“The curse of simultaneity”, Renato Paes Leme, Vasilis Syrgkanis, and Éva Tardos. [LST12]

Francisco Jhonatas

Universidade Estadual de Campinas

June 30, 2017

Introdução

- Simultaneidade é usualmente difícil ou impossível de se alcançar nas implementações

Introdução

- Simultaneidade é usualmente difícil ou impossível de se alcançar nas implementações
- Os autores propõem analisar o Preço de Anarquia Sequencial (*SPoA*)

Introdução

- Simultaneidade é usualmente difícil ou impossível de se alcançar nas implementações
- Os autores propõem analisar o Preço de Anarquia Sequencial (*SPoA*)
- Três jogos foram analisados
  - *Cost Sharing Games*
  - *Unrelated Machine Scheduling Games*
  - *Consensus/Cut Games*

Definições - Jogos Sequenciais

- Vamos considerar jogos sequenciais com informação completa (*Extensive Form Games*)
  - *Subgame Perfect Equilibrium - SPE*
  - *SPE* sempre existe e pode ser calculado através de indução retrativa (*backwards induction*)
  - todo *SPE* é um equilíbrio puro

Definições - Jogos Sequenciais

- Vamos considerar jogos sequenciais com informação completa (*Extensive Form Games*)
  - *Subgame Perfect Equilibrium - SPE*
  - *SPE* sempre existe e pode ser calculado através de indução retrativa (*backwards induction*)
  - todo *SPE* é um equilíbrio puro
- n jogadores

Definições - Jogos Sequenciais

- Vamos considerar jogos sequenciais com informação completa (*Extensive Form Games*)
  - *Subgame Perfect Equilibrium - SPE*
  - *SPE* sempre existe e pode ser calculado através de indução retrativa (*backwards induction*)
  - todo *SPE* é um equilíbrio puro
- n jogadores
- $A = \{A_1, \ldots, A_n\}$

Definições - Jogos Sequenciais

- Vamos considerar jogos sequenciais com informação completa (*Extensive Form Games*)
  - *Subgame Perfect Equilibrium - SPE*
  - *SPE* sempre existe e pode ser calculado através de indução retrativa (*backwards induction*)
  - todo *SPE* é um equilíbrio puro
- n jogadores
- $A = \{A_1, \ldots, A_n\}$
- $u_i : \times_i A_i \to \mathbb{R}, \forall i \in [n]$

Definições - Jogos Sequenciais

- Vamos considerar jogos sequenciais com informação completa (*Extensive Form Games*)
  - *Subgame Perfect Equilibrium - SPE*
  - *SPE* sempre existe e pode ser calculado através de indução retrativa (*backwards induction*)
  - todo *SPE* é um equilíbrio puro
- n jogadores
- $A = \{A_1, \ldots, A_n\}$
- $u_i : \times_i A_i \rightarrow \mathbb{R}, \forall i \in [n]$
- uma ordenação de jogadores (ex.: $1, 2, \ldots, n$)

Definições - Jogos Sequenciais

- Vamos considerar jogos sequenciais com informação completa (*Extensive Form Games*)
  - *Subgame Perfect Equilibrium - SPE*
  - *SPE* sempre existe e pode ser calculado através de indução retrativa (*backwards induction*)
  - todo *SPE* é um equilíbrio puro
- n jogadores
- $A = \{A_1, \ldots, A_n\}$
- $u_i : \times_i A_i \rightarrow \mathbb{R}, \forall i \in [n]$
- uma ordenação de jogadores (ex.: $1, 2, \ldots, n$)
- Rodada i, o jogador i observa as ações escolhidas pelos jogadores $1, 2, \ldots, i-1$ e escolhe uma ação $a_i \in A_i$
  - $s_i : A_1 \times \ldots \times A_{i-1} \rightarrow A_i$

Definições - Jogos Sequenciais

- Seja a função social $W : \times_i A_i \to \mathbb{R}_+$
- *Utility Games*
  - $SPoA = \max_{a \in SPE} \frac{W^*}{W(a)}$
- *Cost Games*
  - $SPoA = \max_{a \in SPE} \frac{W(a)}{W^*}$

Machine Cost Sharing Games

- conjunto N com n tarefas
- conjunto R com m máquinas

Machine Cost Sharing Games

- conjunto N com n tarefas
- conjunto R com m máquinas
- $\forall i \in N : R_i$

Machine Cost Sharing Games

- conjunto N com n tarefas
- conjunto R com m máquinas
- $\forall i \in N : R_i$
- $\forall r \in R$ é associada com uma função de custo decrescente $\gamma_r(x)$

Machine Cost Sharing Games

- conjunto N com n tarefas
- conjunto R com m máquinas
- $\forall i \in N : R_i$
- $\forall r \in R$ é associada com uma função de custo decrescente $\gamma_r(x)$
- tarefa i é um jogador e sua estratégia é escolher uma máquina $s_i \in R_i$

