

- Seja  $i_1$  a instalação que congelou  $\alpha_j$
- Seja  $i_2 \in \mathcal{F}'$  a distância no máximo  $2t_{i_1}$  de  $i_1$ 
  - ▶ ou fechamos  $i_1$  por causa de  $i_2$  ou  $i_1 = i_2$
- $d_{ii_2} \leq d_{ij} + d_{i_1j} + d_{i_1i_2} \leq t_i + 3\alpha_j < 2t_i$
- então  $i$  e  $i_2$  não podem ambas estarem abertas...

Assim,  $\sum_{j \in S_i} \alpha_j \geq \sum_{j \in S_i} t_i/3 \geq (f_i + \sum_{j \in S_i} d_{ij})/3$

# Monotonicidade Cruzada

Falta mostrar que o esquema de compartilhamento de custos é monotônico cruzado

# Monotonicidade Cruzada

Falta mostrar que o esquema de compartilhamento de custos é monotônico cruzado

Seja  $S, T \subseteq \mathcal{A}$ . Se  $j \in S$  está congelado em um tempo  $t$  na execução de  $\text{FLCostShare}(S)$ , então  $j$  está congelado no tempo  $t$  na execução de  $\text{FLCostShare}(S \cup T)$

# Monotonicidade Cruzada

Falta mostrar que o esquema de compartilhamento de custos é monotônico cruzado

Seja  $S, T \subseteq \mathcal{A}$ . Se  $j \in S$  está congelado em um tempo  $t$  na execução de  $\text{FLCostShare}(S)$ , então  $j$  está congelado no tempo  $t$  na execução de  $\text{FLCostShare}(S \cup T)$

- I.e.,  $\xi(j, S) \geq \xi(j, S \cup T)$

## Recapitulando

Existe uma alocação de custos  $1/1,463$ -orçamento balanceada para o Jogo da Localização de Instalações

## Recapitulando

Existe uma alocação de custos  $1/1,463$ -orçamento balanceada para o Jogo da Localização de Instalações

- Basta calcular o dual da formulação



## Recapitulando

Existe uma alocação de custos  $1/1,463$ -orçamento balanceada para o Jogo da Localização de Instalações

- Basta calcular o dual da formulação

O  $1/1,52$ -núcleo do Jogo da Localização de Instalações pode ser vazio

## Recapitulando

Existe uma alocação de custos  $1/1,463$ -orçamento balanceada para o Jogo da Localização de Instalações

- Basta calcular o dual da formulação

O  $1/1,52$ -núcleo do Jogo da Localização de Instalações pode ser vazio

- por causa do gap de integralidade

## Recapitulando

Existe uma alocação de custos  $1/1,463$ -orçamento balanceada para o Jogo da Localização de Instalações

- Basta calcular o dual da formulação

O  $1/1,52$ -núcleo do Jogo da Localização de Instalações pode ser vazio

- por causa do gap de integralidade

Existe um esquema de compartilhamento de custos monotônico cruzado  $1/3$ -orçamento balanceado para o Jogo da Localização de Instalações

## Recapitulando

Existe uma alocação de custos  $1/1,463$ -orçamento balanceada para o Jogo da Localização de Instalações

- Basta calcular o dual da formulação

O  $1/1,52$ -núcleo do Jogo da Localização de Instalações pode ser vazio

- por causa do gap de integralidade

Existe um esquema de compartilhamento de custos monotônico cruzado  $1/3$ -orçamento balanceado para o Jogo da Localização de Instalações

- e não dá para fazer melhor que isso

## Outra forma de alocar custos

O núcleo é um conceito interessante para jogos cooperativos

## Outra forma de alocar custos

O núcleo é um conceito interessante para jogos cooperativos

- Mas pode ser vazio...

## Outra forma de alocar custos

O núcleo é um conceito interessante para jogos cooperativos

- Mas pode ser vazio...
- Ou pode ter mais de um elemento

## Outra forma de alocar custos

O núcleo é um conceito interessante para jogos cooperativos

- Mas pode ser vazio...
- Ou pode ter mais de um elemento
  - ▶ como escolher entre eles?



## Outra forma de alocar custos

O núcleo é um conceito interessante para jogos cooperativos

- Mas pode ser vazio...
- Ou pode ter mais de um elemento
  - ▶ como escolher entre eles?

