Algoritmos de Aproximação

Segundo Semestre de 2012

Localização de facilidades

Problema: Dados conjunto F de facilidades, conjunto C de clientes, custo f_i para cada i em F, custo $c_{ij} \geq 0$ para cada i em F e cada j em C, encontrar conjunto $F' \subseteq F$ que minimize $\sum_{i \in F'} f_i + \sum_{j \in C} \min\{c_{ij} \ : \ i \in F'\}$.

Versão métrica: $c_{ij} \leq c_{i\ell} + c_{k\ell} + c_{kj}$ para todo i, j, k, ℓ.

Localização de facilidades

Problema: Dados conjunto F de facilidades, conjunto C de clientes, custo f_i para cada i em F, custo $c_{ij} \geq 0$ para cada i em F e cada j em C, encontrar conjunto $F' \subseteq F$ que minimize $\sum_{i \in F'} f_i + \sum_{j \in C} \min\{c_{ij} \ : \ i \in F'\}$.

Versão métrica: $c_{ij} \leq c_{i\ell} + c_{k\ell} + c_{kj}$ para todo i, j, k, ℓ.

Variáveis:

- y_i: indica se a facilidade i é aberta ou não.
- x_{ij}: indica se o cliente j se conecta à facilidade i.

Encontrar x e y que

minimizem $\sum_{i \in F} f_i \, y_i + \sum_{i \in F, \ j \in C} c_{ij} \, x_{ij}$

sujeito a $\sum_{i \in F} x_{ij} = 1$ para todo j em C

$x_{ij} \leq y_i$ para todo i em F e j em C

$x_{ij}, y_i \in \{0, 1\}$ para todo i em F e j em C

Relaxação linear e seu dual

Primal (P):

minimizar $\sum_{i \in F} f_i \, y_i + \sum_{i \in F,\ j \in C} c_{ij} \, x_{ij}$

sujeito a $\sum_{i \in F} x_{ij} = 1$ para todo j em C

$x_{ij} \leq y_i$ para todo i em F e j em C

$x_{ij} \geq 0$ para todo i em F e j em C

$y_i \geq 0$ para todo i em F.

Dual (D):

maximizar $\sum_{j \in C} v_j$

sujeito a $\sum_{j \in C} w_{ij} \leq f_i$ para todo i em F

$v_j - w_{ij} \leq c_{ij}$ para todo i em F e j em C

$w_{ij} \geq 0$ para todo i em F e j em C.

- v_j: indica quanto o cliente j paga para se conectar.
- w_{ij}: indica quanto j pagaria para a facilidade i abrir.

Nova definição de vizinhança

Facilidade i é vizinha do cliente j se $v_j^* \geq c_{ij}$.

Nova definição de vizinhança

Facilidade i é vizinha do cliente j se $v_j^* \geq c_{ij}$.

Seja $N(j) = \{i \in F \ : \ i$ é vizinha de $j\}$.

Seja $N(i) = \{j \in C \ : \ i$ é vizinha de $j\}$.

Nova definição de vizinhança

Facilidade i é vizinha do cliente j se $v_j^* \geq c_{ij}$.

Seja $N(j) = \{i \in F \ : \ i$ é vizinha de $j\}$.

Seja $N(i) = \{j \in C \ : \ i$ é vizinha de $j\}$.

Cliente j contribui para facilidade i se $w_{ij}^* > 0$.

Nova definição de vizinhança

Facilidade i é vizinha do cliente j se $v_j^* \geq c_{ij}$.

Seja $N(j) = \{i \in F \ : \ i \text{ é vizinha de } j\}$.

Seja $N(i) = \{j \in C \ : \ i \text{ é vizinha de } j\}$.

Cliente j contribui para facilidade i se $w_{ij}^* > 0$.

Dado v^*, tome $w_{ij}^* = \max(0, v_j^* - c_{ij})$.
Durante o algoritmo, (v^*, w^*) será solução dual viável.

Nova definição de vizinhança

Facilidade i é vizinha do cliente j se $v_j^* \geq c_{ij}$.

Seja $N(j) = \{i \in F \ : \ i \text{ é vizinha de } j\}$.

Seja $N(i) = \{j \in C \ : \ i \text{ é vizinha de } j\}$.

Cliente j contribui para facilidade i se $w_{ij}^* > 0$.

Dado v^*, tome $w_{ij}^* = \max(0, v_j^* - c_{ij})$.
Durante o algoritmo, (v^*, w^*) será solução dual viável.

Se j contribui para i, então $j \in N(i)$,
pois $w_{ij}^* > 0$ implica que $v_j^* > c_{ij}$.

Nova definição de vizinhança

Facilidade i é vizinha do cliente j se $v_j^* \geq c_{ij}$.

Seja $N(j) = \{i \in F \ : \ i \text{ é vizinha de } j\}$.

Seja $N(i) = \{j \in C \ : \ i \text{ é vizinha de } j\}$.

Cliente j contribui para facilidade i se $w_{ij}^* > 0$.

Dado v^*, tome $w_{ij}^* = \max(0, v_j^* - c_{ij})$.
Durante o algoritmo, (v^*, w^*) será solução dual viável.

Se j contribui para i, então $j \in N(i)$,
pois $w_{ij}^* > 0$ implica que $v_j^* > c_{ij}$.

Ademais, se $j \in N(i)$, então $v_j^* = c_{ij} + w_{ij}^*$.

Facilidades justas

Facilidade i é vizinha do cliente j se $v_j^* \geq c_{ij}$.

Cliente j contribui para facilidade i se $w_{ij}^* > 0$.

Dado v^*, tome $w_{ij}^* = \max(0, v_j^* - c_{ij})$.

Se j contribui para i, então j é vizinho de i.
Se j é vizinho de i, então $v_j^* = c_{ij} + w_{ij}^*$.

Facilidades justas

Facilidade i é vizinha do cliente j se $v_j^* \geq c_{ij}$.

Cliente j contribui para facilidade i se $w_{ij}^* > 0$.

Dado v^*, tome $w_{ij}^* = \max(0, v_j^* - c_{ij})$.

Se j contribui para i, então j é vizinho de i.
Se j é vizinho de i, então $v_j^* = c_{ij} + w_{ij}^*$.

Solução viável (v^*, w^*) é maximal se

não podemos aumentar v_j^* de qq valor positivo e obter novos w_{ij}^* que formem uma solução viável.

Facilidades justas

Facilidade i é vizinha do cliente j se $v_j^* \geq c_{ij}$.

Cliente j contribui para facilidade i se $w_{ij}^* > 0$.

Dado v^*, tome $w_{ij}^* = \max(0, v_j^* - c_{ij})$.

Se j contribui para i, então j é vizinho de i.
Se j é vizinho de i, então $v_j^* = c_{ij} + w_{ij}^*$.

Solução viável (v^*, w^*) é maximal se

não podemos aumentar v_j^* de qq valor positivo e obter novos w_{ij}^* que formem uma solução viável.

Seja $T = \{i \in F : \sum_{j \in C} w_{ij}^* = f_i\}$.

Se (v^*, w^*) é maximal,
então todo cliente é vizinho de alguma facilidade em T.

Facilidades justas

Facilidade i é vizinha do cliente j se $v_j^* \geq c_{ij}$.

Seja $N(j) = \{i \in F \ : \ i$ é vizinha de $j\}$.
Seja $N(i) = \{j \in C \ : \ i$ é vizinha de $j\}$.

Seja $T = \{i \in F : \sum_{j \in C} w_{ij}^* = f_i\}$.

Se (v^*, w^*) é maximal,
então todo cliente é vizinho de alguma facilidade em T.

Facilidades justas

Facilidade i é vizinha do cliente j se $v_j^* \geq c_{ij}$.

Seja $N(j) = \{i \in F \ : \ i$ é vizinha de $j\}$.
Seja $N(i) = \{j \in C \ : \ i$ é vizinha de $j\}$.

Seja $T = \{i \in F : \sum_{j \in C} w_{ij}^* = f_i\}$.

Se (v^*, w^*) é maximal,
então todo cliente é vizinho de alguma facilidade em T.

Para cada i em T,

$$f_i + \sum_{j \in N(i)} c_{ij} \ = \ \sum_{j \in N(i)} (w_{ij}^* + c_{ij}) \ = \ \sum_{j \in N(i)} v_j^*.$$

Facilidades justas

Facilidade i é vizinha do cliente j se $v_j^* \geq c_{ij}$.

