Leilões combinatórios

n participantes $\qquad$ m itens

Valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

Leilões combinatórios

n participantes m itens

Valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

Restrições:

- *free-disposal*
  $v_i(S) \leq v_i(T)$ para todo $S \subseteq T$ e todo i.
- normalização
  $v_i(\emptyset) = 0$ para todo i.

Leilões combinatórios

n participantes $\qquad$ m itens

Valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

Restrições:

- *free-disposal*
  $v_i(S) \leq v_i(T)$ para todo $S \subseteq T$ e todo i.
- normalização
  $v_i(\emptyset) = 0$ para todo i.

Alocação: conjuntos $S_1, \ldots, S_n$ de itens tq
$S_i \cap S_j = \emptyset$ para todo $i \neq j$.

Leilões combinatórios

n participantes $\qquad$ m itens

Valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

Restrições:

- *free-disposal*
  $v_i(S) \leq v_i(T)$ para todo $S \subseteq T$ e todo i.
- normalização
  $v_i(\emptyset) = 0$ para todo i.

Alocação: conjuntos $S_1, \ldots, S_n$ de itens tq
$S_i \cap S_j = \emptyset$ para todo $i \neq j$.

Bem-estar social: $\sum_{i=1}^{n} v_i(S_i)$.

Leilões combinatórios

n participantes $\qquad$ m itens

Valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

Alocação: conjuntos $S_1, \ldots, S_n$ de itens tq
$S_i \cap S_j = \emptyset$ para todo $i \neq j$.

Bem-estar social: $\sum_{i=1}^{n} v_i(S_i)$.

Leilões combinatórios

n participantes m itens

Valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

Alocação: conjuntos $S_1, \ldots, S_n$ de itens tq
$S_i \cap S_j = \emptyset$ para todo $i \neq j$.

Bem-estar social: $\sum_{i=1}^{n} v_i(S_i)$.

Alocação socialmente eficiente:
maximiza o bem-estar social.

Leilões combinatórios

n participantes m itens

Valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

Alocação: conjuntos $S_1, \ldots, S_n$ de itens tq
$S_i \cap S_j = \emptyset$ para todo $i \neq j$.

Bem-estar social: $\sum_{i=1}^{n} v_i(S_i)$.

Alocação socialmente eficiente:
maximiza o bem-estar social.

Valorações são informação privada.

Queremos métodos eficientes e à prova de estratégia
para maximizar o bem-estar social.

Caso geral

n participantes $\qquad$ m itens

Valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

Alocação: conjuntos $S_1, \ldots, S_n$ de itens tq
$S_i \cap S_j = \emptyset$ para todo $i \neq j$.

Objetivo: maximizar o bem-estar social $\sum_{i=1}^{n} v_i(S_i)$.

Caso geral

n participantes m itens

Valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

Alocação: conjuntos $S_1, \ldots, S_n$ de itens tq
$S_i \cap S_j = \emptyset$ para todo $i \neq j$.

Objetivo: maximizar o bem-estar social $\sum_{i=1}^{n} v_i(S_i)$.

Vamos descrever esse problema como um MIP.

Caso geral

n participantes m itens

Valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

Alocação: conjuntos $S_1, \ldots, S_n$ de itens tq
$S_i \cap S_j = \emptyset$ para todo $i \neq j$.

Objetivo: maximizar o bem-estar social $\sum_{i=1}^{n} v_i(S_i)$.

Vamos descrever esse problema como um MIP.

MIP: programa linear inteiro.

Formulação linear inteira

n participantes m itens

Valorações: valor $v_i(S)$ para cada $i \in [n]$ e $S \subseteq [m]$.

Alocação: conjuntos $S_1, \ldots, S_n$ de itens tq
$S_i \cap S_j = \emptyset$ para todo $i \neq j$.

Objetivo: maximizar o bem-estar social $\sum_{i=1}^{n} v_i(S_i)$.

Formulação linear inteira

n participantes $\qquad$ m itens

Valorações: valor $v_i(S)$ para cada $i \in [n]$ e $S \subseteq [m]$.

