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 $\mathcal{S}$

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 $j = 1, \ldots, m$

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 $j = 1, \ldots, m$

Este é o programa primal, que denotaremos por (P).

Arredondamento determinístico

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

$x^* \leftarrow$ solução da relaxação linear (P)

para cada e em E **faça**

$\qquad f_e \leftarrow |\{j : e \in S_j\}|$

$f \leftarrow \max\{f_e : e \in E\}$

$I \leftarrow \{j : x^*_j \geq 1/f\}$

devolva I

Arredondamento determinístico

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

1 $\quad x^* \leftarrow$ solução da relaxação linear (P)

2 $\quad$ **para** cada e em E **faça**

3 $\qquad f_e \leftarrow |\{j : e \in S_j\}|$

4 $\quad f \leftarrow \max\{f_e : e \in E\}$

5 $\quad I \leftarrow \{j : x_j^* \geq 1/f\}$

6 $\quad$ **devolva** I

Lema: I é uma cobertura.

Prova: Como $\sum_{j:e\in S_j} x_j^* \geq 1$ tem exatamente $f_e \leq f$ termos, um deles tem que valer pelo menos $1/f$. ∎

Arredondamento determinístico

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

1 $x^* \leftarrow$ solução da relaxação linear (P)
2 **para** cada e em E **faça**
3 $\quad f_e \leftarrow |\{j : e \in S_j\}|$
4 $f \leftarrow \max\{f_e : e \in E\}$
5 $I \leftarrow \{j : x_j^* \geq 1/f\}$
6 **devolva** I

Lema: I é uma cobertura.

Teorema: ARREDDET é uma f-aproximação.

Prova: $c(I) = \sum_{j \in I} c_j \leq \sum_j (c_j\, f\, x_j^*) = f\, z^*_{\mathrm{LP}} \leq f\, \mathrm{opt}$. ∎

Arredondamento de solução dual

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

1 $y^* \leftarrow$ solução do dual da relaxação linear (P)

2 $I' \leftarrow \{j : \sum_{e \in S_j} y^*_e = c_j\}$

3 **devolva** I'

Arredondamento de solução dual

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

1 $y^* \leftarrow$ solução do dual da relaxação linear (P)

2 $I' \leftarrow \{j : \sum_{e \in S_j} y^*_e = c_j\}$

3 **devolva** I'

O consumo de tempo é polinomial.

Arredondamento de solução dual

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

1 $y^* \leftarrow$ solução do dual da relaxação linear (P)

2 $I' \leftarrow \{j : \sum_{e \in S_j} y^*_e = c_j\}$

3 **devolva** I'

O consumo de tempo é polinomial.

Lema: I é uma cobertura.

Prova feita na aula.

Arredondamento de solução dual

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

1 $y^* \leftarrow$ solução do dual da relaxação linear (P)

2 $I' \leftarrow \{j : \sum_{e \in S_j} y^*_e = c_j\}$

3 **devolva** I'

O consumo de tempo é polinomial.

Lema: I é uma cobertura.

Prova feita na aula.

Teorema: ARREDDUAL é uma f-aproximação.

Prova feita na aula.

Algoritmo primal-dual

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

para cada e em E **faça** $y_e \leftarrow 0$

$I \leftarrow \{j : \sum_{e \in S_j} y_e = c_j\}$

enquanto existe e' que não é coberto por I **faça**

$\quad \epsilon \leftarrow \min\{c_j - \sum_{e \in S_j} y_e : j$ tal que $e' \in S_j\}$

$\quad y_{e'} \leftarrow y_{e'} + \epsilon$

$\quad I \leftarrow \{j : \sum_{e \in S_j} y_e = c_j\}$

devolva I

Algoritmo primal-dual

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

para cada e em E **faça** $y_e \leftarrow 0$
$I \leftarrow \{j : \sum_{e \in S_j} y_e = c_j\}$
enquanto existe e' que não é coberto por I **faça**
    $\epsilon \leftarrow \min\{c_j - \sum_{e \in S_j} y_e : j$ tal que $e' \in S_j\}$
    $y_{e'} \leftarrow y_{e'} + \epsilon$
    $I \leftarrow \{j : \sum_{e \in S_j} y_e = c_j\}$
devolva I

Consumo de tempo polinomial, sem resolução de PL!