Seja $N(j) = \{i \in F \ : \ i$ é vizinha de $j\}$.
Seja $N(i) = \{j \in C \ : \ i$ é vizinha de $j\}$.

Seja $T = \{i \in F : \sum_{j \in C} w_{ij}^* = f_i\}$.

Se (v^*, w^*) é maximal,
então todo cliente é vizinho de alguma facilidade em T.

Para cada i em T,

$$f_i + \sum_{j \in N(i)} c_{ij} \ = \ \sum_{j \in N(i)} (w_{ij}^* + c_{ij}) \ = \ \sum_{j \in N(i)} v_j^*.$$

Problema: alguns j estão em mais de um $N(i)$.

Facilidades justas

Facilidade i é vizinha do cliente j se $v_j^* \geq c_{ij}$.

Seja $N(j) = \{i \in F \ : \ i$ é vizinha de $j\}$.
Seja $N(i) = \{j \in C \ : \ i$ é vizinha de $j\}$.

Seja $T = \{i \in F : \sum_{j \in C} w_{ij}^* = f_i\}$.

Se (v^*, w^*) é maximal,
então todo cliente é vizinho de alguma facilidade em T.

Para cada i em T,

$$f_i + \sum_{j \in N(i)} c_{ij} \ = \ \sum_{j \in N(i)} (w_{ij}^* + c_{ij}) \ = \ \sum_{j \in N(i)} v_j^*.$$

Problema: alguns j estão em mais de um $N(i)$.

Vamos escolher $T' \subseteq T$ para abrir.

Ideia do algoritmo primal-dual

Comece com $S = C$ e $T = \emptyset$ ($v^* = w^* = 0$).

Ideia do algoritmo primal-dual

Comece com $S = C$ e $T = \emptyset$ ($v^* = w^* = 0$).

Aumente v_j^* uniformemente para $j \in S$ até algum $v_j^* = c_{ij}$, daí comece a aumentar simultaneamente w_{ij}^*.

Ideia do algoritmo primal-dual

Comece com $S = C$ e $T = \emptyset$ ($v^* = w^* = 0$).

Aumente v_j^* uniformemente para $j \in S$ até algum $v_j^* = c_{ij}$, daí comece a aumentar simultaneamente w_{ij}^*.

Pare quando

- j se torna vizinho de uma facilidade em T ou
- desigualdade de (D) fica justa para alguma facilidade i.

Ideia do algoritmo primal-dual

Comece com $S = C$ e $T = \emptyset$ ($v^* = w^* = 0$).

Aumente v_j^* uniformemente para $j \in S$ até algum $v_j^* = c_{ij}$, daí comece a aumentar simultaneamente w_{ij}^*.

Pare quando

- j se torna vizinho de uma facilidade em T ou
- desigualdade de (D) fica justa para alguma facilidade i.

Em cada caso,

- remova j de S ou
- inclua i em T e remova $N(i)$ de S.

Ideia do algoritmo primal-dual

Comece com $S = C$ e $T = \emptyset$ ($v^* = w^* = 0$).

Aumente v_j^* uniformemente para $j \in S$ até algum $v_j^* = c_{ij}$, daí comece a aumentar simultaneamente w_{ij}^*.

Pare quando

- j se torna vizinho de uma facilidade em T ou
- desigualdade de (D) fica justa para alguma facilidade i.

Em cada caso,

- remova j de S ou
- inclua i em T e remova $N(i)$ de S.

Quando $S = \emptyset$ e todo cliente é vizinho de algum i em T, construa T' gulosamente (basicamente $N(i)$ disjuntos).

Algoritmo

Primal-Dual (F, C, f, c)

1 $v \leftarrow 0$ $w \leftarrow 0$
2 $S \leftarrow C$ $T \leftarrow \emptyset$
3 **enquanto** $S \neq \emptyset$ **faça**
4 aumente v_j para $j \in S$ e w_{ij} para $i \in N(j)$ e $j \in S$.
5 **se** $j \in S$ é vizinho de $i \in T$ **então** $S \leftarrow S \setminus \{j\}$
6 **se** a desigualdade do dual para $i \notin T$ é justa
7 **então** $T \leftarrow T \cup \{i\}$ $S \leftarrow S \setminus N(i)$

Algoritmo

Primal-Dual (F, C, f, c)

1 $v \leftarrow 0$ $w \leftarrow 0$
2 $S \leftarrow C$ $T \leftarrow \emptyset$
3 **enquanto** $S \neq \emptyset$ **faça**
4 aumente v_j para $j \in S$ e w_{ij} para $i \in N(j)$ e $j \in S$.
5 **se** $j \in S$ é vizinho de $i \in T$ **então** $S \leftarrow S \setminus \{j\}$
6 **se** a desigualdade do dual para $i \notin T$ é justa
7 **então** $T \leftarrow T \cup \{i\}$ $S \leftarrow S \setminus N(i)$
8 $T' \leftarrow \emptyset$
9 **enquanto** $T \neq \emptyset$ **faça**
10 seja i um elemento de T
11 $T' \leftarrow T' \cup \{i\}$
12 $T \leftarrow T \setminus \{h \in T : \exists j \in C, w_{ij} > 0, w_{hj} > 0\}$
13 **devolva** T'

Razão de aproximação

Teorema: Primal-Dual é uma 3-aproximação.

Razão de aproximação

Teorema: Primal-Dual é uma 3-aproximação.

Lema: Se j não é vizinho de facilidade em T', então existe i em T' tq $c_{ij} \leq 3\, v_j$.

Razão de aproximação

Teorema: Primal-Dual é uma 3-aproximação.

Lema: Se j não é vizinho de facilidade em T', então existe i em T' tq $c_{ij} \leq 3\, v_j$.

Prova do lema depois.

Razão de aproximação

Teorema: Primal-Dual é uma 3-aproximação.

Lema: Se j não é vizinho de facilidade em T', então existe i em T' tq $c_{ij} \leq 3\, v_j$.

Prova do lema depois.

Prova do teorema:
$A(i) \subseteq N(i)$: clientes atribuídos a cada $i \in T'$.

Razão de aproximação

Teorema: Primal-Dual é uma 3-aproximação.

Lema: Se j não é vizinho de facilidade em T', então existe i em T' tq $c_{ij} \leq 3\, v_j$.

Prova do lema depois.

Prova do teorema:
$A(i) \subseteq N(i)$: clientes atribuídos a cada $i \in T'$.

$$\sum_{i \in T'} \left(f_i + \sum_{j \in A(i)} c_{ij} \right) = \sum_{i \in T'} \sum_{j \in A(i)} (w_{ij} + c_{ij}) = \sum_{i \in T'} \sum_{j \in A(i)} v_j,$$

Razão de aproximação

Teorema: Primal-Dual é uma 3-aproximação.

Lema: Se j não é vizinho de facilidade em T', então existe i em T' tq $c_{ij} \leq 3\, v_j$.

Prova do lema depois.

Prova do teorema:
$A(i) \subseteq N(i)$: clientes atribuídos a cada $i \in T'$.

$$\sum_{i \in T'} \left(f_i + \sum_{j \in A(i)} c_{ij} \right) = \sum_{i \in T'} \sum_{j \in A(i)} (w_{ij} + c_{ij}) = \sum_{i \in T'} \sum_{j \in A(i)} v_j,$$

pois se $i \in T'$, então $\sum_{j \in A(i)} w_{ij} = f_i$ e
se $w_{ij} > 0$, então $j \in A(i)$.

Razão de aproximação

Lema: Se j não é vizinho de facilidade em T', então existe i em T' tq $c_{ij} \leq 3\, v_j$.

Teorema: Primal-Dual é uma 3-aproximação.

Prova: $A(i)$: clientes atribuídos a cada $i \in T'$.

$$\sum_{i \in T'} \left(f_i + \sum_{j \in A(i)} c_{ij} \right) \;=\; \sum_{i \in T'} \sum_{j \in A(i)} v_j .$$

Razão de aproximação

Lema: Se j não é vizinho de facilidade em T', então existe i em T' tq $c_{ij} \leq 3\, v_j$.