Machine Cost Sharing Games

- conjunto N com n tarefas
- conjunto R com m máquinas
- $\forall i \in N : R_i$
- $\forall r \in R$ é associada com uma função de custo decrescente $\gamma_r(x)$
- tarefa i é um jogador e sua estratégia é escolher uma máquina $s_i \in R_i$
- Dado um perfil de estratégias s:

$$c_i(s) = \gamma_{s_i}(n_{s_i}), \text{where } n_r = |\{j \in N : s_j = r\}|$$

Machine Cost Sharing Games

- conjunto N com n tarefas
- conjunto R com m máquinas
- $\forall i \in N : R_i$
- $\forall r \in R$ é associada com uma função de custo decrescente $\gamma_r(x)$
- tarefa i é um jogador e sua estratégia é escolher uma máquina $s_i \in R_i$
- Dado um perfil de estratégias s:

$$c_i(s) = \gamma_{s_i}(n_{s_i}), \text{where } n_r = |\{j \in N : s_j = r\}|$$

- $\forall r \in R : \gamma_r(x) = c_r/x$ (alocação de custo justa)

Machine Cost Sharing Games

- Em sua versão simultânea, sob a função de custo social $C(s) = \sum_i c_i(s)$
  - $PoA = n$
    - Ex.:$m = 2(c_{m_1} = 1 + \epsilon, c_{m_2} = n)$
    - $\forall i \in N : R_i = \{m_1, m_2\}$

Definição

Dizemos que as máquinas possuem custos genéricos se

$$c_r/k \neq c'_r/k', \forall r \neq r', 1 \leq k, k' \leq n.$$

Definição

Dizemos que as máquinas possuem custos genéricos se

$$c_r/k \neq c'_r/k', \forall r \neq r', 1 \leq k, k' \leq n.$$

Teorema (1)

Para qualquer jogo de compartilhamento de custo de máquinas com alocação de custo justa e custos genéricos, existe um único SPE que está a um fator de $O(\log n)$ do ótimo. Além disso, o SPE pode ser computado através de um algoritmo guloso natural. Quando os custos não são genéricos, podem existir mais de um SPE mas o limitante do SPoA ainda é mantido.

Machine Cost Sharing Games

Teorema (1)

Para qualquer jogo de compartilhamento de custo de máquinas com alocação de custo justa e custos genéricos, existe um único SPE que está a um fator de $O(\log n)$ do ótimo. Além disso, o SPE pode ser computado através de um algoritmo guloso natural. Quando os custos não são genéricos, podem existir mais de um SPE mas o limitante do SPoA ainda é mantido.

Proof.

- Encontrar s que minimiza $C(s)$ pode ser modelado como o problema da cobertura de conjuntos
  - jogadores são elementos
  - máquinas são representadas pelo conjunto de jogadores que elas podem servir

Teorema (1)

Para qualquer jogo de compartilhamento de custo de máquinas com alocação de custo justa e custos genéricos, existe um único SPE que está a um fator de $O(\log n)$ do ótimo. Além disso, o SPE pode ser computado através de um algoritmo guloso natural. Quando os custos não são genéricos, podem existir mais de um SPE mas o limitante do SPoA ainda é mantido.

Proof.

- Encontrar s que minimiza $C(s)$ pode ser modelado como o problema da cobertura de conjuntos
  - jogadores são elementos
  - máquinas são representadas pelo conjunto de jogadores que elas podem servir

Algorithm 1 ApproxAlg for Set Cover

```
ENQUANTO existir elementos não cobertos FAÇA
    escolha um conjunto que possui a menor razão de custo sob o número de elementos não coberto
FIM ENQUANTO
```

Machine Cost Sharing Games

Teorema (1)

Para qualquer jogo de compartilhamento de custo de máquinas com alocação de custo justa e custos genéricos, existe um único SPE que está a um fator de $O(\log n)$ do ótimo. Além disso, o SPE pode ser computado através de um algoritmo guloso natural. Quando os custos não são genéricos, podem existir mais de um SPE mas o limitante do SPoA ainda é mantido.

Proof.

- Temos que mostrar que o resultado do algoritmo é o único SPE

Machine Cost Sharing Games

Teorema (1)

Para qualquer jogo de compartilhamento de custo de máquinas com alocação de custo justa e custos genéricos, existe um único SPE que está a um fator de $O(\log n)$ do ótimo. Além disso, o SPE pode ser computado através de um algoritmo guloso natural. Quando os custos não são genéricos, podem existir mais de um SPE mas o limitante do SPoA ainda é mantido.

Proof.

- Temos que mostrar que o resultado do algoritmo é o único SPE
- Calculamos para $t = n, n-1, \ldots, 1$, a melhor ação para t (ação única pela suposição dos custos genéricos) e mostramos que no resultado da indução retrativa, todos os jogadores jogam de acordo com o resultado do algoritmo

Machine Cost Sharing Games

Teorema (1)

Para qualquer jogo de compartilhamento de custo de máquinas com alocação de custo justa e custos genéricos, existe um único SPE que está a um fator de $O(\log n)$ do ótimo. Além disso, o SPE pode ser computado através de um algoritmo guloso natural. Quando os custos não são genéricos, podem existir mais de um SPE mas o limitante do SPoA ainda é mantido.

Proof.

- Temos que mostrar que o resultado do algoritmo é o único SPE
- Calculamos para $t = n, n-1, \ldots, 1$, a melhor ação para t (ação única pela suposição dos custos genéricos) e mostramos que no resultado da indução retrativa, todos os jogadores jogam de acordo com o resultado do algoritmo
- Sejam
  - $r_1, r_2, \ldots, r_k$ as máquinas na ordem que foram escolhidas pelo algoritmo
  - N_j os jogadores que foram alocados para a máquina r_j

Machine Cost Sharing Games

Teorema (1)

Para qualquer jogo de compartilhamento de custo de máquinas com alocação de custo justa e custos genéricos, existe um único SPE que está a um fator de $O(\log n)$ do ótimo. Além disso, o SPE pode ser computado através de um algoritmo guloso natural. Quando os custos não são genéricos, podem existir mais de um SPE mas o limitante do SPoA ainda é mantido.

