Algoritmos de Aproximação

Segundo Semestre de 2012

Cobertura por conjuntos

Instância:

- conjunto base finito E
- coleção $\mathcal{S} = \{S_1, \ldots, S_m\}$ de subconjuntos de E
- custo $c_j > 0$ para cada S_j em $\mathcal{S}$

Objetivo: encontrar cobertura $\mathcal{S}' \subseteq \mathcal{S}$ de E de custo mínimo.

Cobertura por conjuntos

Instância:

- conjunto base finito E
- coleção $\mathcal{S} = \{S_1, \ldots, S_m\}$ de subconjuntos de E
- custo $c_j > 0$ para cada S_j em $\mathcal{S}$

Objetivo: encontrar cobertura $\mathcal{S}' \subseteq \mathcal{S}$ de E de custo mínimo.

Relaxação linear:

Dados E, $\mathcal{S}$ e c, encontrar x que

minimize $\sum_j c_j\, x_j$

sujeito a $\sum_{j:e \in S_j} x_j \geq 1$ para cada e em E

$x_j \geq 0$ para $j = 1, \ldots, m$

Arredondamento probabilístico

ARREDPROB $(E, \mathcal{S}, c)$ $\quad \triangleright\ \mathcal{S} = \{S_1, \ldots, S_m\}$

```
1   x* ← solução da relaxação linear (P)
2   I ← ∅
3   para j ← 1 até m faça
4       r ← RANDOM(0, 1)      ▷ devolve um real em [0, 1)
                                com distribuição uniforme
5       se r < x*_j
6           então I ← I ∪ {j}
7   devolva I
```

Arredondamento probabilístico

ARREDPROB $(E, \mathcal{S}, c)$ $\quad \triangleright\ \mathcal{S} = \{S_1, \ldots, S_m\}$

```
1   x* ← solução da relaxação linear (P)
2   I ← ∅
3   para j ← 1 até m faça
4       r ← RANDOM(0, 1)      ▷ devolve um real em [0, 1)
                                com distribuição uniforme
5       se r < x*_j
6           então I ← I ∪ {j}
7   devolva I
```

Consumo de tempo polinomial.

Arredondamento probabilístico

ARREDPROB $(E, \mathcal{S}, c)$ $\quad \triangleright\ \mathcal{S} = \{S_1, \ldots, S_m\}$

```
1   x* ← solução da relaxação linear (P)
2   I ← ∅
3   para j ← 1 até m faça
4       r ← RANDOM(0, 1)      ▷ devolve um real em [0, 1)
                                com distribuição uniforme
5       se r < x*_j
6           então I ← I ∪ {j}
7   devolva I
```

Consumo de tempo polinomial.

Qual é a probabilidade de I ser uma cobertura?

Arredondamento probabilístico

ARREDPROB $(E, \mathcal{S}, c)$ $\triangleright$ $\mathcal{S} = \{S_1, \ldots, S_m\}$

1 $x^* \leftarrow$ solução da relaxação linear (P)
2 $I \leftarrow \emptyset$
3 **para** $j \leftarrow 1$ **até** m **faça**
4 $r \leftarrow$ RANDOM$(0, 1)$ $\triangleright$ devolve um real em $[0, 1)$ com distribuição uniforme
5 **se** $r < x^*_j$
6 **então** $I \leftarrow I \cup \{j\}$
7 **devolva** I

Consumo de tempo polinomial.

Qual é a probabilidade de I ser uma cobertura?

Qual é o custo esperado de I?

Qual é o custo esperado de I?