Teorema: Primal-Dual é uma 3-aproximação.

Prova: $A(i)$: clientes atribuídos a cada $i \in T'$.

$$\sum_{i \in T'} \left(f_i + \sum_{j \in A(i)} c_{ij} \right) \;=\; \sum_{i \in T'} \sum_{j \in A(i)} v_j .$$

Z: clientes que não são vizinhos de facilidade em T'.

Razão de aproximação

Lema: Se j não é vizinho de facilidade em T', então existe i em T' tq $c_{ij} \leq 3\, v_j$.

Teorema: Primal-Dual é uma 3-aproximação.

Prova: $A(i)$: clientes atribuídos a cada $i \in T'$.

$$\sum_{i \in T'} \left(f_i + \sum_{j \in A(i)} c_{ij} \right) \;=\; \sum_{i \in T'} \sum_{j \in A(i)} v_j .$$

Z: clientes que não são vizinhos de facilidade em T'.

Pelo lema, custo de atribuir $j \in Z$ à facilidade de T' é $\leq 3\, v_j$.

Razão de aproximação

Lema: Se j não é vizinho de facilidade em T', então existe i em T' tq $c_{ij} \leq 3\, v_j$.

Teorema: Primal-Dual é uma 3-aproximação.

Prova: $A(i)$: clientes atribuídos a cada $i \in T'$.

$$\sum_{i \in T'} \left(f_i + \sum_{j \in A(i)} c_{ij} \right) \;=\; \sum_{i \in T'} \sum_{j \in A(i)} v_j .$$

Z: clientes que não são vizinhos de facilidade em T'.

Pelo lema, custo de atribuir $j \in Z$ à facilidade de T' é $\leq 3\, v_j$.

$$\begin{aligned} \sum_{i \in T'} \left(f_i + \sum_{j \in A(i)} c_{ij} \right) + 3 \sum_{j \in Z} v_j \;&=\; \sum_{i \in T'} \sum_{j \in A(i)} v_j + 3 \sum_{j \in Z} v_j \\ &\leq\; 3 \sum_{j \in C} v_j \;\leq\; 3\,\mathrm{OPT}. \end{aligned}$$

Razão de aproximação

Lema: Se j não é vizinho de facilidade em T', então existe i em T' tq $c_{ij} \leq 3\, v_j$.

Razão de aproximação

Lema: Se j não é vizinho de facilidade em T', então existe i em T' tq $c_{ij} \leq 3\, v_j$.

Prova:

Razão de aproximação

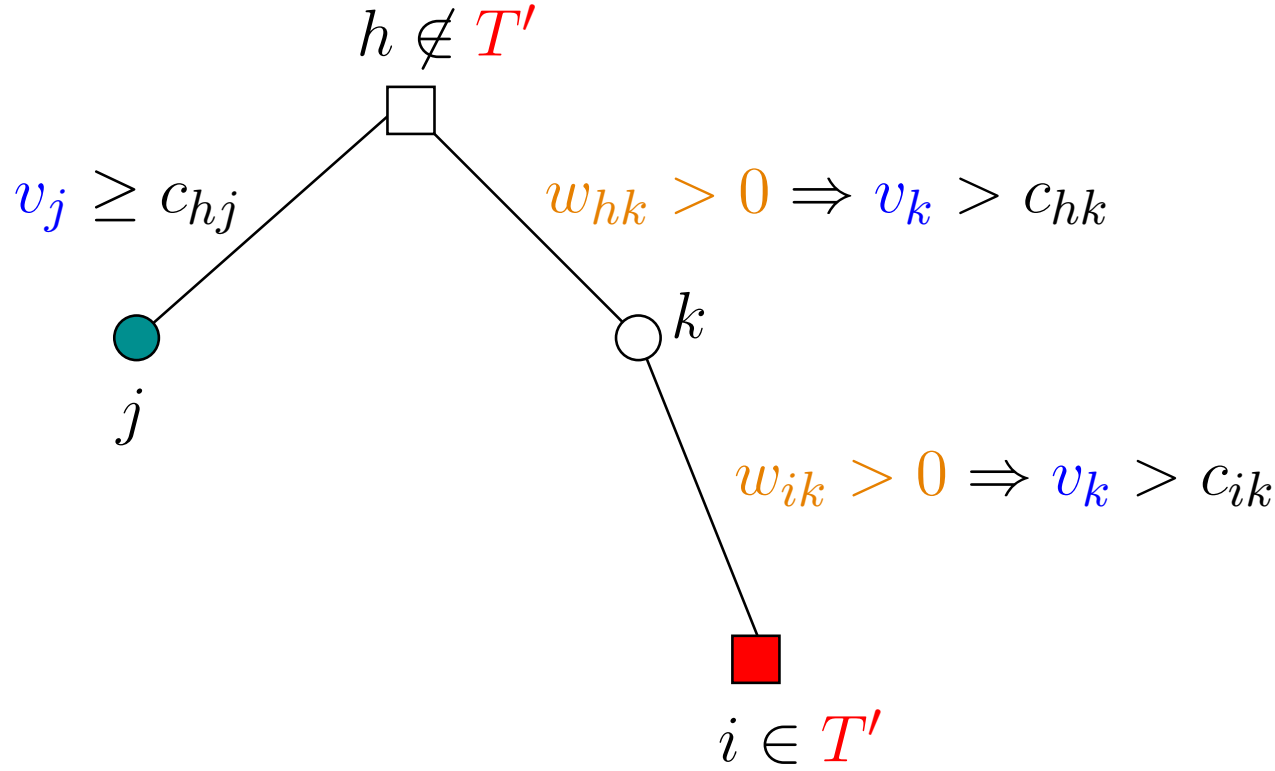
Lema: Se j não é vizinho de facilidade em T', então existe i em T' tq $c_{ij} \leq 3\, v_j$.

Prova: Quando v_j parou de crescer, $j \in N(h)$ para algum $h \in T \setminus T'$.

Razão de aproximação

Lema: Se j não é vizinho de facilidade em T', então existe i em T' tq $c_{ij} \leq 3\, v_j$.

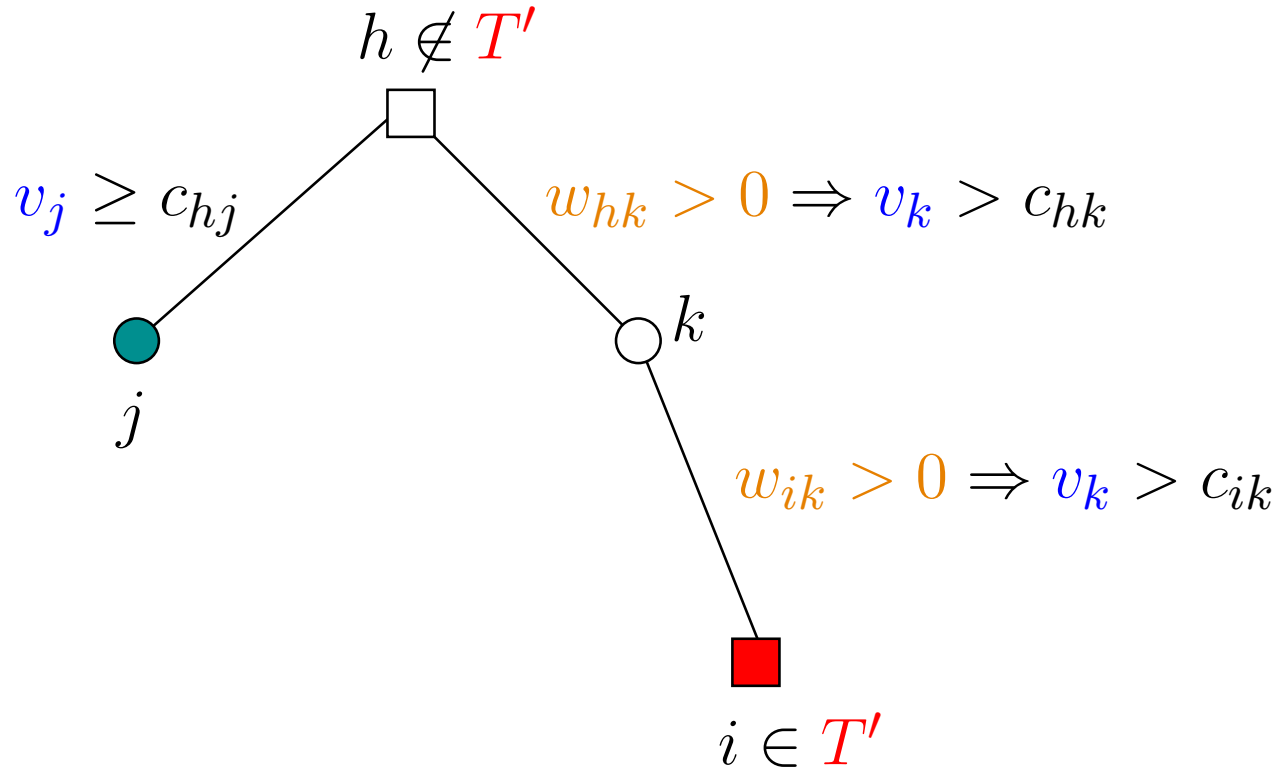
Prova: Quando v_j parou de crescer, $j \in N(h)$ para algum $h \in T \setminus T'$.
Existe cliente k vizinho de h e $i \in T'$.



