Alocação de casas

Alocação de bens indivisíveis

n agentes, cada um com uma única casa,
e uma ordem de preferência em todas as casas

Alocação de casas

Alocação de bens indivisíveis

n agentes, cada um com uma única casa,
e uma ordem de preferência em todas as casas

Objetivo: realocar as casas de um modo apropriado

Alocação de casas

Alocação de bens indivisíveis

n agentes, cada um com uma única casa,
e uma ordem de preferência em todas as casas

Objetivo: realocar as casas de um modo apropriado

As ordens de preferência são arbitrárias,
mas a odem sobre as alocações é restrita:

Alocação de casas

Alocação de bens indivisíveis

n agentes, cada um com uma única casa,
e uma ordem de preferência em todas as casas

Objetivo: realocar as casas de um modo apropriado

As ordens de preferência são arbitrárias,
mas a odem sobre as alocações é restrita:

Qualquer alocação que deixa o agente
com uma mesma casa é equivalente.

Alocação de casas

Alocação de bens indivisíveis

n agentes, cada um com uma única casa,
e uma ordem de preferência em todas as casas

Objetivo: realocar as casas de um modo apropriado

As ordens de preferência são arbitrárias,
mas a odem sobre as alocações é restrita:

Qualquer alocação que deixa o agente
com uma mesma casa é equivalente.

Teorema de Gibbard-Satterthwaite: Se f é uma função de escolha social à prova de estratégia sobre A, com $|A| \geq 3$, então f é uma ditadura.

Alocação de casas

Numere os agentes de 1 a n.

Uma alocação a é uma permutação de 1 a n:
a_i é a casa que vai ficar com o agente i.

Alocação de casas

Numere os agentes de 1 a n.

Uma alocação a é uma permutação de 1 a n:
a_i é a casa que vai ficar com o agente i.

$\succ_i$: ordem de preferência do agente i para as casas

Alocação de casas

Numere os agentes de 1 a n.

Uma alocação a é uma permutação de 1 a n:
a_i é a casa que vai ficar com o agente i.

$\succ_i$: ordem de preferência do agente i para as casas

A: conjunto de todas as alocações

Para $S \subseteq [n]$, seja $A(S) \; = \; \{z \in A : z_i \in S, \forall i \in S\}$.

Alocação de casas

Numere os agentes de 1 a n.

Uma alocação a é uma permutação de 1 a n:
a_i é a casa que vai ficar com o agente i.

$\succ_i$: ordem de preferência do agente i para as casas

A: conjunto de todas as alocações

Para $S \subseteq [n]$, seja $A(S) \ = \ \{z \in A : z_i \in S, \forall i \in S\}$.

$A(S)$: alocações onde agentes de S trocam casas entre si.

Alocação de casas

Numere os agentes de 1 a n.

Uma alocação a é uma permutação de 1 a n:
a_i é a casa que vai ficar com o agente i.

$\succ_i$: ordem de preferência do agente i para as casas

A: conjunto de todas as alocações

Para $S \subseteq [n]$, seja $A(S) \ = \ \{z \in A : z_i \in S, \forall i \in S\}$.

$A(S)$: alocações onde agentes de S trocam casas entre si.

S é uma coalisão bloqueadora para uma alocação a em A
se existe z em $A(S)$ tq $z_i \succ_i a_i$ ou $z_i = a_i$ para todo i em S
e $z_j \succ_j a_j$ para pelo menos um j em S.

Alocação de casas

Numere os agentes de 1 a n.

Uma alocação a é uma permutação de 1 a n:
a_i é a casa que vai ficar com o agente i.

$\succ_i$: ordem de preferência do agente i para as casas

A: conjunto de todas as alocações

Para $S \subseteq [n]$, seja $A(S) \; = \; \{z \in A : z_i \in S, \forall i \in S\}$.

$A(S)$: alocações onde agentes de S trocam casas entre si.

S é uma coalisão bloqueadora para uma alocação a em A
se existe z em $A(S)$ tq $z_i \succ_i a_i$ ou $z_i = a_i$ para todo i em S
e $z_j \succ_j a_j$ para pelo menos um j em S.

Core: conjunto de alocações sem coalisão bloqueadora.