Alocação: conjuntos $S_1, \ldots, S_n$ de itens tq
$S_i \cap S_j = \emptyset$ para todo $i \neq j$.

Objetivo: maximizar o bem-estar social $\sum_{i=1}^{n} v_i(S_i)$.

Variáveis binárias: $x_{i,S} = 1$ sse i recebe S.

Formulação linear inteira

n participantes $\quad$ m itens

Valorações: valor $v_i(S)$ para cada $i \in [n]$ e $S \subseteq [m]$.

Alocação: conjuntos $S_1, \ldots, S_n$ de itens tq
$S_i \cap S_j = \emptyset$ para todo $i \neq j$.

Objetivo: maximizar o bem-estar social $\sum_{i=1}^{n} v_i(S_i)$.

Variáveis binárias: $x_{i,S} = 1$ sse i recebe S.

MIP: encontrar x que

$$\begin{array}{lll}
\text{maximize} & \sum_{i=1}^{n} \sum_{S\subseteq[m]} x_{i,S}\, v_i(S) & \\
\text{sujeito a} & \sum_{S\subseteq[m]} x_{i,S} \leq 1 & \text{para cada } i \in [n] \\
& \sum_{i=1}^{n} \sum_{S\subseteq[m]:j\in S} x_{i,S} \leq 1 & \text{para cada } j \in [m] \\
& x_{i,S} \in \{0,1\} & \text{para cada } i \in [n] \text{ e cada } S \subseteq [m].
\end{array}$$

Formulação linear inteira

Variáveis binárias: $x_{i,S} = 1$ sse i recebe S.

MIP: encontrar x que

$$
\begin{array}{lll}
\text{maximize} & \sum_{i=1}^{n} \sum_{S \subseteq [m]} x_{i,S} v_i(S) & \\
\text{sujeito a} & \sum_{S \subseteq [m]} x_{i,S} \leq 1 & \text{para cada } i \in [n] \\
& \sum_{i=1}^{n} \sum_{S \subseteq [m] : j \in S} x_{i,S} \leq 1 & \text{para cada } j \in [m] \\
& x_{i,S} \in \{0,1\} \quad \text{para cada } i \in [n] \text{ e cada } S \subseteq [m]. &
\end{array}
$$

- Primeira desigualdade garante que cada participante recebe no máximo um subconjunto.

Formulação linear inteira

Variáveis binárias: $x_{i,S} = 1$ sse i recebe S.

MIP: encontrar x que

$$
\begin{array}{lll}
\text{maximize} & \sum_{i=1}^{n} \sum_{S \subseteq [m]} x_{i,S} v_i(S) & \\
\text{sujeito a} & \sum_{S \subseteq [m]} x_{i,S} \leq 1 & \text{para cada } i \in [n] \\
& \sum_{i=1}^{n} \sum_{S \subseteq [m] : j \in S} x_{i,S} \leq 1 & \text{para cada } j \in [m] \\
& x_{i,S} \in \{0, 1\} \quad \text{para cada } i \in [n] \text{ e cada } S \subseteq [m]. &
\end{array}
$$

- Primeira desigualdade garante que cada participante recebe no máximo um subconjunto.
- Segunda desigualdade garante que cada item vai para no máximo um participante.

Formulação linear inteira

Variáveis binárias: $x_{i,S} = 1$ sse i recebe S.

MIP: encontrar x que

$$\begin{array}{lll}
\text{maximize} & \sum_{i=1}^{n} \sum_{S \subseteq [m]} x_{i,S} v_i(S) & \\
\text{sujeito a} & \sum_{S \subseteq [m]} x_{i,S} \leq 1 & \text{para cada } i \in [n] \\
& \sum_{i=1}^{n} \sum_{S \subseteq [m] : j \in S} x_{i,S} \leq 1 & \text{para cada } j \in [m] \\
& x_{i,S} \in \{0,1\} \quad \text{para cada } i \in [n] \text{ e cada } S \subseteq [m]. &
\end{array}$$

- Primeira desigualdade garante que cada participante recebe no máximo um subconjunto.
- Segunda desigualdade garante que cada item vai para no máximo um participante.
- Função objetivo calcula o bem-estar social máximo.

