Jogo de conexão global

Seja $G = (V, E)$ um digrafo com custo c_e para cada e em E.

Jogo de conexão global

Seja $G = (V, E)$ um digrafo com custo c_e para cada e em E.

k jogadores, cada um com um par de vértices (s_i, t_i).

Jogo de conexão global

Seja $G = (V, E)$ um digrafo com custo c_e para cada e em E.

k jogadores, cada um com um par de vértices (s_i, t_i).

Estratégia do jogador i: caminho de s_i a t_i em G.

Jogo de conexão global

Seja $G = (V, E)$ um digrafo com custo c_e para cada e em E.

k jogadores, cada um com um par de vértices (s_i, t_i).

Estratégia do jogador i: caminho de s_i a t_i em G.

custo para i do vetor de estratégias $S = (P_1, \ldots, P_k)$:

$$\mathrm{custo}_i(S) \;=\; \sum_{e \in P_i} \frac{c_e}{k_e},$$

onde k_e é o número de caminhos em S que usam e.

Jogo de conexão global

Seja $G = (V, E)$ um digrafo com custo c_e para cada e em E.

k jogadores, cada um com um par de vértices (s_i, t_i).

Estratégia do jogador i: caminho de s_i a t_i em G.

custo para i do vetor de estratégias $S = (P_1, \ldots, P_k)$:

$$\mathrm{custo}_i(S) \;=\; \sum_{e \in P_i} \frac{c_e}{k_e},$$

onde k_e é o número de caminhos em S que usam e.

Custo social $\mathrm{cs}(S)$: soma do custo das arestas em $\cup_i P_i$.

Problema de Steiner Generalizado

Seja $G = (V, E)$ um digrafo com custo c_e para cada e em E, e pares de vértices (s_i, t_i) para $i = 1, \ldots, k$.

Problema de Steiner Generalizado

Seja $G = (V, E)$ um digrafo com custo c_e para cada e em E, e pares de vértices (s_i, t_i) para $i = 1, \ldots, k$.

Problema: Encontrar um subgrafo de G de custo mínimo que contenha um caminho de s_i a t_i para $i = 1, \ldots, k$.

Problema de Steiner Generalizado

Seja $G = (V, E)$ um digrafo com custo c_e para cada e em E, e pares de vértices (s_i, t_i) para $i = 1, \ldots, k$.

Problema: Encontrar um subgrafo de G de custo mínimo que contenha um caminho de s_i a t_i para $i = 1, \ldots, k$.

Esse problema é NP-difícil, mas
é com ele que vamos comparar o custo dos equilíbrios.

Problema de Steiner Generalizado

Seja $G = (V, E)$ um digrafo com custo c_e para cada e em E, e pares de vértices (s_i, t_i) para $i = 1, \ldots, k$.

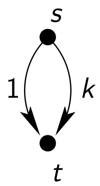
Problema: Encontrar um subgrafo de G de custo mínimo que contenha um caminho de s_i a t_i para $i = 1, \ldots, k$.

Esse problema é NP-difícil, mas
é com ele que vamos comparar o custo dos equilíbrios.

Ou seja, o custo social ótimo do jogo é o custo de uma solução do correspondente problema de Steiner generalizado.

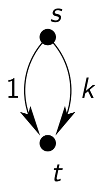
Exemplo 1

k jogadores, todos com o mesmo s e t.



Exemplo 1

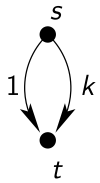
k jogadores, todos com o mesmo s e t.



Ótimo: todos pela aresta de custo 1.

Exemplo 1

k jogadores, todos com o mesmo s e t.

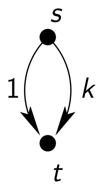


Ótimo: todos pela aresta de custo 1.

Isso é um equilíbrio: $\mathrm{PoE} = 1$

Exemplo 1

k jogadores, todos com o mesmo s e t.



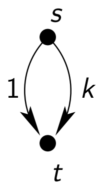
Ótimo: todos pela aresta de custo 1.

Isso é um equilíbrio: $\mathrm{PoE} = 1$

Outro equilíbrio: todos pela aresta de custo k...

Exemplo 1

k jogadores, todos com o mesmo s e t.



Ótimo: todos pela aresta de custo 1.

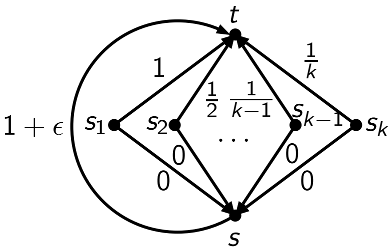
Isso é um equilíbrio: $\mathrm{PoE} = 1$

Outro equilíbrio: todos pela aresta de custo k...

$\mathrm{PoA} = k$

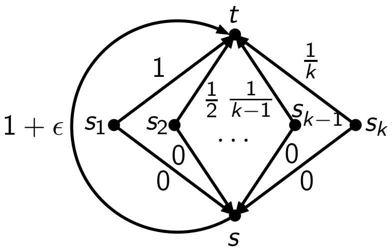
Exemplo 2

k jogadores, todos com o mesmo t.



Exemplo 2

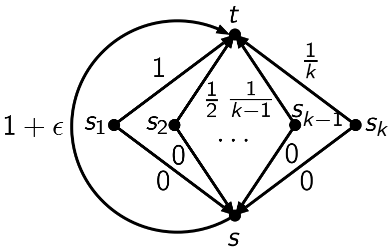
k jogadores, todos com o mesmo t.



Ótimo: todos por s, a um custo total de $1 + \epsilon$.

Exemplo 2

k jogadores, todos com o mesmo t.

