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 $i \neq j$.

Objetivo: encontrar alocação que
maximize o 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 $i \neq j$.

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

Vamos descrever esse problema como um IP.

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.

MIP: encontrar x que
maximizem $\sum_{i=1}^n \sum_{S \subseteq [m]} x_{i,S} v_i(S)$
sujeitos 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 $i \in [n]$
$x_{i,S} \in \{0, 1\}$ para cada $i \in [n]$ e cada $S \subseteq [m]$.

Formulação linear inteira

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

MIP: encontrar x que

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

sujeitos 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 $i \in [n]$

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

- Primeira 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

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

sujeitos 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 $i \in [n]$

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

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

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

sujeitos 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 $i \in [n]$

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

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

Relaxação linear inteira

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

LP: encontrar x que

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

sujeitos 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 $S \subseteq [m]$.

Relaxação linear inteira

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

LP: encontrar x que

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

sujeitos 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 $i \in [n]$

$x_{i,S} \geq 0$ 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

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

sujeitos 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 $i \in [n]$

$x_{i,S} \geq 0$ 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 $i \in [n]$ e $S \subseteq [m]$

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

Relaxação linear inteira

LP: encontrar x que

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

sujeitos 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 $i \in [n]$

$x_{i,S} \geq 0$ 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 $S \subseteq [m]$

$u_i \geq 0$, $p_j \geq 0$ 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 estas interpretações.

Equilíbrio Walrasiano

Dado 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.

Equilíbrio Walrasiano

Dado 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

Dado 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 $u_i(S) = v_i(S) - \sum_{j \in S} p_j$.

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

Dado 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 $u_i(S) = v_i(S) - \sum_{j \in S} p_j$.

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

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 é economicamente eficiente.

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 é economicamente eficiente.

Ou seja, ele é um ótimo do LP!

Equilíbrio Walrasiano

Dado 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 é economicamente eficiente.

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

Equilíbrio Walrasiano

Dado 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 é economicamente eficiente.

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.

Relaxação linear inteira

LP: encontrar x que
maximizem $\sum_{i=1}^{n} \sum_{S \subseteq [m]} x_{i,S} v_i(S)$
sujeitos 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 $i \in [n]$
$x_{i,S} \geq 0$ 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 $S \subseteq [m]$
$u_i \geq 0$, $p_j \geq 0$ 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 estas interpretações.

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.

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.

Neste caso porém o valor de $\{a, b\}$ seria 4 e $\{a, b\}$ não seria uma demanda para Bob.

Linguagens de apostas

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

Linguagens de apostas

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

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

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

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

$v(S)$: 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.

Linguagens de apostas

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

OR: valoração (S_1, p_1) OR $\cdots$ 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) XOR $\cdots$ XOR (S_k, p_k)

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

Qualquer valoração pode ser escrita com o XOR.

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:

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

Podemos simular XOR usando OR:

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

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

Combinações de OR e XOR

Dadas valorações u e v, temos as seguintes valorações:

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

Podemos simular XOR usando OR:

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

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

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

Combinações de OR e XOR

Dadas valorações u e v, temos as seguintes valorações:

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

Podemos simular XOR usando OR:

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

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

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

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

Consequências

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

Consequências

Qualquer valoração XOR/OR 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 XOR/OR 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 XOR/OR 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 XOR/OR!

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 $i \neq j$.

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

Leilões ascendentes

Subclasse dos chamados leilões iterativos.

Leilões ascendentes

Subclasse dos chamados leilões iterativos.

Preços só podem subir.

Leilões ascendentes

Subclasse dos chamados leilões iterativos.

Preços só podem subir.

Ideia:

Leiloeiro começa com os preços nulos (ou no mínimo).

Leilões ascendentes

Subclasse dos chamados leilões iterativos.

Preços só podem subir.

Ideia:

Leiloeiro começa com os preços nulos (ou no mínimo).

Cada participante aposta em um dos seus conjuntos mais desejados de itens, ou seja, em uma demanda.

Leilões ascendentes

Subclasse dos chamados leilões iterativos.

Preços só podem subir.

Ideia:

Leiloeiro começa com os preços nulos (ou no mínimo).

Cada participante aposta em um dos seus conjuntos mais desejados de itens, ou seja, em uma demanda.

