Jogo de balanceamento de carga

Dados:

- n tarefas
- m máquinas
- w_i: peso da tarefa i
- s_j: velocidade da máquina j

Jogo de balanceamento de carga

Dados:

- n tarefas
- m máquinas
- w_i: peso da tarefa i
- s_j: velocidade da máquina j

Cada jogador controla uma tarefa.

Conjuntos de estratégias $S_i = [m]$:
o jogador escolhe a qual máquina atribuir sua tarefa.

Jogo de balanceamento de carga

Dados:

- n tarefas
- m máquinas
- w_i: peso da tarefa i
- s_j: velocidade da máquina j

Cada jogador controla uma tarefa.

Conjuntos de estratégias $S_i = [m]$:
o jogador escolhe a qual máquina atribuir sua tarefa.

Vetor de estratégias: atribuição $A : [n] \to [m]$.

Jogo de balanceamento de carga

Dados:

- n tarefas
- m máquinas
- w_i: peso da tarefa i
- s_j: velocidade da máquina j

Cada jogador controla uma tarefa.

Conjuntos de estratégias $S_i = [m]$:
o jogador escolhe a qual máquina atribuir sua tarefa.

Vetor de estratégias: atribuição $A : [n] \to [m]$.

Custo de A para i, onde $j = A(i)$: $\text{custo}_i(A) = \sum_{k:A(k)=j} \frac{w_k}{s_j}$.

Jogo de balanceamento de carga

Vimos na aula passada:

Jogo de balanceamento de carga

Vimos na aula passada:

Jogo com estratégias puras sempre tem equilíbrio.

Jogo de balanceamento de carga

Vimos na aula passada:

Jogo com estratégias puras sempre tem equilíbrio.

Para m máquinas idênticas,
o preço da anarquia é $\left(2 - \frac{2}{m+1}\right)$.

Jogo de balanceamento de carga

Vimos na aula passada:

Jogo com estratégias puras sempre tem equilíbrio.

Para m máquinas idênticas,
o preço da anarquia é $\left(2 - \frac{2}{m+1}\right)$.

Nesta aula:

- tempo de convergência para máquinas idênticas

Jogo de balanceamento de carga

Vimos na aula passada:

Jogo com estratégias puras sempre tem equilíbrio.

Para m máquinas idênticas,

o preço da anarquia é $\left(2 - \frac{2}{m+1}\right)$.

Nesta aula:

- tempo de convergência para máquinas idênticas
- preço da anarquia para máquinas relacionadas

Jogo de balanceamento de carga

Vimos na aula passada:

Jogo com estratégias puras sempre tem equilíbrio.

Para m máquinas idênticas,
o preço da anarquia é $\left(2 - \frac{2}{m+1}\right)$.

Nesta aula:

- tempo de convergência para máquinas idênticas
- preço da anarquia para máquinas relacionadas
- tempo de convergência para máquinas relacionadas

Tempo de convergência

Para máquinas idênticas,
há sequência curta de melhoras
de qualquer atribuição inicial para um equilíbrio.

Tempo de convergência

Para máquinas idênticas,
há sequência curta de melhoras
de qualquer atribuição inicial para um equilíbrio.

Política da melhor resposta de peso máximo:

Tempo de convergência

Para máquinas idênticas,
há sequência curta de melhoras
de qualquer atribuição inicial para um equilíbrio.

Política da melhor resposta de peso máximo:

Ative apenas tarefas insatisfeitas de peso máximo.

Tempo de convergência

Para máquinas idênticas,
há sequência curta de melhoras
de qualquer atribuição inicial para um equilíbrio.

Política da melhor resposta de peso máximo:

Ative apenas tarefas insatisfeitas de peso máximo.

Uma tarefa ativa muda para melhor máquina.

Tempo de convergência

Para máquinas idênticas,
há sequência curta de melhoras
de qualquer atribuição inicial para um equilíbrio.

Política da melhor resposta de peso máximo:

Ative apenas tarefas insatisfeitas de peso máximo.

Uma tarefa ativa muda para melhor máquina.

Teorema: A política de melhor resposta de peso máximo atinge um equilíbrio depois de no máximo n passos.

Tempo de convergência

Para máquinas idênticas,
há sequência curta de melhoras
de qualquer atribuição inicial para um equilíbrio.

Política da melhor resposta de peso máximo:

Ative apenas tarefas insatisfeitas de peso máximo.

Uma tarefa ativa muda para melhor máquina.

