Algoritmos de Aproximação

Segundo Semestre de 2012

Escalonamento de uma máquina

Dados: n tarefas $([n] = \{1, \ldots, n\})$
duração $d[i]$ da tarefa i $(i = 1, \ldots, n)$
momento de chegada $r[i]$ da tarefa i $(i = 1, \ldots, n)$

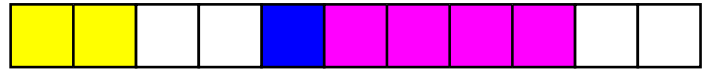
um escalonamento é uma ordenação de $[n]$.

Escalonamento de uma máquina

Dados: n tarefas $([n] = \{1, \ldots, n\})$
duração $d[i]$ da tarefa i $(i = 1, \ldots, n)$
momento de chegada $r[i]$ da tarefa i $(i = 1, \ldots, n)$

um escalonamento é uma ordenação de $[n]$.

Exemplo: $d_1 = 2$, $r_1 = 0$, $d_2 = 1$, $r_2 = 4$, $d_3 = 4$, $r_3 = 1$.

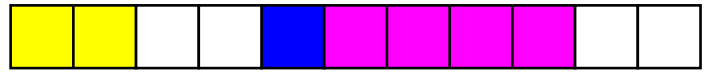


Escalonamento de uma máquina

Dados: n tarefas $([n] = \{1, \ldots, n\})$
duração $d[i]$ da tarefa i $(i = 1, \ldots, n)$
momento de chegada $r[i]$ da tarefa i $(i = 1, \ldots, n)$

um escalonamento é uma ordenação de $[n]$.

Exemplo: $d_1 = 2$, $r_1 = 0$, $d_2 = 1$, $r_2 = 4$, $d_3 = 4$, $r_3 = 1$.



$C_1 = 2$, $C_2 = 5$, $C_3 = 9$.

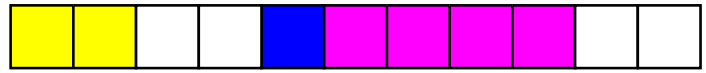
Encontrar escalonamento com
soma dos tempos de conclusão mínima.

Escalonamento de uma máquina

Dados: n tarefas $([n] = \{1, \ldots, n\})$
duração $d[i]$ da tarefa i $(i = 1, \ldots, n)$
momento de chegada $r[i]$ da tarefa i $(i = 1, \ldots, n)$

um escalonamento é uma ordenação de $[n]$.

Exemplo: $d_1 = 2$, $r_1 = 0$, $d_2 = 1$, $r_2 = 4$, $d_3 = 4$, $r_3 = 1$.



$C_1 = 2$, $C_2 = 5$, $C_3 = 9$.

Encontrar escalonamento com
soma dos tempos de conclusão mínima.

Mesmo que minimizar a média dos tempos de conclusão.

Escalonamento com interrupções

Dados: n tarefas $([n] = \{1, \ldots, n\})$
duração $d[i]$ da tarefa i $(i = 1, \ldots, n)$
momento de chegada $r[i]$ da tarefa i $(i = 1, \ldots, n)$

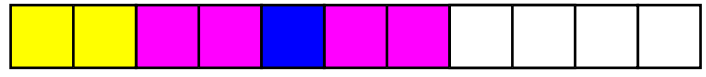
Escalonamento com (possíveis) interrupções
(with preemption).

Escalonamento com interrupções

Dados: n tarefas $([n] = \{1, \ldots, n\})$
duração $d[i]$ da tarefa i $(i = 1, \ldots, n)$
momento de chegada $r[i]$ da tarefa i $(i = 1, \ldots, n)$

Escalonamento com (possíveis) interrupções
(with preemption).

Exemplo: $d_1 = 2$, $r_1 = 0$, $d_2 = 1$, $r_2 = 4$, $d_3 = 4$, $r_3 = 1$.

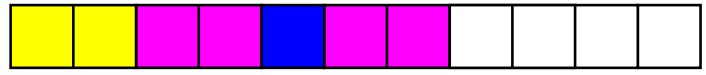


Escalonamento com interrupções

Dados: n tarefas $([n] = \{1, \ldots, n\})$
duração $d[i]$ da tarefa i $(i = 1, \ldots, n)$
momento de chegada $r[i]$ da tarefa i $(i = 1, \ldots, n)$

Escalonamento com (possíveis) interrupções
(with preemption).

Exemplo: $d_1 = 2$, $r_1 = 0$, $d_2 = 1$, $r_2 = 4$, $d_3 = 4$, $r_3 = 1$.



$C_1^P = 2$, $C_2^P = 5$, $C_3^P = 7$.

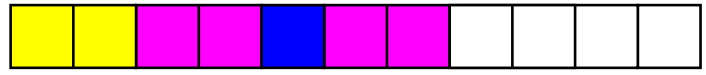
Encontrar escalonamento com interrupções com
soma dos tempos de conclusão mínimo.

Escalonamento com interrupções

Dados: n tarefas $([n] = \{1, \ldots, n\})$
duração $d[i]$ da tarefa i $(i = 1, \ldots, n)$
momento de chegada $r[i]$ da tarefa i $(i = 1, \ldots, n)$

Escalonamento com (possíveis) interrupções
(with preemption).