ARREDPROB $(E, \mathcal{S}, c)$ $\quad \triangleright\ \mathcal{S} = \{S_1, \ldots, S_m\}$

1 $x^* \leftarrow$ solução da relaxação linear (P)
2 $I \leftarrow \emptyset$
3 **para** $j \leftarrow 1$ **até** m **faça**
4 $\quad r \leftarrow$ RANDOM$(0, 1)$
5 $\quad$ **se** $r < x^*_j$
6 $\quad\quad$ **então** $I \leftarrow I \cup \{j\}$
7 **devolva** I

Qual é o custo esperado de I?

```
ARREDPROB (E, S, c)        ▷ S = {S_1, ..., S_m}
1   x* ← solução da relaxação linear (P)
2   I ← ∅
3   para j ← 1 até m faça
4       r ← RANDOM(0, 1)
5       se r < x*_j
6           então I ← I ∪ {j}
7   devolva I
```

X_j: variável aleatória que indica se j entrou em I ou não.

Vale que $\Pr[X_j = 1] = x^*_j$.

Qual é o custo esperado de I?

ARREDPROB $(E, \mathcal{S}, c)$ $\quad \triangleright\ \mathcal{S} = \{S_1, \ldots, S_m\}$

1 $x^* \leftarrow$ solução da relaxação linear (P)
2 $I \leftarrow \emptyset$
3 **para** $j \leftarrow 1$ **até** m **faça**
4 $\quad r \leftarrow$ RANDOM$(0, 1)$
5 $\quad$ **se** $r < x^*_j$
6 $\quad\quad$ **então** $I \leftarrow I \cup \{j\}$
7 **devolva** I

X_j: variável aleatória que indica se j entrou em I ou não.

Vale que $\Pr[X_j = 1] = x^*_j$.

Logo $\mathrm{E}[c(I)] = \sum_{j=1}^{m} c_j \Pr[X_j = 1] = \sum_{j=1}^{m} c_j\, x^*_j \leq \mathrm{opt}$.

Probabilidade de I ser uma cobertura

```
ARREDPROB (E, S, c)        ▷ S = {S_1, ..., S_m}
1   x* ← solução da relaxação linear (P)
2   I ← ∅
3   para j ← 1 até m faça
4       r ← RANDOM(0, 1)
5       se r < x*_j
6           então I ← I ∪ {j}
7   devolva I
```

Probabilidade de I ser uma cobertura

```
ARREDPROB (E, S, c)        ▷ S = {S_1, ..., S_m}
1   x* ← solução da relaxação linear (P)
2   I ← ∅
3   para j ← 1 até m faça
4         r ← RANDOM(0, 1)
5         se r < x*_j
6               então I ← I ∪ {j}
7   devolva I
```

Para cada e' em E,

$$\Pr[e' \text{ não é coberto}] = \prod_{j:e' \in S_j} \Pr[X_j = 0] = \prod_{j:e' \in S_j} (1 - x_j^*)$$

$$\leq \prod_{j:e' \in S_j} e^{-x_j^*} = e^{-\sum_{j:e' \in S_j} x_j^*} \leq e^{-1}.$$

Probabilidade de I ser uma cobertura

```
ARREDPROB (E, S, c)        ▷ S = {S_1, ..., S_m}
1   x* ← solução da relaxação linear (P)
2   I ← ∅
3   para j ← 1 até m faça
4         r ← RANDOM(0, 1)
5         se r < x*_j
6               então I ← I ∪ {j}
7   devolva I
```

Para cada e' em E, temos que $\Pr[e' \text{ não é coberto}] \leq e^{-1}$.

Probabilidade de I ser uma cobertura

ARREDPROB $(E, \mathcal{S}, c)$ $\quad \triangleright\ \mathcal{S} = \{S_1, \ldots, S_m\}$

1 $x^* \leftarrow$ solução da relaxação linear (P)
2 $I \leftarrow \emptyset$
3 **para** $j \leftarrow 1$ **até** m **faça**
4 $\quad r \leftarrow$ RANDOM$(0,1)$
5 $\quad$ **se** $x^*_j < r$
6 $\quad\quad$ **então** $I \leftarrow I \cup \{j\}$
7 **devolva** I

repita $b \ln n$ vezes

Para cada e' em E, temos que $\Pr[e' \text{ não é coberto}] \leq e^{-1}$.

Queremos que essa probabilidade seja bem menor.