Leiloeiro aumenta alguns preços de algum modo, até decidir por uma alocação.

Leilões ascendentes

Preços por itens

Leilões ascendentes

Preços por itens

Valoração v satisfaz a propriedade dos substitutos se, para todo par de preços dos itens $q \geq p$,

Leilões ascendentes

Preços por itens

Valoração v satisfaz a propriedade dos substitutos se, para todo par de preços dos itens $q \geq p$, a demanda com preços q contem todos os itens da demanda com preço p que mantiveram seus preços.

Leilões ascendentes

Preços por itens

Valoração v satisfaz a propriedade dos substitutos se, para todo par de preços dos itens $q \geq p$, a demanda com preços q contem todos os itens da demanda com preço p que mantiveram seus preços.

Formalmente: para todo $A \in \arg\min_S \{v(S) - \sum_{j \in S} p_j\}$, existe $D \in \arg\min_S \{v(S) - \sum_{j \in S} q_j\}$ tal que $D \supseteq \{j \in A : p_j = q_j\}$.

Leilões ascendentes

Preços por itens

Valoração v satisfaz a propriedade dos substitutos se, para todo par de preços dos itens $q \geq p$, a demanda com preços q contem todos os itens da demanda com preço p que mantiveram seus preços.

Formalmente: para todo $A \in \arg\min_S\{v(S) - \sum_{j \in S} p_j\}$, existe $D \in \arg\min_S\{v(S) - \sum_{j \in S} q_j\}$ tal que $D \supseteq \{j \in A : p_j = q_j\}$.

Apenas itens cujo preço subiu podem sair da demanda.

Leilões ascendentes

Preços por itens

Valoração v satisfaz a propriedade dos substitutos se, para todo par de preços dos itens $q \geq p$, a demanda com preços q contem todos os itens da demanda com preço p que mantiveram seus preços.

Formalmente: para todo $A \in \arg\min_S \{v(S) - \sum_{j \in S} p_j\}$, existe $D \in \arg\min_S \{v(S) - \sum_{j \in S} q_j\}$ tal que $D \supseteq \{j \in A : p_j = q_j\}$.

Apenas itens cujo preço subiu podem sair da demanda.

Todo item está na demanda de algum participante.

Leilão de preços ascendentes de itens

PREÇOSASCENDENTES (m, n)

para $j \leftarrow 1$ **até** m **faça** $p_j \leftarrow 0$
para $i \leftarrow 1$ **até** n **faça** $S_i \leftarrow \emptyset$
para $i \leftarrow 1$ **até** n **faça**
    seja D_i uma demanda de i para preços
        p_j para $j \in S_i$ e $p_j + \epsilon$ para $j \notin S_i$
    se $D_i \neq S_i$ para algum i
        então para $j \in D_i \setminus S_i$ **faça** $p_j \leftarrow p_j + \epsilon$
            $S_i \leftarrow D_i$
            para $k \leftarrow 1$ **até** n **faça**
                se $k \neq i$ **então** $S_k \leftarrow S_k \setminus D_i$
devolva $S_1, \ldots, S_n$

Leilão de preços ascendentes de itens

PreçosAscendentes (m, n)

para $j \leftarrow 1$ **até** m **faça** $p_j \leftarrow 0$
para $i \leftarrow 1$ **até** n **faça** $S_i \leftarrow \emptyset$
para $i \leftarrow 1$ **até** n **faça**
    seja D_i uma demanda de i para preços
        p_j para $j \in S_i$ e $p_j + \epsilon$ para $j \notin S_i$
    se $D_i \neq S_i$ para algum i
        então para $j \in D_i \setminus S_i$ **faça** $p_j \leftarrow p_j + \epsilon$
            $S_i \leftarrow D_i$
            para $k \leftarrow 1$ **até** n **faça**
                se $k \neq i$ **então** $S_k \leftarrow S_k \setminus D_i$
devolva $S_1, \ldots, S_n$

Termina quando nenhum item que está tentativamente com um participante está na demanda de outro.