Exemplo: $d_1 = 2$, $r_1 = 0$, $d_2 = 1$, $r_2 = 4$, $d_3 = 4$, $r_3 = 1$.



$C_1^P = 2$, $C_2^P = 5$, $C_3^P = 7$.

Encontrar escalonamento com interrupções com soma dos tempos de conclusão mínimo.

SRPT: shortest remaining processing time

Escalonamento com interrupções

OPT: ótimo do problema original

Escalonamento com interrupções

OPT: ótimo do problema original

Lema: Para os tempos de conclusão $C_1^P, \ldots, C_n^P$ de um escalonamento ótimo com interrupções, vale que

$$\sum_{j=1}^{n} C_j^P \leq \mathrm{OPT}.$$

Escalonamento com interrupções

OPT: ótimo do problema original

Lema: Para os tempos de conclusão $C_1^P, \ldots, C_n^P$ de um escalonamento ótimo com interrupções, vale que

$$\sum_{j=1}^{n} C_j^P \leq \mathrm{OPT}.$$

Prova: Todo escalonamento sem interrupção é um escalonamento também com interrupção.

Algoritmo

Obtenha os tempos de conclusão $C_1^P, \ldots, C_n^P$ de um escalonamento ótimo com interrupções, com a regra SRPT.

Reordene as tarefas de modo que $C_1^P \leq \cdots \leq C_n^P$.

Algoritmo

Obtenha os tempos de conclusão $C_1^P, \ldots, C_n^P$ de um escalonamento ótimo com interrupções, com a regra SRPT.

Reordene as tarefas de modo que $C_1^P \leq \cdots \leq C_n^P$.

Obtenha um escalonamento sem interrupções, escalonando as tarefas na ordem $1, \ldots, n$.

Algoritmo

Obtenha os tempos de conclusão $C_1^P, \ldots, C_n^P$ de um escalonamento ótimo com interrupções, com a regra SRPT.

Reordene as tarefas de modo que $C_1^P \leq \cdots \leq C_n^P$.

Obtenha um escalonamento sem interrupções, escalonando as tarefas na ordem $1, \ldots, n$.

Chame de $C_1^N, \ldots, C_n^N$ os tempos de conclusão resultantes.

Algoritmo

Obtenha os tempos de conclusão $C_1^P, \ldots, C_n^P$ de um escalonamento ótimo com interrupções, com a regra SRPT.

Reordene as tarefas de modo que $C_1^P \leq \cdots \leq C_n^P$.

Obtenha um escalonamento sem interrupções, escalonando as tarefas na ordem $1, \ldots, n$.

Chame de $C_1^N, \ldots, C_n^N$ os tempos de conclusão resultantes.

Lema: Para $j = 1, \ldots, n$, temos que $C_j^N \leq 2\, C_j^P$.

Prova vista em aula.

Prova do lema

Lema: Para $j = 1, \ldots, n$, temos que $C_j^N \leq 2\, C_j^P$.

Prova: Note que

$$C_j^P \geq \max_{k=1,\ldots,j} r_k \quad \text{e} \quad C_j^P \geq \sum_{k=1}^{j} d_k.$$

Prova do lema

Lema: Para $j = 1, \ldots, n$, temos que $C_j^N \leq 2\, C_j^P$.

Prova: Note que

$$C_j^P \geq \max_{k=1,\ldots,j} r_k \quad \text{e} \quad C_j^P \geq \sum_{k=1}^{j} d_k.$$

Claro que $C_j^N \geq \max_{k=1,\ldots,j} r_k$.

Prova do lema

Lema: Para $j = 1, \ldots, n$, temos que $C_j^N \leq 2\, C_j^P$.

Prova: Note que

$$C_j^P \geq \max_{k=1,\ldots,j} r_k \quad \text{e} \quad C_j^P \geq \sum_{k=1}^{j} d_k.$$

Claro que $C_j^N \geq \max_{k=1,\ldots,j} r_k$.

Observe que não pode haver tempo ocioso entre $\max_{k=1,\ldots,j} r_k$ e C_j^N. Logo,

$$C_j^N \leq \max_{k=1,\ldots,j} r_k + \sum_{k=1}^{j} d_k \leq 2\, C_j^P. \blacksquare$$

Conclusão

Algoritmo:

Obtenha os tempos de conclusão $C_1^P, \ldots, C_n^P$ de um escalonamento ótimo com interrupções, com a regra SRPT.

Reordene as tarefas de modo que $C_1^P \leq \cdots \leq C_n^P$.

Obtenha um escalonamento sem interrupções, escalonando as tarefas na ordem $1, \ldots, n$.

Chame de $C_1^N, \ldots, C_n^N$ os tempos de conclusão resultantes.

Conclusão

Algoritmo:

Obtenha os tempos de conclusão $C_1^P, \ldots, C_n^P$ de um escalonamento ótimo com interrupções, com a regra SRPT.

Reordene as tarefas de modo que $C_1^P \leq \cdots \leq C_n^P$.

Obtenha um escalonamento sem interrupções, escalonando as tarefas na ordem $1, \ldots, n$.

Chame de $C_1^N, \ldots, C_n^N$ os tempos de conclusão resultantes.

Teorema: O algoritmo acima é uma 2-aproximação.

Conclusão

Algoritmo:

Obtenha os tempos de conclusão $C_1^P, \ldots, C_n^P$ de um escalonamento ótimo com interrupções, com a regra SRPT.