Razão de aproximação

Lema: Se j não é vizinho de facilidade em T', então existe i em T' tq $c_{ij} \leq 3\, v_j$.

Prova: Quando v_j parou de crescer, $j \in N(h)$ para algum $h \in T \setminus T'$.
Existe cliente k vizinho de h e $i \in T'$, e $v_k \leq v_j$.



Inaproximabilidade

- Reduções de problemas NP-completos
- Reduções que preservam aproximações
- Reduções a partir do PCP
- Reduções do Label Cover
- Reduções do Unique Games

Inaproximabilidade

- Reduções de problemas NP-completos (já vimos)
- Reduções que preservam aproximações (hoje)
- Reduções a partir do PCP
- Reduções do Label Cover
- Reduções do Unique Games

Inaproximabilidade

- Reduções de problemas NP-completos (já vimos)
- Reduções que preservam aproximações (hoje)
- Reduções a partir do PCP (quarta que vem)
- Reduções do Label Cover
- Reduções do Unique Games (quarta que vem)

Reduções de problemas NP-completos

- Redução do 2-Partição para o Bin Packing
  (α-aproximação com $\alpha < \frac{3}{2}$ implica que P=NP.)

Reduções de problemas NP-completos

- Redução do 2-Partição para o Bin Packing
  (α-aproximação com $\alpha < \frac{3}{2}$ implica que P=NP.)

- Redução do Circuito Hamiltoniano para o TSP
  (α-aproximação com $\alpha = \mathrm{O}(2^n)$ implica que P=NP.)

Reduções de problemas NP-completos

- Redução do 2-Partição para o Bin Packing
  (α-aproximação com $\alpha < \frac{3}{2}$ implica que P=NP.)
- Redução do Circuito Hamiltoniano para o TSP
  (α-aproximação com $\alpha = \mathrm{O}(2^n)$ implica que P=NP.)
- Redução do Conjunto Dominante Mínimo para o Problema dos 2-Centros
  (α-aproximação com $\alpha < 2$ implica que P=NP.)

Reduções que preservam aproximação

- Redução do MAX E3SAT para o MAX 2SAT

Reduções que preservam aproximação

- Redução do MAX E3SAT para o MAX 2SAT
- Redução do MAX E3SAT para o Conjunto Independente

Reduções que preservam aproximação

- Redução do MAX E3SAT para o MAX 2SAT
- Redução do MAX E3SAT para o Conjunto Independente
- Redução do Conjunto Independente para o próprio

MAX SAT

Variáveis booleanas $x_1, \ldots, x_n$.

Literal: variável x_i ou sua negação $\bar{x}_i$

Cláusula: conjunto de literais nas variáveis $x_1, \ldots, x_n$.

MAX SAT

Variáveis booleanas $x_1, \ldots, x_n$.

Literal: variável x_i ou sua negação $\bar{x}_i$

Cláusula: conjunto de literais nas variáveis $x_1, \ldots, x_n$.

MAX SAT: Dadas cláusulas $C_1, \ldots, C_m$ nas variáveis $x_1, \ldots, x_n$, e pesos $w_j \geq 0$ para cada C_j, encontrar uma atribuição para as variáveis que maximize o peso das cláusulas satisfeitas.

MAX SAT

Variáveis booleanas $x_1, \ldots, x_n$.

Literal: variável x_i ou sua negação $\bar{x}_i$

Cláusula: conjunto de literais nas variáveis $x_1, \ldots, x_n$.

MAX SAT: Dadas cláusulas $C_1, \ldots, C_m$ nas variáveis $x_1, \ldots, x_n$, e pesos $w_j \geq 0$ para cada C_j, encontrar uma atribuição para as variáveis que maximize o peso das cláusulas satisfeitas.

MAX E3SAT: as cláusulas têm exatamente três literais.

MAX SAT

Variáveis booleanas $x_1, \ldots, x_n$.

Literal: variável x_i ou sua negação $\bar{x}_i$

Cláusula: conjunto de literais nas variáveis $x_1, \ldots, x_n$.

MAX SAT: Dadas cláusulas $C_1, \ldots, C_m$ nas variáveis $x_1, \ldots, x_n$, e pesos $w_j \geq 0$ para cada C_j, encontrar uma atribuição para as variáveis que maximize o peso das cláusulas satisfeitas.

MAX E3SAT: as cláusulas têm exatamente três literais.

MAX 2SAT: as cláusulas têm no máximo dois literais.

Redução MAX E3SAT p/ MAX 2SAT

Troque a cláusula $C_j = (x_1 \vee x_2 \vee x_3)$ pelas 10 cláusulas

Redução MAX E3SAT p/ MAX 2SAT

Troque a cláusula $C_j = (x_1 \vee x_2 \vee x_3)$ pelas 10 cláusulas x_1, x_2, x_3, $\bar{x}_1 \vee \bar{x}_2$, $\bar{x}_2 \vee \bar{x}_3$, $\bar{x}_1 \vee \bar{x}_3$, y_j, $x_1 \vee \bar{y}_j$, $x_2 \vee \bar{y}_j$, $x_3 \vee \bar{y}_j$.

y_j: variável nova.

Redução MAX E3SAT p/ MAX 2SAT

Troque a cláusula $C_j = (x_1 \vee x_2 \vee x_3)$ pelas 10 cláusulas
x_1, x_2, x_3, $\bar{x}_1 \vee \bar{x}_2$, $\bar{x}_2 \vee \bar{x}_3$, $\bar{x}_1 \vee \bar{x}_3$, y_j, $x_1 \vee \bar{y}_j$, $x_2 \vee \bar{y}_j$, $x_3 \vee \bar{y}_j$.

y_j: variável nova.

Atribuição que satisfaz C_j corresponde à
atribuição que satisfaz 7 das 10 cláusulas acima.

Redução MAX E3SAT p/ MAX 2SAT

Troque a cláusula $C_j = (x_1 \vee x_2 \vee x_3)$ pelas 10 cláusulas
x_1, x_2, x_3, $\bar{x}_1 \vee \bar{x}_2$, $\bar{x}_2 \vee \bar{x}_3$, $\bar{x}_1 \vee \bar{x}_3$, y_j, $x_1 \vee \bar{y}_j$, $x_2 \vee \bar{y}_j$, $x_3 \vee \bar{y}_j$.

y_j: variável nova.

Atribuição que satisfaz C_j corresponde à
atribuição que satisfaz 7 das 10 cláusulas acima.

Atribuição que não satisfaz C_j corresponde à
atribuição que satisfaz 6 das 10 cláusulas acima.

Redução MAX E3SAT p/ MAX 2SAT

Troque a cláusula $C_j = (x_1 \vee x_2 \vee x_3)$ pelas 10 cláusulas
x_1, x_2, x_3, $\bar{x}_1 \vee \bar{x}_2$, $\bar{x}_2 \vee \bar{x}_3$, $\bar{x}_1 \vee \bar{x}_3$, y_j, $x_1 \vee \bar{y}_j$, $x_2 \vee \bar{y}_j$, $x_3 \vee \bar{y}_j$.

y_j: variável nova.

Atribuição que satisfaz C_j corresponde à
atribuição que satisfaz 7 das 10 cláusulas acima.