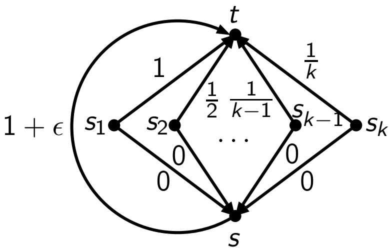


Ótimo: todos por s, a um custo total de $1 + \epsilon$.

Mas isso não é um equilíbrio: k está insatisfeito.

Exemplo 2

k jogadores, todos com o mesmo t.



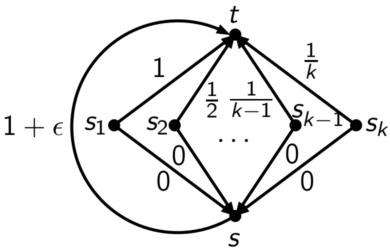
Ótimo: todos por s, a um custo total de $1+\epsilon$.

Mas isso não é um equilíbrio: k está insatisfeito.

Único equilíbrio: todos diretamente para t...

Exemplo 2

k jogadores, todos com o mesmo t.



Ótimo: todos por s, a um custo total de $1+\epsilon$.

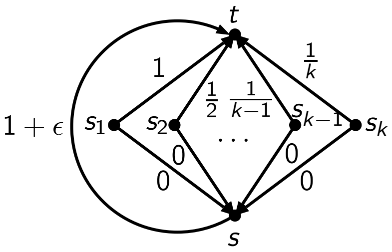
Mas isso não é um equilíbrio: k está insatisfeito.

Único equilíbrio: todos diretamente para t...

$\mathrm{cs} = \sum_{i=1}^{k} \frac{1}{i} = H_k$ e portanto $\mathrm{PoE} = \mathrm{PoA} = H_k$.

Exemplo 2

k jogadores, todos com o mesmo t.



Ótimo: todos por s, a um custo total de $1+\epsilon$.

Mas isso não é um equilíbrio: k está insatisfeito.

Único equilíbrio: todos diretamente para t...

$\text{cs} = \sum_{i=1}^{k} \frac{1}{i} = H_k$ e portanto $\text{PoE} = \text{PoA} = H_k$.

Será que pode ser pior que isso?

Nesta aula

Teorema: Qualquer instância do jogo de conexão global tem um equilíbrio puro.

Nesta aula

Teorema: Qualquer instância do jogo de conexão global tem um equilíbrio puro.

Teorema: O preço da estabilidade do jogo de conexão global com k jogadores é no máximo H_k.

Nesta aula

Teorema: Qualquer instância do jogo de conexão global tem um equilíbrio puro.

Teorema: O preço da estabilidade do jogo de conexão global com k jogadores é no máximo H_k.

A prova destes resultados usa função potencial.

Nesta aula

Teorema: Qualquer instância do jogo de conexão global tem um equilíbrio puro.

Teorema: O preço da estabilidade do jogo de conexão global com k jogadores é no máximo H_k.

A prova destes resultados usa função potencial.

A prova é mais geral, para jogos de potencial.

Função Potencial

Seja $G = (V, E)$ um digrafo com custo c_e para cada e em E, e os pares de vértices (s_i, t_i) dos jogadores $i = 1, \ldots, k$.

Função Potencial

Seja $G = (V, E)$ um digrafo com custo c_e para cada e em E, e os pares de vértices (s_i, t_i) dos jogadores $i = 1, \ldots, k$.

Seja $S = (P_1, \ldots, P_k)$ um vetor de estratégias.

Seja k_e o número de caminhos em S que usam e.

Função Potencial

Seja $G = (V, E)$ um digrafo com custo c_e para cada e em E, e os pares de vértices (s_i, t_i) dos jogadores $i = 1, \ldots, k$.

Seja $S = (P_1, \ldots, P_k)$ um vetor de estratégias.

Seja k_e o número de caminhos em S que usam e.

Para cada aresta e, defina $\psi_e(S) \;=\; c_e\, H_{k_e}$.

Função Potencial

Seja $G = (V, E)$ um digrafo com custo c_e para cada e em E, e os pares de vértices (s_i, t_i) dos jogadores $i = 1, \ldots, k$.

Seja $S = (P_1, \ldots, P_k)$ um vetor de estratégias.

Seja k_e o número de caminhos em S que usam e.

Para cada aresta e, defina $\psi_e(S) \; = \; c_e \, H_{k_e}$.

Seja $\psi(S) \; = \; \sum_e \psi_e(S)$.

Função Potencial

Seja $G = (V, E)$ um digrafo com custo c_e para cada e em E, e os pares de vértices (s_i, t_i) dos jogadores $i = 1, \ldots, k$.

Seja $S = (P_1, \ldots, P_k)$ um vetor de estratégias.

Seja k_e o número de caminhos em S que usam e.

Para cada aresta e, defina $\psi_e(S) \; = \; c_e \, H_{k_e}$.

Seja $\psi(S) \; = \; \sum_e \psi_e(S)$.

ψ é chamada de função potencial, e será usada para provar uma delimitação superior no PoE do jogo.

Função Potencial

Seja $G = (V, E)$ um digrafo com custo c_e para cada e em E, e os pares de vértices (s_i, t_i) dos jogadores $i = 1, \ldots, k$.

Seja $S = (P_1, \ldots, P_k)$ um vetor de estratégias.

Seja k_e o número de caminhos em S que usam e.

Para cada aresta e, defina $\psi_e(S) \ = \ c_e \, H_{k_e}$.

Seja $\psi(S) \ = \ \sum_e \psi_e(S)$.