Algoritmo primal-dual

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

para cada e em E **faça** $y_e \leftarrow 0$
$I \leftarrow \{j : \sum_{e \in S_j} y_e = c_j\}$
enquanto existe e' que não é coberto por I **faça**
  $\epsilon \leftarrow \min\{c_j - \sum_{e \in S_j} y_e : j$ tal que $e' \in S_j\}$
  $y_{e'} \leftarrow y_{e'} + \epsilon$
  $I \leftarrow \{j : \sum_{e \in S_j} y_e = c_j\}$
devolva I

Consumo de tempo polinomial, sem resolução de PL!

Claro que I é uma cobertura.

Algoritmo primal-dual

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

1 **para** cada e em E **faça** $y_e \leftarrow 0$

2 $I \leftarrow \{j : \sum_{e \in S_j} y_e = c_j\}$

3 **enquanto** existe e' que não é coberto por I **faça**

4 $\quad \epsilon \leftarrow \min\{c_j - \sum_{e \in S_j} y_e : j$ tal que $e' \in S_j\}$

5 $\quad y_{e'} \leftarrow y_{e'} + \epsilon$

6 $\quad I \leftarrow \{j : \sum_{e \in S_j} y_e = c_j\}$

7 **devolva** I

Consumo de tempo polinomial, sem resolução de PL!

Claro que I é uma cobertura.

Teorema: PRIMALDUAL é uma f-aproximação.

A mesma prova do ARREDDUAL funciona!

Algoritmo guloso

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

$I \leftarrow \emptyset$
para $j \leftarrow 1$ **até** m **faça** $\hat{S}_j \leftarrow S_j$
enquanto I não é uma cobertura **faça**
    $k \leftarrow \arg\min\{\frac{c_j}{|\hat{S}_j|} : j$ é tal que $\hat{S}_j \neq \emptyset\}$
    $I \leftarrow I \cup \{k\}$
    para $j \leftarrow 1$ **até** m **faça**
        $\hat{S}_j \leftarrow \hat{S}_j \setminus S_k$
devolva I

Algoritmo guloso

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

$I \leftarrow \emptyset$
para $j \leftarrow 1$ **até** m **faça** $\hat{S}_j \leftarrow S_j$
enquanto I não é uma cobertura **faça**
    $k \leftarrow \arg\min\{\frac{c_j}{|\hat{S}_j|} : j$ é tal que $\hat{S}_j \neq \emptyset\}$
    $I \leftarrow I \cup \{k\}$
    para $j \leftarrow 1$ **até** m **faça**
        $\hat{S}_j \leftarrow \hat{S}_j \setminus S_k$
devolva I

Consumo de tempo polinomial, sem resolução de PL!

Algoritmo guloso

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

$I \leftarrow \emptyset$

para $j \leftarrow 1$ **até** m **faça** $\hat{S}_j \leftarrow S_j$

enquanto I não é uma cobertura **faça**

  $k \leftarrow \arg\min\{\frac{c_j}{|\hat{S}_j|}$: j é tal que $\hat{S}_j \neq \emptyset\}$

  $I \leftarrow I \cup \{k\}$

  para $j \leftarrow 1$ **até** m **faça**

    $\hat{S}_j \leftarrow \hat{S}_j \setminus S_k$

devolva I

Consumo de tempo polinomial, sem resolução de PL!

Claro que I é uma cobertura.

Algoritmo guloso

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

$I \leftarrow \emptyset$

para $j \leftarrow 1$ **até** m **faça** $\hat{S}_j \leftarrow S_j$

enquanto I não é uma cobertura **faça**

  $k \leftarrow \arg\min\{\frac{c_j}{|\hat{S}_j|} : j$ é tal que $\hat{S}_j \neq \emptyset\}$

  $I \leftarrow I \cup \{k\}$

  para $j \leftarrow 1$ **até** m **faça**

    $\hat{S}_j \leftarrow \hat{S}_j \setminus S_k$

devolva I

Consumo de tempo polinomial, sem resolução de PL!

Claro que I é uma cobertura.

Teorema: GULOSO é uma H_n-aproximação, onde $n = |E|$.
Prova feita na aula.