Leilão de preços ascendentes de itens

PREÇOSASCENDENTES (m, n)

para $j \leftarrow 1$ **até** m **faça** $p_j \leftarrow 0$
para $i \leftarrow 1$ **até** n **faça** $S_i \leftarrow \emptyset$
para $i \leftarrow 1$ **até** n **faça**
    seja D_i uma demanda de i para preços
        p_j para $j \in S_i$ e $p_j + \epsilon$ para $j \notin S_i$
    se $D_i \neq S_i$ para algum i
        então para $j \in D_i \setminus S_i$ **faça** $p_j \leftarrow p_j + \epsilon$
            $S_i \leftarrow D_i$
            para $k \leftarrow 1$ **até** n **faça**
                se $k \neq i$ **então** $S_k \leftarrow S_k \setminus D_i$
devolva $S_1, \ldots, S_n$

Termina próximo a um equilíbrio Walrasiano.

Equilíbrio ϵ-Walrasiano

Alocação $S_1, \ldots, S_n$ e preços $p_1, \ldots, p_m$ são equilíbrio ϵ-Walrasiano se

- $\cup_i S_i \supseteq \{j : p_j > 0\}$ e
- S_i para cada i é uma demanda de i para os preços p_j para $j \in S_i$ e $p_j + \epsilon$ para $j \notin S_i$.

Equilíbrio ϵ-Walrasiano

Alocação $S_1, \ldots, S_n$ e preços $p_1, \ldots, p_m$ são equilíbrio ϵ-Walrasiano se

- $\cup_i S_i \supseteq \{j : p_j > 0\}$ e
- S_i para cada i é uma demanda de i para os preços p_j para $j \in S_i$ e $p_j + \epsilon$ para $j \notin S_i$.

Teorema: Para participantes com valoração de substitutos, o algoritmo PREÇOSASCENDENTES produz um equilíbrio ϵ-Walrasiano. Assim a alocação produzida atinge um bem-estar social a $n\epsilon$ do bem-estar social ótimo.

Equilíbrio ϵ-Walrasiano

Alocação $S_1, \ldots, S_n$ e preços $p_1, \ldots, p_m$
são equilíbrio ϵ-Walrasiano se

- $\cup_i S_i \supseteq \{j : p_j > 0\}$ e
- S_i para cada i é uma demanda de i para os preços p_j para $j \in S_i$ e $p_j + \epsilon$ para $j \notin S_i$.

Teorema: Para participantes com valoração de substitutos, o algoritmo PreçosAscendentes produz um equilíbrio ϵ-Walrasiano. Assim a alocação produzida atinge um bem-estar social a $n\epsilon$ do bem-estar social ótimo.

Afirmação:
Para todo participante i, durante todo o algoritmo, $S_i \subseteq D_i$.

Equilíbrio ϵ-Walrasiano

Alocação $S_1, \ldots, S_n$ e preços $p_1, \ldots, p_m$ são equilíbrio ϵ-Walrasiano se

- $\cup_i S_i \supseteq \{j : p_j > 0\}$ e
- S_i para cada i é uma demanda de i para os preços p_j para $j \in S_i$ e $p_j + \epsilon$ para $j \notin S_i$.

Teorema: Para participantes com valoração de substitutos, o algoritmo PREÇOSASCENDENTES produz um equilíbrio ϵ-Walrasiano. Assim a alocação produzida atinge um bem-estar social a $n\epsilon$ do bem-estar social ótimo.

Infelizmente, não é à prova de estratégia.

Exemplo

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

	$v(a)$	$v(b)$	$v(ab)$
Alice	4	4	4
Bob	5	5	5

Exemplo

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

	$v(a)$	$v(b)$	$v(ab)$
Alice	4	4	4
Bob	5	5	5

Para estas valorações, o leilão do algoritmo termina no equilíbrio Walrasiano, com preços $p_a = p_b = 4$, e Bob recebendo os dois itens, com utilidade 2.

Exemplo

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

	$v(a)$	$v(b)$	$v(ab)$
Alice	4	4	4
Bob	5	5	5

Para estas valorações, o leilão do algoritmo termina no equilíbrio Walrasiano, com preços $p_a = p_b = 4$, e Bob recebendo os dois itens, com utilidade 2.

Se Bob declarasse sua valoração apenas por a, então o algoritmo deixaria a com Bob e b com Alice, os dois pelo preço nulo.