Proof.

- Basta mostrarmos que os jogadores não desejam desviar da solução do algoritmo em seu turno

Machine Cost Sharing Games

Teorema (1)

Para qualquer jogo de compartilhamento de custo de máquinas com alocação de custo justa e custos genéricos, existe um único SPE que está a um fator de $O(\log n)$ do ótimo. Além disso, o SPE pode ser computado através de um algoritmo guloso natural. Quando os custos não são genéricos, podem existir mais de um SPE mas o limitante do SPoA ainda é mantido.

Proof.

- Basta mostrarmos que os jogadores não desejam desviar da solução do algoritmo em seu turno
- Vamos considerar primeiro N_1

Machine Cost Sharing Games

Teorema (1)

Para qualquer jogo de compartilhamento de custo de máquinas com alocação de custo justa e custos genéricos, existe um único SPE que está a um fator de $O(\log n)$ do ótimo. Além disso, o SPE pode ser computado através de um algoritmo guloso natural. Quando os custos não são genéricos, podem existir mais de um SPE mas o limitante do SPoA ainda é mantido.

Proof.

- Basta mostrarmos que os jogadores não desejam desviar da solução do algoritmo em seu turno
- Vamos considerar primeiro N_1
  - jogadores possuem custo c_{r_1}/N_1

Machine Cost Sharing Games

Teorema (1)

Para qualquer jogo de compartilhamento de custo de máquinas com alocação de custo justa e custos genéricos, existe um único SPE que está a um fator de $O(\log n)$ do ótimo. Além disso, o SPE pode ser computado através de um algoritmo guloso natural. Quando os custos não são genéricos, podem existir mais de um SPE mas o limitante do SPoA ainda é mantido.

Proof.

- Basta mostrarmos que os jogadores não desejam desviar da solução do algoritmo em seu turno
- Vamos considerar primeiro N_1
  - jogadores possuem custo c_{r_1}/N_1
  - menor custo que um jogador pode ter

Machine Cost Sharing Games

Teorema (1)

Para qualquer jogo de compartilhamento de custo de máquinas com alocação de custo justa e custos genéricos, existe um único SPE que está a um fator de $O(\log n)$ do ótimo. Além disso, o SPE pode ser computado através de um algoritmo guloso natural. Quando os custos não são genéricos, podem existir mais de um SPE mas o limitante do SPoA ainda é mantido.

Proof.

- Basta mostrarmos que os jogadores não desejam desviar da solução do algoritmo em seu turno
- Vamos considerar primeiro N_1
  - jogadores possuem custo c_{r_1}/N_1
  - menor custo que um jogador pode ter
  - último jogador de N_1 escolherá r_1 (dado que os outros em N_1 também escolheram r_1)

Machine Cost Sharing Games

Teorema (1)

Para qualquer jogo de compartilhamento de custo de máquinas com alocação de custo justa e custos genéricos, existe um único SPE que está a um fator de $O(\log n)$ do ótimo. Além disso, o SPE pode ser computado através de um algoritmo guloso natural. Quando os custos não são genéricos, podem existir mais de um SPE mas o limitante do SPoA ainda é mantido.

Proof.

- Basta mostrarmos que os jogadores não desejam desviar da solução do algoritmo em seu turno
- Vamos considerar primeiro N_1
  - jogadores possuem custo c_{r_1}/N_1
  - menor custo que um jogador pode ter
  - último jogador de N_1 escolherá r_1 (dado que os outros em N_1 também escolheram r_1) , o penúltimo,...
  - independente das decisões dos jogadores fora de N_1

Machine Cost Sharing Games

Teorema (1)

Para qualquer jogo de compartilhamento de custo de máquinas com alocação de custo justa e custos genéricos, existe um único SPE que está a um fator de $O(\log n)$ do ótimo. Além disso, o SPE pode ser computado através de um algoritmo guloso natural. Quando os custos não são genéricos, podem existir mais de um SPE mas o limitante do SPoA ainda é mantido.

Proof.

- jogadores em N_2

Machine Cost Sharing Games

Teorema (1)

Para qualquer jogo de compartilhamento de custo de máquinas com alocação de custo justa e custos genéricos, existe um único SPE que está a um fator de $O(\log n)$ do ótimo. Além disso, o SPE pode ser computado através de um algoritmo guloso natural. Quando os custos não são genéricos, podem existir mais de um SPE mas o limitante do SPoA ainda é mantido.

Proof.

- jogadores em N_2
  - (jogadores de N_1 escolheram r_1)

Teorema (1)

Para qualquer jogo de compartilhamento de custo de máquinas com alocação de custo justa e custos genéricos, existe um único SPE que está a um fator de $O(\log n)$ do ótimo. Além disso, o SPE pode ser computado através de um algoritmo guloso natural. Quando os custos não são genéricos, podem existir mais de um SPE mas o limitante do SPoA ainda é mantido.

Proof.