Lema: $\mathrm{custo}(S) \ \leq \ \psi(S) \ \leq \ H_k \, \mathrm{custo}(S)$.

Função Potencial

Seja $G = (V, E)$ um digrafo com custo c_e para cada e em E, e os pares de vértices (s_i, t_i) dos jogadores $i = 1, \ldots, k$.

Seja $S = (P_1, \ldots, P_k)$ um vetor de estratégias.

Seja k_e o número de caminhos em S que usam e.

Para cada aresta e, defina $\psi_e(S) \; = \; c_e \, H_{k_e}$.

Seja $\psi(S) \; = \; \sum_e \psi_e(S)$.

Lema: $\mathrm{custo}(S) \; \leq \; \psi(S) \; \leq \; H_k \, \mathrm{custo}(S)$.

Prova: Para cada e em S, $\psi_e(S) \geq c_e$ e $\psi_e(S) \leq c_e \, H_k$. $\square$

Função Potencial Exata

Para cada aresta e, defina $\psi_e(S) \;=\; c_e\, H_{k_e}$.

Seja $\psi(S) \;=\; \sum_e \psi_e(S)$.

Lema: $\mathrm{custo}(S) \;\leq\; \psi(S) \;\leq\; H_k\, \mathrm{custo}(S)$.

Função Potencial Exata

Para cada aresta e, defina $\psi_e(S) \; = \; c_e \, H_{k_e}$.

Seja $\psi(S) \; = \; \sum_e \psi_e(S)$.

Lema: $\text{custo}(S) \; \leq \; \psi(S) \; \leq \; H_k \, \text{custo}(S)$.

Lema: Seja $P'_i \neq P_i$ um caminho alternativo para i e seja $S' = (S_{-i}, P'_i)$. Então $\psi(S) - \psi(S') \; = \; \text{custo}_i(S) - \text{custo}_i(S')$.

Função Potencial Exata

Para cada aresta e, defina $\psi_e(S) \; = \; c_e \, H_{k_e}$.

Seja $\psi(S) \; = \; \sum_e \psi_e(S)$.

Lema: $\text{custo}(S) \; \leq \; \psi(S) \; \leq \; H_k \, \text{custo}(S)$.

Lema: Seja $P'_i \neq P_i$ um caminho alternativo para i e seja $S' = (S_{-i}, P'_i)$. Então $\psi(S) - \psi(S') \; = \; \text{custo}_i(S) - \text{custo}_i(S')$.

Isso equivale a $\psi(S) - \psi(S') \; = \; u_i(S') - u_i(S)$.

Função Potencial Exata

Para cada aresta e, defina $\psi_e(S) \; = \; c_e \, H_{k_e}$.

Seja $\psi(S) \; = \; \sum_e \psi_e(S)$.

Lema: $\text{custo}(S) \; \leq \; \psi(S) \; \leq \; H_k \, \text{custo}(S)$.

Lema: Seja $P'_i \neq P_i$ um caminho alternativo para i e seja $S' = (S_{-i}, P'_i)$. Então $\psi(S) - \psi(S') \; = \; \text{custo}_i(S) - \text{custo}_i(S')$.

Isso equivale a $\psi(S) - \psi(S') \; = \; u_i(S') - u_i(S)$.

Uma função potencial com tal propriedade é dita exata.

Função Potencial Exata

$$\psi_e(S) \;=\; c_e\, H_{k_e} \quad \text{e} \quad \psi(S) \;=\; \textstyle\sum_e \psi_e(S).$$

Lema: Seja $P'_i \neq P_i$ um caminho alternativo para i e seja $S' = (S_{-i}, P'_i)$. Então $\psi(S) - \psi(S') \;=\; \mathrm{custo}_i(S) - \mathrm{custo}_i(S')$.

Função Potencial Exata

$$\psi_e(S) \; = \; c_e \, H_{k_e} \quad \text{e} \quad \psi(S) \; = \; \textstyle\sum_e \psi_e(S).$$

Lema: Seja $P'_i \neq P_i$ um caminho alternativo para i e seja $S' = (S_{-i}, P'_i)$. Então $\psi(S) - \psi(S') \; = \; \mathrm{custo}_i(S) - \mathrm{custo}_i(S')$.

Prova: Se e está em ambos P_i e P'_i ou em nenhum, então i paga o mesmo por e em S e em S', e $\psi_e(S) = \psi_e(S')$.

Função Potencial Exata

$$\psi_e(S) \;=\; c_e\, H_{k_e} \quad \text{e} \quad \psi(S) \;=\; \textstyle\sum_e \psi_e(S).$$

Lema: Seja $P'_i \neq P_i$ um caminho alternativo para i e seja $S' = (S_{-i}, P'_i)$. Então $\psi(S) - \psi(S') \;=\; \mathrm{custo}_i(S) - \mathrm{custo}_i(S')$.

Prova: Se e está em ambos P_i e P'_i ou em nenhum, então i paga o mesmo por e em S e em S', e $\psi_e(S) = \psi_e(S')$.

Se $e \in P_i \setminus P'_i$, então i deixa de pagar $\frac{c_e}{k_e}$ de S para S'.

Função Potencial Exata

$$\psi_e(S) \ = \ c_e\, H_{k_e} \quad \text{e} \quad \psi(S) \ = \ \textstyle\sum_e \psi_e(S).$$

Lema: Seja $P'_i \neq P_i$ um caminho alternativo para i e seja $S' = (S_{-i}, P'_i)$. Então $\psi(S) - \psi(S') \ = \ \mathrm{custo}_i(S) - \mathrm{custo}_i(S')$.