Reordene as tarefas de modo que $C_1^P \leq \cdots \leq C_n^P$.

Obtenha um escalonamento sem interrupções, escalonando as tarefas na ordem $1, \ldots, n$.

Chame de $C_1^N, \ldots, C_n^N$ os tempos de conclusão resultantes.

Teorema: O algoritmo acima é uma 2-aproximação.

Prova:

$$\sum_{j=1}^{n} C_j^N \leq 2 \sum_{j=1}^{n} C_j^P \leq 2\,\mathrm{OPT}.$$

Versão com pesos

Dados: n tarefas $([n] = \{1, \ldots, n\})$
duração $d[i]$ da tarefa i $(i = 1, \ldots, n)$
momento de chegada $r[i]$ da tarefa i $(i = 1, \ldots, n)$
peso $w[i]$ da tarefa i $(i = 1, \ldots, n)$

Versão com pesos

Dados: n tarefas $([n] = \{1, \ldots, n\})$
duração $d[i]$ da tarefa i $(i = 1, \ldots, n)$
momento de chegada $r[i]$ da tarefa i $(i = 1, \ldots, n)$
peso $w[i]$ da tarefa i $(i = 1, \ldots, n)$

Um escalonamento é uma ordenação de $[n]$.

Encontrar escalonamento com
soma ponderada dos tempos de conclusão mínima.

Versão com pesos

Dados: n tarefas $([n] = \{1, \ldots, n\})$
duração $d[i]$ da tarefa i $(i = 1, \ldots, n)$
momento de chegada $r[i]$ da tarefa i $(i = 1, \ldots, n)$
peso $w[i]$ da tarefa i $(i = 1, \ldots, n)$

Um escalonamento é uma ordenação de $[n]$.

Encontrar escalonamento com
soma ponderada dos tempos de conclusão mínima.

Versão com interrupções com pesos é NP-difícil...

Versão com pesos

Dados: n tarefas $([n] = \{1, \ldots, n\})$
duração $d[i]$ da tarefa i $(i = 1, \ldots, n)$
momento de chegada $r[i]$ da tarefa i $(i = 1, \ldots, n)$
peso $w[i]$ da tarefa i $(i = 1, \ldots, n)$

Um escalonamento é uma ordenação de $[n]$.

Encontrar escalonamento com
soma ponderada dos tempos de conclusão mínima.

Versão com interrupções com pesos é NP-difícil...

Vamos usar um PL para obter um bom escalonamento.

Relaxação linear

Encontrar $C_1, \ldots, C_n$ que minimizem $\sum_{j=1}^{n} w_j C_j$, sujeitos a

Relaxação linear

Encontrar $C_1, \ldots, C_n$ que minimizem $\sum_{j=1}^{n} w_j C_j$, sujeitos a

$$C_j \geq r_j + d_j \qquad \text{para todo } j$$

Relaxação linear

Encontrar $C_1, \ldots, C_n$ que minimizem $\sum_{j=1}^{n} w_j C_j$, sujeitos a

$$C_j \geq r_j + d_j \qquad \text{para todo } j$$

e... vamos deduzir uma segunda restrição, menos óbvia.

Seja $S \subseteq [n]$ e considere a soma $\sum_{j \in S} d_j C_j$.

Relaxação linear

Encontrar $C_1, \ldots, C_n$ que minimizem $\sum_{j=1}^{n} w_j C_j$, sujeitos a

$$C_j \geq r_j + d_j \qquad \text{para todo } j$$

e... vamos deduzir uma segunda restrição, menos óbvia.

Seja $S \subseteq [n]$ e considere a soma $\sum_{j \in S} d_j C_j$.

Essa soma é o menor possível se $r_j = 0$ para todo $j \in S$ e as tarefas de S vem antes de todas as demais.

Relaxação linear

Encontrar $C_1, \ldots, C_n$ que minimizem $\sum_{j=1}^{n} w_j\, C_j$, sujeitos a

$$C_j \geq r_j + d_j \qquad \text{para todo } j$$

e... vamos deduzir uma segunda restrição, menos óbvia.

Seja $S \subseteq [n]$ e considere a soma $\sum_{j \in S} d_j\, C_j$.

Essa soma é o menor possível se $r_j = 0$ para todo $j \in S$ e as tarefas de S vem antes de todas as demais.

Neste caso,

$$\sum_{j \in S} d_j\, C_j = \sum_{j,k \in S, j \leq k} d_j\, d_k = \frac{1}{2} \left(\sum_{j \in S} d_j\right)^2 + \frac{1}{2} \sum_{j \in S} d_j^2 \geq \frac{1}{2} \left(\sum_{j \in S} d_j\right)^2.$$

Relaxação linear

Seja $d(S) = \sum_{j \in S} d_j$.

O PL fica: encontrar $C_1, \ldots, C_n$ que

minimizem $\sum_{j=1}^{n} w_j C_j$,
sujeitos a

$$C_j \geq r_j + d_j \qquad \text{para todo } j$$

$$\sum_{j \in S} d_j C_j \geq \tfrac{1}{2} d(S)^2 \qquad \text{para todo } S \subseteq [n].$$

Relaxação linear

Seja $d(S) = \sum_{j \in S} d_j$.