- jogadores em N_2
  - (jogadores de N_1 escolheram r_1)
  - então, pela definição do algoritmo, o menor custo possível para os jogadores em N_2 é c_{r_2}/N_2

Teorema (1)

Para qualquer jogo de compartilhamento de custo de máquinas com alocação de custo justa e custos genéricos, existe um único SPE que está a um fator de $O(\log n)$ do ótimo. Além disso, o SPE pode ser computado através de um algoritmo guloso natural. Quando os custos não são genéricos, podem existir mais de um SPE mas o limitante do SPoA ainda é mantido.

Proof.

- jogadores em N_2
  - (jogadores de N_1 escolheram r_1)
  - então, pela definição do algoritmo, o menor custo possível para os jogadores em N_2 é c_{r_2}/N_2
  - empregamos o mesmo argumento utilizado anteriormente

$\square$

Observação (2)

O resultante SPE é independente da ordem em que os jogadores jogam. Além disso, os jogadores não precisam saber a ordem dos outros jogadores para encontrar sua escolha ótima. Consequentemente, o SPE também é um equilibrio de Nash do jogo.

- O algoritmo guloso também funciona para funções de custo decrescentes mais gerais
  - escolher máquina r com menor $\gamma_r(d_r)$, onde d_r é o número de jogadores não cobertos que podem ser alocados para a máquina r.

- O algoritmo guloso também funciona para funções de custo decrescentes mais gerais
  - escolher máquina r com menor $\gamma_r(d_r)$, onde d_r é o número de jogadores não cobertos que podem ser alocados para a máquina r.
- Neste cenário mais geral, o $SPoA \leq$ o melhor limitante superior do PoS que pode ser derivado pelo método potencial

Machine Cost Sharing Games

- O algoritmo guloso também funciona para funções de custo decrescentes mais gerais
  - escolher máquina r com menor $\gamma_r(d_r)$, onde d_r é o número de jogadores não cobertos que podem ser alocados para a máquina r.
- Neste cenário mais geral, o $SPoA \leq$ o melhor limitante superior do PoS que pode ser derivado pelo método potencial

Teorema (3)

Para jogos de compartilhamento de custo de máquinas com funções de custo decrescentes arbitrárias, o bem-estar social de qualquer SPE é no máximo o potencial da solução ótima.

Unrelated Machine Scheduling

- conjunto N com n tarefas
- conjunto M com m máquinas não-relacionadas

Unrelated Machine Scheduling

- conjunto N com n tarefas
- conjunto M com m máquinas não-relacionadas
- t_{ji}: tempo de processamento da tarefa j na máquina i

Unrelated Machine Scheduling

- conjunto N com n tarefas
- conjunto M com m máquinas não-relacionadas
- t_{ji}: tempo de processamento da tarefa j na máquina i
- Objetivo: $\min \max_{i \in M} \sum_{j;\phi(j)=i} t_{ji}$

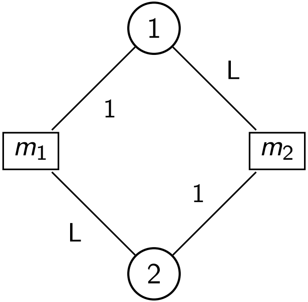
Unrelated Machine Scheduling

- conjunto N com n tarefas
- conjunto M com m máquinas não-relacionadas
- t_{ji}: tempo de processamento da tarefa j na máquina i
- Objetivo: $\min \max_{i \in M} \sum_{j; \phi(j)=i} t_{ji}$
- Cada jogador (tarefa) i:
  - escolhe em qual máquina aloca a tarefa.
  - $S_i \subseteq M$
  - custo de i é a carga da máquina na qual ele foi alocado

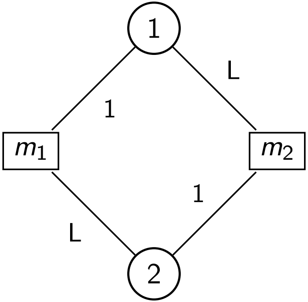
Unrelated Machine Scheduling

- Esse jogo sempre possui equilíbrio puro de Nash [Vöc07]

- Esse jogo sempre possui equilíbrio puro de Nash [Vöc07]
- PoA pode ser arbitrariamente grande. Considere exemplo abaixo com $L >> 1$

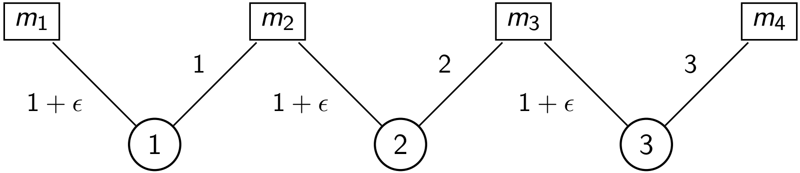


- Esse jogo sempre possui equilíbrio puro de Nash [Vöc07]
- PoA pode ser arbitrariamente grande. Considere exemplo abaixo com $L >> 1$

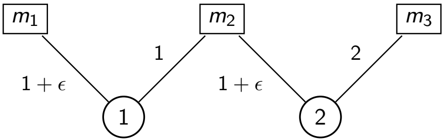


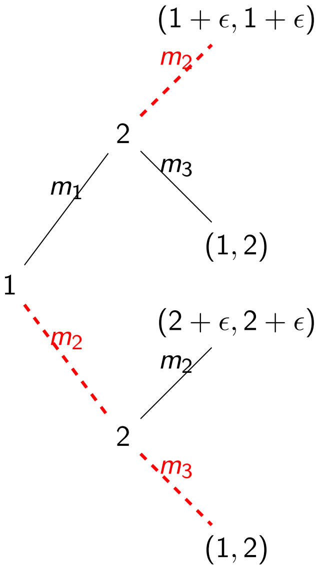
- $\Omega(n) \leq SPoA \leq O(m2^n)$

- Considere a generalização do exemplo a seguir:



Unrelated Machine Scheduling - limitante inferior





Unrelated Machine Scheduling - limitante superior

Teorema (4)

O SPoA dos jogos de balanceamento de carga em máquinas não-relacionadas é limitado por $O(m2^n)$.