Relaxação linear inteira

Relaxamos a restrição de integralidade em $x_{i,S}$.

LP: encontrar x que

maximize $\sum_{i=1}^{n} \sum_{S \subseteq [m]} x_{i,S} v_i(S)$

sujeito a $\sum_{S \subseteq [m]} x_{i,S} \leq 1$ para cada $i \in [n]$

$\sum_{i=1}^{n} \sum_{S \subseteq [m] : j \in S} x_{i,S} \leq 1$ para cada $j \in [m]$

$0 \leq x_{i,S} \leq 1$ para cada $i \in [n]$ e cada $S \subseteq [m]$.

Relaxação linear inteira

Relaxamos a restrição de integralidade em $x_{i,S}$.

LP: encontrar x que

maximize $\sum_{i=1}^{n} \sum_{S \subseteq [m]} x_{i,S} v_i(S)$

sujeito a $\sum_{S \subseteq [m]} x_{i,S} \leq 1$ para cada $i \in [n]$

$\sum_{i=1}^{n} \sum_{S \subseteq [m] : j \in S} x_{i,S} \leq 1$ para cada $j \in [m]$

$0 \leq x_{i,S} \leq 1$ para cada $i \in [n]$ e cada $S \subseteq [m]$.

Como é o dual deste LP?

Relaxação linear inteira

Relaxamos a restrição de integralidade em $x_{i,S}$.

LP: encontrar x que

maximize $\sum_{i=1}^{n} \sum_{S \subseteq [m]} x_{i,S} v_i(S)$

sujeito a $\sum_{i=1}^{n} \sum_{S \subseteq [m] : j \in S} x_{i,S} \leq 1$ para cada $j \in [m]$

$\sum_{S \subseteq [m]} x_{i,S} \leq 1$ para cada $i \in [n]$

$0 \leq x_{i,S} \leq 1$ para cada $i \in [n]$ e cada $S \subseteq [m]$.

Como é o dual deste LP?

Dual: encontrar u e p que

minimizem $\sum_{i=1}^{n} u_i + \sum_{j \in [m]} p_j$

sujeitos a $u_i + \sum_{j \in S} p_j \geq v_i(S)$ para cada $i \in [n]$ e $S \subseteq [m]$

$u_i \geq 0, \quad p_j \geq 0$ para cada $i \in [n]$ e $j \in [m]$.

Relaxação linear inteira

LP: encontrar x que

maximize $\sum_{i=1}^{n} \sum_{S \subseteq [m]} x_{i,S} v_i(S)$

sujeito a $\sum_{S \subseteq [m]} x_{i,S} \leq 1$ para cada $i \in [n]$

$\sum_{i=1}^{n} \sum_{S \subseteq [m] : j \in S} x_{i,S} \leq 1$ para cada $j \in [m]$

$0 \leq x_{i,S} \leq 1$ para cada $i \in [n]$ e cada $S \subseteq [m]$.

Dual: encontrar u e p que

minimizem $\sum_{i=1}^{n} u_i + \sum_{j \in [m]} p_j$

sujeitos a $u_i + \sum_{j \in S} p_j \geq v_i(S)$ para cada $S \subseteq [m]$

$u_i \geq 0, \quad p_j \geq 0$ para cada $i \in [n]$ e $j \in [m]$.

Relaxação linear inteira

LP: encontrar x que

maximize $\sum_{i=1}^{n} \sum_{S \subseteq [m]} x_{i,S} v_i(S)$

sujeito a $\sum_{S \subseteq [m]} x_{i,S} \leq 1$ para cada $i \in [n]$

$\sum_{i=1}^{n} \sum_{S \subseteq [m] : j \in S} x_{i,S} \leq 1$ para cada $j \in [m]$

$0 \leq x_{i,S} \leq 1$ para cada $i \in [n]$ e cada $S \subseteq [m]$.

Dual: encontrar u e p que

minimizem $\sum_{i=1}^{n} u_i + \sum_{j \in [m]} p_j$

sujeitos a $u_i + \sum_{j \in S} p_j \geq v_i(S)$ para cada $S \subseteq [m]$

$u_i \geq 0, \quad p_j \geq 0$ para cada $i \in [n]$ e $j \in [m]$.