Teorema: A política de melhor resposta de peso máximo atinge um equilíbrio depois de no máximo n passos.

Prova: Tarefa que migra nunca mais fica insatisfeita.

Tempo de convergência

Máquinas idênticas

Política da melhor resposta de peso máximo:
Ative apenas tarefas insatisfeitas de peso máximo.
Uma tarefa ativa muda para melhor máquina.

Teorema: A política de melhor resposta de peso máximo atinge um equilíbrio depois de no máximo n passos.

Prova: Tarefa que migra nunca mais fica insatisfeita.

Tempo de convergência

Máquinas idênticas

Política da melhor resposta de peso máximo:
Ative apenas tarefas insatisfeitas de peso máximo.
Uma tarefa ativa muda para melhor máquina.

Teorema: A política de melhor resposta de peso máximo atinge um equilíbrio depois de no máximo n passos.

Prova: Tarefa que migra nunca mais fica insatisfeita.

- A carga mínima não diminui.

Tempo de convergência

Máquinas idênticas

Política da melhor resposta de peso máximo:
Ative apenas tarefas insatisfeitas de peso máximo.
Uma tarefa ativa muda para melhor máquina.

Teorema: A política de melhor resposta de peso máximo atinge um equilíbrio depois de no máximo n passos.

Prova: Tarefa que migra nunca mais fica insatisfeita.

- A carga mínima não diminui.
- Tarefas ativas têm peso não crescente.

Tempo de convergência

Máquinas idênticas

Política da melhor resposta de peso máximo:
Ative apenas tarefas insatisfeitas de peso máximo.
Uma tarefa ativa muda para melhor máquina.

Teorema: A política de melhor resposta de peso máximo atinge um equilíbrio depois de no máximo n passos.

Prova: Tarefa que migra nunca mais fica insatisfeita.

- A carga mínima não diminui.
- Tarefas ativas têm peso não crescente.

Atinge equilíbrio no máximo em n passos. ∎

Caso de máquinas relacionadas

Teorema: Para o jogo de balanceamento de carga em m máquinas relacionadas, PoA$(m) = \Theta\left(\frac{\lg m}{\lg \lg m}\right)$.

Caso de máquinas relacionadas

Teorema: Para o jogo de balanceamento de carga em m máquinas relacionadas, PoA$(m) = \Theta\left(\frac{\lg m}{\lg \lg m}\right)$.

Primeira parte:
Mostrar que para jogos $J = (n, m, w, s)$ e atribuições $A : [n] \to [m]$ em equilíbrio vale que

$$\mathrm{custo}(A) = \mathrm{O}\left(\frac{\lg m}{\lg \lg m}\right)\mathrm{OPT}(J).$$

Caso de máquinas relacionadas

Teorema: Para o jogo de balanceamento de carga em m máquinas relacionadas, PoA$(m) = \Theta\left(\frac{\lg m}{\lg \lg m}\right)$.

Primeira parte:
Mostrar que para jogos $J = (n, m, w, s)$ e atribuições $A : [n] \to [m]$ em equilíbrio vale que

$$\text{custo}(A) = \mathrm{O}\left(\frac{\lg m}{\lg \lg m}\right)\mathrm{OPT}(J).$$

Segunda parte:
Apresentar jogo $J = (n, m, w, s)$ e atribuição $A : [n] \to [m]$ em equilíbrio tal que

$$\text{custo}(A) = \Omega\left(\frac{\lg m}{\lg \lg m}\right)\mathrm{OPT}(J).$$

Primeira parte

Lema: Para jogos $J = (n, m, w, s)$ e atribuições $A : [n] \to [m]$ em equilíbrio, vale que

$$\text{custo}(A) = \mathrm{O}\Big(\frac{\lg m}{\lg \lg m}\Big)\mathrm{OPT}(J).$$

Primeira parte

Lema: Para jogos $J = (n, m, w, s)$ e atribuições $A : [n] \to [m]$ em equilíbrio, vale que

$$\text{custo}(A) = \mathrm{O}\Big(\frac{\lg m}{\lg \lg m}\Big)\mathrm{OPT}(J).$$

Prova: Seja $c = \lfloor \text{custo}(A)/\mathrm{OPT}(J) \rfloor$.

Primeira parte

Lema: Para jogos $J = (n, m, w, s)$ e atribuições $A : [n] \to [m]$ em equilíbrio, vale que

$$\mathrm{custo}(A) = \mathrm{O}\Big(\frac{\lg m}{\lg \lg m}\Big)\mathrm{OPT}(J).$$

Prova: Seja $c = \lfloor \mathrm{custo}(A)/\mathrm{OPT}(J) \rfloor$.