Algoritmo TTC

TTC: top trading cycle

Grafo orientado:

- um vértice para cada agente
- arco de i para j se a casa de i é a favorita de j

Algoritmo TTC

TTC: top trading cycle

Grafo orientado:

- um vértice para cada agente
- arco de i para j se a casa de i é a favorita de j

Cada vértice tem grau de entrada 1:
há um circuito em cada componente.

Algoritmo TTC

TTC: top trading cycle

Grafo orientado:

- um vértice para cada agente
- arco de i para j se a casa de i é a favorita de j

Cada vértice tem grau de entrada 1:
há um circuito em cada componente.

Circuitos são disjuntos nos vértices então.

Algoritmo TTC

TTC: top trading cycle

Grafo orientado:

- um vértice para cada agente
- arco de i para j se a casa de i é a favorita de j

Cada vértice tem grau de entrada 1:
há um circuito em cada componente.

Circuitos são disjuntos nos vértices então.

Seja $N = [n]$ e N_1 o conjunto de agentes em circuitos.

Execute as permutas de casas dos circuitos em N_1.

Algoritmo TTC

TTC: top trading cycle

Grafo orientado:

- um vértice para cada agente
- arco de i para j se a casa de i é a favorita de j

Cada vértice tem grau de entrada 1:
há um circuito em cada componente.

Circuitos são disjuntos nos vértices então.

Seja $N = [n]$ e N_1 o conjunto de agentes em circuitos.

Execute as permutas de casas dos circuitos em N_1.

Repita: novo grafo com agentes de $N \setminus N_1$, e
arco de i para j se a casa de i é a favorita de j em $N \setminus N_1$.

Algoritmo TTC

Grafo orientado:

- um vértice para cada agente
- arco de i para j se a casa de i é a favorita de j

Cada vértice tem grau de entrada 1:
há um circuito em cada componente.

Seja $N = [n]$ e N_1 o conjunto de agentes em circuitos.

Execute as permutas de casas dos circuitos em N_1.

Repita: novo grafo com agentes de $N \setminus N_1$, e
arco de i para j se a casa de i é a favorita de j em $N \setminus N_1$.

Algoritmo TTC

Grafo orientado:

- um vértice para cada agente
- arco de i para j se a casa de i é a favorita de j

Cada vértice tem grau de entrada 1:
há um circuito em cada componente.

Seja $N = [n]$ e N_1 o conjunto de agentes em circuitos.

Execute as permutas de casas dos circuitos em N_1.

Repita: novo grafo com agentes de $N \setminus N_1$, e
arco de i para j se a casa de i é a favorita de j em $N \setminus N_1$.

N_2: conjunto de agentes em circuitos do novo grafo.

Execute as permutas de casas dos circuitos em N_2.

Algoritmo TTC

Teorema: O core do problema da alocação de casas consiste exatamente de uma alocação.

Algoritmo TTC

Teorema: O core do problema da alocação de casas consiste exatamente de uma alocação.

Prova feita na aula.

Algoritmo TTC

Teorema: O core do problema da alocação de casas consiste exatamente de uma alocação.

Prova feita na aula.

Mas esse algoritmo garante que os agentes declararão as suas verdadeiras preferências?

Algoritmo TTC

Teorema: O core do problema da alocação de casas consiste exatamente de uma alocação.

Prova feita na aula.

Mas esse algoritmo garante que os agentes declararão as suas verdadeiras preferências?

Teorema: O mecanismo TTC é à prova de estratégia.

Algoritmo TTC

Teorema: O core do problema da alocação de casas consiste exatamente de uma alocação.

Prova feita na aula.

Mas esse algoritmo garante que os agentes declararão as suas verdadeiras preferências?

Teorema: O mecanismo TTC é à prova de estratégia.

Prova feita na aula.

Casamentos estáveis

Atribuição de estudantes a universidades,
médicos a programas de residências.

Casamentos estáveis

Atribuição de estudantes a universidades,
médicos a programas de residências.

H - conjunto de homens

M - conjunto de mulheres

Casamentos estáveis

Atribuição de estudantes a universidades,
médicos a programas de residências.

H - conjunto de homens

M - conjunto de mulheres

Cada homem tem ordem estrita de preferência sobre M.

Casamentos estáveis

Atribuição de estudantes a universidades,
médicos a programas de residências.