Basta repetirmos $b \ln n$ vezes as linhas 4-6, para um $b \geq 2$, onde $n := |E|$.

Probabilidade de I ser uma cobertura

```
4        r ← RANDOM(0, 1)
5        se x*_j < r
6            então I ← I ∪ {j}
```

repita $b \ln n$ vezes

Para cada e' em E, temos que

$$\Pr[e' \text{ não é coberto}] \leq e^{-b \ln n} = \frac{1}{n^b}.$$

Probabilidade de I ser uma cobertura

```
4        r ← RANDOM(0, 1)
5        se x*_j < r
6            então I ← I ∪ {j}
```

repita $b \ln n$ vezes

Para cada e' em E, temos que

$$\Pr[e' \text{ não é coberto}] \leq e^{-b \ln n} = \frac{1}{n^b}.$$

$$\begin{aligned} \Pr[I \text{ não é cobertura}] &\leq \sum_{e' \in E} \Pr[e' \text{ não é coberto}] \\ &\leq n \frac{1}{n^b} = \frac{1}{n^{b-1}}. \end{aligned}$$

Probabilidade de I ser uma cobertura

```
4        r ← RANDOM(0, 1)
5        se x*_j < r
6            então I ← I ∪ {j}
```

repita $b \ln n$ vezes

Para cada e' em E, temos que

$$\Pr[e' \text{ não é coberto}] \leq e^{-b \ln n} = \frac{1}{n^b}.$$

$$\begin{aligned} \Pr[I \text{ não é cobertura}] &\leq \sum_{e' \in E} \Pr[e' \text{ não é coberto}] \\ &\leq n \frac{1}{n^b} = \frac{1}{n^{b-1}}. \end{aligned}$$

Dizemos que I é uma cobertura com alta probabilidade.

Custo esperado do novo I

ARREDPROB $(E, \mathcal{S}, c)$ $\quad \triangleright\ \mathcal{S} = \{S_1, \ldots, S_m\}$

1 $x^* \leftarrow$ solução da relaxação linear (P)
2 $I \leftarrow \emptyset$
3 **para** $j \leftarrow 1$ **até** m **faça**
4 $\quad r \leftarrow$ RANDOM$(0,1)$
5 $\quad$ **se** $x_j^* < r$
6 $\quad\quad$ **então** $I \leftarrow I \cup \{j\}$
7 **devolva** I

repita $b \ln n$ vezes

Custo esperado do novo I

ARREDPROB $(E, \mathcal{S}, c)$ $\quad \triangleright\ \mathcal{S} = \{S_1, \ldots, S_m\}$

```
1   x* ← solução da relaxação linear (P)
2   I ← ∅
3   para j ← 1 até m faça
4       r ← RANDOM(0, 1)
5       se x*_j < r
6           então I ← I ∪ {j}
7   devolva I
```

repita $b \ln n$ vezes

X_j: variável aleatória que indica se j entrou em I ou não.

Vale que $\Pr[X_j = 1] \leq (b \ln n) x^*_j$. Logo...

Custo esperado do novo I

ARREDPROB $(E, \mathcal{S}, c)$ $\quad \triangleright\ \mathcal{S} = \{S_1, \ldots, S_m\}$

```
1   x* ← solução da relaxação linear (P)
2   I ← ∅
3   para j ← 1 até m faça
4       r ← RANDOM(0, 1)
5       se x*_j < r
6           então I ← I ∪ {j}
7   devolva I
```

repita $b \ln n$ vezes

X_j: variável aleatória que indica se j entrou em I ou não.

Vale que $\Pr[X_j = 1] \leq (b \ln n) x^*_j$. Logo...

$$\mathrm{E}[c(I)] = \sum_{j=1}^{m} c_j \Pr[X_j = 1] \leq b \ln n \sum_{j=1}^{m} c_j \, x^*_j \leq (b \ln n) \mathrm{opt}.$$

Custo esperado do novo I

X_j: variável aleatória que indica se j entrou em I ou não.

