Algoritmos de Aproximação

Segundo Semestre de 2012

Localização de facilidades

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

$$\sum_{i \in F'} f_i + \sum_{j \in C} \min\{c_{ij} \ : \ i \in F'\}.$$

Localização de facilidades

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

$$\sum_{i \in F'} f_i + \sum_{j \in C} \min\{c_{ij} \; : \; i \in F'\}.$$

Tão difícil de aproximar quanto o SET COVER.

Localização de facilidades

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

$$\sum_{i \in F'} f_i + \sum_{j \in C} \min\{c_{ij} \ : \ i \in F'\}.$$

Tão difícil de aproximar quanto o SET COVER.

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

Localização de facilidades

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

$$\sum_{i \in F'} f_i + \sum_{j \in C} \min\{c_{ij} \ : \ i \in F'\}.$$

Tão difícil de aproximar quanto o SET COVER.

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

Vamos ver uma 4-aproximação para a versão métrica.

Formulação inteira

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

Formulação inteira

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

Variáveis:

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

Formulação inteira

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

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

$$
\begin{array}{lll}
\text{minimizem} & \sum_{i \in F} f_i \, y_i + \sum_{i \in F,\, j \in C} c_{ij} \, x_{ij} & \\
\text{sujeito a} & \sum_{i \in F} x_{ij} = 1 & \text{para todo } j \text{ em } C \\
 & x_{ij} \leq y_i & \text{para todo } i \text{ em } F \text{ e } j \text{ em } C \\
 & x_{ij} \in \{0, 1\} & \text{para todo } i \text{ em } F \text{ e } j \text{ em } C \\
 & y_i \in \{0, 1\} & \text{para todo } i \text{ em } F.
\end{array}
$$

Relaxação linear

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

$$
\begin{array}{lll}
\text{minimizar} & \sum_{i \in F} f_i \, y_i + \sum_{i \in F,\ j \in C} c_{ij} \, x_{ij} & \\
\text{sujeito a} & \sum_{i \in F} x_{ij} = 1 & \text{para todo } j \text{ em } C \\
& x_{ij} \leq y_i & \text{para todo } i \text{ em } F \text{ e } j \text{ em } C \\
& x_{ij} \geq 0 & \text{para todo } i \text{ em } F \text{ e } j \text{ em } C \\
& y_i \geq 0 & \text{para todo } i \text{ em } F.
\end{array}
$$

Relaxação linear

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

$$
\begin{array}{lll}
\text{minimizar} & \sum_{i\in F} f_i\, y_i + \sum_{i\in F,\ j\in C} c_{ij}\, x_{ij} & \\
\text{sujeito a} & \sum_{i\in F} x_{ij} = 1 & \text{para todo } j \text{ em } C \\
& x_{ij} \leq y_i & \text{para todo } i \text{ em } F \text{ e } j \text{ em } C \\
& x_{ij} \geq 0 & \text{para todo } i \text{ em } F \text{ e } j \text{ em } C \\
& y_i \geq 0 & \text{para todo } i \text{ em } F.
\end{array}
$$

Dual:

$$
\begin{array}{lll}
\text{maximizar} & \sum_{j\in C} v_j & \\
\text{sujeito a} & \sum_{j\in C} w_{ij} \leq f_i & \text{para todo } i \text{ em } F \\
& v_j - w_{ij} \leq c_{ij} & \text{para todo } i \text{ em } F \text{ e } j \text{ em } C \\
& w_{ij} \geq 0 & \text{para todo } i \text{ em } F \text{ e } j \text{ em } C.
\end{array}
$$

Relaxação linear

Primal:

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.

Relaxação linear

Primal:

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:

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.

Relaxação linear

Primal:

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:

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.

Relaxação linear

(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

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

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

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

(D) maximizar $\sum_{j\in C} v_j$

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

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

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

Relaxação linear

(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.

(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.