Prova: Se e está em ambos P_i e P'_i ou em nenhum, então i paga o mesmo por e em S e em S', e $\psi_e(S) = \psi_e(S')$.

Se $e \in P_i \setminus P'_i$, então i deixa de pagar $\frac{c_e}{k_e}$ de S para S'.
Por outro lado, $k'_e = k_e - 1$ e $\psi_e(S') = \psi_e(S) - \dfrac{c_e}{k_e}$.

Função Potencial Exata

$$\psi_e(S) \;=\; c_e\, H_{k_e} \quad \text{e} \quad \psi(S) \;=\; \textstyle\sum_e \psi_e(S).$$

Lema: Seja $P'_i \neq P_i$ um caminho alternativo para i e seja $S' = (S_{-i}, P'_i)$. Então $\psi(S) - \psi(S') \;=\; \mathrm{custo}_i(S) - \mathrm{custo}_i(S')$.

Prova: Se e está em ambos P_i e P'_i ou em nenhum, então i paga o mesmo por e em S e em S', e $\psi_e(S) = \psi_e(S')$.

Se $e \in P_i \setminus P'_i$, então i deixa de pagar $\frac{c_e}{k_e}$ de S para S'.
Por outro lado, $k'_e = k_e - 1$ e $\psi_e(S') = \psi_e(S) - \dfrac{c_e}{k_e}$.

Se $e \in P'_i \setminus P_i$, então i paga $\frac{c_e}{k_e+1}$ a mais de S para S'.

Função Potencial Exata

$$\psi_e(S) \ = \ c_e\, H_{k_e} \quad \text{e} \quad \psi(S) \ = \ \textstyle\sum_e \psi_e(S).$$

Lema: Seja $P'_i \neq P_i$ um caminho alternativo para i e seja $S' = (S_{-i}, P'_i)$. Então $\psi(S) - \psi(S') \ = \ \text{custo}_i(S) - \text{custo}_i(S')$.

Prova: Se e está em ambos P_i e P'_i ou em nenhum, então i paga o mesmo por e em S e em S', e $\psi_e(S) = \psi_e(S')$.

Se $e \in P_i \setminus P'_i$, então i deixa de pagar $\frac{c_e}{k_e}$ de S para S'.
Por outro lado, $k'_e = k_e - 1$ e $\psi_e(S') = \psi_e(S) - \dfrac{c_e}{k_e}$.

Se $e \in P'_i \setminus P_i$, então i paga $\frac{c_e}{k_e+1}$ a mais de S para S'.
Por outro lado, $k'_e = k_e + 1$ e $\psi_e(S') = \psi_e(S) + \dfrac{c_e}{k_e+1}$.

Função Potencial Exata

$$\psi_e(S) \; = \; c_e \, H_{k_e} \quad \text{e} \quad \psi(S) \; = \; \textstyle\sum_e \psi_e(S).$$

Lema: Seja $P'_i \neq P_i$ um caminho alternativo para i e seja $S' = (S_{-i}, P'_i)$. Então $\psi(S) - \psi(S') \; = \; \mathrm{custo}_i(S) - \mathrm{custo}_i(S')$.

Prova: Se e está em ambos P_i e P'_i ou em nenhum, então i paga o mesmo por e em S e em S', e $\psi_e(S) = \psi_e(S')$.

Se $e \in P_i \setminus P'_i$, então i deixa de pagar $\frac{c_e}{k_e}$ de S para S'.
Por outro lado, $k'_e = k_e - 1$ e $\psi_e(S') = \psi_e(S) - \dfrac{c_e}{k_e}$.

Se $e \in P'_i \setminus P_i$, então i paga $\frac{c_e}{k_e+1}$ a mais de S para S'.
Por outro lado, $k'_e = k_e + 1$ e $\psi_e(S') = \psi_e(S) + \dfrac{c_e}{k_e+1}$.

Assim sendo $\psi(S) - \psi(S') \; = \; \mathrm{custo}_i(S) - \mathrm{custo}_i(S')$. $\square$

Jogos de Potencial

Jogo de potencial: jogo que admite uma função potencial (exata).

Jogos de Potencial

Jogo de potencial: jogo que admite uma função potencial (exata).

Lema anterior: jogos de conexão global são de potencial.

Jogos de Potencial

Jogo de potencial: jogo que admite uma função potencial (exata).

Lema anterior: jogos de conexão global são de potencial.

Jogos de congestionamento:

Dados n jogadores, um conjunto E de recursos,
uma função custo $c_e : \{1, \ldots, n\} \to \mathbb{Z}^+$ para cada $e \in E$,
cada um dos n jogadores tem como estratégias subconjuntos de E.

Jogos de Potencial

Jogo de potencial: jogo que admite uma função potencial (exata).

Lema anterior: jogos de conexão global são de potencial.

Jogos de congestionamento:

Dados n jogadores, um conjunto E de recursos,
uma função custo $c_e : \{1, \ldots, n\} \to \mathbb{Z}^+$ para cada $e \in E$,
cada um dos n jogadores tem como estratégias subconjuntos de E.

O custo de i para o vetor de estratégias $S = (S_1, \ldots, S_n)$ é

$$\sum_{e \in S_i} c_e(\ell_e(S)),$$

onde $\ell_e(S)$ é o número de conjuntos de S contendo e.

Jogos de Potencial

Jogo de potencial: jogo que admite uma função potencial (exata).

Jogos de Potencial

Jogo de potencial: jogo que admite uma função potencial (exata).

Todo jogo de potencial finito tem um equilíbrio puro:

Jogos de Potencial

Jogo de potencial: jogo que admite uma função potencial (exata).