Vale que $\Pr[X_j = 1] \leq (b \ln n) x_j^*$. Logo...

$$\mathrm{E}[c(I)] = \sum_{j=1}^{m} c_j \Pr[X_j = 1] \leq b \ln n \sum_{j=1}^{m} c_j \, x_j^* \leq (b \ln n) \mathrm{opt}.$$

Custo esperado do novo I

X_j: variável aleatória que indica se j entrou em I ou não.

Vale que $\Pr[X_j = 1] \leq (b \ln n) x_j^*$. Logo...

$$\mathrm{E}[c(I)] = \sum_{j=1}^{m} c_j \Pr[X_j = 1] \leq b \ln n \sum_{j=1}^{m} c_j \, x_j^* \leq (b \ln n)\mathrm{opt}.$$

Exercício: Deduza uma delimitação deste tipo para

$$\mathrm{E}[c(I) | I \text{ é uma cobertura}].$$

Fizemos em sala isso.

Arredondamento probabilístico

ARREDPROB $(E, \mathcal{S}, c)$ $\quad \triangleright\ \mathcal{S} = \{S_1, \ldots, S_m\}$

1 $x^* \leftarrow$ solução da relaxação linear (P)
2 $I \leftarrow \emptyset$
3 **para** $j \leftarrow 1$ **até** m **faça**
4 **repita** $b \ln |E|$ vezes
5 $r \leftarrow$ RANDOM$(0, 1)$
6 **se** $r < x^*_j$
7 **então** $I \leftarrow I \cup \{j\}$
8 **devolva** I

Arredondamento probabilístico

ARREDPROB $(E, \mathcal{S}, c)$ $\quad \triangleright\ \mathcal{S} = \{S_1, \ldots, S_m\}$

```
1   x* ← solução da relaxação linear (P)
2   I ← ∅
3   para j ← 1 até m faça
4       repita b ln |E| vezes
5           r ← RANDOM(0, 1)
6           se r < x*_j
7               então I ← I ∪ {j}
8   devolva I
```

Teorema: ARREDPROB$(E, \mathcal{S}, c)$ é uma $\mathrm{O}(\lg n)$-aproximação probabilística para o SET COVER com alta probabilidade, onde $n = |E|$.

Algoritmos gulosos

Vamos começar com um exemplo:

- Escalonamento com deadlines em uma máquina

Material do capítulo 2 do WS

Escalonamento com deadlines

Dados: n tarefas

momento de chegada $r[i]$ da tarefa i $(i = 1, \ldots, n)$

prazo de entrega $p[i]$ da tarefa i $(i = 1, \ldots, n)$

duração $d[i]$ da tarefa i $(i = 1, \ldots, n)$

Escalonamento com deadlines

Dados: n tarefas

momento de chegada $r[i]$ da tarefa i $(i = 1, \ldots, n)$

prazo de entrega $p[i]$ da tarefa i $(i = 1, \ldots, n)$

duração $d[i]$ da tarefa i $(i = 1, \ldots, n)$

Um escalonamento é um vetor s onde
$s[i]$ é o momento de início do processamento da tarefa i
$(i = 1, \ldots, n)$.

Escalonamento com deadlines

Dados: n tarefas

momento de chegada $r[i]$ da tarefa i $(i = 1, \ldots, n)$

prazo de entrega $p[i]$ da tarefa i $(i = 1, \ldots, n)$

duração $d[i]$ da tarefa i $(i = 1, \ldots, n)$

Um escalonamento é um vetor s onde
$s[i]$ é o momento de início do processamento da tarefa i
$(i = 1, \ldots, n)$.

$C[i]$: momento em que a tarefa i é completada, $s[i] + d[i]$.

$L[i]$: atraso, dado por $C[i] - p[i]$

Escalonamento com deadlines

Dados: n tarefas

momento de chegada $r[i]$ da tarefa i $(i = 1, \ldots, n)$

prazo de entrega $p[i]$ da tarefa i $(i = 1, \ldots, n)$

duração $d[i]$ da tarefa i $(i = 1, \ldots, n)$

Um escalonamento é um vetor s onde
$s[i]$ é o momento de início do processamento da tarefa i
$(i = 1, \ldots, n)$.