Lema: Sejam (x^*, y^*) e e (v^*, w^*) soluções ótimas do primal e do dual. Se $x^*_{ij} > 0$ então $c_{ij} \leq v^*_j$.

Relaxação linear

(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.

(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.

Lema: Sejam (x^*, y^*) e e (v^*, w^*) soluções ótimas do primal e do dual. Se $x^*_{ij} > 0$ então $c_{ij} \leq v^*_j$.

Prova: Por folgas complementares, se $x^*_{ij} > 0$, então $v^*_j - w^*_{ij} = c_{ij}$.

Relaxação linear

(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

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

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

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

(D) maximizar $\sum_{j\in C} v_j$

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

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

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

Lema: Sejam (x^*, y^*) e e (v^*, w^*) soluções ótimas do primal e do dual. Se $x^*_{ij} > 0$ então $c_{ij} \le v^*_j$.

Prova: Por folgas complementares, se $x^*_{ij} > 0$, então $v^*_j - w^*_{ij} = c_{ij}$. Como $w^*_{ij} \ge 0$, temos que $c_{ij} \le v^*_j$.

Como usar as soluções de (P) e (D)

Lema: Sejam (x^*, y^*) e (v^*, w^*) soluções ótimas do primal e do dual. Se $x^*_{ij} > 0$ então $c_{ij} \leq v^*_j$.

Como usar as soluções de (P) e (D)

Lema: Sejam (x^*, y^*) e (v^*, w^*) soluções ótimas do primal e do dual. Se $x^*_{ij} > 0$ então $c_{ij} \leq v^*_j$.

Facilidade i é vizinha do cliente j se $x^*_{ij} > 0$.

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

Como usar as soluções de (P) e (D)

Lema: Sejam (x^*, y^*) e (v^*, w^*) soluções ótimas do primal e do dual. Se $x^*_{ij} > 0$ então $c_{ij} \leq v^*_j$.

Facilidade i é vizinha do cliente j se $x^*_{ij} > 0$.

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

Se S é um conjunto de facilidades e todo cliente é vizinho de uma facilidade em S, pelo lema o custo de conexão dos clientes a S é no máximo $\sum_{j \in C} v^*_j \leq \mathrm{OPT}$.

Como usar as soluções de (P) e (D)

Lema: Sejam (x^*, y^*) e (v^*, w^*) soluções ótimas do primal e do dual. Se $x^*_{ij} > 0$ então $c_{ij} \leq v^*_j$.

Facilidade i é vizinha do cliente j se $x^*_{ij} > 0$.

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

Se S é um conjunto de facilidades e todo cliente é vizinho de uma facilidade em S, pelo lema o custo de conexão dos clientes a S é no máximo $\sum_{j \in C} v^*_j \leq \mathrm{OPT}$.

Infelizmente o custo das facilidades de um conjunto S assim pode ser muito alto.

Como usar as soluções de (P) e (D)

Facilidade i é vizinha do cliente j se $x^*_{ij} > 0$.

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

Se S é um conjunto de facilidades e todo cliente é vizinho de uma facilidade em S, pelo lema o custo de conexão dos clientes a S é no máximo $\sum_{j \in C} v^*_j \leq \mathrm{OPT}$.

Infelizmente o custo de um tal S pode ser muito alto.

Como usar as soluções de (P) e (D)

Facilidade i é vizinha do cliente j se $x^*_{ij} > 0$.
Seja $N(j) = \{i \in F \ : \ i$ é vizinha de $j\}$.

Se S é um conjunto de facilidades e todo cliente é vizinho de uma facilidade em S, pelo lema o custo de conexão dos clientes a S é no máximo $\sum_{j \in C} v^*_j \leq \mathrm{OPT}$.

Infelizmente o custo de um tal S pode ser muito alto.

Se um conjunto $F' \subseteq F$ pode ser particionado em conjuntos $F_1, F_2, \ldots$ onde $F_k = N(j_k)$ para algum cliente j_k,

Como usar as soluções de (P) e (D)

Facilidade i é vizinha do cliente j se $x^*_{ij} > 0$.
Seja $N(j) = \{i \in F \ : \ i \text{ é vizinha de } j\}$.