H - conjunto de homens

M - conjunto de mulheres

Cada homem tem ordem estrita de preferência sobre M.

Adicione homem/mulher fictício para
representar a possibilidade de ficar solteiro.

Assim $|H| = |M|$.

Casamentos estáveis

Atribuição de estudantes a universidades,
médicos a programas de residências.

H - conjunto de homens

M - conjunto de mulheres

Cada homem tem ordem estrita de preferência sobre M.

Adicione homem/mulher fictício para
representar a possibilidade de ficar solteiro.

Assim $|H| = |M|$.

Emparelhamento de H em M:
alocação de homens a mulheres.

Casamentos estáveis

H - conjunto de homens M - conjunto de mulheres

Cada homem tem ordem estrita de preferência sobre M.

Emparelhamento de H em M:
alocação de homens a mulheres.

Casamentos estáveis

H - conjunto de homens M - conjunto de mulheres

Cada homem tem ordem estrita de preferência sobre M.

Emparelhamento de H em M:
alocação de homens a mulheres.

Um emparelhamento é instável se
existem homens h e h' e mulheres m e m' tq
m é emparelhada com h, m' com h', mas $m' \succ_h m$ e $h \succ_{m'} h'$.

Casamentos estáveis

H - conjunto de homens M - conjunto de mulheres

Cada homem tem ordem estrita de preferência sobre M.

Emparelhamento de H em M:
alocação de homens a mulheres.

Um emparelhamento é instável se
existem homens h e h' e mulheres m e m' tq
m é emparelhada com h, m' com h', mas $m' \succ_h m$ e $h \succ_{m'} h'$.

Neste caso, h e m' prefeririam se casar um com o outro.

Casamentos estáveis

H - conjunto de homens $\qquad$ M - conjunto de mulheres

Cada homem tem ordem estrita de preferência sobre M.

Emparelhamento de H em M:
alocação de homens a mulheres.

Um emparelhamento é instável se
existem homens h e h' e mulheres m e m' tq
m é emparelhada com h, m' com h', mas $m' \succ_h m$ e $h \succ_{m'} h'$.

Neste caso, h e m' preferiam se casar um com o outro.

O par (h, m') é um par bloqueador,
e emparelhamento é estável se não tem par bloqueador.

Casamentos estáveis: exemplo

Ordens de preferências para $n = 3$:

Casamentos estáveis: exemplo

Ordens de preferências para $n = 3$:

$\succ_{h_1}$	$\succ_{h_2}$	$\succ_{h_3}$	$\succ_{m_1}$	$\succ_{m_2}$	$\succ_{m_3}$
m_2	m_1	m_1	h_1	h_3	h_1
m_1	m_3	m_2	h_3	h_1	h_3
m_3	m_2	m_3	h_2	h_2	h_2

Casamentos estáveis: exemplo

Ordens de preferências para $n = 3$:

$\succ_{h_1}$	$\succ_{h_2}$	$\succ_{h_3}$	$\succ_{m_1}$	$\succ_{m_2}$	$\succ_{m_3}$
m_2	m_1	m_1	h_1	h_3	h_1
m_1	m_3	m_2	h_3	h_1	h_3
m_3	m_2	m_3	h_2	h_2	h_2

O emparelhamento $\{(h_1, m_1), (h_2, m_2), (h_3, m_3)\}$ é instável, pois (h_1, m_2) é um par bloqueador.

Casamentos estáveis: exemplo

Ordens de preferências para $n = 3$:

$\succ_{h_1}$	$\succ_{h_2}$	$\succ_{h_3}$	$\succ_{m_1}$	$\succ_{m_2}$	$\succ_{m_3}$
m_2	m_1	m_1	h_1	h_3	h_1
m_1	m_3	m_2	h_3	h_1	h_3
m_3	m_2	m_3	h_2	h_2	h_2

O emparelhamento $\{(h_1, m_1), (h_2, m_2), (h_3, m_3)\}$ é instável, pois (h_1, m_2) é um par bloqueador.

Já o emparelhamento $\{(h_1, m_1), (h_3, m_2), (h_2, m_3)\}$ é estável.