Os nomes u_i e p_j são propositais, pois,
quando há solução ótima inteira,
estes valores têm exatamente as respectivas interpretações.

Equilíbrio Walrasiano

Dados preços $p_1, \ldots, p_m$ para os itens,
uma demanda para o participante i é
um conjunto $S \subseteq [m]$ que maximiza sua utilidade.

Equilíbrio Walrasiano

Dados preços $p_1, \ldots, p_m$ para os itens,
uma demanda para o participante i é
um conjunto $S \subseteq [m]$ que maximiza sua utilidade.

A utilidade de i para S é $u_i(S) = v_i(S) - \sum_{j \in S} p_j$.

Equilíbrio Walrasiano

Dados preços $p_1, \ldots, p_m$ para os itens,
uma demanda para o participante i é
um conjunto $S \subseteq [m]$ que maximiza sua utilidade.

A utilidade de i para S é $u_i(S) = v_i(S) - \sum_{j \in S} p_j$.

Ou seja, uma demanda é um conjunto S tal que

$$u_i(S) \; = \; v_i(S) - \sum_{j \in S} p_j \; \geq \; v_i(S') - \sum_{j \in S'} p_j,$$

para qualquer $S' \subseteq [m]$.

Equilíbrio Walrasiano

Dados preços $p_1, \ldots, p_m$ para os itens,
uma demanda para o participante i é
um conjunto $S \subseteq [m]$ que maximiza sua utilidade.

A utilidade de i para S é $u_i(S) = v_i(S) - \sum_{j \in S} p_j$.

Ou seja, uma demanda é um conjunto S tal que

$$u_i(S) \;=\; v_i(S) - \sum_{j \in S} p_j \;\geq\; v_i(S') - \sum_{j \in S'} p_j,$$

para qualquer $S' \subseteq [m]$.

Equilíbrio Walrasiano: preços em que todo participante recebe
uma demanda e itens não vendidos têm preço nulo.

Equilíbrio Walrasiano

Exemplo onde não há equilíbrio Walrasiano:

Dois itens, a e b, e dois jogadores, Alice e Bob, com as seguintes valorações

	$v(a)$	$v(b)$	$v(ab)$
Alice	2	2	2
Bob	0	0	3

Equilíbrio Walrasiano

Exemplo onde não há equilíbrio Walrasiano:

Dois itens, a e b, e dois jogadores, Alice e Bob, com as seguintes valorações

	$v(a)$	$v(b)$	$v(ab)$
Alice	2	2	2
Bob	0	0	3

O ótimo é dar $\{a, b\}$ para Bob e deixar Alice sem nada.

Equilíbrio Walrasiano

Exemplo onde não há equilíbrio Walrasiano:

Dois itens, a e b, e dois jogadores, Alice e Bob, com as seguintes valorações

	$v(a)$	$v(b)$	$v(ab)$
Alice	2	2	2
Bob	0	0	3

O ótimo é dar $\{a, b\}$ para Bob e deixar Alice sem nada.

Mas o conjunto vazio só é uma demanda para Alice se os preços de a e b forem pelo menos 2 e o preço de $\{a, b\}$ for pelo menos 4.

Equilíbrio Walrasiano

Exemplo onde não há equilíbrio Walrasiano:

Dois itens, a e b, e dois jogadores, Alice e Bob, com as seguintes valorações

	$v(a)$	$v(b)$	$v(ab)$
Alice	2	2	2
Bob	0	0	3

O ótimo é dar $\{a, b\}$ para Bob e deixar Alice sem nada.

Mas o conjunto vazio só é uma demanda para Alice se os preços de a e b forem pelo menos 2 e o preço de $\{a, b\}$ for pelo menos 4.

Mas, neste caso, $\{a, b\}$ não seria uma demanda para Bob.

Equilíbrio Walrasiano

Dados preços $p_1, \ldots, p_m$ para os itens,
uma demanda para o participante i é
um conjunto $S \subseteq [m]$ que maximize sua utilidade.

A utilidade de i para S é $u_i(S) = v_i(S) - \sum_{j \in S} p_j$.