$C[i]$: momento em que a tarefa i é completada, $s[i] + d[i]$.

$L[i]$: atraso, dado por $C[i] - p[i]$

$L_{\max}$: atraso máximo, ou seja, $\max_i L[i]$

Escalonamento com deadlines

Dados: n tarefas

momento de chegada $r[i]$ da tarefa i $(i = 1, \ldots, n)$

prazo de entrega $p[i]$ da tarefa i $(i = 1, \ldots, n)$

duração $d[i]$ da tarefa i $(i = 1, \ldots, n)$

Um escalonamento é um vetor s onde
$s[i]$ é o momento de início do processamento da tarefa i
$(i = 1, \ldots, n)$.

$C[i]$: momento em que a tarefa i é completada, $s[i] + d[i]$.

$L[i]$: atraso, dado por $C[i] - p[i]$

$L_{\max}$: atraso máximo, ou seja, $\max_i L[i]$

Objetivo: encontrar escalonamento com $L_{\max}$ mínimo.

Escalonamento com deadlines

Problemas...

Se a instância tem atraso máximo $L_{\max} = 0$, qualquer aproximação tem que produzir um tal escalonamento.

Porém decidir se existe ou não um escalonamento com $L_{\max} \leq 0$ é um problema (fortemente) NP-difícil.

Vamos então considerar o caso em que $r[i] \geq 0$ e $p[i] < 0$ para toda tarefa i. Para este caso, $L_{\max} > 0$ para todo escalonamento.

Vamos mostrar uma 2-aproximação para este caso.

Algoritmo guloso

Tarefa i está disponível no momento t se $r[i] \leq t$.

Política EDD: Sempre que a máquina fica ociosa, escalone uma tarefa disponível com o menor prazo possível.

EDD: earliest due date

Algoritmo guloso

Tarefa i está disponível no momento t se $r[i] \leq t$.

Política EDD: Sempre que a máquina fica ociosa, escalone uma tarefa disponível com o menor prazo possível.

EDD: earliest due date

Teorema: A política EDD é uma 2-aproximação para o caso em que $p[i] < 0$ para todo i.

Prova feita na aula.

Algoritmo guloso

ESCALONAMENTO-GULOSO (n, r, p, d)

```
1   S ← {1, ..., n}
2   t ← 0
3   enquanto S ≠ ∅ faça
4       t' ← min{r[i] : i ∈ S}
5       t ← max{t, t'}
6       j ← arg min{p[i] : i ∈ S e r[i] ≤ t}
7       s[j] ← t
8       t ← t + d[j]
9       S ← S \ {j}
10  devolva s
```

Teorema: ESCALONAMENTO-GULOSO (n, r, d, p) é uma 2-aproximação para o caso em que $p[i] < 0$ para todo i.

Delimitação para opt

S: conjunto de tarefas

$r(S)$: $\min\{r[i] : i \in S\}$
$d(S)$: $\sum_{i \in S} d[i]$
$p(S)$: $\max\{p[i] : i \in S\}$

Delimitação para opt

S: conjunto de tarefas

$r(S)$: $\min\{r[i] : i \in S\}$
$d(S)$: $\sum_{i \in S} d[i]$
$p(S)$: $\max\{p[i] : i \in S\}$

Lema: Para cada conjunto S de tarefas,

$$L^*_{\max} \geq r(S) + d(S) - p(S)$$

Delimitação para opt

S: conjunto de tarefas

$r(S)$: $\min\{r[i] : i \in S\}$
$d(S)$: $\sum_{i \in S} d[i]$
$p(S)$: $\max\{p[i] : i \in S\}$

Lema: Para cada conjunto S de tarefas,

$$L^*_{\max} \geq r(S) + d(S) - p(S)$$

Prova: Considere um escalonamento s^* ótimo.

Seja i a tarefa de S com $s^*[i]$ mínimo.
Claro que $s^*[i] \geq r(S)$.