Todo jogo de potencial finito tem um equilíbrio puro:

Seja S tal que $\psi(S)$ é mínimo.

Jogos de Potencial

Jogo de potencial: jogo que admite uma função potencial (exata).

Todo jogo de potencial finito tem um equilíbrio puro:

Seja S tal que $\psi(S)$ é mínimo.

S é um equilíbrio pois,

$$\psi(S) \leq \psi(S') \text{ para todo } S' = (S_{-i}, S'_i).$$

Jogos de Potencial

Jogo de potencial: jogo que admite uma função potencial (exata).

Todo jogo de potencial finito tem um equilíbrio puro:

Seja S tal que $\psi(S)$ é mínimo.

S é um equilíbrio pois,

$$\psi(S) \leq \psi(S') \text{ para todo } S' = (S_{-i}, S'_i).$$

Logo, como $\psi(S) - \psi(S') = u_i(S') - u_i(S)$,
$u_i(S') \leq u_i(S)$ para todo $S' = (S_{-i}, S'_i)$.

Jogos de Potencial

Jogo de potencial: jogo que admite uma função potencial (exata).

Jogos de congestionamento:

Dados n jogadores, um conjunto E de recursos,
funções c_e para cada $e \in E$, vetor de estratégias $S = (S_1, \ldots, S_n)$,
com cada $S_i \subseteq E$, o custo de i para S é $\sum_{e \in S_i} c_e(\ell_e(S))$,
onde $\ell_e(S)$ é o número de conjuntos de S contendo e.

Jogos de Potencial

Jogo de potencial: jogo que admite uma função potencial (exata).

Jogos de congestionamento:

Dados n jogadores, um conjunto E de recursos,
funções c_e para cada $e \in E$, vetor de estratégias $S = (S_1, \ldots, S_n)$,
com cada $S_i \subseteq E$, o custo de i para S é $\sum_{e \in S_i} c_e(\ell_e(S))$,
onde $\ell_e(S)$ é o número de conjuntos de S contendo e.

Rosenthal (1973) provou que
todo jogo de congestionamento é um jogo de potencial.

Jogos de Potencial

Jogo de potencial: jogo que admite uma função potencial (exata).

Jogos de congestionamento:

Dados n jogadores, um conjunto E de recursos,
funções c_e para cada $e \in E$, vetor de estratégias $S = (S_1, \ldots, S_n)$,
com cada $S_i \subseteq E$, o custo de i para S é $\sum_{e \in S_i} c_e(\ell_e(S))$,
onde $\ell_e(S)$ é o número de conjuntos de S contendo e.

Rosenthal (1973) provou que
todo jogo de congestionamento é um jogo de potencial.

Monderer e Shapley (1996) provaram o inverso:

para todo jogo de potencial,
há um jogo de congestionamento com a mesma função potencial.

Preço da Estabilidade

Teorema: Se, para um jogo de potencial com potencial ψ, existem números A e B tais que

$$\frac{\mathrm{custo}(S)}{A} \leq \psi(S) \leq B\,\mathrm{custo}(S).$$

para todo vetor de estratégias S, então $\mathrm{PoE} \leq AB$.

Preço da Estabilidade

Teorema: Se, para um jogo de potencial com potencial ψ, existem números A e B tais que

$$\frac{\text{custo}(S)}{A} \leq \psi(S) \leq B\,\text{custo}(S).$$

para todo vetor de estratégias S, então $\text{PoE} \leq AB$.

Prova: Seja S tq $\psi(S)$ é mínimo e S^* com custo mínimo.

Preço da Estabilidade

Teorema: Se, para um jogo de potencial com potencial ψ, existem números A e B tais que

$$\frac{\text{custo}(S)}{A} \leq \psi(S) \leq B\,\text{custo}(S).$$

para todo vetor de estratégias S, então $\text{PoE} \leq AB$.

Prova: Seja S tq $\psi(S)$ é mínimo e S^* com custo mínimo.

Por hipótese,

$$\frac{\text{custo}(S)}{A} \;\leq\; \psi(S) \;\leq\; \psi(S^*) \;\leq\; B\,\text{custo}(S^*).$$

Preço da Estabilidade

Teorema: Se, para um jogo de potencial com potencial ψ, existem números A e B tais que

$$\frac{\text{custo}(S)}{A} \leq \psi(S) \leq B\,\text{custo}(S).$$

para todo vetor de estratégias S, então $\text{PoE} \leq AB$.

Prova: Seja S tq $\psi(S)$ é mínimo e S^* com custo mínimo.

Por hipótese,

$$\frac{\text{custo}(S)}{A} \;\leq\; \psi(S) \;\leq\; \psi(S^*) \;\leq\; B\,\text{custo}(S^*).$$

Logo $\text{custo}(S) \leq AB\,\text{custo}(S^*)$. $\square$

Preço da Estabilidade

Teorema: Se, para um jogo de potencial com potencial ψ, existem números A e B tais que

$$\frac{\mathrm{custo}(S)}{A} \leq \psi(S) \leq B \, \mathrm{custo}(S).$$

para todo vetor de estratégias S, então $\mathrm{PoE} \leq AB$.

Preço da Estabilidade

Teorema: Se, para um jogo de potencial com potencial ψ, existem números A e B tais que

$$\frac{\mathrm{custo}(S)}{A} \leq \psi(S) \leq B\,\mathrm{custo}(S).$$

para todo vetor de estratégias S, então $\mathrm{PoE} \leq AB$.

Para jogos de conexão global com k jogadores, vale

Lema: $\mathrm{custo}(S) \ \leq \ \psi(S) \ \leq \ H_k\,\mathrm{custo}(S)$.