Equilíbrio Walrasiano: preços em que todo participante
recebe uma demanda e itens não vendidos têm preço nulo.

Equilíbrio Walrasiano

Dados preços $p_1, \ldots, p_m$ para os itens,
uma demanda para o participante i é
um conjunto $S \subseteq [m]$ que maximize sua utilidade.

A utilidade de i para S é $u_i(S) = v_i(S) - \sum_{j \in S} p_j$.

Equilíbrio Walrasiano: preços em que todo participante recebe uma demanda e itens não vendidos têm preço nulo.

Primeiro Teorema do Bem-Estar Social: Se existe equilíbrio Walrasiano, então ele maximiza o bem-estar social.

Equilíbrio Walrasiano

Dados preços $p_1, \dots, p_m$ para os itens,
uma demanda para o participante i é
um conjunto $S \subseteq [m]$ que maximize sua utilidade.

A utilidade de i para S é $u_i(S) = v_i(S) - \sum_{j \in S} p_j$.

Equilíbrio Walrasiano: preços em que todo participante recebe uma demanda e itens não vendidos têm preço nulo.

Primeiro Teorema do Bem-Estar Social: Se existe equilíbrio Walrasiano, então ele maximiza o bem-estar social.

Ou seja, ele é um ótimo do LP!

Equilíbrio Walrasiano

Dados preços $p_1, \ldots, p_m$ para os itens,
uma demanda para o participante i é
um conjunto $S \subseteq [m]$ que maximize sua utilidade
$u_i(S) = v_i(S) - \sum_{j \in S} p_j$.

Equilíbrio Walrasiano: preços em que todo participante recebe uma demanda e itens não vendidos têm preço nulo.

Primeiro Teorema do Bem-Estar Social: Se existe equilíbrio Walrasiano, então ele maximiza o bem-estar social.

Equilíbrio Walrasiano

Dados preços $p_1, \ldots, p_m$ para os itens,
uma demanda para o participante i é
um conjunto $S \subseteq [m]$ que maximize sua utilidade
$u_i(S) = v_i(S) - \sum_{j \in S} p_j$.

Equilíbrio Walrasiano: preços em que todo participante recebe uma demanda e itens não vendidos têm preço nulo.

Primeiro Teorema do Bem-Estar Social: Se existe equilíbrio Walrasiano, então ele maximiza o bem-estar social.

Segundo Teorema do Bem-Estar Social:
Se existe solução inteira ótima para o LP,
então ela corresponde a um equilíbrio Walrasiano.

Equilíbrio Walrasiano

Dados preços $p_1, \ldots, p_m$ para os itens,
uma demanda para o participante i é
um conjunto $S \subseteq [m]$ que maximize sua utilidade
$u_i(S) = v_i(S) - \sum_{j \in S} p_j$.

Equilíbrio Walrasiano: preços em que todo participante recebe uma demanda e itens não vendidos têm preço nulo.

Primeiro Teorema do Bem-Estar Social: Se existe equilíbrio Walrasiano, então ele maximiza o bem-estar social.

Segundo Teorema do Bem-Estar Social:
Se existe solução inteira ótima para o LP,
então ela corresponde a um equilíbrio Walrasiano.

Equilíbrio Walrasiano

Dados preços $p_1, \ldots, p_m$ para os itens,
uma demanda para o participante i é
um conjunto $S \subseteq [m]$ que maximize sua utilidade
$u_i(S) = v_i(S) - \sum_{j \in S} p_j$.

Equilíbrio Walrasiano: preços em que todo participante recebe uma demanda e itens não vendidos têm preço nulo.

Primeiro Teorema do Bem-Estar Social: Se existe equilíbrio Walrasiano, então ele maximiza o bem-estar social.

Segundo Teorema do Bem-Estar Social:
Se existe solução inteira ótima para o LP,
então ela corresponde a um equilíbrio Walrasiano.

Corolário: Um equilíbrio Walrasiano existe
sse o LP tem solução inteira.