Teorema (4)

O SPoA dos jogos de balanceamento de carga em máquinas não-relacionadas é limitado por $O(m2^n)$.

Proof.

- $\vec{L_0} \in \mathbb{R}_+^M$: carga inicial de cada máquina

Unrelated Machine Scheduling - limitante superior

Teorema (4)

O SPoA dos jogos de balanceamento de carga em máquinas não-relacionadas é limitado por $O(m2^n)$.

Proof.

- $\vec{L_0} \in \mathbb{R}_+^M$: carga inicial de cada máquina
- $SPE(\vec{L_0}, k)$: *makespan* do SPE quando os jogadores $k, k+1, \ldots, n$ jogam a partir de $\vec{L_0}$

Unrelated Machine Scheduling - limitante superior

Teorema (4)

O SPoA dos jogos de balanceamento de carga em máquinas não-relacionadas é limitado por $O(m2^n)$.

Proof.

- $\vec{L_0} \in \mathbb{R}_+^M$: carga inicial de cada máquina
- $SPE(\vec{L_0}, k)$: *makespan* do SPE quando os jogadores $k, k+1, \ldots, n$ jogam a partir de $\vec{L_0}$
- $t_j^* = \min_{i \in M} t_{ji}$

Unrelated Machine Scheduling - limitante superior

Teorema (4)

O SPoA dos jogos de balanceamento de carga em máquinas não-relacionadas é limitado por $O(m2^n)$.

Proof.

- $\vec{L_0} \in \mathbb{R}_+^M$: carga inicial de cada máquina
- $SPE(\vec{L_0}, k)$: *makespan* do SPE quando os jogadores $k, k+1, \ldots, n$ jogam a partir de $\vec{L_0}$
- $t_j^* = \min_{i \in M} t_{ji}$
- Hipótese de indução:

$$\forall \vec{L_0} \in \mathbb{R}_+^M, SPE(\vec{L_0}, k) \leq ||\vec{L_0}||_\infty + 2^{n-k} \sum_{j=k}^{n} t_j^*$$

Unrelated Machine Scheduling - limitante superior

Teorema (4)

O SPoA dos jogos de balanceamento de carga em máquinas não-relacionadas é limitado por $O(m2^n)$.

Proof.

- O teorema segue: $\vec{L_0} = \vec{0}$, $k = 1$, $\sum_{j=k}^{n} t_j^* \leq mOPT$

Unrelated Machine Scheduling - limitante superior

Teorema (4)

O SPoA dos jogos de balanceamento de carga em máquinas não-relacionadas é limitado por $O(m2^n)$.

Proof.

- O teorema segue: $\vec{L_0} = \vec{0}$, $k = 1$, $\sum_{j=k}^{n} t_j^* \leq mOPT$

$$SPE(\vec{0}, 1) \leq 2^{n-1} \sum_{j=1}^{n} t_j^* \leq OPT \cdot m2^{n-1}$$

Teorema (4)

O SPoA dos jogos de balanceamento de carga em máquinas não-relacionadas é limitado por $O(m2^n)$.

Proof.

- O teorema segue: $\vec{L_0} = \vec{0}$, $k = 1$, $\sum_{j=k}^{n} t_j^* \leq mOPT$

$$SPE(\vec{0}, 1) \leq 2^{n-1} \sum_{j=1}^{n} t_j^* \leq OPT \cdot m2^{n-1}$$

- Base: $k = n$

$$makespan \leq ||\vec{L_0}||_\infty + t_n^*$$

Unrelated Machine Scheduling - limitante superior

Teorema (4)

O SPoA dos jogos de balanceamento de carga em máquinas não-relacionadas é limitado por $O(m2^n)$.

Proof.

- Suponha que a hipótese é válida para $k+1, \ldots, n$

Unrelated Machine Scheduling - limitante superior

Teorema (4)

O SPoA dos jogos de balanceamento de carga em máquinas não-relacionadas é limitado por $O(m2^n)$.

Proof.

- Suponha que a hipótese é válida para $k+1, \ldots, n$
- Jogador k
  - pode escolher a máquina em que ele terá t_k^*

Unrelated Machine Scheduling - limitante superior

Teorema (4)

O SPoA dos jogos de balanceamento de carga em máquinas não-relacionadas é limitado por $O(m2^n)$.

Proof.

- Suponha que a hipótese é válida para $k+1, \ldots, n$
- Jogador k
  - pode escolher a máquina em que ele terá t_k^*
  - Seja $\vec{L_1^*}$ o novo vetor de cargas após essa jogada

Unrelated Machine Scheduling - limitante superior

Teorema (4)

O SPoA dos jogos de balanceamento de carga em máquinas não-relacionadas é limitado por $O(m2^n)$.