Preço da Estabilidade

Teorema: Se, para um jogo de potencial com potencial ψ, existem números A e B tais que

$$\frac{\mathrm{custo}(S)}{A} \leq \psi(S) \leq B\,\mathrm{custo}(S).$$

para todo vetor de estratégias S, então $\mathrm{PoE} \leq AB$.

Para jogos de conexão global com k jogadores, vale

Lema: $\mathrm{custo}(S) \;\leq\; \psi(S) \;\leq\; H_k\,\mathrm{custo}(S)$.

Portanto, $\mathrm{PoE} \leq H_k$.

Busca por um equilíbrio

Política de melhor resposta converge para um equilíbrio.

Busca por um equilíbrio

Política de melhor resposta converge para um equilíbrio.

Mas quanto tempo leva?

Busca por um equilíbrio

Política de melhor resposta converge para um equilíbrio.

Mas quanto tempo leva?

É possível calcular eficientemente S tq $\psi(S)$ é mínimo local?

Busca por um equilíbrio

Política de melhor resposta converge para um equilíbrio.

Mas quanto tempo leva?

É possível calcular eficientemente S tq $\psi(S)$ é mínimo local?

Problemas de otimização

Busca por um equilíbrio

Política de melhor resposta converge para um equilíbrio.

Mas quanto tempo leva?

É possível calcular eficientemente S tq $\psi(S)$ é mínimo local?

Problemas de otimização

Para uma instância I, cada solução viável S tem um custo $c_I(S)$.

Busca por um equilíbrio

Política de melhor resposta converge para um equilíbrio.

Mas quanto tempo leva?

É possível calcular eficientemente S tq $\psi(S)$ é mínimo local?

Problemas de otimização

Para uma instância I, cada solução viável S tem um custo $c_I(S)$.

Um oráculo decide se
um objeto é uma solução viável e qual o seu custo.

Busca por um equilíbrio

Política de melhor resposta converge para um equilíbrio.

Mas quanto tempo leva?

É possível calcular eficientemente S tq $\psi(S)$ é mínimo local?

Problemas de otimização

Para uma instância I, cada solução viável S tem um custo $c_I(S)$.

Um oráculo decide se
um objeto é uma solução viável e qual o seu custo.

O objetivo é, dada uma instância,
encontrar uma solução viável de custo mínimo.

Problemas de otimização local

Para cada solução viável S de uma instância I, temos
uma vizinhança $N_I(S)$ que é um conjunto de soluções viáveis de I.

Problemas de otimização local

Para cada solução viável S de uma instância I, temos
uma vizinhança $N_I(S)$ que é um conjunto de soluções viáveis de I.

Solução localmente ótima para I:
solução viável S tq $c_I(S) \leq c_I(S')$ para toda $S' \in N_I(S)$.

Problemas de otimização local

Para cada solução viável S de uma instância I, temos uma vizinhança $N_I(S)$ que é um conjunto de soluções viáveis de I.

Solução localmente ótima para I:
solução viável S tq $c_I(S) \leq c_I(S')$ para toda $S' \in N_I(S)$.

Problemas de otimização local: Dada instância I, encontrar uma solução localmente ótima para I.

Problemas de otimização local

Para cada solução viável S de uma instância I, temos
uma vizinhança $N_I(S)$ que é um conjunto de soluções viáveis de I.

Solução localmente ótima para I:
solução viável S tq $c_I(S) \leq c_I(S')$ para toda $S' \in N_I(S)$.

Problemas de otimização local: Dada instância I,
encontrar uma solução localmente ótima para I.

Problema de busca local polinomial (PLS):
Problema de otimização local para o qual temos um oráculo que,
para toda instância I e solução viável S, decide se S
é localmente ótima e, se não for, devolve S' com $c_I(S') < c_I(S)$.

Problemas de otimização local

Para cada solução viável S de uma instância I, temos
uma vizinhança $N_I(S)$ que é um conjunto de soluções viáveis de I.

Solução localmente ótima para I:
solução viável S tq $c_I(S) \leq c_I(S')$ para toda $S' \in N_I(S)$.

Problemas de otimização local: Dada instância I,
encontrar uma solução localmente ótima para I.

Problema de busca local polinomial (PLS):
Problema de otimização local para o qual temos um oráculo que,
para toda instância I e solução viável S, decide se S
é localmente ótima e, se não for, devolve S' com $c_I(S') < c_I(S)$.

(oráculo = algoritmo polinomial)

Exemplo de problema em PLS

Dada uma fórmula booleana SAT com custo nas cláusulas, encontrar uma atribuição local ótima.

Exemplo de problema em PLS

Dada uma fórmula booleana SAT com custo nas cláusulas, encontrar uma atribuição local ótima.

Custo de uma atribuição:
soma dos custos das cláusulas insatisfeitas.

Exemplo de problema em PLS

Dada uma fórmula booleana SAT com custo nas cláusulas, encontrar uma atribuição local ótima.

Custo de uma atribuição:
soma dos custos das cláusulas insatisfeitas.

Vizinhança de uma atribuição:
atribuições que diferem apenas no valor de uma variável.

Exemplo de problema em PLS

Dada uma fórmula booleana SAT com custo nas cláusulas, encontrar uma atribuição local ótima.

Custo de uma atribuição:
soma dos custos das cláusulas insatisfeitas.

Vizinhança de uma atribuição:
atribuições que diferem apenas no valor de uma variável.

Atribuição ótima local:
Mudar o valor de uma única variável
não aumenta o custo das cláusulas satisfeitas.