Vamos ver outra forma de alocar custos

## Outra forma de alocar custos

O núcleo é um conceito interessante para jogos cooperativos

- Mas pode ser vazio...
- Ou pode ter mais de um elemento
  - ▶ como escolher entre eles?

Vamos ver outra forma de alocar custos

Uma forma simples de alocar custos entre os jogadores:

## Outra forma de alocar custos

O núcleo é um conceito interessante para jogos cooperativos

- Mas pode ser vazio...
- Ou pode ter mais de um elemento
  - ▶ como escolher entre eles?

Vamos ver outra forma de alocar custos

Uma forma simples de alocar custos entre os jogadores:

- Ordene os jogadores como  $a_1, a_2, \dots, a_n$

## Outra forma de alocar custos

O núcleo é um conceito interessante para jogos cooperativos

- Mas pode ser vazio...
- Ou pode ter mais de um elemento
  - ▶ como escolher entre eles?

Vamos ver outra forma de alocar custos

Uma forma simples de alocar custos entre os jogadores:

- Ordene os jogadores como  $a_1, a_2, \dots, a_n$
- Cobre de  $a_1$  o valor  $c(\{a_1\})$

## Outra forma de alocar custos

O núcleo é um conceito interessante para jogos cooperativos

- Mas pode ser vazio...
- Ou pode ter mais de um elemento
  - ▶ como escolher entre eles?

Vamos ver outra forma de alocar custos

Uma forma simples de alocar custos entre os jogadores:

- Ordene os jogadores como  $a_1, a_2, \dots, a_n$
- Cobre de  $a_1$  o valor  $c(\{a_1\})$
- Cobre de  $a_2$  o valor  $c(\{a_2, a_1\}) - c(\{a_1\})$

## Outra forma de alocar custos

O núcleo é um conceito interessante para jogos cooperativos

- Mas pode ser vazio...
- Ou pode ter mais de um elemento
  - ▶ como escolher entre eles?

Vamos ver outra forma de alocar custos

Uma forma simples de alocar custos entre os jogadores:

- Ordene os jogadores como  $a_1, a_2, \dots, a_n$
- Dobre de  $a_1$  o valor  $c(\{a_1\})$
- Dobre de  $a_2$  o valor  $c(\{a_2, a_1\}) - c(\{a_1\})$
- ...

## Outra forma de alocar custos

O núcleo é um conceito interessante para jogos cooperativos

- Mas pode ser vazio...
- Ou pode ter mais de um elemento
  - ▶ como escolher entre eles?

Vamos ver outra forma de alocar custos

Uma forma simples de alocar custos entre os jogadores:

- Ordene os jogadores como  $a_1, a_2, \dots, a_n$
- Cobre de  $a_1$  o valor  $c(\{a_1\})$
- Cobre de  $a_2$  o valor  $c(\{a_2, a_1\}) - c(\{a_1\})$
- ...
- Cobre de  $a_n$  o valor  $c(\mathcal{A}) - c(\mathcal{A} \setminus \{a_n\})$

## Outra forma de alocar custos

O núcleo é um conceito interessante para jogos cooperativos

- Mas pode ser vazio...
- Ou pode ter mais de um elemento
  - ▶ como escolher entre eles?

Vamos ver outra forma de alocar custos

Uma forma simples de alocar custos entre os jogadores:

- Ordene os jogadores como  $a_1, a_2, \dots, a_n$
- Dobre de  $a_1$  o valor  $c(\{a_1\})$
- Dobre de  $a_2$  o valor  $c(\{a_2, a_1\}) - c(\{a_1\})$
- ...
- Dobre de  $a_n$  o valor  $c(\mathcal{A}) - c(\mathcal{A} \setminus \{a_n\})$

Essa alocação de custos é orçamento balanceada



## Outra forma de alocar custos

O núcleo é um conceito interessante para jogos cooperativos

- Mas pode ser vazio...
- Ou pode ter mais de um elemento
  - ▶ como escolher entre eles?