Atribuição que não satisfaz C_j corresponde à
atribuição que satisfaz 6 das 10 cláusulas acima.

m: número de cláusulas na instância I do MAX E3SAT.

$k^* = \mathrm{OPT}(I)$.

Redução MAX E3SAT p/ MAX 2SAT

Troque a cláusula $C_j = (x_1 \vee x_2 \vee x_3)$ pelas 10 cláusulas
x_1, x_2, x_3, $\bar{x}_1 \vee \bar{x}_2$, $\bar{x}_2 \vee \bar{x}_3$, $\bar{x}_1 \vee \bar{x}_3$, y_j, $x_1 \vee \bar{y}_j$, $x_2 \vee \bar{y}_j$, $x_3 \vee \bar{y}_j$.

y_j: variável nova.

Atribuição que satisfaz C_j corresponde à
atribuição que satisfaz 7 das 10 cláusulas acima.

Atribuição que não satisfaz C_j corresponde à
atribuição que satisfaz 6 das 10 cláusulas acima.

m: número de cláusulas na instância I do MAX E3SAT.

$k^* = \mathrm{OPT}(I)$.

Então $\mathrm{OPT}(I') = 7k^* + 6(m - k^*) = k^* + 6m$,
onde I' é a instância correspondente do MAX 2SAT.

Redução MAX E3SAT p/ MAX 2SAT

Se A é uma α-aproximação para o MAX 2SAT,
que produz uma atribuição que satisfaz 7 cláusulas das 10,
para $\bar{k}$ dos C_j, a atribuição derivada satisfaz $\bar{k}$ dos C_j.

Redução MAX E3SAT p/ MAX 2SAT

Se A é uma α-aproximação para o MAX 2SAT,
que produz uma atribuição que satisfaz 7 cláusulas das 10,
para $\bar{k}$ dos C_j, a atribuição derivada satisfaz $\bar{k}$ dos C_j.

$$\begin{aligned} \mathrm{OPT}(I) - \bar{k} &= k^* - \bar{k} = 7k^* + 6(m - k^*) - (7\bar{k} + 6(m - \bar{k})) \\ &\leq \mathrm{OPT}(I') - \alpha\,\mathrm{OPT}(I') = (1 - \alpha)\mathrm{OPT}(I'). \end{aligned}$$

Redução MAX E3SAT p/ MAX 2SAT

Se A é uma α-aproximação para o MAX 2SAT,
que produz uma atribuição que satisfaz 7 cláusulas das 10,
para $\bar{k}$ dos C_j, a atribuição derivada satisfaz $\bar{k}$ dos C_j.

$$\begin{aligned} \mathrm{OPT}(I) - \bar{k} &= k^* - \bar{k} = 7k^* + 6(m - k^*) - (7\bar{k} + 6(m - \bar{k})) \\ &\leq \mathrm{OPT}(I') - \alpha\,\mathrm{OPT}(I') = (1-\alpha)\mathrm{OPT}(I'). \end{aligned}$$

Logo $\bar{k} \geq \mathrm{OPT}(I) - (1-\alpha)\mathrm{OPT}(I')$.

Redução MAX E3SAT p/ MAX 2SAT

Se A é uma α-aproximação para o MAX 2SAT,
que produz uma atribuição que satisfaz 7 cláusulas das 10,
para $\bar{k}$ dos C_j, a atribuição derivada satisfaz $\bar{k}$ dos C_j.

$$\begin{aligned} \mathrm{OPT}(I) - \bar{k} &= k^* - \bar{k} = 7k^* + 6(m - k^*) - (7\bar{k} + 6(m - \bar{k})) \\ &\leq \mathrm{OPT}(I') - \alpha\,\mathrm{OPT}(I') = (1-\alpha)\mathrm{OPT}(I'). \end{aligned}$$

Logo $\bar{k} \geq \mathrm{OPT}(I) - (1-\alpha)\mathrm{OPT}(I')$.

Para deduzir uma razão de aproximação para o MAX E3SAT, precisamos relacionar $\mathrm{OPT}(I')$ e $\mathrm{OPT}(I)$.

Redução MAX E3SAT p/ MAX 2SAT

Se A é uma α-aproximação para o MAX 2SAT,
que produz uma atribuição que satisfaz 7 cláusulas das 10,
para $\bar{k}$ dos C_j, tal atribuição satisfaz $\bar{k}$ dos C_j.

Vale que $\bar{k} \geq \mathrm{OPT}(I) - (1-\alpha)\mathrm{OPT}(I')$.

Redução MAX E3SAT p/ MAX 2SAT

Se A é uma α-aproximação para o MAX 2SAT,
que produz uma atribuição que satisfaz 7 cláusulas das 10,
para $\bar{k}$ dos C_j, tal atribuição satisfaz $\bar{k}$ dos C_j.

Vale que $\bar{k} \geq \mathrm{OPT}(I) - (1-\alpha)\mathrm{OPT}(I')$.

Sabemos que $\mathrm{OPT}(I) = k^* \geq \frac{7}{8}\, m$.

Redução MAX E3SAT p/ MAX 2SAT

Se A é uma α-aproximação para o MAX 2SAT,
que produz uma atribuição que satisfaz 7 cláusulas das 10,
para $\bar{k}$ dos C_j, tal atribuição satisfaz $\bar{k}$ dos C_j.

Vale que $\bar{k} \geq \mathrm{OPT}(I) - (1-\alpha)\mathrm{OPT}(I')$.

Sabemos que $\mathrm{OPT}(I) = k^* \geq \frac{7}{8}\, m$.

Como $\mathrm{OPT}(I') = 7k^* + 6(m - k^*) = k^* + 6m$,
temos que $\mathrm{OPT}(I') \leq k^* + 6\frac{8}{7}k^* = \frac{55}{7}\mathrm{OPT}(I)$.

Redução MAX E3SAT p/ MAX 2SAT

Se A é uma α-aproximação para o MAX 2SAT,
que produz uma atribuição que satisfaz 7 cláusulas das 10,
para $\bar{k}$ dos C_j, tal atribuição satisfaz $\bar{k}$ dos C_j.

Vale que $\bar{k} \geq \mathrm{OPT}(I) - (1-\alpha)\mathrm{OPT}(I')$.

Sabemos que $\mathrm{OPT}(I) = k^* \geq \frac{7}{8}\, m$.

Como $\mathrm{OPT}(I') = 7k^* + 6(m - k^*) = k^* + 6m$,
temos que $\mathrm{OPT}(I') \leq k^* + 6\frac{8}{7}k^* = \frac{55}{7}\mathrm{OPT}(I)$.

Logo $\bar{k} \geq \mathrm{OPT}(I) - (1-\alpha)\frac{55}{7}\mathrm{OPT}(I) = \left(\frac{55}{7}\alpha - \frac{48}{7}\right)\mathrm{OPT}(I)$.

Redução MAX E3SAT p/ MAX 2SAT

Se A é uma α-aproximação para o MAX 2SAT,
que produz uma atribuição que satisfaz 7 cláusulas das 10,
para $\bar{k}$ dos C_j, tal atribuição satisfaz $\bar{k}$ dos C_j.

Vale que $\bar{k} \geq \mathrm{OPT}(I) - (1-\alpha)\mathrm{OPT}(I')$.

Sabemos que $\mathrm{OPT}(I) = k^* \geq \frac{7}{8}\, m$.

Como $\mathrm{OPT}(I') = 7k^* + 6(m - k^*) = k^* + 6m$,
temos que $\mathrm{OPT}(I') \leq k^* + 6\frac{8}{7}k^* = \frac{55}{7}\mathrm{OPT}(I)$.

Logo $\bar{k} \geq \mathrm{OPT}(I) - (1-\alpha)\frac{55}{7}\mathrm{OPT}(I) = \left(\frac{55}{7}\alpha - \frac{48}{7}\right)\mathrm{OPT}(I)$.