Linguagens de apostas

Átomos: ofertas (S, p) (como em objetivo único)

Linguagens de apostas

Átomos: ofertas (S, p) (como em objetivo único)

OR: valoração (S_1, p_1) $\mathrm{OR} \cdots \mathrm{OR}$ (S_k, p_k)

$v(S)$: valor máximo das coleções válidas (disjuntas de S_i's)
(o valor da coleção é a soma dos valores)

Linguagens de apostas

Átomos: ofertas (S, p) (como em objetivo único)

OR: valoração $(S_1, p_1)\ \mathrm{OR} \cdots \mathrm{OR}\ (S_k, p_k)$

$v(S)$: valor máximo das coleções válidas (disjuntas de S_i's)
(o valor da coleção é a soma dos valores)

XOR: valoração $(S_1, p_1)\ \mathrm{XOR} \cdots \mathrm{XOR}\ (S_k, p_k)$

$v(S)$: $\max\{p_i : S_i \subseteq S\}$

Linguagens de apostas

Átomos: ofertas (S, p) (como em objetivo único)

OR: valoração $(S_1, p_1) \text{ OR} \cdots \text{OR } (S_k, p_k)$

$v(S)$: valor máximo das coleções válidas (disjuntas de S_i's)
(o valor da coleção é a soma dos valores)

XOR: valoração $(S_1, p_1) \text{ XOR} \cdots \text{XOR } (S_k, p_k)$

$v(S)$: $\max\{p_i : S_i \subseteq S\}$

Qualquer valoração pode ser escrita com o XOR.
(Lembre-se do free-disposal.)

Linguagens de apostas

Átomos: ofertas (S, p) (como em objetivo único)

OR: valoração (S_1, p_1) $\mathrm{OR} \cdots \mathrm{OR}$ (S_k, p_k)

$v(S)$: valor máximo das coleções válidas (disjuntas de S_i's)
(o valor da coleção é a soma dos valores)

XOR: valoração (S_1, p_1) $\mathrm{XOR} \cdots \mathrm{XOR}$ (S_k, p_k)

$v(S)$: $\max\{p_i : S_i \subseteq S\}$

Qualquer valoração pode ser escrita com o XOR.
(Lembre-se do free-disposal.)

OR representam valorações superaditivas:
$v(S \cup T) \geq v(S) + v(T)$ para S e T disjuntos

Combinações de OR e XOR

Dadas valorações u e v, temos as seguintes valorações:

- $(v \text{ XOR } u)(S) := \max\{v(S), u(S)\}$;
- $(v \text{ OR } u)(S) := \max\{v(R) + u(T) : R, T \subseteq S,\ R \cap T = \emptyset\}$.

Combinações de OR e XOR

Dadas valorações u e v, temos as seguintes valorações:

- $(v \text{ XOR } u)(S) := \max\{v(S), u(S)\}$;
- $(v \text{ OR } u)(S) := \max\{v(R) + u(T) : R, T \subseteq S,\ R \cap T = \emptyset\}$.

Exemplo: Fórmula $((\{a, b\}, 3) \text{ XOR } (\{c\}, 2)) \text{ OR } (\{d\}, 5)$ atribui valor 3 para o conjunto $\{a, b, c\}$ e 8 para $\{a, b, d\}$.

Combinações de OR e XOR

Dadas valorações u e v, temos as seguintes valorações:

- $(v \text{ XOR } u)(S) := \max\{v(S), u(S)\}$;
- $(v \text{ OR } u)(S) := \max\{v(R) + u(T) : R, T \subseteq S,\ R \cap T = \emptyset\}$.

Exemplo: Fórmula $((\{a, b\}, 3) \text{ XOR } (\{c\}, 2)) \text{ OR } (\{d\}, 5)$ atribui valor 3 para o conjunto $\{a, b, c\}$ e 8 para $\{a, b, d\}$.

Valoração simétrica: $v(S)$ depende apenas de $|S|$.

Valoração decrescente (*downward sloping*):

$$v(S) = \sum_{j=1}^{|S|} p_j, \text{ onde } p_1 \geq p_2 \geq \cdots \geq p_m.$$

Combinações de OR e XOR

Dadas valorações u e v, temos as seguintes valorações:

- $(v \text{ XOR } u)(S) := \max\{v(S), u(S)\}$;
- $(v \text{ OR } u)(S) := \max\{v(R) + u(T) : R, T \subseteq S,\ R \cap T = \emptyset\}$.