Vamos ver outra forma de alocar custos

Uma forma simples de alocar custos entre os jogadores:

- Ordene os jogadores como  $a_1, a_2, \dots, a_n$
- Cobre de  $a_1$  o valor  $c(\{a_1\})$
- Cobre de  $a_2$  o valor  $c(\{a_2, a_1\}) - c(\{a_1\})$
- ...
- Cobre de  $a_n$  o valor  $c(\mathcal{A}) - c(\mathcal{A} \setminus \{a_n\})$

Essa alocação de custos é orçamento balanceada

- é chamada de compartilhamento de custos incremental

## Outra forma de alocar custos

O núcleo é um conceito interessante para jogos cooperativos

- Mas pode ser vazio...
- Ou pode ter mais de um elemento
  - ▶ como escolher entre eles?

Vamos ver outra forma de alocar custos

Uma forma simples de alocar custos entre os jogadores:

- Ordene os jogadores como  $a_1, a_2, \dots, a_n$
- Cobre de  $a_1$  o valor  $c(\{a_1\})$
- Cobre de  $a_2$  o valor  $c(\{a_2, a_1\}) - c(\{a_1\})$
- ...
- Cobre de  $a_n$  o valor  $c(\mathcal{A}) - c(\mathcal{A} \setminus \{a_n\})$

Essa alocação de custos é orçamento balanceada

- é chamada de compartilhamento de custos incremental
- mas não é anônima, pode não ser muito justa...

## O Valor de Shapley

O Valor de Shapley resolve o problema da anonimidade usando aleatoriedade

## O Valor de Shapley

O Valor de Shapley resolve o problema da anonimidade usando aleatoriedade

- escolhemos uma ordem aleatória dos jogadores

## O Valor de Shapley

O Valor de Shapley resolve o problema da anonimidade usando aleatoriedade

- escolhemos uma ordem aleatória dos jogadores
- cobramos dele o seu valor esperado

## O Valor de Shapley

O Valor de Shapley resolve o problema da anonimidade usando aleatoriedade

- escolhemos uma ordem aleatória dos jogadores
- cobramos dele o seu valor esperado

Qual a probabilidade do jogador  $j$  ocupar a posição  $s + 1$  e um conjunto  $S$  um conjunto  $S$  ocupar as  $s$  primeiras posições?

## O Valor de Shapley

O Valor de Shapley resolve o problema da anonimidade usando aleatoriedade

- escolhemos uma ordem aleatória dos jogadores
- cobramos dele o seu valor esperado

Qual a probabilidade do jogador  $j$  ocupar a posição  $s + 1$  e um conjunto  $S$  um conjunto  $S$  ocupar as  $s$  primeiras posições?

$$\frac{s!(n - 1 - s!)}{n!}$$

## O Valor de Shapley

O Valor de Shapley resolve o problema da anonimidade usando aleatoriedade

- escolhemos uma ordem aleatória dos jogadores
- cobramos dele o seu valor esperado

Qual a probabilidade do jogador  $j$  ocupar a posição  $s + 1$  e um conjunto  $S$  um conjunto  $S$  ocupar as  $s$  primeiras posições?

$$\frac{s!(n - 1 - s!)}{n!}$$

O Valor de Shapley define a seguinte alocação de custos  $\phi$



## O Valor de Shapley

O Valor de Shapley resolve o problema da anonimidade usando aleatoriedade

- escolhemos uma ordem aleatória dos jogadores
- cobramos dele o seu valor esperado

Qual a probabilidade do jogador  $j$  ocupar a posição  $s + 1$  e um conjunto  $S$  um conjunto  $S$  ocupar as  $s$  primeiras posições?

$$\frac{s!(n - 1 - s!)}{n!}$$

O Valor de Shapley define a seguinte alocação de custos  $\phi$

$$\phi_i(c) = \sum_{s=0}^{n-1} \sum_{\substack{S \subseteq \mathcal{A} \setminus \{i\} \\ |S|=s}} \frac{s!(n - 1 - s!)}{n!} (c(S \cup \{i\}) - c(S))$$

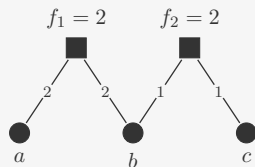
## O Valor de Shapley

A alocação de custos do Valor de Shapley não precisa estar no núcleo mesmo quando o núcleo é não-vazio