O PL fica: encontrar $C_1, \ldots, C_n$ que

minimizem $\sum_{j=1}^{n} w_j C_j$,
sujeitos a

$$C_j \geq r_j + d_j \qquad \text{para todo } j$$
$$\sum_{j \in S} d_j C_j \geq \tfrac{1}{2} d(S)^2 \qquad \text{para todo } S \subseteq [n].$$

Alerta: Esse PL tem um número exponencial de restrições!

Relaxação linear

Seja $d(S) = \sum_{j \in S} d_j$.

O PL fica: encontrar $C_1, \ldots, C_n$ que

minimizem $\sum_{j=1}^{n} w_j C_j$,
sujeitos a

$$C_j \geq r_j + d_j \qquad \text{para todo } j$$
$$\sum_{j \in S} d_j C_j \geq \tfrac{1}{2} d(S)^2 \qquad \text{para todo } S \subseteq [n].$$

Alerta: Esse PL tem um número exponencial de restrições!

Vamos supor que conseguimos resolvê-lo
em tempo polinomial assim mesmo.

Algoritmo

Seja $C_1^*, \ldots, C_n^*$ uma solução ótima do PL.

Reordene as tarefas de modo que $C_1^* \leq \cdots \leq C_n^*$.

Algoritmo

Seja $C_1^*, \ldots, C_n^*$ uma solução ótima do PL.

Reordene as tarefas de modo que $C_1^* \leq \cdots \leq C_n^*$.

Obtenha um escalonamento, escalonando as tarefas na ordem $1, \ldots, n$.

Algoritmo

Seja $C_1^*, \ldots, C_n^*$ uma solução ótima do PL.

Reordene as tarefas de modo que $C_1^* \leq \cdots \leq C_n^*$.

Obtenha um escalonamento, escalonando as tarefas na ordem $1, \ldots, n$.

Chame de $C_1^N, \ldots, C_n^N$ os tempos de conclusão resultantes.

Algoritmo

Seja $C_1^*, \ldots, C_n^*$ uma solução ótima do PL.

Reordene as tarefas de modo que $C_1^* \leq \cdots \leq C_n^*$.

Obtenha um escalonamento, escalonando as tarefas na ordem $1, \ldots, n$.

Chame de $C_1^N, \ldots, C_n^N$ os tempos de conclusão resultantes.

Teorema: O algoritmo acima é uma 3-aproximação.

Prova: Se mostrarmos que $C_j^N \leq 3\, C_j^*$ para todo j, então

$$\sum_{j=1}^{n} w_j\, C_j^N \leq 3 \sum_{j=1}^{n} w_j\, C_j^* \leq 3\, \mathrm{OPT}.$$

Prova do lema

Seja $C_1^*, \ldots, C_n^*$ uma solução ótima do PL.

Reordene as tarefas de modo que $C_1^* \leq \cdots \leq C_n^*$.

Escalone as tarefas na ordem $1, \ldots, n$.

Chame de $C_1^N, \ldots, C_n^N$ os tempos de conclusão resultantes.

Lema: Para $j = 1, \ldots, n$, temos que $C_j^N \leq 3\, C_j^P$.

Prova do lema

Seja $C_1^*, \ldots, C_n^*$ uma solução ótima do PL.

Reordene as tarefas de modo que $C_1^* \leq \cdots \leq C_n^*$.

Escalone as tarefas na ordem $1, \ldots, n$.

Chame de $C_1^N, \ldots, C_n^N$ os tempos de conclusão resultantes.

Lema: Para $j = 1, \ldots, n$, temos que $C_j^N \leq 3\, C_j^P$.

Prova: Como antes, não pode haver tempo ocioso entre $\max_{k=1,\ldots,j} r_k$ e C_j^N. Logo, $C_j^N \leq \max_{k=1,\ldots,j} r_k + \sum_{k=1}^{j} d_k$.

Prova do lema

Seja $C_1^*, \ldots, C_n^*$ uma solução ótima do PL.

Reordene as tarefas de modo que $C_1^* \leq \cdots \leq C_n^*$.

Escalone as tarefas na ordem $1, \ldots, n$.

Chame de $C_1^N, \ldots, C_n^N$ os tempos de conclusão resultantes.

Lema: Para $j = 1, \ldots, n$, temos que $C_j^N \leq 3\, C_j^P$.

Prova: Como antes, não pode haver tempo ocioso entre $\max_{k=1,\ldots,j} r_k$ e C_j^N. Logo, $C_j^N \leq \max_{k=1,\ldots,j} r_k + \sum_{k=1}^{j} d_k$.

Seja $\ell \in [j]$ tal que $r_\ell = \max_{k=1,\ldots,j} r_k$.

Claro que $C_j^* \geq C_\ell^*$, pela ordenação, e $C_j^* \geq r_\ell$.

Prova do lema

Seja $C_1^*, \ldots, C_n^*$ uma solução ótima do PL.

Reordene as tarefas de modo que $C_1^* \leq \cdots \leq C_n^*$.

Escalone as tarefas na ordem $1, \ldots, n$.

Chame de $C_1^N, \ldots, C_n^N$ os tempos de conclusão resultantes.

Lema: Para $j = 1, \ldots, n$, temos que $C_j^N \leq 3\, C_j^P$.

Prova: Como antes, não pode haver tempo ocioso entre $\max_{k=1,\ldots,j} r_k$ e C_j^N. Logo, $C_j^N \leq \max_{k=1,\ldots,j} r_k + \sum_{k=1}^{j} d_k$.