α-aprox. MAX 2SAT $\Longrightarrow$ $\left(\frac{55}{7}\alpha - \frac{48}{7}\right)$-aprox. MAX E3SAT

Redução MAX E3SAT p/ MAX 2SAT

α-aprox. MAX 2SAT $\Longrightarrow$ $(\frac{55}{7}\alpha - \frac{48}{7})$-aprox. MAX E3SAT

Redução MAX E3SAT p/ MAX 2SAT

α-aprox. MAX 2SAT $\Longrightarrow$ $(\frac{55}{7}\alpha - \frac{48}{7})$-aprox. MAX E3SAT

Lembre-se do seguinte

Teorema: A menos que P $=$ NP, não existe uma α-aproximação para MAX E3SAT com $\alpha > \frac{7}{8}$.

(Esse vem do PCP.)

Redução MAX E3SAT p/ MAX 2SAT

α-aprox. MAX 2SAT $\Longrightarrow$ $(\frac{55}{7}\alpha - \frac{48}{7})$-aprox. MAX E3SAT

Lembre-se do seguinte

Teorema: A menos que P $=$ NP, não existe uma α-aproximação para MAX E3SAT com $\alpha > \frac{7}{8}$.

(Esse vem do PCP.)

Então

Teorema: A menos que P $=$ NP, não existe uma α-aproximação para MAX 2SAT com $\alpha > \frac{433}{440} \approx 0{,}984$.

L-redução

Dados dois problemas de otimização Π e Π',
temos uma L-redução de Π para Π' se, para um par de números a e b positivos,

L-redução

Dados dois problemas de otimização Π e Π', temos uma L-redução de Π para Π' se, para um par de números a e b positivos,

- para cada instância I de Π, podemos calcular em tempo polinomial uma instância I' de Π';

L-redução

Dados dois problemas de otimização Π e Π', temos uma L-redução de Π para Π' se, para um par de números a e b positivos,

- para cada instância I de Π, podemos calcular em tempo polinomial uma instância I' de Π';
- $\mathrm{OPT}(I') \leq a\,\mathrm{OPT}(I')$;

L-redução

Dados dois problemas de otimização Π e Π',
temos uma L-redução de Π para Π' se, para um par de números a e b positivos,

- para cada instância I de Π, podemos calcular em tempo polinomial uma instância I' de Π';
- $\mathrm{OPT}(I') \leq a\,\mathrm{OPT}(I')$;
- dada solução de valor v' para I', podemos calcular em tempo polinomial solução para I' de valor v tal que

$$|\mathrm{OPT}(I) - v| \;\leq\; b|\mathrm{OPT}(I') - v'|.$$

L-redução

Dados dois problemas de otimização Π e Π', temos uma L-redução de Π para Π' se, para um par de números a e b positivos,

- para cada instância I de Π, podemos calcular em tempo polinomial uma instância I' de Π';
- $\mathrm{OPT}(I') \leq a\,\mathrm{OPT}(I')$;
- dada solução de valor v' para I', podemos calcular em tempo polinomial solução para I' de valor v tal que

$$|\mathrm{OPT}(I) - v| \;\leq\; b|\mathrm{OPT}(I') - v'|.$$

Para problemas de maximização,

$$\begin{aligned} v \;&\geq\; \mathrm{OPT}(I) - b(\mathrm{OPT}(I') - v') \;\geq\; \mathrm{OPT}(I) - b(1-\alpha)\mathrm{OPT}(I') \\ &\geq\; \mathrm{OPT}(I)(1 - ab(1-\alpha)). \end{aligned}$$

L-redução

Dados dois problemas de otimização Π e Π',
temos uma L-redução de Π para Π' se, para um par de números a e b positivos,

- para cada instância I de Π, podemos calcular em tempo polinomial uma instância I' de Π';
- $\mathrm{OPT}(I') \leq a\,\mathrm{OPT}(I')$;
- dada solução de valor v' para I', podemos calcular em tempo polinomial solução para I' de valor v tal que

$$|\mathrm{OPT}(I) - v| \;\leq\; b|\mathrm{OPT}(I') - v'|.$$

Para problemas de maximização, $v \geq \mathrm{OPT}(I)(1 - ab(1-\alpha))$.

L-redução

Dados dois problemas de otimização Π e Π',
temos uma L-redução de Π para Π' se, para um par de números a e b positivos,

- para cada instância I de Π, podemos calcular em tempo polinomial uma instância I' de Π';
- $\mathrm{OPT}(I') \leq a\,\mathrm{OPT}(I')$;
- dada solução de valor v' para I', podemos calcular em tempo polinomial solução para I' de valor v tal que

$$|\mathrm{OPT}(I) - v| \;\leq\; b|\mathrm{OPT}(I') - v'|.$$

Para problemas de maximização, $v \geq \mathrm{OPT}(I)(1 - ab(1-\alpha))$.

Quanto menor ab, melhor o resultado.

L-redução

Dados dois problemas de otimização Π e Π',
temos uma L-redução de Π para Π' se, para um par de números a e b positivos,

- para cada instância I de Π, podemos calcular em tempo polinomial uma instância I' de Π';
- $\mathrm{OPT}(I') \leq a\,\mathrm{OPT}(I')$;
- dada solução de valor v' para I', podemos calcular em tempo polinomial solução para I' de valor v tal que

$$|\mathrm{OPT}(I) - v| \;\leq\; b|\mathrm{OPT}(I') - v'|.$$

Para problemas de maximização, $v \geq \mathrm{OPT}(I)(1 - ab(1-\alpha))$.
Para problemas de minimização, $v \leq \mathrm{OPT}(I)(1 + ab(1-\alpha))$.

Quanto menor ab, melhor o resultado.

L-redução

Dados dois problemas de otimização Π e Π',
temos uma L-redução de Π para Π' se, para um par de números a e b positivos,

- para cada instância I de Π, podemos calcular em tempo polinomial uma instância I' de Π';
- $\mathrm{OPT}(I') \leq a\,\mathrm{OPT}(I)$;
- dada solução de valor v' para I', podemos calcular em tempo polinomial solução para I' de valor v tal que

$$|\mathrm{OPT}(I) - v| \;\leq\; b|\mathrm{OPT}(I') - v'|.$$

Quanto menor ab, melhor o resultado.

L-redução

Dados dois problemas de otimização Π e Π',
temos uma L-redução de Π para Π' se, para um par de números a e b positivos,

- para cada instância I de Π, podemos calcular em tempo polinomial uma instância I' de Π';
- $\mathrm{OPT}(I') \leq a\,\mathrm{OPT}(I)$;
- dada solução de valor v' para I', podemos calcular em tempo polinomial solução para I' de valor v tal que

$$|\mathrm{OPT}(I) - v| \;\leq\; b|\mathrm{OPT}(I') - v'|.$$

Quanto menor ab, melhor o resultado.

Redução anterior: $a = \frac{55}{7}$ $\quad(\mathrm{OPT}(I') \leq \frac{55}{7}\mathrm{OPT}(I))$
e $b = 1$ $\quad(7k^* + 6(m - k^*) - (7\bar{k} + 6(m - \bar{k})) = k^* - \bar{k})$.

Redução melhor

Redução anterior: $a = \frac{55}{7}$ e $b = 1$.

Redução melhor

Redução anterior: $a = \frac{55}{7}$ e $b = 1$.

Troque a cláusula $C_j = (x_1 \vee x_2 \vee x_3)$ pelas 7 cláusulas

Redução melhor

Redução anterior: $a = \frac{55}{7}$ e $b = 1$.

Troque a cláusula $C_j = (x_1 \vee x_2 \vee x_3)$ pelas 7 cláusulas $x_1 \vee x_3$, $\bar{x}_1 \vee \bar{x}_2$, $x_1 \vee \bar{y}_j$, $\bar{x}_1 \vee y_j$, $x_3 \vee \bar{y}_j$, $\bar{x}_3 \vee y_j$, $x_2 \vee y_j$, cada uma com peso $1/2$, exceto a última, que tem peso 1.

Redução melhor

Redução anterior: $a = \frac{55}{7}$ e $b = 1$.

Troque a cláusula $C_j = (x_1 \vee x_2 \vee x_3)$ pelas 7 cláusulas $x_1 \vee x_3$, $\bar{x}_1 \vee \bar{x}_2$, $x_1 \vee \bar{y}_j$, $\bar{x}_1 \vee y_j$, $x_3 \vee \bar{y}_j$, $\bar{x}_3 \vee y_j$, $x_2 \vee y_j$, cada uma com peso $1/2$, exceto a última, que tem peso 1.