Vamos mostrar que $c \leq \Gamma^{-1}(m)$,
onde Γ^{-1} denota a inversa da função gama,
que é tal que $\Gamma(k) = (k-1)!$.

Primeira parte

Lema: Para jogos $J = (n, m, w, s)$ e atribuições $A : [n] \to [m]$ em equilíbrio, vale que

$$\text{custo}(A) = \mathrm{O}\Big(\frac{\lg m}{\lg \lg m}\Big)\mathrm{OPT}(J).$$

Prova: Seja $c = \lfloor \text{custo}(A)/\mathrm{OPT}(J) \rfloor$.

Vamos mostrar que $c \leq \Gamma^{-1}(m)$, onde Γ^{-1} denota a inversa da função gama, que é tal que $\Gamma(k) = (k-1)!$.

Sabe-se que $\Gamma^{-1}(m) = \Theta\big(\frac{\lg m}{\lg \lg m}\big)$, logo isso é suficiente para completar a prova.

Primeira parte

Lema: Para jogos $J = (n, m, w, s)$ e atribuições $A : [n] \to [m]$ em equilíbrio, vale que

$$\mathrm{custo}(A) = \mathrm{O}\Big(\frac{\lg m}{\lg \lg m}\Big).$$

Prova: Seja $c = \lfloor \mathrm{custo}(A)/\mathrm{OPT}(J) \rfloor$.

Vamos mostrar que $c \leq \Gamma^{-1}(m) = \Theta\big(\frac{\lg m}{\lg \lg m}\big)$, onde Γ é a função gama e $\Gamma(k) = (k-1)!$.

Primeira parte

Lema: Para jogos $J = (n, m, w, s)$ e atribuições $A : [n] \to [m]$ em equilíbrio, vale que

$$\text{custo}(A) = \mathrm{O}\Big(\frac{\lg m}{\lg \lg m}\Big).$$

Prova: Seja $c = \lfloor \text{custo}(A) / \text{OPT}(J) \rfloor$.

Vamos mostrar que $c \leq \Gamma^{-1}(m) = \Theta\big(\frac{\lg m}{\lg \lg m}\big)$, onde Γ é a função gama e $\Gamma(k) = (k-1)!$.

SPG, $s_1 \geq s_2 \geq \cdots \geq s_m$, e seja $L = [1, 2, \ldots, m]$ a lista das máquinas em ordem de velocidade.

Primeira parte

Lema: Para jogos $J = (n, m, w, s)$ e atribuições $A : [n] \to [m]$ em equilíbrio, vale que

$$\text{custo}(A) = \mathrm{O}\Big(\frac{\lg m}{\lg \lg m}\Big).$$

Prova: Seja $c = \lfloor \text{custo}(A)/\text{OPT}(J) \rfloor$.

Vamos mostrar que $c \leq \Gamma^{-1}(m) = \Theta\big(\frac{\lg m}{\lg \lg m}\big)$, onde Γ é a função gama e $\Gamma(k) = (k-1)!$.

SPG, $s_1 \geq s_2 \geq \cdots \geq s_m$, e seja $L = [1, 2, \ldots, m]$ a lista das máquinas em ordem de velocidade.

Para $k \in \{0, \ldots, c-1\}$,
L_k: maior prefixo de L de máquinas com carga $\geq k\text{OPT}(J)$.