Seja $\ell \in [j]$ tal que $r_\ell = \max_{k=1,\ldots,j} r_k$.

Claro que $C_j^* \geq C_\ell^*$, pela ordenação, e $C_j^* \geq r_\ell$.

Vamos argumentar que $C_j^* \geq \frac{1}{2}\, d([j])$.

Prova do lema

Lema: Para $j = 1, \ldots, n$, temos que $C_j^N \leq 3\, C_j^P$.

Prova: Como antes, não pode haver tempo ocioso entre $\max_{k=1,\ldots,j} r_k$ e C_j^N. Logo, $C_j^N \leq \max_{k=1,\ldots,j} r_k + \sum_{k=1}^{j} d_k$.

Seja $\ell \in [j]$ tal que $r_\ell = \max_{k=1,\ldots,j} r_k$.

Claro que $C_j^* \geq C_\ell^*$, pela ordenação, e $C_j^* \geq r_\ell$.

Prova do lema

Lema: Para $j = 1, \ldots, n$, temos que $C_j^N \leq 3\, C_j^P$.

Prova: Como antes, não pode haver tempo ocioso entre $\max_{k=1,\ldots,j} r_k$ e C_j^N. Logo, $C_j^N \leq \max_{k=1,\ldots,j} r_k + \sum_{k=1}^{j} d_k$.

Seja $\ell \in [j]$ tal que $r_\ell = \max_{k=1,\ldots,j} r_k$.

Claro que $C_j^* \geq C_\ell^*$, pela ordenação, e $C_j^* \geq r_\ell$.

Se $C_j^* \geq \frac{1}{2}\, d([j])$, então

$$C_j^N \leq r_\ell + d([j]) \leq 3\, C_j^*$$

e concluímos o lema.

Conclusão

Por que $C_j^* \geq \frac{1}{2}\, d([j])$?

Conclusão

Por que $C_j^* \geq \frac{1}{2}\, d([j])$?

Seja $S = [j]$. Como C^* é viável, $\sum_{k \in S} d_k\, C_k^* \geq \frac{1}{2}\, d(S)^2$.

Conclusão

Por que $C_j^* \geq \frac{1}{2}\, d([j])$?

Seja $S = [j]$. Como C^* é viável, $\sum_{k \in S} d_k\, C_k^* \geq \frac{1}{2}\, d(S)^2$.

Então $C_j^*\, d(S) = C_j^*\, \sum_{k \in S} d_k \geq \sum_{k \in S} d_k\, C_k^* \geq \frac{1}{2}\, d(S)^2$.

Conclusão

Por que $C_j^* \geq \frac{1}{2}\, d([j])$?

Seja $S = [j]$. Como C^* é viável, $\sum_{k \in S} d_k\, C_k^* \geq \frac{1}{2}\, d(S)^2$.

Então $C_j^*\, d(S) = C_j^* \sum_{k \in S} d_k \geq \sum_{k \in S} d_k\, C_k^* \geq \frac{1}{2}\, d(S)^2$.

Ou seja, $C_j^* \geq \frac{1}{2}\, d(S)$.

Conclusão

Por que $C_j^* \geq \frac{1}{2}\, d([j])$?

Seja $S = [j]$. Como C^* é viável, $\sum_{k \in S} d_k\, C_k^* \geq \frac{1}{2}\, d(S)^2$.

Então $C_j^*\, d(S) = C_j^* \sum_{k \in S} d_k \geq \sum_{k \in S} d_k\, C_k^* \geq \frac{1}{2}\, d(S)^2$.

Ou seja, $C_j^* \geq \frac{1}{2}\, d(S)$.

Sem contar o fato de que precisamos resolver o tal PL...

Teorema: O algoritmo apresentado é uma 3-aproximação.

Mas como resolver tal PL?

Seja $d(S) = \sum_{j \in S} d_j$.

O PL fica: encontrar $C_1, \ldots, C_n$ que

minimizem $\sum_{j=1}^{n} w_j C_j$,
sujeitos a

$$C_j \geq r_j + d_j \qquad \text{para todo } j$$

$$\sum_{j \in S} d_j C_j \geq \frac{1}{2} d(S)^2 \qquad \text{para todo } S \subseteq [n].$$

Mas como resolver tal PL?

Seja $d(S) = \sum_{j \in S} d_j$.

O PL fica: encontrar $C_1, \ldots, C_n$ que

minimizem $\sum_{j=1}^{n} w_j C_j$,
sujeitos a

$$C_j \geq r_j + d_j \qquad \text{para todo } j$$

$$\sum_{j \in S} d_j C_j \geq \tfrac{1}{2} d(S)^2 \qquad \text{para todo } S \subseteq [n].$$

Método dos elipsóides.

Mas como resolver tal PL?

Seja $d(S) = \sum_{j \in S} d_j$.

O PL fica: encontrar $C_1, \ldots, C_n$ que

minimizem $\sum_{j=1}^{n} w_j C_j$,
sujeitos a

$$C_j \geq r_j + d_j \qquad \text{para todo } j$$
$$\sum_{j \in S} d_j C_j \geq \tfrac{1}{2} d(S)^2 \qquad \text{para todo } S \subseteq [n].$$

Método dos elipsóides.