Proof.

- Suponha que a hipótese é válida para $k+1, \ldots, n$
- Jogador k
  - pode escolher a máquina em que ele terá t_k^*
  - Seja $\vec{L_1^*}$ o novo vetor de cargas após essa jogada
  - $||\vec{L_1^*}||_\infty \leq ||\vec{L_0}||_\infty + t_k^*$

Unrelated Machine Scheduling - limitante superior

Teorema (4)

O SPoA dos jogos de balanceamento de carga em máquinas não-relacionadas é limitado por $O(m2^n)$.

Proof.

- Suponha que a hipótese é válida para $k+1, \ldots, n$
- Jogador k
  - pode escolher a máquina em que ele terá t_k^*
  - Seja $\vec{L_1^*}$ o novo vetor de cargas após essa jogada
  - $||\vec{L_1^*}||_\infty \leq ||\vec{L_0}||_\infty + t_k^*$
  - Pela hipótese de indução, o *makespan* (e consequentemente o custo do jogador k) é no máximo

$$||\vec{L_1^*}||_\infty + 2^{n-k-1} \sum_{j=k+1}^{n} t_j^*$$

Unrelated Machine Scheduling - limitante superior

Teorema (4)

O SPoA dos jogos de balanceamento de carga em máquinas não-relacionadas é limitado por $O(m2^n)$.

Proof.

- Suponha que a hipótese é válida para $k+1,\ldots,n$
- Jogador k
  - pode escolher a máquina em que ele terá t_k^*
  - Seja $\vec{L_1^*}$ o novo vetor de cargas após essa jogada
  - $||\vec{L_1^*}||_\infty \leq ||\vec{L_0}||_\infty + t_k^*$
  - Pela hipótese de indução, o *makespan* (e consequentemente o custo do jogador k) é no máximo

$$||\vec{L_1^*}||_\infty + 2^{n-k-1}\sum_{j=k+1}^{n} t_j^* \leq ||\vec{L_0}||_\infty + t_k^* + 2^{n-k-1}\sum_{j=k+1}^{n} t_j^*$$

Unrelated Machine Scheduling - limitante superior

Teorema (4)

O SPoA dos jogos de balanceamento de carga em máquinas não-relacionadas é limitado por $O(m2^n)$.

Proof.

- No equilíbrio, o jogador k pode escolher uma outra máquina i

Unrelated Machine Scheduling - limitante superior

Teorema (4)

O SPoA dos jogos de balanceamento de carga em máquinas não-relacionadas é limitado por $O(m2^n)$.

Proof.

- No equilíbrio, o jogador k pode escolher uma outra máquina i
- Seja $\vec{L_1}$ o novo vetor de cargas após essa jogada

Unrelated Machine Scheduling - limitante superior

Teorema (4)

O SPoA dos jogos de balanceamento de carga em máquinas não-relacionadas é limitado por $O(m2^n)$.

Proof.

- No equilíbrio, o jogador k pode escolher uma outra máquina i
- Seja $\vec{L_1}$ o novo vetor de cargas após essa jogada
- $||\vec{L_1}||_\infty \leq ||\vec{L_0}||_\infty + t_k^* + 2^{n-k-1}\sum_{j=k+1}^{n} t_j^*$

Unrelated Machine Scheduling - limitante superior

Teorema (4)

O SPoA dos jogos de balanceamento de carga em máquinas não-relacionadas é limitado por $O(m2^n)$.

Proof.

- No equilíbrio, o jogador k pode escolher uma outra máquina i
- Seja $\vec{L_1}$ o novo vetor de cargas após essa jogada
- $||\vec{L_1}||_\infty \leq ||\vec{L_0}||_\infty + t_k^* + 2^{n-k-1}\sum_{j=k+1}^{n} t_j^*$
- Pela hipótese de indução

$$SPE(\vec{L_1}, k+1) \leq ||\vec{L_1}||_\infty + 2^{n-k-1}\sum_{j=k+1}^{n} t_j^*$$

□

Unrelated Machine Scheduling - limitante superior

Teorema (4)

O SPoA dos jogos de balanceamento de carga em máquinas não-relacionadas é limitado por $O(m2^n)$.

Proof.

- No equilíbrio, o jogador k pode escolher uma outra máquina i
- Seja $\vec{L_1}$ o novo vetor de cargas após essa jogada
- $||\vec{L_1}||_\infty \leq ||\vec{L_0}||_\infty + t_k^* + 2^{n-k-1} \sum_{j=k+1}^{n} t_j^*$
- Pela hipótese de indução

$$
\begin{aligned}
SPE(\vec{L_1}, k+1) &\leq ||\vec{L_1}||_\infty + 2^{n-k-1} \sum_{j=k+1}^{n} t_j^* \\
&\leq ||\vec{L_0}||_\infty + t_k^* + 2^{n-k-1} \sum_{j=k+1}^{n} t_j^* + 2^{n-k-1} \sum_{j=k+1}^{n} t_j^*
\end{aligned}
$$

$\square$

Unrelated Machine Scheduling - limitante superior

Teorema (4)

O SPoA dos jogos de balanceamento de carga em máquinas não-relacionadas é limitado por $O(m2^n)$.

Proof.