Primeira parte

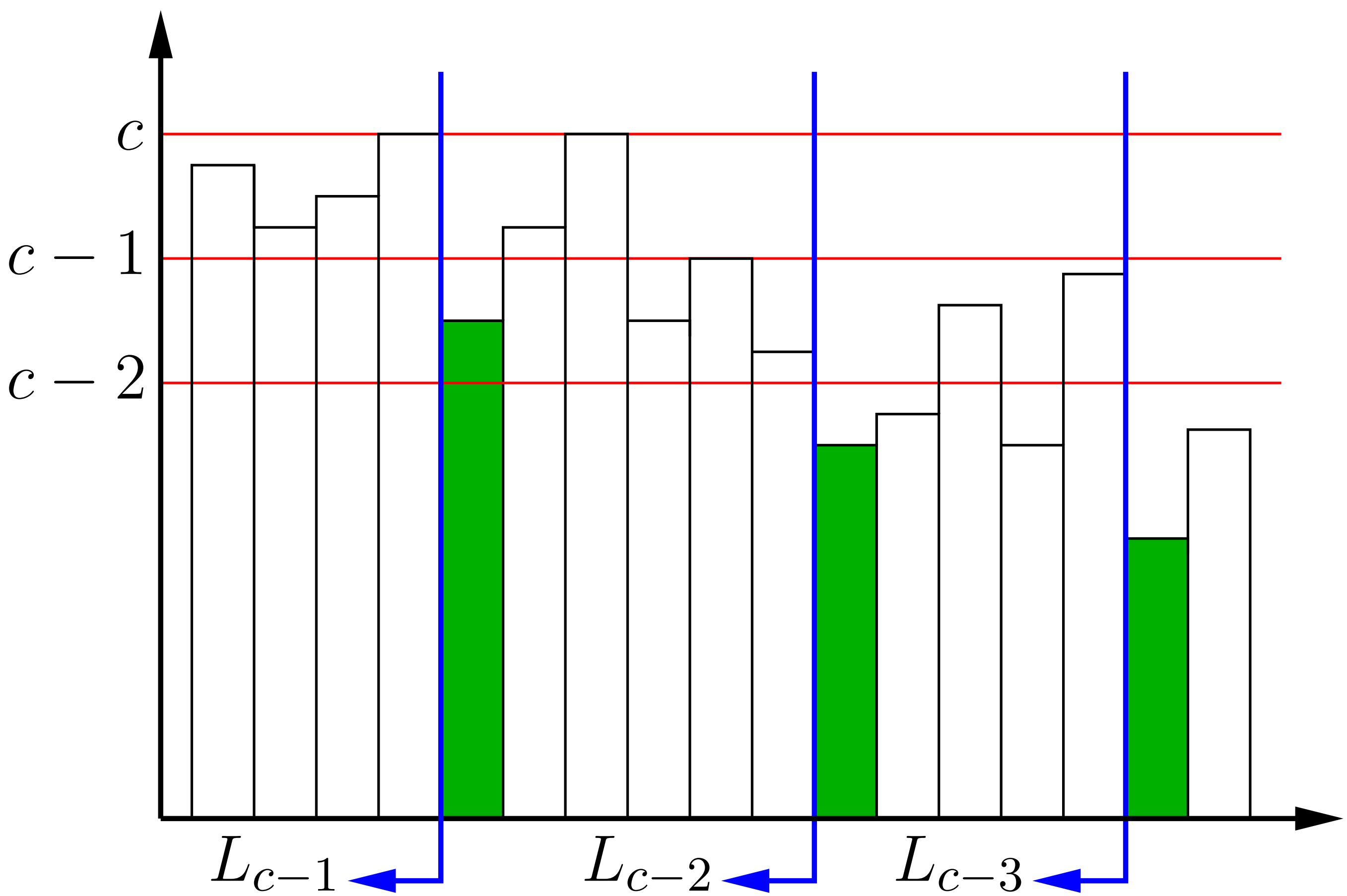
$$\Gamma(k) = (k-1)! \qquad s_1 \geq s_2 \geq \cdots \geq s_m \qquad L = [1, 2, \ldots, m]$$

Para $k \in \{0, \ldots, c-1\}$,
L_k: maior prefixo de L de máquinas com carga $\geq k\mathrm{OPT}(J)$.

Primeira parte

$\Gamma(k) = (k-1)!$ $\qquad s_1 \geq s_2 \geq \cdots \geq s_m$ $\qquad L = [1, 2, \ldots, m]$

Para $k \in \{0, \ldots, c-1\}$,
L_k: maior prefixo de L de máquinas com carga $\geq k\mathrm{OPT}(J)$.



Primeira parte

$\Gamma(k) = (k-1)!$ $\qquad s_1 \geq s_2 \geq \cdots \geq s_m$ $\qquad L = [1, 2, \ldots, m]$

Para $k \in \{0, \ldots, c-1\}$,
L_k: maior prefixo de L de máquinas com carga $\geq k\mathrm{OPT}(J)$.

Primeira parte

$\Gamma(k) = (k-1)!$ $\qquad s_1 \geq s_2 \geq \cdots \geq s_m$ $\qquad L = [1, 2, \ldots, m]$

Para $k \in \{0, \ldots, c-1\}$,
L_k: maior prefixo de L de máquinas com carga $\geq k\mathrm{OPT}(J)$.