Problema da separação.

Problema da separação

Seja $d(S) = \sum_{j \in S} d_j$.

O PL fica: encontrar $C_1, \ldots, C_n$ que

minimizem $\sum_{j=1}^{n} w_j C_j$,
sujeitos a

$$C_j \geq r_j + d_j \qquad \text{para todo } j$$

$$\sum_{j \in S} d_j C_j \geq \tfrac{1}{2} d(S)^2 \qquad \text{para todo } S \subseteq [n].$$

Problema da separação

Seja $d(S) = \sum_{j \in S} d_j$.

O PL fica: encontrar $C_1, \ldots, C_n$ que

minimizem $\sum_{j=1}^{n} w_j C_j$,
sujeitos a

$$C_j \geq r_j + d_j \qquad \text{para todo } j$$
$$\sum_{j \in S} d_j C_j \geq \tfrac{1}{2} d(S)^2 \qquad \text{para todo } S \subseteq [n].$$

Dado $C_1, \ldots, C_n$ candidato a viável para o PL acima, como determinar se $C_1, \ldots, C_n$ é viável ou apresentar uma desigualdade do PL que não está satisfeita?

Problema da separação

Encontrar $C_1, \ldots, C_n$ que

minimizem $\sum_{j=1}^{n} w_j C_j$,

sujeitos a

$$C_j \geq r_j + d_j \qquad \text{para todo } j$$

$$\sum_{j \in S} d_j C_j \geq \frac{1}{2} d(S)^2 \qquad \text{para todo } S \subseteq [n].$$

Problema da separação

Encontrar $C_1, \ldots, C_n$ que

minimizem $\sum_{j=1}^{n} w_j C_j$,
sujeitos a

$$C_j \geq r_j + d_j \qquad \text{para todo } j$$
$$\sum_{j \in S} d_j C_j \geq \frac{1}{2} d(S)^2 \qquad \text{para todo } S \subseteq [n].$$

Reordene os índices de modo que $C_1 \leq \cdots \leq C_n$.

Problema da separação

Encontrar $C_1, \ldots, C_n$ que

minimizem $\sum_{j=1}^{n} w_j C_j$,
sujeitos a

$$C_j \geq r_j + d_j \qquad \text{para todo } j$$
$$\sum_{j \in S} d_j C_j \geq \tfrac{1}{2} d(S)^2 \qquad \text{para todo } S \subseteq [n].$$

Reordene os índices de modo que $C_1 \leq \cdots \leq C_n$.

Verifique a segunda desigualdade para cada $S_i = [i]$.

Se não for satisfeita para algum S_i,
claro que $C_1, \ldots, C_n$ não é viável.

Problema da separação

Encontrar $C_1, \ldots, C_n$ que

minimizem $\sum_{j=1}^{n} w_j C_j$,
sujeitos a

$$C_j \geq r_j + d_j \qquad \text{para todo } j$$
$$\sum_{j \in S} d_j C_j \geq \frac{1}{2} d(S)^2 \qquad \text{para todo } S \subseteq [n].$$

Reordene os índices de modo que $C_1 \leq \cdots \leq C_n$.

Verifique a segunda desigualdade para cada $S_i = [i]$.

Se não for satisfeita para algum S_i,
claro que $C_1, \ldots, C_n$ não é viável.

Se for satisfeita para $i = 1, \ldots, n$,
então $C_1, \ldots, C_n$ é viável

Problema da separação

Encontrar $C_1, \ldots, C_n$ que

minimizem $\sum_{j=1}^{n} w_j C_j$,

sujeitos a

$$C_j \geq r_j + d_j \qquad \text{para todo } j$$

$$\sum_{j \in S} d_j C_j \geq \frac{1}{2} d(S)^2 \qquad \text{para todo } S \subseteq [n].$$

Problema da separação

Encontrar $C_1, \ldots, C_n$ que

minimizem $\sum_{j=1}^{n} w_j C_j$,

sujeitos a

$$C_j \geq r_j + d_j \qquad \text{para todo } j$$

$$\sum_{j \in S} d_j C_j \geq \frac{1}{2} d(S)^2 \qquad \text{para todo } S \subseteq [n].$$

Seja $S_i = [i]$.

Lema: Se $\sum_{j \in S_i} d_j C_j \geq \frac{1}{2} d(S_i)^2$ para $i = 1, \ldots, n$,
então $\sum_{j \in S} d_j C_j \geq \frac{1}{2} d(S)^2$ para todo $S \subseteq [n]$.

Problema da separação

Encontrar $C_1, \ldots, C_n$ que

minimizem $\sum_{j=1}^{n} w_j C_j$,
sujeitos a

$$C_j \geq r_j + d_j \qquad \text{para todo } j$$
$$\sum_{j \in S} d_j C_j \geq \frac{1}{2} d(S)^2 \qquad \text{para todo } S \subseteq [n].$$

Seja $S_i = [i]$.

Lema: Se $\sum_{j \in S_i} d_j C_j \geq \frac{1}{2} d(S_i)^2$ para $i = 1, \ldots, n$,
então $\sum_{j \in S} d_j C_j \geq \frac{1}{2} d(S)^2$ para todo $S \subseteq [n]$.

Prova feita na aula.

Prova do lema

Seja $S_i = [i]$.