- No equilíbrio, o jogador k pode escolher uma outra máquina i
- Seja $\vec{L_1}$ o novo vetor de cargas após essa jogada
- $||\vec{L_1}||_\infty \leq ||\vec{L_0}||_\infty + t_k^* + 2^{n-k-1} \sum_{j=k+1}^{n} t_j^*$
- Pela hipótese de indução

$$\begin{aligned} SPE(\vec{L_1}, k+1) &\leq ||\vec{L_1}||_\infty + 2^{n-k-1} \sum_{j=k+1}^{n} t_j^* \\ &\leq ||\vec{L_0}||_\infty + t_k^* + 2^{n-k-1} \sum_{j=k+1}^{n} t_j^* + 2^{n-k-1} \sum_{j=k+1}^{n} t_j^* \\ &\leq ||\vec{L_0}||_\infty + 2^{n-k} \sum_{j=k}^{n} t_j^* = SPE(\vec{L_0}, k) \end{aligned}$$

□

Unrelated Machine Scheduling - limitante superior

- Bilò et al. [BFMM12] melhorarm o gap do SPoA

$$\Omega(2^{\Omega(\sqrt{n})}) \leq SPoA \leq O(2^n)$$

- Bilò et al. [BFMM12] melhorarm o gap do SPoA

$$\Omega(2^{\Omega(\sqrt{n})}) \leq SPoA \leq O(2^n)$$

- Limitante superior

$$OPT \geq \max_{j \in N} t_j^*$$

Consensus and Cut Games

- Grafo ponderado $G = (V, E, w)$
  - jogadores: $|V| = n$
  - $w : E \to \mathbb{R}_+$
- $\forall i \in [n], S_i = \{R, B\}$

Consensus and Cut Games

- Grafo ponderado $G = (V, E, w)$
  - jogadores: $|V| = n$
  - $w : E \to \mathbb{R}_+$
- $\forall i \in [n], S_i = \{R, B\}$
- Dizemos que um vetor peso w é genérico, se $w_i \neq 0$

Consensus and Cut Games

- Grafo ponderado $G = (V, E, w)$
  - jogadores: $|V| = n$
  - $w : E \to \mathbb{R}_+$
- $\forall i \in [n], S_i = \{R, B\}$
- Dizemos que um vetor peso w é genérico, se $w_i \neq 0$
- Consensus Games (*Cost Games*)
  - $c_i(s)$: soma dos pesos das arestas de i para jogadores com diferente cor
  - A solução ótima corresponde a todos os jogadores escolherem a mesma cor

Consensus and Cut Games

- Grafo ponderado $G = (V, E, w)$
  - jogadores: $|V| = n$
  - $w : E \to \mathbb{R}_+$
- $\forall i \in [n], S_i = \{R, B\}$
- Dizemos que um vetor peso w é genérico, se $w_i \neq 0$
- Consensus Games (*Cost Games*)
  - $c_i(s)$: soma dos pesos das arestas de i para jogadores com diferente cor
  - A solução ótima corresponde a todos os jogadores escolherem a mesma cor
  - Na versão simultânea, existem instâncias que apresentam equilíbrios puros de Nash que não são iguais ao ótimo

Consensus and Cut Games

- Grafo ponderado $G = (V, E, w)$
  - jogadores: $|V| = n$
  - $w : E \rightarrow \mathbb{R}_+$
- $\forall i \in [n], S_i = \{R, B\}$
- Dizemos que um vetor peso w é genérico, se $w_i \neq 0$
- Consensus Games (*Cost Games*)
  - $c_i(s)$: soma dos pesos das arestas de i para jogadores com diferente cor
  - A solução ótima corresponde a todos os jogadores escolherem a mesma cor
  - Na versão simultânea, existem instâncias que apresentam equilíbrios puros de Nash que não são iguais ao ótimo

Observação (5)

The unique SPE in generic consensus games is the optimal outcome.

- Cut Games(*Utility Games*)
  - $u_i(s)$: soma dos pesos das arestas de i para jogadores com diferente cor

Consensus and Cut Games

- Cut Games(*Utility Games*)
  - $u_i(s)$: soma dos pesos das arestas de i para jogadores com diferente cor
- Versão simultânea apresenta $PoA = 2$.

Consensus and Cut Games

- Cut Games(*Utility Games*)
  - $u_i(s)$: soma dos pesos das arestas de i para jogadores com diferente cor
- Versão simultânea apresenta $PoA = 2$.
- "Racionalidade sequencial" não melhora o preço de anarquia

- Cut Games(*Utility Games*)
  - $u_i(s)$: soma dos pesos das arestas de i para jogadores com diferente cor
- Versão simultânea apresenta $PoA = 2$.
- "Racionalidade sequencial" não melhora o preço de anarquia

Teorema (6)

The SPoA of sequential cut games is at most 4.

Consensus and Cut Games

Teorema (6)

The SPoA of sequential cut games is at most 4.

Proof.

- Considere os jogadores na ordem em que eles aparecem

Consensus and Cut Games

Teorema (6)

The SPoA of sequential cut games is at most 4.

Proof.

- Considere os jogadores na ordem em que eles aparecem
- $E_k : \{(i, k) | i < k\}$

Consensus and Cut Games

Teorema (6)

The SPoA of sequential cut games is at most 4.

Proof.