Exemplo de problema em PLS

Dada uma fórmula booleana SAT com custo nas cláusulas, encontrar uma atribuição local ótima.

Custo de uma atribuição:
soma dos custos das cláusulas insatisfeitas.

Vizinhança de uma atribuição:
atribuições que diferem apenas no valor de uma variável.

Atribuição ótima local:
Mudar o valor de uma única variável
não aumenta o custo das cláusulas satisfeitas.

Esse problema é PLS-completo.

Complexidade computacional

Jogos de congestionamento estão em PLS.

Complexidade computacional

Jogos de congestionamento estão em PLS.

Solução viável e custo: vetor de estratégia com seu custo.

Complexidade computacional

Jogos de congestionamento estão em PLS.

Solução viável e custo: vetor de estratégia com seu custo.

Vizinhança do vetor S de estratégias:
vetores S' obtidos mudando a estratégia de um jogador.

Complexidade computacional

Jogos de congestionamento estão em PLS.

Solução viável e custo: vetor de estratégia com seu custo.

Vizinhança do vetor S de estratégias:
vetores S' obtidos mudando a estratégia de um jogador.

Redução do SAT:

Instância: k variáveis $x_1, \ldots, x_k$
n cláusulas $C_1, \ldots, C_n$ com custos $w_1, \ldots, w_n$.

Jogo de congestionamento: k jogadores
Recursos: cláusulas

Estratégias para i:
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ verdade ou
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ falso.

Redução do SAT

Dada uma fórmula booleana SAT, com peso nas cláusulas, encontrar uma atribuição local ótima.

Instância: k variáveis $x_1, \ldots, x_k$
n cláusulas $C_1, \ldots, C_n$ com pesos $w_1, \ldots, w_n$.

Jogo de congestionamento: k jogadores
Recursos: cláusulas

Estratégias para i:
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ verdade ou
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ falso.

Seja A um vetor de estratégias.

Redução do SAT

Dada uma fórmula booleana SAT, com peso nas cláusulas, encontrar uma atribuição local ótima.

Instância: k variáveis $x_1, \ldots, x_k$
n cláusulas $C_1, \ldots, C_n$ com pesos $w_1, \ldots, w_n$.

Jogo de congestionamento: k jogadores
Recursos: cláusulas

Estratégias para i:
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ verdade ou
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ falso.

Seja A um vetor de estratégias.

$$c_j(A) = \begin{cases} 0 & \text{se } C_j \in A \\ w_j & \text{se } C_j \notin A. \end{cases}$$

Jogo de congestionamento

Um jogador para cada variável.
Cada cláusula é um recurso.

Estratégias para i:
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ verdade ou
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ falso.

Seja A um vetor de estratégias.

$$c_j(A) \; = \; \begin{cases} 0 & \text{se } C_j \in A \\ w_j & \text{se } C_j \notin A. \end{cases}$$

Jogo de congestionamento

Um jogador para cada variável.
Cada cláusula é um recurso.

Estratégias para i:
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ verdade ou
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ falso.

Seja A um vetor de estratégias.

$$c_j(A) \;=\; \begin{cases} 0 & \text{se } C_j \in A \\ w_j & \text{se } C_i \notin A. \end{cases}$$

Custo social de A: $\mathrm{cs}(A) = \sum_j c_j(A)$.

Jogo de congestionamento

Um jogador para cada variável.
Cada cláusula é um recurso.

Estratégias para i:
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ verdade ou
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ falso.

Seja A um vetor de estratégias.

$$c_j(A) = \begin{cases} 0 & \text{se } C_j \in A \\ w_j & \text{se } C_i \notin A. \end{cases}$$

Custo social de A: $\mathrm{cs}(A) = \sum_j c_j(A)$.

$\mathrm{custo}_i(A) = \sum_{j : x_i \in L(C_j)} c_j(A)$.

Jogo de congestionamento

Um jogador para cada variável.
Cada cláusula é um recurso.

Estratégias para i:
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ verdade ou
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ falso.

Seja A um vetor de estratégias.

$$c_j(A) \;=\; \begin{cases} 0 & \text{se } C_j \in A \\ w_j & \text{se } C_i \notin A. \end{cases}$$

Custo social de A: $\mathrm{cs}(A) = \sum_j c_j(A)$.

$\mathrm{custo}_i(A) = \sum_{j:x_i \in L(C_j)} c_j(A)$.

A': vetor de estratégia quando i troca sua estratégia em A.

$\mathrm{custo}_i(A') - \mathrm{custo}_i(A)$:
peso das cláusulas perdidas pela troca $-$ peso das cláusulas ganhas.

Jogo de congestionamento

Um jogador para cada variável.
Cada cláusula é um recurso.

Estratégias para i:
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ verdade ou
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ falso.

Seja A um vetor de estratégias.
Seja $c_j(A) = 0$ se $C_j \in A$ e $c_j(A) = w_j$ caso contrário.
Custo social de A: $\mathrm{cs}(A) = \sum_j c_j(A)$.
Então $\mathrm{custo}_i(A) = \sum_{j: x_i \in L(C_j)} c_j(A)$.

Jogo de congestionamento

Um jogador para cada variável.
Cada cláusula é um recurso.

Estratégias para i:
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ verdade ou
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ falso.

Seja A um vetor de estratégias.
Seja $c_j(A) = 0$ se $C_j \in A$ e $c_j(A) = w_j$ caso contrário.
Custo social de A: $\mathrm{cs}(A) = \sum_j c_j(A)$.
Então $\mathrm{custo}_i(A) = \sum_{j: x_i \in L(C_j)} c_j(A)$.

Função potencial $\psi(A)$: peso das cláusulas fora de A.