Lema: Se $\sum_{j \in S_i} d_j \, C_j \geq \frac{1}{2} \, d(S_i)^2$ para $i = 1, \ldots, n$, então $\sum_{j \in S} d_j \, C_j \geq \frac{1}{2} \, d(S)^2$ para todo $S \subseteq [n]$.

Prova do lema

Seja $S_i = [i]$.

Lema: Se $\sum_{j \in S_i} d_j \, C_j \geq \frac{1}{2} \, d(S_i)^2$ para $i = 1, \ldots, n$,
então $\sum_{j \in S} d_j \, C_j \geq \frac{1}{2} \, d(S)^2$ para todo $S \subseteq [n]$.

Prova: Suponha que $\sum_{j \in S} d_j \, C_j < \frac{1}{2} \, d(S)^2$.

Vamos mostrar que $\sum_{j \in S_i} d_j \, C_j < \frac{1}{2} \, d(S_i)^2$ para algum i.

Prova do lema

Seja $S_i = [i]$.

Lema: Se $\sum_{j \in S_i} d_j \, C_j \geq \frac{1}{2} \, d(S_i)^2$ para $i = 1, \ldots, n$,
então $\sum_{j \in S} d_j \, C_j \geq \frac{1}{2} \, d(S)^2$ para todo $S \subseteq [n]$.

Prova: Suponha que $\sum_{j \in S} d_j \, C_j < \frac{1}{2} \, d(S)^2$.

Vamos mostrar que $\sum_{j \in S_i} d_j \, C_j < \frac{1}{2} \, d(S_i)^2$ para algum i.

Alteraremos S diminuindo a diferença $\sum_{j \in S} d_j \, C_j - \frac{1}{2} \, d(S)^2$.

Prova do lema

Seja $S_i = [i]$.

Lema: Se $\sum_{j \in S_i} d_j\, C_j \geq \frac{1}{2}\, d(S_i)^2$ para $i = 1, \ldots, n$,
então $\sum_{j \in S} d_j\, C_j \geq \frac{1}{2}\, d(S)^2$ para todo $S \subseteq [n]$.

Prova: Suponha que $\sum_{j \in S} d_j\, C_j < \frac{1}{2}\, d(S)^2$.

Vamos mostrar que $\sum_{j \in S_i} d_j\, C_j < \frac{1}{2}\, d(S_i)^2$ para algum i.

Alteraremos S diminuindo a diferença $\sum_{j \in S} d_j\, C_j - \frac{1}{2}\, d(S)^2$.

A remoção de uma tarefa k de S diminui a diferença acima se $-d_k\, C_k + d_k\, d(S - k) + \frac{1}{2}\, d_k^2 < 0$.
Ou seja, se $C_k > d(S - k) + \frac{1}{2}\, d_k$.

Prova do lema

Seja $S_i = [i]$.

Lema: Se $\sum_{j \in S_i} d_j \, C_j \geq \frac{1}{2} \, d(S_i)^2$ para $i = 1, \ldots, n$,
então $\sum_{j \in S} d_j \, C_j \geq \frac{1}{2} \, d(S)^2$ para todo $S \subseteq [n]$.

Prova: Suponha que $\sum_{j \in S} d_j \, C_j < \frac{1}{2} \, d(S)^2$.

Vamos mostrar que $\sum_{j \in S_i} d_j \, C_j < \frac{1}{2} \, d(S_i)^2$ para algum i.

Alteraremos S diminuindo a diferença $\sum_{j \in S} d_j \, C_j - \frac{1}{2} \, d(S)^2$.

A inserção de uma tarefa k de S diminui a diferença acima
se $d_k \, C_k - d_k \, d(S) - \frac{1}{2} \, d_k^2 < 0$.
Ou seja, se $C_k < d(S) + \frac{1}{2} \, d_k$.

Prova do lema

Seja $S_i = [i]$.

Lema: Se $\sum_{j \in S_i} d_j \, C_j \geq \frac{1}{2} \, d(S_i)^2$ para $i = 1, \ldots, n$,
então $\sum_{j \in S} d_j \, C_j \geq \frac{1}{2} \, d(S)^2$ para todo $S \subseteq [n]$.

Prova: Suponha que $\sum_{j \in S} d_j \, C_j < \frac{1}{2} \, d(S)^2$.

Vamos mostrar que $\sum_{j \in S_i} d_j \, C_j < \frac{1}{2} \, d(S_i)^2$ para algum i.

Alteraremos S diminuindo a diferença $\sum_{j \in S} d_j \, C_j - \frac{1}{2} \, d(S)^2$.

Podemos remover k de S se $C_k > d(S - k) + \frac{1}{2} \, d_k$.

Podemos inserir k em S se $C_k < d(S) + \frac{1}{2} \, d_k$.

Prova do lema

Seja $S_i = [i]$.

Lema: Se $\sum_{j \in S_i} d_j \, C_j \geq \frac{1}{2} \, d(S_i)^2$ para $i = 1, \ldots, n$, então $\sum_{j \in S} d_j \, C_j \geq \frac{1}{2} \, d(S)^2$ para todo $S \subseteq [n]$.

Prova: Suponha que $\sum_{j \in S} d_j \, C_j < \frac{1}{2} \, d(S)^2$.

Podemos remover k de S se $C_k > d(S - k) + \frac{1}{2} \, d_k$.
Podemos inserir k em S se $C_k < d(S) + \frac{1}{2} \, d_k$.