Atribuição que satisfaz C_j corresponde à atribuição que satisfaz cláusulas acima que somam peso $3{,}5$.

Redução melhor

Redução anterior: $a = \frac{55}{7}$ e $b = 1$.

Troque a cláusula $C_j = (x_1 \vee x_2 \vee x_3)$ pelas 7 cláusulas $x_1 \vee x_3$, $\bar{x}_1 \vee \bar{x}_2$, $x_1 \vee \bar{y}_j$, $\bar{x}_1 \vee y_j$, $x_3 \vee \bar{y}_j$, $\bar{x}_3 \vee y_j$, $x_2 \vee y_j$, cada uma com peso $1/2$, exceto a última, que tem peso 1.

Atribuição que satisfaz C_j corresponde à atribuição que satisfaz cláusulas acima que somam peso $3{,}5$.

Atribuição que não satisfaz C_j corresponde à atribuição que satisfaz cláusulas acima que somam peso $2{,}5$.

Redução melhor

Redução anterior: $a = \frac{55}{7}$ e $b = 1$.

Troque a cláusula $C_j = (x_1 \vee x_2 \vee x_3)$ pelas 7 cláusulas $x_1 \vee x_3$, $\bar{x}_1 \vee \bar{x}_2$, $x_1 \vee \bar{y}_j$, $\bar{x}_1 \vee y_j$, $x_3 \vee \bar{y}_j$, $\bar{x}_3 \vee y_j$, $x_2 \vee y_j$, cada uma com peso $1/2$, exceto a última, que tem peso 1.

Atribuição que satisfaz C_j corresponde à atribuição que satisfaz cláusulas acima que somam peso $3{,}5$.

Atribuição que não satisfaz C_j corresponde à atribuição que satisfaz cláusulas acima que somam peso $2{,}5$.

Isso resulta numa L-redução com $a = 1 + \frac{5}{2} \cdot \frac{8}{7} = \frac{27}{7}$ e $b = 1$.

Redução melhor

Redução anterior: $a = \frac{55}{7}$ e $b = 1$.

Troque a cláusula $C_j = (x_1 \vee x_2 \vee x_3)$ pelas 7 cláusulas $x_1 \vee x_3$, $\bar{x}_1 \vee \bar{x}_2$, $x_1 \vee \bar{y}_j$, $\bar{x}_1 \vee y_j$, $x_3 \vee \bar{y}_j$, $\bar{x}_3 \vee y_j$, $x_2 \vee y_j$, cada uma com peso $1/2$, exceto a última, que tem peso 1.

Atribuição que satisfaz C_j corresponde à atribuição que satisfaz cláusulas acima que somam peso $3{,}5$.

Atribuição que não satisfaz C_j corresponde à atribuição que satisfaz cláusulas acima que somam peso $2{,}5$.

Isso resulta numa L-redução com $a = 1 + \frac{5}{2} \cdot \frac{8}{7} = \frac{27}{7}$ e $b = 1$.

Teorema: A menos que P $=$ NP, não existe uma α-aproximação para MAX 2SAT com $\alpha > \frac{209}{216} \approx 0{,}968$.

MAX E3SAT p/ Conjunto Independente

Conjunto Independente: Dado um grafo G, encontrar um conjunto independente de tamanho máximo em G.

MAX E3SAT p/ Conjunto Independente

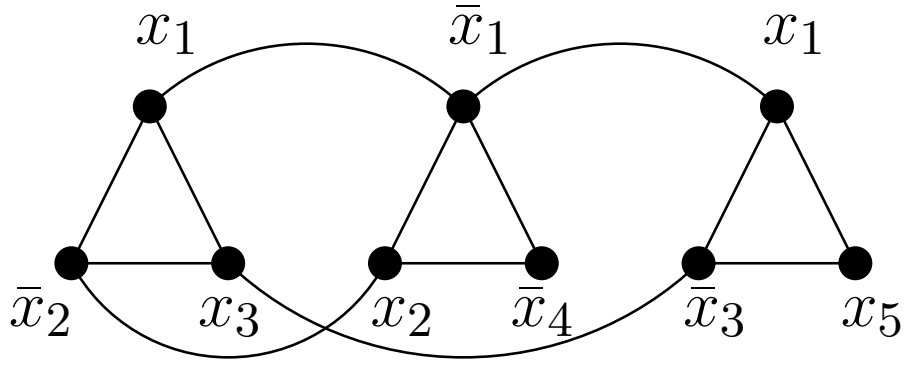
Conjunto Independente: Dado um grafo G, encontrar um conjunto independente de tamanho máximo em G.

Para $\Phi = (x_1 \vee \bar{x}_2 \vee x_3)(\bar{x}_1 \vee x_2 \vee x_4)(x_1 \vee \bar{x}_3 \vee x_5)$

MAX E3SAT p/ Conjunto Independente

Conjunto Independente: Dado um grafo G, encontrar um conjunto independente de tamanho máximo em G.

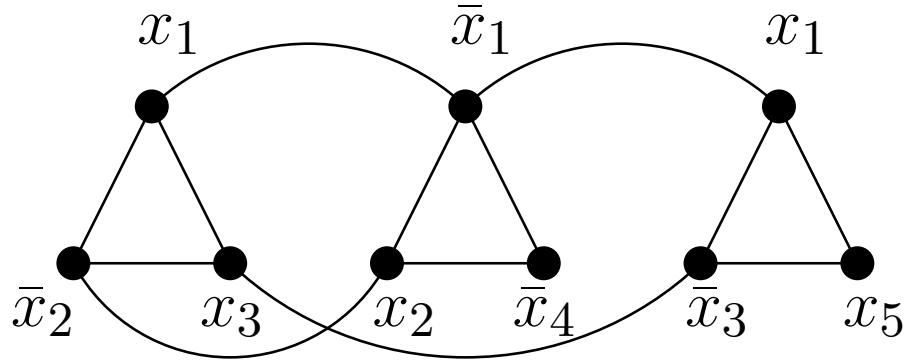
Para $\Phi = (x_1 \vee \bar{x}_2 \vee x_3)(\bar{x}_1 \vee x_2 \vee x_4)(x_1 \vee \bar{x}_3 \vee x_5)$



MAX E3SAT p/ Conjunto Independente

Conjunto Independente: Dado um grafo G, encontrar um conjunto independente de tamanho máximo em G.

Para $\Phi = (x_1 \vee \bar{x}_2 \vee x_3)(\bar{x}_1 \vee x_2 \vee x_4)(x_1 \vee \bar{x}_3 \vee x_5)$

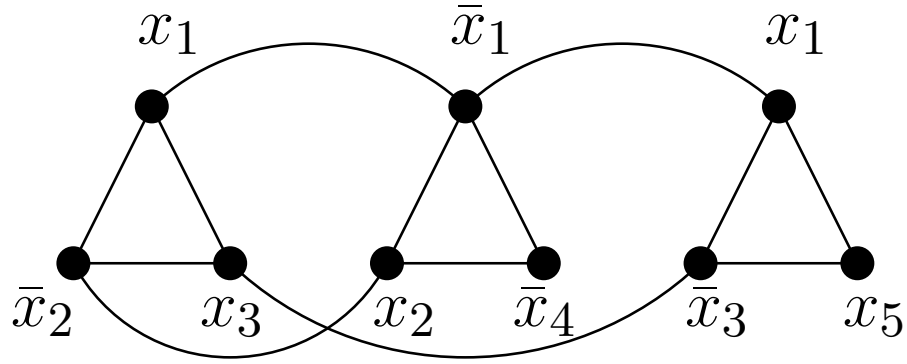


Note que $\mathrm{OPT}(\phi) = \mathrm{OPT}(G)$ e de um conjunto independente de tamanho v', podemos obter uma atribuição que satisfaz $v \geq v'$ cláusulas.

MAX E3SAT p/ Conjunto Independente

Conjunto Independente: Dado um grafo G, encontrar um conjunto independente de tamanho máximo em G.

Para $\Phi = (x_1 \vee \bar{x}_2 \vee x_3)(\bar{x}_1 \vee x_2 \vee x_4)(x_1 \vee \bar{x}_3 \vee x_5)$



Note que $\mathrm{OPT}(\phi) = \mathrm{OPT}(G)$ e de um conjunto independente de tamanho v', podemos obter uma atribuição que satisfaz $v \geq v'$ cláusulas.