Jogo de congestionamento

Um jogador para cada variável.
Cada cláusula é um recurso.

Estratégias para i:
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ verdade ou
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ falso.

Seja A um vetor de estratégias.
Seja $c_j(A) = 0$ se $C_j \in A$ e $c_j(A) = w_j$ caso contrário.
Custo social de A: $\mathrm{cs}(A) = \sum_j c_j(A)$.
Então $\mathrm{custo}_i(A) = \sum_{j : x_i \in L(C_j)} c_j(A)$.

Função potencial $\psi(A)$: peso das cláusulas fora de A.

A': vetor de estratégia quando i troca sua estratégia em A.

$\mathrm{custo}_i(A') - \mathrm{custo}_i(A) = \psi(A) - \psi(A')$.

Jogo de congestionamento

Um jogador para cada variável.
Cada cláusula é um recurso.

Estratégias para i:
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ verdade ou
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ falso.

Seja A um vetor de estratégias.
Seja $c_j(A) = 0$ se $C_j \in A$ e $c_j(A) = w_j$ caso contrário.
Custo social de A: $\mathrm{cs}(A) = \sum_j c_j(A)$.
Então $\mathrm{custo}_i(A) = \sum_{j:x_i \in L(C_j)} c_j(A)$.

Função potencial $\psi(A)$: peso das cláusulas fora de A.

A': vetor de estratégia quando i troca sua estratégia em A.

$\mathrm{custo}_i(A') - \mathrm{custo}_i(A) = \psi(A) - \psi(A')$.

Equilíbrio: mínimo local de ψ.

Redução do SAT

Dada uma fórmula booleana SAT com custos nas cláusulas, encontrar uma atribuição local ótima.

Instância: k variáveis $x_1, \ldots, x_k$
n cláusulas $C_1, \ldots, C_n$ com pesos $w_1, \ldots, w_n$.

Jogo de congestionamento: k jogadores
Recursos: cláusulas

Estratégias para i:
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ verdade ou
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ falso.

Seja A um vetor de estratégias.
Seja $c_j(A) = 0$ se $C_j \in A$ e $c_j(A) = w_j$ caso contrário.
Então $\mathrm{custo}_i(A) = \sum_{j: x_i \in L(C_j)} c_j(A)$.

Redução do SAT

Dada uma fórmula booleana SAT com custos nas cláusulas, encontrar uma atribuição local ótima.

Instância: k variáveis $x_1, \ldots, x_k$
n cláusulas $C_1, \ldots, C_n$ com pesos $w_1, \ldots, w_n$.

Jogo de congestionamento: k jogadores
Recursos: cláusulas

Estratégias para i:
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ verdade ou
conjunto de cláusulas satisfeitas por $x_i \leftarrow$ falso.

Seja A um vetor de estratégias.
Seja $c_j(A) = 0$ se $C_j \in A$ e $c_j(A) = w_j$ caso contrário.
Então $\mathrm{custo}_i(A) = \sum_{j:x_i \in L(C_j)} c_j(A)$.

Equilíbrio sse atribuição local ótima para o SAT. $\square$

Regras de divisões de custo

De volta ao jogo de conexão global:

Regras de divisões de custo

De volta ao jogo de conexão global:

Consideramos a divisão uniforme do custo da aresta.

Regras de divisões de custo

De volta ao jogo de conexão global:

Consideramos a divisão uniforme do custo da aresta.

Existem regras mais razoáveis?

Com PoE melhor?

Para as quais seja fácil determinar equilíbrio?

Regras de divisões de custo

De volta ao jogo de conexão global:

Consideramos a divisão uniforme do custo da aresta.

Existem regras mais razoáveis?

Com PoE melhor?

Para as quais seja fácil determinar equilíbrio?

Projeto de mecanismos

Regras de divisões de custo

Vetor de estratégias $S = (S_1, \ldots, S_n)$

$J_e := \{i : e \in S_i\}$ (jogadores que usam e)

Regras de divisões de custo

Vetor de estratégias $S = (S_1, \ldots, S_n)$

$J_e := \{i : e \in S_i\}$ (jogadores que usam e)

Condições sobre as regras

- justa: $\mathrm{custo}_e(i, J_e) = 0$ se $i \notin J_e$.
- balanceada com orçamento: $\sum_i \mathrm{custo}_e(i, J_e) = c_e$.

Regras de divisões de custo

Vetor de estratégias $S = (S_1, \ldots, S_n)$

$J_e := \{i : e \in S_i\}$ (jogadores que usam e)

Condições sobre as regras

- justa: $\mathrm{custo}_e(i, J_e) = 0$ se $i \notin J_e$.
- balanceada com orçamento: $\sum_i \mathrm{custo}_e(i, J_e) = c_e$.

Esquema de divisão cego (*oblivious*):

$$\mathrm{custo}_e(i, J_e) \text{ depende apenas de } c_e \text{ e } J_e.$$

Regras de divisões de custo

Vetor de estratégias $S = (S_1, \dots, S_n)$

$J_e := \{i : e \in S_i\}$ (jogadores que usam e)

Condições sobre as regras

- justa: $\mathrm{custo}_e(i, J_e) = 0$ se $i \notin J_e$.
- balanceada com orçamento: $\sum_i \mathrm{custo}_e(i, J_e) = c_e$.

Esquema de divisão cego (*oblivious*):

$$\mathrm{custo}_e(i, J_e) \text{ depende apenas de } c_e \text{ e } J_e.$$

Teorema: Existe esquema de divisão não oblívio em que o jogo de conexão global com $t_i = t$ para todo i tem preço da anarquia no máximo 2.