Vamos provar que $|L_k| \geq (k+1)|L_{k+1}|$ e $|L_{c-1}| \geq 1$.

Primeira parte

$\Gamma(k) = (k-1)!$ $\qquad s_1 \geq s_2 \geq \cdots \geq s_m$ $\qquad L = [1, 2, \ldots, m]$

Para $k \in \{0, \ldots, c-1\}$,
L_k: maior prefixo de L de máquinas com carga $\geq k\mathrm{OPT}(J)$.

Vamos provar que $|L_k| \geq (k+1)|L_{k+1}|$ e $|L_{c-1}| \geq 1$.

Disso segue que $|L_0| \geq (c-1)! = \Gamma(c)$.

Primeira parte

$\Gamma(k) = (k-1)!$ $\qquad s_1 \geq s_2 \geq \cdots \geq s_m$ $\qquad L = [1, 2, \ldots, m]$

Para $k \in \{0, \ldots, c-1\}$,
L_k: maior prefixo de L de máquinas com carga $\geq k\mathrm{OPT}(J)$.

Vamos provar que $|L_k| \geq (k+1)|L_{k+1}|$ e $|L_{c-1}| \geq 1$.

Disso segue que $|L_0| \geq (c-1)! = \Gamma(c)$.

Como $L_0 = L$, temos que $m \geq \Gamma(c)$ e portanto $c \leq \Gamma^{-1}(m)$.

Primeira parte

$\Gamma(k) = (k-1)!$ $\qquad s_1 \geq s_2 \geq \cdots \geq s_m$ $\qquad L = [1, 2, \ldots, m]$

Para $k \in \{0, \ldots, c-1\}$,
L_k: maior prefixo de L de máquinas com carga $\geq k\mathrm{OPT}(J)$.

Vamos provar que $|L_k| \geq (k+1)|L_{k+1}|$ e $|L_{c-1}| \geq 1$.

Disso segue que $|L_0| \geq (c-1)! = \Gamma(c)$.

Como $L_0 = L$, temos que $m \geq \Gamma(c)$ e portanto $c \leq \Gamma^{-1}(m)$.

Resta provar a recorrência.

Prova da recorrência

Vamos provar que $|L_k| \geq (k+1)|L_{k+1}|$ e $|L_{c-1}| \geq 1$.

Prova da recorrência

Vamos provar que $|L_k| \geq (k+1)|L_{k+1}|$ e $|L_{c-1}| \geq 1$.

Primeiro, mostremos que $|L_{c-1}| \geq 1$.

Suponha que não, ou seja, que $L_{c-1} = \emptyset$.

Prova da recorrência

Vamos provar que $|L_k| \geq (k+1)|L_{k+1}|$ e $|L_{c-1}| \geq 1$.

Primeiro, mostremos que $|L_{c-1}| \geq 1$.

Suponha que não, ou seja, que $L_{c-1} = \emptyset$.

Então a carga $\ell_1 < (c-1)\mathrm{OPT}(J)$.

Prova da recorrência

Vamos provar que $|L_k| \geq (k+1)|L_{k+1}|$ e $|L_{c-1}| \geq 1$.

Primeiro, mostremos que $|L_{c-1}| \geq 1$.

Suponha que não, ou seja, que $L_{c-1} = \emptyset$.

Então a carga $\ell_1 < (c-1)\mathrm{OPT}(J)$.

Seja i tarefa atribuída a uma máquina com carga $c\,\mathrm{OPT}(J)$.

Prova da recorrência

Vamos provar que $|L_k| \geq (k+1)|L_{k+1}|$ e $|L_{c-1}| \geq 1$.

Primeiro, mostremos que $|L_{c-1}| \geq 1$.

Suponha que não, ou seja, que $L_{c-1} = \emptyset$.

Então a carga $\ell_1 < (c-1)\mathrm{OPT}(J)$.

Seja i tarefa atribuída a uma máquina com carga $c\,\mathrm{OPT}(J)$.

$$\ell_1 + \frac{w_i}{s_1} \;<\; (c-1)\,\mathrm{OPT}(J) + \mathrm{OPT}(J) \;=\; c\,\mathrm{OPT}(J).$$

Prova da recorrência

Vamos provar que $|L_k| \geq (k+1)|L_{k+1}|$ e $|L_{c-1}| \geq 1$.

Primeiro, mostremos que $|L_{c-1}| \geq 1$.

Suponha que não, ou seja, que $L_{c-1} = \emptyset$.

Então a carga $\ell_1 < (c-1)\mathrm{OPT}(J)$.