## O Valor de Shapley

A alocação de custos do Valor de Shapley não precisa estar no núcleo mesmo quando o núcleo é não-vazio

Vimos um Jogo de Localização de Instalações com núcleo não-vazio e  $c(\{a\}) = 4$ ,  $c(\{b\}) = 3$ ,  $c(\{c\}) = 3$ ,  $c(\{a, b\}) = 6$ ,  $c(\{b, c\}) = 4$ ,  $c(\{a, c\}) = 7$  e  $c(\{a, b, c\}) = 8$



## O Valor de Shapley

A alocação de custos do Valor de Shapley não precisa estar no núcleo mesmo quando o núcleo é não-vazio

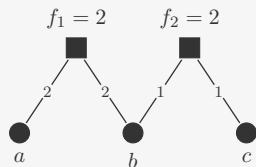
Vimos um Jogo de Localização de Instalações com núcleo não-vazio e  $c(\{a\}) = 4$ ,  $c(\{b\}) = 3$ ,  $c(\{c\}) = 3$ ,  $c(\{a, b\}) = 6$ ,  $c(\{b, c\}) = 4$ ,  $c(\{a, c\}) = 7$  e  $c(\{a, b, c\}) = 8$

Porém,

$$\phi_a = \frac{1}{3} \cdot 4 + \frac{1}{6} \cdot 3 + \frac{1}{6} \cdot 4 + \frac{1}{3} \cdot 4 = \frac{23}{6}$$

$$\phi_b = \frac{1}{3} \cdot 3 + \frac{1}{6} \cdot 2 + \frac{1}{6} \cdot 1 + \frac{1}{3} \cdot 1 = \frac{11}{6}$$

$$\phi_c = \frac{1}{3} \cdot 3 + \frac{1}{6} \cdot 3 + \frac{1}{6} \cdot 1 + \frac{1}{3} \cdot 2 = \frac{7}{3},$$



## O Valor de Shapley

A alocação de custos do Valor de Shapley não precisa estar no núcleo mesmo quando o núcleo é não-vazio

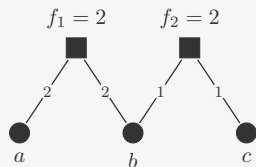
Vimos um Jogo de Localização de Instalações com núcleo não-vazio e  $c(\{a\}) = 4$ ,  $c(\{b\}) = 3$ ,  $c(\{c\}) = 3$ ,  $c(\{a, b\}) = 6$ ,  $c(\{b, c\}) = 4$ ,  $c(\{a, c\}) = 7$  e  $c(\{a, b, c\}) = 8$

Porém,

$$\phi_a = \frac{1}{3} \cdot 4 + \frac{1}{6} \cdot 3 + \frac{1}{6} \cdot 4 + \frac{1}{3} \cdot 4 = \frac{23}{6}$$

$$\phi_b = \frac{1}{3} \cdot 3 + \frac{1}{6} \cdot 2 + \frac{1}{6} \cdot 1 + \frac{1}{3} \cdot 1 = \frac{11}{6}$$

$$\phi_c = \frac{1}{3} \cdot 3 + \frac{1}{6} \cdot 3 + \frac{1}{6} \cdot 1 + \frac{1}{3} \cdot 2 = \frac{7}{3},$$



mas  $\phi_b + \phi_c = 25/6 > 4 = c(\{b, c\})$

## Caracterização Axiomática

Fixe um conjunto  $\mathcal{A}$  de  $n$  jogadores. O **valor** é uma função que atribui, para cada função de custo  $c$ , um vetor  $\phi(c) \in \mathcal{R}^n$ .

## Caracterização Axiomática

Fixe um conjunto  $\mathcal{A}$  de  $n$  jogadores. O **valor** é uma função que atribui, para cada função de custo  $c$ , um vetor  $\phi(c) \in \mathcal{R}^n$ .

Três propriedades interessantes que um valor pode ter:

# Caracterização Axiomática

Fixe um conjunto  $\mathcal{A}$  de  $n$  jogadores. O **valor** é uma função que atribui, para cada função de custo  $c$ , um vetor  $\phi(c) \in \mathcal{R}^n$ .