Casamentos estáveis: exemplo

Ordens de preferências para $n = 3$:

$\succ_{h_1}$	$\succ_{h_2}$	$\succ_{h_3}$	$\succ_{m_1}$	$\succ_{m_2}$	$\succ_{m_3}$
m_2	m_1	m_1	h_1	h_3	h_1
m_1	m_3	m_2	h_3	h_1	h_3
m_3	m_2	m_3	h_2	h_2	h_2

O emparelhamento $\{(h_1, m_1), (h_2, m_2), (h_3, m_3)\}$ é instável, pois (h_1, m_2) é um par bloqueador.

Já o emparelhamento $\{(h_1, m_1), (h_3, m_2), (h_2, m_3)\}$ é estável.

Dada as listas de preferências de todos, existe emparelhamento estável?

Como encontrá-lo, se existe?

Algoritmo da aceitação postergada

Versão com proposta masculina:

Algoritmo da aceitação postergada

Versão com proposta masculina:

Cada homem propõe à primeira mulher de sua lista.

Algoritmo da aceitação postergada

Versão com proposta masculina:

Cada homem propõe à primeira mulher de sua lista.

Cada mulher que recebeu mais de uma proposta recusa todas, exceto pela do homem mais alto em sua lista.
Para esse, posterga a sua resposta.

Algoritmo da aceitação postergada

Versão com proposta masculina:

Cada homem propõe à primeira mulher de sua lista.

Cada mulher que recebeu mais de uma proposta recusa todas, exceto pela do homem mais alto em sua lista.
Para esse, posterga a sua resposta.

Nova rodada:

Cada homem que teve sua proposta recusada propõe à próxima mulher de sua lista.

Algoritmo da aceitação postergada

Versão com proposta masculina:

Cada homem propõe à primeira mulher de sua lista.

Cada mulher que recebeu mais de uma proposta recusa todas, exceto pela do homem mais alto em sua lista.
Para esse, posterga a sua resposta.

Nova rodada:

Cada homem que teve sua proposta recusada propõe à próxima mulher de sua lista.

Cada mulher com mais de uma proposta recusa todas, exceto pela do homem mais alto em sua lista.

Eventualmente recusa proposta recebida em rodada anterior.

Algoritmo com proposta masculina

Cada homem propõe à primeira mulher de sua lista.

Cada mulher que recebeu mais de uma proposta recusa todas, exceto pela do homem mais alto em sua lista.
Para esse, posterga a sua resposta.

Cada homem que teve sua proposta recusada propõe à próxima mulher de sua lista.

Cada mulher com mais de uma proposta recusa todas, exceto pela do homem mais alto em sua lista.

Algoritmo com proposta masculina

Cada homem propõe à primeira mulher de sua lista.

Cada mulher que recebeu mais de uma proposta recusa todas,
exceto pela do homem mais alto em sua lista.
Para esse, posterga a sua resposta.

Cada homem que teve sua proposta recusada
propõe à próxima mulher de sua lista.

Cada mulher com mais de uma proposta recusa todas,
exceto pela do homem mais alto em sua lista.

Repete esse processo,
sempre progredindo nas listas dos homens.

Algoritmo com proposta masculina

Cada homem propõe à primeira mulher de sua lista.

Cada mulher que recebeu mais de uma proposta recusa todas,
exceto pela do homem mais alto em sua lista.
Para esse, posterga a sua resposta.

Cada homem que teve sua proposta recusada
propõe à próxima mulher de sua lista.

Cada mulher com mais de uma proposta recusa todas,
exceto pela do homem mais alto em sua lista.

Repete esse processo,
sempre progredindo nas listas dos homens.

O processo então termina (não mais que n^2 rodadas).

Algoritmo no exemplo

$\succ_{h_1}$	$\succ_{h_2}$	$\succ_{h_3}$	$\succ_{m_1}$	$\succ_{m_2}$	$\succ_{m_3}$
m_2	m_1	m_1	h_1	h_3	h_1
m_1	m_3	m_2	h_3	h_1	h_3
m_3	m_2	m_3	h_2	h_2	h_2