Seja i tarefa atribuída a uma máquina com carga $c\,\mathrm{OPT}(J)$.

$$\ell_1 + \frac{w_i}{s_1} \;<\; (c-1)\,\mathrm{OPT}(J) + \mathrm{OPT}(J) \;=\; c\,\mathrm{OPT}(J).$$

Ou seja, i tem incentivo para mudar para a máquina 1.

Prova da recorrência

Vamos provar que $|L_k| \geq (k+1)|L_{k+1}|$ e $|L_{c-1}| \geq 1$.

Primeiro, mostremos que $|L_{c-1}| \geq 1$.

Suponha que não, ou seja, que $L_{c-1} = \emptyset$.

Então a carga $\ell_1 < (c-1)\mathrm{OPT}(J)$.

Seja i tarefa atribuída a uma máquina com carga $c\,\mathrm{OPT}(J)$.

$$\ell_1 + \frac{w_i}{s_1} \;<\; (c-1)\,\mathrm{OPT}(J) + \mathrm{OPT}(J) \;=\; c\,\mathrm{OPT}(J).$$

Ou seja, i tem incentivo para mudar para a máquina 1.

Contradição pois A é um equilíbrio.

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Lema: $A^*(i) \in L_k$ para toda tarefa i tq $A(i) \in L_{k+1}$.

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Lema: $A^*(i) \in L_k$ para toda tarefa i tq $A(i) \in L_{k+1}$.

Prova: Se $L_k = L$, nada a provar.

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Lema: $A^*(i) \in L_k$ para toda tarefa i tq $A(i) \in L_{k+1}$.

Prova: Se $L_k = L$, nada a provar.

Seja q máquina de menor índice em $L \setminus L_k$.

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Lema: $A^*(i) \in L_k$ para toda tarefa i tq $A(i) \in L_{k+1}$.

Prova: Se $L_k = L$, nada a provar.

Seja q máquina de menor índice em $L \setminus L_k$.

Temos que $\ell_q < k\,\mathrm{OPT}(J)$.

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Lema: $A^*(i) \in L_k$ para toda tarefa i tq $A(i) \in L_{k+1}$.

Prova: Se $L_k = L$, nada a provar.

Seja q máquina de menor índice em $L \setminus L_k$.

Temos que $\ell_q < k\,\mathrm{OPT}(J)$.

$\ell_{A(i)} \geq (k+1)\,\mathrm{OPT}(J)$ para toda tarefa i tq $A(i) \in L_{k+1}$.

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Lema: $A^*(i) \in L_k$ para toda tarefa i tq $A(i) \in L_{k+1}$.

Prova: Se $L_k = L$, nada a provar.

Seja q máquina de menor índice em $L \setminus L_k$.

Temos que $\ell_q < k\,\mathrm{OPT}(J)$.

$\ell_{A(i)} \geq (k+1)\,\mathrm{OPT}(J)$ para toda tarefa i tq $A(i) \in L_{k+1}$.

Se $w_i \leq s_q\,\mathrm{OPT}(J)$, então i tem incentivo para ir para q:

$$\ell_q + \frac{w_i}{s_q} < k\,\mathrm{OPT}(J) + \mathrm{OPT}(J) = (k+1)\mathrm{OPT}(J) \leq \ell_{A(i)},$$

contradição.

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Lema: $A^*(i) \in L_k$ para toda tarefa i tq $A(i) \in L_{k+1}$.

Prova: Seja q máquina de menor índice em $L \setminus L_k$.

Temos que $\ell_q < k\,\mathrm{OPT}(J)$.

Para toda tarefa i tq $A(i) \in L_{k+1}$,

$$\ell_{A(i)} \geq (k+1)\,\mathrm{OPT}(J) \text{ e } w_i > s_q\,\mathrm{OPT}(J).$$

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Lema: $A^*(i) \in L_k$ para toda tarefa i tq $A(i) \in L_{k+1}$.

Prova: Seja q máquina de menor índice em $L \setminus L_k$.

Temos que $\ell_q < k \, \mathrm{OPT}(J)$.

Para toda tarefa i tq $A(i) \in L_{k+1}$,

$$\ell_{A(i)} \geq (k+1) \, \mathrm{OPT}(J) \text{ e } w_i > s_q \, \mathrm{OPT}(J).$$

Por contradição, suponha que $j = A^*(i) \notin L_k$.