Três propriedades interessantes que um valor pode ter:

- **Anonimidade**: trocar os nomes dos jogadores não afeta quanto um jogador paga



# Caracterização Axiomática

Fixe um conjunto  $\mathcal{A}$  de  $n$  jogadores. O **valor** é uma função que atribui, para cada função de custo  $c$ , um vetor  $\phi(c) \in \mathcal{R}^n$ .

Três propriedades interessantes que um valor pode ter:

- **Anonimidade**: trocar os nomes dos jogadores não afeta quanto um jogador paga
  - ▶ para toda permutação  $\pi$  de  $\mathcal{A}$ ,  $\phi_{\pi_i}(\pi(c)) = \phi_i(c)$

# Caracterização Axiomática

Fixe um conjunto  $\mathcal{A}$  de  $n$  jogadores. O **valor** é uma função que atribui, para cada função de custo  $c$ , um vetor  $\phi(c) \in \mathcal{R}^n$ .

Três propriedades interessantes que um valor pode ter:

- **Anonimidade**: trocar os nomes dos jogadores não afeta quanto um jogador paga
  - ▶ para toda permutação  $\pi$  de  $\mathcal{A}$ ,  $\phi_{\pi_i}(\pi(c)) = \phi_i(c)$
- **Dummy**: um jogador que não adiciona custo não deve pagar nada

# Caracterização Axiomática

Fixe um conjunto  $\mathcal{A}$  de  $n$  jogadores. O **valor** é uma função que atribui, para cada função de custo  $c$ , um vetor  $\phi(c) \in \mathcal{R}^n$ .

Três propriedades interessantes que um valor pode ter:

- **Anonimidade**: trocar os nomes dos jogadores não afeta quanto um jogador paga
  - ▶ para toda permutação  $\pi$  de  $\mathcal{A}$ ,  $\phi_{\pi_i}(\pi(c)) = \phi_i(c)$
- **Dummy**: um jogador que não adiciona custo não deve pagar nada
  - ▶ se para todo  $S \subseteq \mathcal{A} \setminus \{i\}$ ,  $c(S) = c(S \cup \{i\})$ , então  $\phi_i(c) = 0$

# Caracterização Axiomática

Fixe um conjunto  $\mathcal{A}$  de  $n$  jogadores. O **valor** é uma função que atribui, para cada função de custo  $c$ , um vetor  $\phi(c) \in \mathcal{R}^n$ .

Três propriedades interessantes que um valor pode ter:

- **Anonimidade**: trocar os nomes dos jogadores não afeta quanto um jogador paga
  - ▶ para toda permutação  $\pi$  de  $\mathcal{A}$ ,  $\phi_{\pi_i}(\pi(c)) = \phi_i(c)$
- **Dummy**: um jogador que não adiciona custo não deve pagar nada
  - ▶ se para todo  $S \subseteq \mathcal{A} \setminus \{i\}$ ,  $c(S) = c(S \cup \{i\})$ , então  $\phi_i(c) = 0$
- **Aditividade**: para quaisquer duas funções de custo  $c_1$  e  $c_2$ ,  
 $\phi(c_1 + c_2) = \phi(c_1) + \phi(c_2)$

# Caracterização Axiomática

Fixe um conjunto  $\mathcal{A}$  de  $n$  jogadores. O **valor** é uma função que atribui, para cada função de custo  $c$ , um vetor  $\phi(c) \in \mathcal{R}^n$ .

Três propriedades interessantes que um valor pode ter:

- **Anonimidade:** trocar os nomes dos jogadores não afeta quanto um jogador paga
  - ▶ para toda permutação  $\pi$  de  $\mathcal{A}$ ,  $\phi_{\pi_i}(\pi(c)) = \phi_i(c)$
- **Dummy:** um jogador que não adiciona custo não deve pagar nada
  - ▶ se para todo  $S \subseteq \mathcal{A} \setminus \{i\}$ ,  $c(S) = c(S \cup \{i\})$ , então  $\phi_i(c) = 0$
- **Aditividade:** para quaisquer duas funções de custo  $c_1$  e  $c_2$ ,  
 $\phi(c_1 + c_2) = \phi(c_1) + \phi(c_2)$

**Teorema:** O Valor de Shapley é o único valor satisfazendo as três propriedades.