Algoritmo no exemplo

$\succ_{h_1}$	$\succ_{h_2}$	$\succ_{h_3}$	$\succ_{m_1}$	$\succ_{m_2}$	$\succ_{m_3}$
m_2	m_1	m_1	h_1	h_3	h_1
m_1	m_3	m_2	h_3	h_1	h_3
m_3	m_2	m_3	h_2	h_2	h_2

h_1 propõe para m_2
h_2 propõe para m_1
h_3 propõe para m_1

Algoritmo no exemplo

$\succ_{h_1}$	$\succ_{h_2}$	$\succ_{h_3}$	$\succ_{m_1}$	$\succ_{m_2}$	$\succ_{m_3}$
m_2	m_1	m_1	h_1	h_3	h_1
m_1	m_3	m_2	h_3	h_1	h_3
m_3	m_2	m_3	h_2	h_2	h_2

h_1 propõe para m_2
h_2 propõe para m_1
h_3 propõe para m_1

m_1 rejeita a proposta de h_2.

Algoritmo no exemplo

$\succ_{h_1}$	$\succ_{h_2}$	$\succ_{h_3}$	$\succ_{m_1}$	$\succ_{m_2}$	$\succ_{m_3}$
m_2	m_1	m_1	h_1	h_3	h_1
m_1	m_3	m_2	h_3	h_1	h_3
m_3	m_2	m_3	h_2	h_2	h_2

h_1 propõe para m_2
h_2 propõe para m_1
h_3 propõe para m_1

m_1 rejeita a proposta de h_2.

h_2 propõe para m_3, e o algoritmo termina.

Algoritmo no exemplo

$\succ_{h_1}$	$\succ_{h_2}$	$\succ_{h_3}$	$\succ_{m_1}$	$\succ_{m_2}$	$\succ_{m_3}$
m_2	m_1	m_1	h_1	h_3	h_1
m_1	m_3	m_2	h_3	h_1	h_3
m_3	m_2	m_3	h_2	h_2	h_2

h_1 propõe para m_2
h_2 propõe para m_1
h_3 propõe para m_1

m_1 rejeita a proposta de h_2.

h_2 propõe para m_3, e o algoritmo termina.

Emparelhamento produzido: $\{(h_1, m_2), (h_2, m_3), (h_3, m_1)\}$

Algoritmo no exemplo

$\succ_{h_1}$	$\succ_{h_2}$	$\succ_{h_3}$	$\succ_{m_1}$	$\succ_{m_2}$	$\succ_{m_3}$
m_2	m_1	m_1	h_1	h_3	h_1
m_1	m_3	m_2	h_3	h_1	h_3
m_3	m_2	m_3	h_2	h_2	h_2

h_1 propõe para m_2
h_2 propõe para m_1
h_3 propõe para m_1

m_1 rejeita a proposta de h_2.

h_2 propõe para m_3, e o algoritmo termina.

Emparelhamento produzido: $\{(h_1, m_2), (h_2, m_3), (h_3, m_1)\}$

Note que tal emparelhamento é estável!

Estabilidade do emparelhamento

Teorema: O emparelhamento produzido pelo algoritmo da aceitação postergada é estável.

Prova feita na aula.

Estabilidade do emparelhamento

Teorema: O emparelhamento produzido pelo algoritmo da aceitação postergada é estável.

Prova feita na aula.

No exemplo, aplicando a versão da proposta feminina, obtemos o emparelhamento $\{(h_1, m_1), (h_2, m_3), (h_3, m_2)\}$.

Estabilidade do emparelhamento

Teorema: O emparelhamento produzido pelo algoritmo da aceitação postergada é estável.

Prova feita na aula.

No exemplo, aplicando a versão da proposta feminina, obtemos o emparelhamento $\{(h_1, m_1), (h_2, m_3), (h_3, m_2)\}$.

Diferente do obtido pela proposta masculina!

Estabilidade do emparelhamento

Teorema: O emparelhamento produzido pelo algoritmo da aceitação postergada é estável.

Prova feita na aula.

No exemplo, aplicando a versão da proposta feminina, obtemos o emparelhamento $\{(h_1, m_1), (h_2, m_3), (h_3, m_2)\}$.

Diferente do obtido pela proposta masculina!

Há alguma diferença significativa entre estes emparelhamentos?

Estabilidade do emparelhamento

Teorema: O emparelhamento produzido pelo algoritmo da aceitação postergada é estável.

Prova feita na aula.

No exemplo, aplicando a versão da proposta feminina, obtemos o emparelhamento $\{(h_1, m_1), (h_2, m_3), (h_3, m_2)\}$.