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Lema: $A^*(i) \in L_k$ para toda tarefa i tq $A(i) \in L_{k+1}$.

Prova: Seja q máquina de menor índice em $L \setminus L_k$.

Temos que $\ell_q < k\,\mathrm{OPT}(J)$.

Para toda tarefa i tq $A(i) \in L_{k+1}$,

$$\ell_{A(i)} \geq (k+1)\,\mathrm{OPT}(J) \text{ e } w_i > s_q\,\mathrm{OPT}(J).$$

Por contradição, suponha que $j = A^*(i) \notin L_k$.

Então, como $s_j \leq s_q$, a carga de j em A^* seria

$$\geq \frac{w_i}{s_j} > \frac{s_q\,\mathrm{OPT}(J)}{s_j} \geq \mathrm{OPT}(J), \quad \text{contradição.} \blacksquare$$

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Lema: $A^*(i) \in L_k$ para toda tarefa i tq $A(i) \in L_{k+1}$.

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Lema: $A^*(i) \in L_k$ para toda tarefa i tq $A(i) \in L_{k+1}$.

Pela definição de L_{k+1}, para cada j em L_{k+1},

$$\sum_{i:A(i)=j} w_i \;\geq\; (k+1)\,\mathrm{OPT}(J)\,s_j,$$

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Lema: $A^*(i) \in L_k$ para toda tarefa i tq $A(i) \in L_{k+1}$.

Pela definição de L_{k+1}, para cada j em L_{k+1},

$$\sum_{i:A(i)=j} w_i \;\geq\; (k+1)\,\mathrm{OPT}(J)\,s_j\,,$$

e o total de peso atribuído por A a máquinas em L_{k+1} é

$$\sum_{j \in L_{k+1}} (k+1)\,\mathrm{OPT}(J)\,s_j\,.$$

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Lema: $A^*(i) \in L_k$ para toda tarefa i tq $A(i) \in L_{k+1}$.

Pela definição de L_{k+1}, para cada j em L_{k+1},

$$\sum_{i:A(i)=j} w_i \;\geq\; (k+1)\,\mathrm{OPT}(J)\,s_j,$$

e o total de peso atribuído por A a máquinas em L_{k+1} é

$$\sum_{j\in L_{k+1}} (k+1)\,\mathrm{OPT}(J)\,s_j.$$

Pelo Lema, A^* atribui tudo isso a L_k sem estourar $\mathrm{OPT}(J)$.

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Lema: $A^*(i) \in L_k$ para toda tarefa i tq $A(i) \in L_{k+1}$.

O total de peso atribuído por A a máquinas em L_{k+1} é

$$\sum_{j \in L_{k+1}} (k+1)\,\mathrm{OPT}(J)\,s_j\,.$$

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Lema: $A^*(i) \in L_k$ para toda tarefa i tq $A(i) \in L_{k+1}$.

O total de peso atribuído por A a máquinas em L_{k+1} é

$$\sum_{j \in L_{k+1}} (k+1)\,\mathrm{OPT}(J)\,s_j\,.$$

Pelo Lema, A^* atribui tudo isso a L_k sem estourar $\mathrm{OPT}(J)$, logo

$$\sum_{j \in L_k} \mathrm{OPT}(J)\,s_j \;\geq\; \sum_{j \in L_{k+1}} (k+1)\,\mathrm{OPT}(J)\,s_j\,,$$

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Lema: $A^*(i) \in L_k$ para toda tarefa i tq $A(i) \in L_{k+1}$.

O total de peso atribuído por A a máquinas em L_{k+1} é

$$\sum_{j \in L_{k+1}} (k+1)\,\mathrm{OPT}(J)\,s_j\,.$$

Pelo Lema, A^* atribui tudo isso a L_k sem estourar $\mathrm{OPT}(J)$, logo

$$\sum_{j \in L_k} \mathrm{OPT}(J)\,s_j \;\geq\; \sum_{j \in L_{k+1}} (k+1)\,\mathrm{OPT}(J)\,s_j\,,$$

que implica que $\sum_{j \in L_k} s_j \;\geq\; \sum_{j \in L_{k+1}} (k+1)s_j\,.$

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Lema: $A^*(i) \in L_k$ para toda tarefa i tq $A(i) \in L_{k+1}$.

Peso atribuído por A a máquinas em L_{k+1} é $\sum_{j \in L_{k+1}} (k+1)\,\mathrm{OPT}(J)\,s_j$.