Prova do lema

Seja $S_i = [i]$.

Lema: Se $\sum_{j \in S_i} d_j \, C_j \geq \frac{1}{2} \, d(S_i)^2$ para $i = 1, \ldots, n$, então $\sum_{j \in S} d_j \, C_j \geq \frac{1}{2} \, d(S)^2$ para todo $S \subseteq [n]$.

Prova: Suponha que $\sum_{j \in S} d_j \, C_j < \frac{1}{2} \, d(S)^2$.

Podemos remover k de S se $C_k > d(S - k) + \frac{1}{2} \, d_k$.
Podemos inserir k em S se $C_k < d(S) + \frac{1}{2} \, d_k$.

Seja ℓ a tarefa de maior índice em S.

Remova ℓ de S se $C_\ell > d(S - \ell) + \frac{1}{2} \, d_\ell$ e repita.

Prova do lema

Seja $S_i = [i]$.

Lema: Se $\sum_{j \in S_i} d_j \, C_j \geq \frac{1}{2} \, d(S_i)^2$ para $i = 1, \ldots, n$, então $\sum_{j \in S} d_j \, C_j \geq \frac{1}{2} \, d(S)^2$ para todo $S \subseteq [n]$.

Prova: Suponha que $\sum_{j \in S} d_j \, C_j < \frac{1}{2} \, d(S)^2$.

Podemos remover k de S se $C_k > d(S - k) + \frac{1}{2} \, d_k$.
Podemos inserir k em S se $C_k < d(S) + \frac{1}{2} \, d_k$.

Seja ℓ a tarefa de maior índice em S.

Remova ℓ de S se $C_\ell > d(S - \ell) + \frac{1}{2} \, d_\ell$ e repita.

Ao final, $C_\ell \leq d(S - \ell) + \frac{1}{2} \, d_\ell$.

Prova do lema

Seja $S_i = [i]$.

Lema: Se $\sum_{j \in S_i} d_j \, C_j \geq \frac{1}{2} \, d(S_i)^2$ para $i = 1, \ldots, n$, então $\sum_{j \in S} d_j \, C_j \geq \frac{1}{2} \, d(S)^2$ para todo $S \subseteq [n]$.

Prova: Suponha que $\sum_{j \in S} d_j \, C_j < \frac{1}{2} \, d(S)^2$.

Podemos remover k de S se $C_k > d(S - k) + \frac{1}{2} \, d_k$.
Podemos inserir k em S se $C_k < d(S) + \frac{1}{2} \, d_k$.

Seja ℓ a tarefa de maior índice em S.

Remova ℓ de S se $C_\ell > d(S - \ell) + \frac{1}{2} \, d_\ell$ e repita.

Ao final, $C_\ell \leq d(S - \ell) + \frac{1}{2} \, d_\ell$.

Vale que $\sum_{j \in S_\ell} d_j \, C_j < \frac{1}{2} \, d(S_\ell)^2$.

Prova do lema

Lema: Se $\sum_{j \in S_i} d_j \, C_j \geq \frac{1}{2} \, d(S_i)^2$ para $i = 1, \ldots, n$, então $\sum_{j \in S} d_j \, C_j \geq \frac{1}{2} \, d(S)^2$ para todo $S \subseteq [n]$.

Prova: Suponha que $\sum_{j \in S} d_j \, C_j < \frac{1}{2} \, d(S)^2$.

Podemos remover k de S se $C_k > d(S - k) + \frac{1}{2} \, d_k$.

Podemos inserir k em S se $C_k < d(S) + \frac{1}{2} \, d_k$.

Seja ℓ a tarefa de maior índice em S.

Remova ℓ de S se $C_\ell > d(S - \ell) + \frac{1}{2} \, d_\ell$ e repita.

Ao final, $C_\ell \leq d(S - \ell) + \frac{1}{2} \, d_\ell$.

Vale que $\sum_{j \in S_\ell} d_j \, C_j < \frac{1}{2} \, d(S_\ell)^2$.

Prova do lema

Lema: Se $\sum_{j \in S_i} d_j \, C_j \geq \frac{1}{2} \, d(S_i)^2$ para $i = 1, \ldots, n$, então $\sum_{j \in S} d_j \, C_j \geq \frac{1}{2} \, d(S)^2$ para todo $S \subseteq [n]$.

Prova: Suponha que $\sum_{j \in S} d_j \, C_j < \frac{1}{2} \, d(S)^2$.

Podemos remover k de S se $C_k > d(S - k) + \frac{1}{2} \, d_k$.

Podemos inserir k em S se $C_k < d(S) + \frac{1}{2} \, d_k$.

Seja ℓ a tarefa de maior índice em S.

Remova ℓ de S se $C_\ell > d(S - \ell) + \frac{1}{2} \, d_\ell$ e repita.

Ao final, $C_\ell \leq d(S - \ell) + \frac{1}{2} \, d_\ell$.

Vale que $\sum_{j \in S_\ell} d_j \, C_j < \frac{1}{2} \, d(S_\ell)^2$.

De fato, se $k < \ell$ e $k \notin S$, então k pode ser inserido em S pois $C_k \leq C_\ell \leq d(S - \ell) + \frac{1}{2} \, d_\ell < d(S) < d(S) + \frac{1}{2} \, d_k$.