Exemplo: Fórmula $((\{a, b\}, 3) \text{ XOR } (\{c\}, 2)) \text{ OR } (\{d\}, 5)$ atribui valor 3 para o conjunto $\{a, b, c\}$ e 8 para $\{a, b, d\}$.

Valoração simétrica: $v(S)$ depende apenas de $|S|$.

Valoração decrescente (*downward sloping*):

$$v(S) = \sum_{j=1}^{|S|} p_j, \text{ onde } p_1 \geq p_2 \geq \cdots \geq p_m.$$

Lema: Fórmulas OX/XOR podem representar qualquer valoração simétrica decrescente sobre m itens com tamanho m^2.

Redução de XOR para OR

Podemos simular XOR usando OR:

Valoração $(S_1, p_1)\ \mathrm{XOR} \cdots \mathrm{XOR}\ (S_k, p_k)$:

$$v(S) = \max\{p_i : S_i \subseteq S\}$$

Redução de XOR para OR

Podemos simular XOR usando OR:

Valoração $(S_1, p_1)\ \mathrm{XOR} \cdots \mathrm{XOR}\ (S_k, p_k)$:

$$v(S) = \max\{p_i : S_i \subseteq S\}$$

Adicione um item artificial x e a valoração
$(S_1', p_1)\ \mathrm{OR} \cdots \mathrm{OR}\ (S_k', p_k)$ com $S_i' = S_i \cup \{x\}$ imita a anterior.

Redução de XOR para OR

Podemos simular XOR usando OR:

Valoração $(S_1, p_1)\ \mathrm{XOR} \cdots \mathrm{XOR}\ (S_k, p_k)$:

$$v(S) = \max\{p_i : S_i \subseteq S\}$$

Adicione um item artificial x e a valoração $(S'_1, p_1)\ \mathrm{OR} \cdots \mathrm{OR}\ (S'_k, p_k)$ com $S'_i = S_i \cup \{x\}$ imita a anterior.

Valoração OR^* para i: fórmula OR com os itens $[m] \cup D_i$, onde D_i é o conjunto de itens artificiais usados apenas por i.

Redução de XOR para OR

Podemos simular XOR usando OR:

Valoração $(S_1, p_1)\ \mathrm{XOR} \cdots \mathrm{XOR}\ (S_k, p_k)$:

$$v(S) = \max\{p_i : S_i \subseteq S\}$$

Adicione um item artificial x e a valoração $(S'_1, p_1)\ \mathrm{OR} \cdots \mathrm{OR}\ (S'_k, p_k)$ com $S'_i = S_i \cup \{x\}$ imita a anterior.

Valoração OR^* para i: fórmula OR com os itens $[m] \cup D_i$, onde D_i é o conjunto de itens artificiais usados apenas por i.

Teorema: Valorações OR/XOR de tamanho s podem ser representadas por valorações OR^* de tamanho s usando no máximo s^2 itens artificiais.

Consequências

Qualquer valoração OR/XOR pode ser escrita com OR^*.

Consequências

Qualquer valoração OR/XOR pode ser escrita com OR^*.

Qualquer algoritmo de determinação de vencedores
pode olhar uma valoração OR^*
como uma valoração OR em mais itens.

Consequências

Qualquer valoração OR/XOR pode ser escrita com OR^*.

Qualquer algoritmo de determinação de vencedores
pode olhar uma valoração OR^*
como uma valoração OR em mais itens.

E qualquer valoração OR pode ser olhada como
uma coleção de participantes de objetivo único.

Consequências

Qualquer valoração OR/XOR pode ser escrita com OR^*.

Qualquer algoritmo de determinação de vencedores
pode olhar uma valoração OR^*
como uma valoração OR em mais itens.

E qualquer valoração OR pode ser olhada como
uma coleção de participantes de objetivo único.

Então algoritmos projetados
para participantes com objetivo único podem ser aplicados
para participantes com valorações OR/XOR!