Ou seja, temos $a = b = 1$.

MAX E3SAT p/ Conjunto Independente

Conjunto Independente: Dado um grafo G, encontrar um conjunto independente de tamanho máximo em G.

L-redução do MAX E3SAT para o Conjunto Independente com $a = b = 1$ implica no seguinte

MAX E3SAT p/ Conjunto Independente

Conjunto Independente: Dado um grafo G, encontrar um conjunto independente de tamanho máximo em G.

L-redução do MAX E3SAT para o Conjunto Independente com $a = b = 1$ implica no seguinte

Teorema: A menos que P $=$ NP, não existe uma α-aproximação para Conjunto Independente com $\alpha > \frac{7}{8}$.

Conjunto Independente para o próprio

Conjunto Independente: Dado grafo $G = (V, E)$, encontrar conjunto independente de tamanho máximo em G.

Conjunto Independente para o próprio

Conjunto Independente: Dado grafo $G = (V, E)$, encontrar conjunto independente de tamanho máximo em G.

Considere $G' = G \times G$, onde $V(G') = V \times V$ e (u_1, v_1) é adjacente a (u_2, v_2) sse $u_1u_2 \in E$ ou $v_1v_2 \in E$.

Conjunto Independente para o próprio

Conjunto Independente: Dado grafo $G = (V, E)$, encontrar conjunto independente de tamanho máximo em G.

Considere $G' = G \times G$, onde $V(G') = V \times V$ e (u_1, v_1) é adjacente a (u_2, v_2) sse $u_1u_2 \in E$ ou $v_1v_2 \in E$.

Se S é conjunto independente em G,
então $S \times S$ é conjunto independente em G'.

Conjunto Independente para o próprio

Conjunto Independente: Dado grafo $G = (V, E)$, encontrar conjunto independente de tamanho máximo em G.

Considere $G' = G \times G$, onde $V(G') = V \times V$ e (u_1, v_1) é adjacente a (u_2, v_2) sse $u_1u_2 \in E$ ou $v_1v_2 \in E$.

Se S é conjunto independente em G,
então $S \times S$ é conjunto independente em G'.

Se S' é conjunto independente em G' e
$S_1 = \{u \in V : \exists(u, w) \in S'\}$ e $S_2 = \{u \in V : \exists(w, u) \in S'\}$,
então ambos S_1 e S_2 são conjuntos independentes em G.

Conjunto Independente para o próprio

Conjunto Independente: Dado grafo $G = (V, E)$, encontrar conjunto independente de tamanho máximo em G.

Considere $G' = G \times G$, onde $V(G') = V \times V$ e (u_1, v_1) é adjacente a (u_2, v_2) sse $u_1u_2 \in E$ ou $v_1v_2 \in E$.

Se S é conjunto independente em G,
então $S \times S$ é conjunto independente em G'.

Se S' é conjunto independente em G' e
$S_1 = \{u \in V : \exists(u, w) \in S'\}$ e $S_2 = \{u \in V : \exists(w, u) \in S'\}$,
então ambos S_1 e S_2 são conjuntos independentes em G.

Conjunto Independente de tamanho k em G sse
conjunto independente de tamanho k^2 em G'.

Conjunto Independente para o próprio

Conjunto Independente: Dado grafo $G = (V, E)$, encontrar conjunto independente de tamanho máximo em G.

Considere $G' = G \times G$, onde $V(G') = V \times V$ e (u_1, v_1) é adjacente a (u_2, v_2) sse $u_1u_2 \in E$ ou $v_1v_2 \in E$.

Se S é conjunto independente em G,
então $S \times S$ é conjunto independente em G'.

Se S' é conjunto independente em G' e
$S_1 = \{u \in V : \exists (u, w) \in S'\}$ e $S_2 = \{u \in V : \exists (w, u) \in S'\}$,
então ambos S_1 e S_2 são conjuntos independentes em G.

Ou seja, $\mathrm{OPT}(G') = \mathrm{OPT}(G)^2$.

Conjunto Independente para o próprio

Conjunto Independente: Dado grafo $G = (V, E)$, encontrar conjunto independente de tamanho máximo em G.

Considere $G' = G \times G$, onde $V(G') = V \times V$ e (u_1, v_1) é adjacente a (u_2, v_2) sse $u_1 v_1 \in E$ ou $u_2 v_2 \in E$.

Vale que $\mathrm{OPT}(G') = \mathrm{OPT}(G)^2$.

Conjunto Independente para o próprio

Conjunto Independente: Dado grafo $G = (V, E)$, encontrar conjunto independente de tamanho máximo em G.

Considere $G' = G \times G$, onde $V(G') = V \times V$ e (u_1, v_1) é adjacente a (u_2, v_2) sse $u_1v_1 \in E$ ou $u_2v_2 \in E$.

Vale que $\mathrm{OPT}(G') = \mathrm{OPT}(G)^2$.

Se existe α-aproximação para Conjunto Independente, ao executá-la em G', obtemos conjunto independente de G' de tamanho $k' \geq \alpha\,\mathrm{OPT}(G') = \alpha\,\mathrm{OPT}(G)^2$,

Conjunto Independente para o próprio

Conjunto Independente: Dado grafo $G = (V, E)$, encontrar conjunto independente de tamanho máximo em G.

Considere $G' = G \times G$, onde $V(G') = V \times V$ e (u_1, v_1) é adjacente a (u_2, v_2) sse $u_1 v_1 \in E$ ou $u_2 v_2 \in E$.

Vale que $\mathrm{OPT}(G') = \mathrm{OPT}(G)^2$.

Se existe α-aproximação para Conjunto Independente, ao executá-la em G', obtemos conjunto independente de G' de tamanho $k' \geq \alpha\,\mathrm{OPT}(G') = \alpha\,\mathrm{OPT}(G)^2$, e podemos extrair conjunto independente de G de tamanho $\geq \sqrt{k'} \geq \sqrt{\alpha}\,\mathrm{OPT}(G)$.

Conjunto Independente para o próprio

Conjunto Independente: Dado grafo $G = (V, E)$, encontrar conjunto independente de tamanho máximo em G.

Considere $G' = G \times G$, onde $V(G') = V \times V$ e (u_1, v_1) é adjacente a (u_2, v_2) sse $u_1v_1 \in E$ ou $u_2v_2 \in E$.

Vale que $\mathrm{OPT}(G') = \mathrm{OPT}(G)^2$.

Se existe α-aproximação para Conjunto Independente, ao executá-la em G', obtemos conjunto independente de G' de tamanho $k' \geq \alpha\,\mathrm{OPT}(G') = \alpha\,\mathrm{OPT}(G)^2$, e podemos extrair conjunto independente de G de tamanho $\geq \sqrt{k'} \geq \sqrt{\alpha}\,\mathrm{OPT}(G)$.

Ou seja, obtemos uma $\sqrt{\alpha}$-aproximação.

Consequência

Conjunto Independente: Dado grafo $G = (V, E)$, encontrar conjunto independente de tamanho máximo em G.

Teorema: Se existe uma α-aproximação para qualquer constante α, com $0 < \alpha < 1$, então existe um esquema de aproximação polinomial para o Conjunto Independente.

Consequência

Conjunto Independente: Dado grafo $G = (V, E)$, encontrar conjunto independente de tamanho máximo em G.

Teorema: Se existe uma α-aproximação para qualquer constante α, com $0 < \alpha < 1$, então existe um esquema de aproximação polinomial para o Conjunto Independente.

Mas provamos há pouco o seguinte

Teorema: A menos que P $=$ NP, não existe uma α-aproximação para Conjunto Independente com $\alpha > \frac{7}{8}$.

Consequência

Conjunto Independente: Dado grafo $G = (V, E)$, encontrar conjunto independente de tamanho máximo em G.

Teorema: Se existe uma α-aproximação para qualquer constante α, com $0 < \alpha < 1$, então existe um esquema de aproximação polinomial para o Conjunto Independente.

Mas provamos há pouco o seguinte

Teorema: A menos que P $=$ NP, não existe uma α-aproximação para Conjunto Independente com $\alpha > \frac{7}{8}$.

Assim deduzimos

Teorema: A menos que P $=$ NP, não existe α-aproximação para Conjunto Independente para nenhuma contante α.