Delimitação para opt

S: conjunto de tarefas

$r(S)$: $\min\{r[i] : i \in S\}$
$d(S)$: $\sum_{i \in S} d[i]$
$p(S)$: $\max\{p[i] : i \in S\}$

Lema: Para cada conjunto S de tarefas,

$$L^*_{\max} \geq r(S) + d(S) - p(S)$$

Prova: Considere um escalonamento s^* ótimo.

Seja i a tarefa de S com $s^*[i]$ mínimo.
Claro que $s^*[i] \geq r(S)$.

Seja j a tarefa de S com $s^*[j]$ máximo.
Claro que $C^*[j] = s^*[j] + d[j] \geq s^*[i] + d(S)$.

Delimitação para opt

S: conjunto de tarefas $\qquad d(S)$: $\sum_{i \in S} d[i]$

$r(S)$: $\min\{r[i] : i \in S\}$ $\qquad p(S)$: $\max\{p[i] : i \in S\}$

Lema: Para cada conjunto S de tarefas,

$$L^*_{\max} \geq r(S) + d(S) - p(S)$$

Prova: Considere um escalonamento s^* ótimo.

Seja i a tarefa de S com $s^*[i]$ mínimo.
Claro que $s^*[i] \geq r(S)$.

Seja j a tarefa de S com $s^*[j]$ máximo.
Claro que $C^*[j] = s^*[j] + d[j] \geq s^*[i] + d(S)$.

Delimitação para opt

S: conjunto de tarefas $\quad d(S)$: $\sum_{i \in S} d[i]$

$r(S)$: $\min\{r[i] : i \in S\}$ $\quad p(S)$: $\max\{p[i] : i \in S\}$

Lema: Para cada conjunto S de tarefas,

$$L^*_{\max} \geq r(S) + d(S) - p(S)$$

Prova: Considere um escalonamento s^* ótimo.

Seja i a tarefa de S com $s^*[i]$ mínimo.
Claro que $s^*[i] \geq r(S)$.

Seja j a tarefa de S com $s^*[j]$ máximo.
Claro que $C^*[j] = s^*[j] + d[j] \geq s^*[i] + d(S)$.

Mas então $C^*[j] \geq r(S) + d(S)$.

Delimitação para opt

S: conjunto de tarefas $\qquad d(S)$: $\sum_{i \in S} d[i]$

$r(S)$: $\min\{r[i] : i \in S\}$ $\qquad p(S)$: $\max\{p[i] : i \in S\}$

Lema: Para cada conjunto S de tarefas,

$$L^*_{\max} \geq r(S) + d(S) - p(S)$$

Prova: Considere um escalonamento s^* ótimo.

Seja i a tarefa de S com $s^*[i]$ mínimo.
Claro que $s^*[i] \geq r(S)$.

Seja j a tarefa de S com $s^*[j]$ máximo.
Claro que $C^*[j] = s^*[j] + d[j] \geq s^*[i] + d(S)$.

Mas então $C^*[j] \geq r(S) + d(S)$.

Logo $L^*_{\max} \geq C^*[j] - p[j] \geq r(S) + d(S) - p(S)$. ■

Análise da razão de aproximação

Lema: Para cada conjunto S de tarefas,

$$L^*_{\max} \geq r(S) + d(S) - p(S)$$

Análise da razão de aproximação

Lema: Para cada conjunto S de tarefas,

$$L^*_{\max} \geq r(S) + d(S) - p(S)$$

Teorema: ESCALONAMENTO-GULOSO (n, r, d, p) é uma 2-aproximação para o caso em que $p[i] < 0$ para todo i.

Prova: Considere o escalonamento s produzido pelo ESCALONAMENTO-GULOSO.

Seja j tal que $L_{\max} = C[j] - p[j]$.

Análise da razão de aproximação

Lema: Para cada conjunto S de tarefas,

$$L^*_{\max} \geq r(S) + d(S) - p(S)$$

Teorema: ESCALONAMENTO-GULOSO (n, r, d, p) é uma 2-aproximação para o caso em que $p[i] < 0$ para todo i.