Como A^* atribui tudo isso a L_k sem estourar $\mathrm{OPT}(J)$, temos que $\sum_{j \in L_k} s_j \;\geq\; \sum_{j \in L_{k+1}} (k+1) s_j$,
ou seja, $\sum_{j \in L_k \setminus L_{k+1}} s_j \;\geq\; \sum_{j \in L_{k+1}} k\, s_j$.

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Lema: $A^*(i) \in L_k$ para toda tarefa i tq $A(i) \in L_{k+1}$.

Peso atribuído por A a máquinas em L_{k+1} é $\sum_{j \in L_{k+1}} (k+1)\,\mathrm{OPT}(J)\,s_j$.

Como A^* atribui tudo isso a L_k sem estourar $\mathrm{OPT}(J)$, temos que $\sum_{j \in L_k} s_j \;\geq\; \sum_{j \in L_{k+1}} (k+1) s_j$, ou seja, $\sum_{j \in L_k \setminus L_{k+1}} s_j \;\geq\; \sum_{j \in L_{k+1}} k\, s_j$.

Seja s^* a velocidade da máquina mais lenta em L_{k+1}, ou seja, $s^* = s_{|L_{k+1}|}$.

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Lema: $A^*(i) \in L_k$ para toda tarefa i tq $A(i) \in L_{k+1}$.

Peso atribuído por A a máquinas em L_{k+1} é $\sum_{j \in L_{k+1}} (k+1)\, \mathrm{OPT}(J)\, s_j$.

Como A^* atribui tudo isso a L_k sem estourar $\mathrm{OPT}(J)$,
temos que $\sum_{j \in L_k} s_j \;\geq\; \sum_{j \in L_{k+1}} (k+1) s_j$,
ou seja, $\sum_{j \in L_k \setminus L_{k+1}} s_j \;\geq\; \sum_{j \in L_{k+1}} k\, s_j$.

Seja s^* a velocidade da máquina mais lenta em L_{k+1},
ou seja, $s^* = s_{|L_{k+1}|}$.

Então $\sum_{j \in L_k} s^* \;\geq\; \sum_{j \in L_{k+1}} (k+1) s^*$,

Prova de que $|L_k| \geq (k+1)|L_{k+1}|$

Seja A^* uma atribuição ótima (de makespan mínimo).

Lema: $A^*(i) \in L_k$ para toda tarefa i tq $A(i) \in L_{k+1}$.

Peso atribuído por A a máquinas em L_{k+1} é $\sum_{j \in L_{k+1}} (k+1)\,\mathrm{OPT}(J)\,s_j$.

Como A^* atribui tudo isso a L_k sem estourar $\mathrm{OPT}(J)$, temos que $\sum_{j \in L_k} s_j \ \geq\ \sum_{j \in L_{k+1}} (k+1) s_j$,
ou seja, $\sum_{j \in L_k \setminus L_{k+1}} s_j \ \geq\ \sum_{j \in L_{k+1}} k\, s_j$.

Seja s^* a velocidade da máquina mais lenta em L_{k+1}, ou seja, $s^* = s_{|L_{k+1}|}$.

Então $\sum_{j \in L_k} s^* \ \geq\ \sum_{j \in L_{k+1}} (k+1) s^*$,

donde concluímos que $|L_k| \leq (k+1)|L_{k+1}|$. ■

Caso de máquinas relacionadas

Teorema: Para o jogo de balanceamento de carga em m máquinas relacionadas, PoA$(m) = \Theta\left(\frac{\lg m}{\lg \lg m}\right)$.

Primeira parte:
Mostrar que para jogos $J = (n, m, w, s)$ e atribuições $A : [n] \to [m]$ em equilíbrio vale que

$$\text{custo}(A) = \mathrm{O}\left(\frac{\lg m}{\lg \lg m}\right).$$

Caso de máquinas relacionadas

Teorema: Para o jogo de balanceamento de carga em m máquinas relacionadas, PoA$(m) = \Theta(\frac{\lg m}{\lg \lg m})$.

Primeira parte:
Mostrar que para jogos $J = (n, m, w, s)$ e atribuições $A : [n] \to [m]$ em equilíbrio vale que

$$\text{custo}(A) = \text{O}(\frac{\lg m}{\lg \lg m}).$$

Segunda parte: (Aula que vem!!!!)
Apresentar jogo $J = (n, m, w, s)$ e atribuição $A : [n] \to [m]$ em equilíbrio tal que

$$\text{custo}(A) = \Omega(\frac{\lg m}{\lg \lg m}).$$