Diferente do obtido pela proposta masculina!

Há alguma diferença significativa entre estes emparelhamentos?

Emparelhamento ν é ótimo-masculino se não há emparelhamento estável μ tq $\mu(h) \succ_h \nu(h)$ ou $\mu(h) = \nu(h)$ para todo h em H e $\mu(j) \succ_j \nu(j)$ para algum j em H.

Emparelhamentos ótimos

Teorema: O emparelhamento produzido pelo algoritmo da aceitação postergada é estável.

Emparelhamento ν é ótimo-masculino se não há emparelhamento estável μ tq $\mu(h) \succ_h \nu(h)$ ou $\mu(h) = \nu(h)$ para todo h em H e $\mu(j) \succ_j \nu(j)$ para algum j em H.

Emparelhamentos ótimos

Teorema: O emparelhamento produzido pelo algoritmo da aceitação postergada é estável.

Emparelhamento ν é ótimo-masculino se não há emparelhamento estável μ tq $\mu(h) \succ_h \nu(h)$ ou $\mu(h) = \nu(h)$ para todo h em H e $\mu(j) \succ_j \nu(j)$ para algum j em H.

Definição de emparelhamento ótimo-feminino é análoga.

Emparelhamentos ótimos

Teorema: O emparelhamento produzido pelo algoritmo da aceitação postergada é estável.

Emparelhamento ν é ótimo-masculino se não há emparelhamento estável μ tq $\mu(h) \succ_h \nu(h)$ ou $\mu(h) = \nu(h)$ para todo h em H e $\mu(j) \succ_j \nu(j)$ para algum j em H.

Definição de emparelhamento ótimo-feminino é análoga.

Teorema: O algoritmo da aceitação postergada com proposta masculina (feminina) produz um escalonamento ótimo-masculino (ótimo-feminino).

Emparelhamentos ótimos

Teorema: O emparelhamento produzido pelo algoritmo da aceitação postergada é estável.

Emparelhamento ν é ótimo-masculino se não há emparelhamento estável μ tq $\mu(h) \succ_h \nu(h)$ ou $\mu(h) = \nu(h)$ para todo h em H e $\mu(j) \succ_j \nu(j)$ para algum j em H.

Definição de emparelhamento ótimo-feminino é análoga.

Teorema: O algoritmo da aceitação postergada com proposta masculina (feminina) produz um escalonamento ótimo-masculino (ótimo-feminino).

Prova feita na aula.

Prova de estratégia

Emparelhamento ν é ótimo-masculino se não há emparelhamento estável μ tq $\mu(h) \succ_h \nu(h)$ ou $\mu(h) = \nu(h)$ para todo h em H e $\mu(j) \succ_j \nu(j)$ para algum j em H.

Definição de emparelhamento ótimo-feminino é análoga.

Teorema: O algoritmo da aceitação postergada com proposta masculina (feminina) produz um escalonamento ótimo-masculino (ótimo-feminino).

Prova de estratégia

Emparelhamento ν é ótimo-masculino se não há emparelhamento estável μ tq $\mu(h) \succ_h \nu(h)$ ou $\mu(h) = \nu(h)$ para todo h em H e $\mu(j) \succ_j \nu(j)$ para algum j em H.

Definição de emparelhamento ótimo-feminino é análoga.

Teorema: O algoritmo da aceitação postergada com proposta masculina (feminina) produz um escalonamento ótimo-masculino (ótimo-feminino).

Teorema: O algoritmo da aceitação postergada com proposta masculina (feminina) é um mecanismo à prova de estratégia para os homens (mulheres) .

Prova de estratégia

Emparelhamento ν é ótimo-masculino se não há emparelhamento estável μ tq $\mu(h) \succ_h \nu(h)$ ou $\mu(h) = \nu(h)$ para todo h em H e $\mu(j) \succ_j \nu(j)$ para algum j em H.

Definição de emparelhamento ótimo-feminino é análoga.

Teorema: O algoritmo da aceitação postergada com proposta masculina (feminina) produz um escalonamento ótimo-masculino (ótimo-feminino).

Teorema: O algoritmo da aceitação postergada com proposta masculina (feminina) é um mecanismo à prova de estratégia para os homens (mulheres) .

Prova feita na aula.