Prova: Considere o escalonamento s produzido pelo ESCALONAMENTO-GULOSO.

Seja j tal que $L_{\max} = C[j] - p[j]$.

Seja S o conjunto maximal de tarefas k tais que $s[k] \leq s[j]$ e a máquina não fica ociosa em nenhum instante do intervalo $[s[i], C[j])$, onde $i = \arg\min\{s[k] : k \in S\}$.

Análise da razão de aproximação

Teorema: ESCALONAMENTO-GULOSO (n, r, d, p) é uma 2-aproximação para o caso em que $p[i] < 0$ para todo i.

Prova: Considere o escalonamento s produzido pelo ESCALONAMENTO-GULOSO.

Seja j tal que $L_{\max} = C[j] - p[j]$.

Seja S o conjunto maximal de tarefas k tais que $s[k] \leq s[j]$ e a máquina não fica ociosa em nenhum instante do intervalo $[s[i], C[j])$, onde $i = \arg\min\{s[k] : k \in S\}$.

Análise da razão de aproximação

Teorema: ESCALONAMENTO-GULOSO (n, r, d, p) é uma 2-aproximação para o caso em que $p[i] < 0$ para todo i.

Prova: Considere o escalonamento s produzido pelo ESCALONAMENTO-GULOSO.

Seja j tal que $L_{\max} = C[j] - p[j]$.

Seja S o conjunto maximal de tarefas k tais que $s[k] \leq s[j]$ e a máquina não fica ociosa em nenhum instante do intervalo $[s[i], C[j])$, onde $i = \arg\min\{s[k] : k \in S\}$.

Do Lema para S, temos que
$L^*_{\max} \geq r(S) + d(S) - p(S) \geq r(S) + d(S)$, pois $p(S) < 0$.

Análise da razão de aproximação

Teorema: ESCALONAMENTO-GULOSO (n, r, d, p) é uma 2-aproximação para o caso em que $p[i] < 0$ para todo i.

Prova: Considere o escalonamento s produzido pelo ESCALONAMENTO-GULOSO.

Seja j tal que $L_{\max} = C[j] - p[j]$.

Seja S o conjunto maximal de tarefas k tais que $s[k] \leq s[j]$ e a máquina não fica ociosa em nenhum instante do intervalo $[s[i], C[j])$, onde $i = \arg\min\{s[k] : k \in S\}$.

Do Lema para S, temos que
$L^*_{\max} \geq r(S) + d(S) - p(S) \geq r(S) + d(S)$, pois $p(S) < 0$.

Do Lema para o conjunto $\{j\}$, temos que
$L^*_{\max} \geq r[j] + d[j] - p[j] \geq -p[j]$, pois $r[j] \geq 0$ e $d[j] > 0$.

Análise da razão de aproximação

Teorema: ESCALONAMENTO-GULOSO (n, r, d, p) é uma 2-aproximação para o caso em que $p[i] < 0$ para todo i.

Prova: Considere o escalonamento s produzido pelo ESCALONAMENTO-GULOSO.

Seja j tal que $L_{\max} = C[j] - p[j]$.

Seja S o conjunto maximal de tarefas k tais que $s[k] \leq s[j]$ e a máquina não fica ociosa em nenhum instante do intervalo $[s[i], C[j])$, onde $i = \arg\min\{s[k] : k \in S\}$.

Do Lema para S, temos que
$L^*_{\max} \geq r(S) + d(S) - p(S) \geq r(S) + d(S)$, pois $p(S) < 0$.

Do Lema para o conjunto $\{j\}$, temos que
$L^*_{\max} \geq r[j] + d[j] - p[j] \geq -p[j]$, pois $r[j] \geq 0$ e $d[j] > 0$.

Logo, $L_{\max} = C[j] - p[j] = r(S) + d(S) - p[j] \leq 2\, L^*_{\max}$. ∎