- Considere os jogadores na ordem em que eles aparecem
- $E_k : \{(i,k) | i < k\}$
- Sejam A, B duas partições dos nós no SPE

Consensus and Cut Games

Teorema (6)

The SPoA of sequential cut games is at most 4.

Proof.

- Considere os jogadores na ordem em que eles aparecem
- $E_k : \{(i,k) | i < k\}$
- Sejam A, B duas partições dos nós no SPE
- Jogador k
  - S.p.d.g.: $w(E_k \cap A) \geq w(E_k \cap B) \geq \frac{1}{2} w(E_k)$

Consensus and Cut Games

Teorema (6)

The SPoA of sequential cut games is at most 4.

Proof.

- Considere os jogadores na ordem em que eles aparecem
- $E_k : \{(i,k) | i < k\}$
- Sejam A, B duas partições dos nós no SPE
- Jogador k
  - S.p.d.g.: $w(E_k \cap A) \geq w(E_k \cap B) \geq \frac{1}{2} w(E_k)$
  - Se jogando no equilíbrio, ele desejar mudar para B, então sua utilidade é de pelo menos $\frac{1}{2} w(E_k)$

Consensus and Cut Games

Teorema (6)

The SPoA of sequential cut games is at most 4.

Proof.

- Considere os jogadores na ordem em que eles aparecem
- $E_k : \{(i,k) | i < k\}$
- Sejam A, B duas partições dos nós no SPE
- Jogador k
  - S.p.d.g.: $w(E_k \cap A) \geq w(E_k \cap B) \geq \frac{1}{2} w(E_k)$
  - Se jogando no equilíbrio, ele desejar mudar para B, então sua utilidade é de pelo menos $\frac{1}{2} w(E_k)$
  - Então, $u_k(SPE) \geq \frac{1}{2} w(E_k)$

Consensus and Cut Games

Teorema (6)

The SPoA of sequential cut games is at most 4.

Proof.

- Considere os jogadores na ordem em que eles aparecem
- $E_k : \{(i,k) | i < k\}$
- Sejam A, B duas partições dos nós no SPE
- Jogador k
  - S.p.d.g.: $w(E_k \cap A) \geq w(E_k \cap B) \geq \frac{1}{2} w(E_k)$
  - Se jogando no equilíbrio, ele desejar mudar para B, então sua utilidade é de pelo menos $\frac{1}{2} w(E_k)$
  - Então, $u_k(SPE) \geq \frac{1}{2} w(E_k)$
- Somando para todos os k

$$2SPE = \sum_k u_k$$

Consensus and Cut Games

Teorema (6)

The SPoA of sequential cut games is at most 4.

Proof.

- Considere os jogadores na ordem em que eles aparecem
- $E_k : \{(i,k) | i < k\}$
- Sejam A, B duas partições dos nós no SPE
- Jogador k
  - S.p.d.g.: $w(E_k \cap A) \geq w(E_k \cap B) \geq \frac{1}{2}w(E_k)$
  - Se jogando no equilíbrio, ele desejar mudar para B, então sua utilidade é de pelo menos $\frac{1}{2}w(E_k)$
  - Então, $u_k(SPE) \geq \frac{1}{2}w(E_k)$
- Somando para todos os k

$$2SPE = \sum_k u_k \geq \frac{1}{2}\sum_k w(E_k)$$

Consensus and Cut Games

Teorema (6)

The SPoA of sequential cut games is at most 4.

Proof.

- Considere os jogadores na ordem em que eles aparecem
- $E_k : \{(i, k) | i < k\}$
- Sejam A, B duas partições dos nós no SPE
- Jogador k
  - S.p.d.g.: $w(E_k \cap A) \geq w(E_k \cap B) \geq \frac{1}{2} w(E_k)$
  - Se jogando no equilíbrio, ele desejar mudar para B, então sua utilidade é de pelo menos $\frac{1}{2} w(E_k)$
  - Então, $u_k(SPE) \geq \frac{1}{2} w(E_k)$
- Somando para todos os k

$$2SPE = \sum_k u_k \geq \frac{1}{2} \sum_k w(E_k) = \frac{1}{2} w(E)$$

Teorema (6)

The SPoA of sequential cut games is at most 4.

Proof.

- Considere os jogadores na ordem em que eles aparecem
- $E_k : \{(i, k) | i < k\}$
- Sejam A, B duas partições dos nós no SPE
- Jogador k
  - S.p.d.g.: $w(E_k \cap A) \geq w(E_k \cap B) \geq \frac{1}{2}w(E_k)$
  - Se jogando no equilíbrio, ele desejar mudar para B, então sua utilidade é de pelo menos $\frac{1}{2}w(E_k)$
  - Então, $u_k(SPE) \geq \frac{1}{2}w(E_k)$
- Somando para todos os k

$$2SPE = \sum_k u_k \geq \frac{1}{2}\sum_k w(E_k) = \frac{1}{2}w(E) \geq \frac{1}{2}OPT$$

References I

- Vittorio Bilò, Michele Flammini, Gianpiero Monaco, and Luca Moscardelli, *Some anomalies of farsighted strategic behavior*, International Workshop on Approximation and Online Algorithms, Springer, 2012, pp. 229–241.
- Renato Paes Leme, Vasilis Syrgkanis, and Éva Tardos, *The curse of simultaneity*, Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, ACM, 2012, pp. 60–67.
- Berthold Vöcking, *Selfish load balancing*, Algorithmic game theory 20 (2007), 517–542.