Se S é um conjunto de facilidades e todo cliente é vizinho de uma facilidade em S, pelo lema o custo de conexão dos clientes a S é no máximo $\sum_{j \in C} v^*_j \leq \mathrm{OPT}$.

Infelizmente o custo de um tal S pode ser muito alto.

Se um conjunto $F' \subseteq F$ pode ser particionado em conjuntos $F_1, F_2, \ldots$ onde $F_k = N(j_k)$ para algum cliente j_k, então abrindo a facilidade i_k mais barata de $N(j_k)$ tem-se

Como usar as soluções de (P) e (D)

Facilidade i é vizinha do cliente j se $x^*_{ij} > 0$.
Seja $N(j) = \{i \in F \ : \ i \text{ é vizinha de } j\}$.

Se S é um conjunto de facilidades e todo cliente é vizinho de uma facilidade em S, pelo lema o custo de conexão dos clientes a S é no máximo $\sum_{j \in C} v^*_j \leq \mathrm{OPT}$.

Infelizmente o custo de um tal S pode ser muito alto.

Se um conjunto $F' \subseteq F$ pode ser particionado em conjuntos $F_1, F_2, \ldots$ onde $F_k = N(j_k)$ para algum cliente j_k, então abrindo a facilidade i_k mais barata de $N(j_k)$ tem-se

$$f_{i_k} = f_{i_k} \sum_{i \in N(j_k)} x^*_{ij_k} \leq \sum_{i \in N(j_k)} f_i \, x^*_{ij_k} \leq \sum_{i \in N(j_k)} f_i \, y^*_i .$$

Como usar as soluções de (P) e (D)

Facilidade i é vizinha do cliente j se $x^*_{ij} > 0$.
Seja $N(j) = \{i \in F \ : \ i$ é vizinha de $j\}$.

Se um conjunto $F' \subseteq F$ pode ser particionado em conjuntos $F_1, F_2, \ldots$ onde $F_k = N(j_k)$ para algum cliente j_k, então abrindo a facilidade i_k mais barata de $N(j_k)$ tem-se

$$f_{i_k} = f_{i_k} \sum_{i \in N(j_k)} x^*_{ij_k} \le \sum_{i \in N(j_k)} f_i \, x^*_{ij_k} \le \sum_{i \in N(j_k)} f_i \, y^*_i .$$

Como usar as soluções de (P) e (D)

Facilidade i é vizinha do cliente j se $x^*_{ij} > 0$.
Seja $N(j) = \{i \in F \ : \ i \text{ é vizinha de } j\}$.

Se um conjunto $F' \subseteq F$ pode ser particionado em conjuntos $F_1, F_2, \ldots$ onde $F_k = N(j_k)$ para algum cliente j_k, então abrindo a facilidade i_k mais barata de $N(j_k)$ tem-se

$$f_{i_k} = f_{i_k} \sum_{i \in N(j_k)} x^*_{ij_k} \leq \sum_{i \in N(j_k)} f_i \, x^*_{ij_k} \leq \sum_{i \in N(j_k)} f_i \, y^*_i .$$

Somando-se para todas as facilidades abertas

$$\sum_k f_{i_k} \leq \sum_k \sum_{i \in N(j_k)} f_i \, y^*_i = \sum_{i \in F'} f_i \, y^*_i \leq \sum_{i \in F} f_i \, y^*_i .$$

Problema...

Se um conjunto $F' \subseteq F$ pode ser particionado em conjuntos $F_1, F_2, \ldots$ onde $F_k = N(j_k)$ para algum cliente j_k, então abrindo a facilidade i_k mais barata de $N(j_k)$ tem-se

$$\sum_k f_{i_k} \leq \sum_{i \in F} f_i \, y_i^* .$$

Problema...

Se um conjunto $F' \subseteq F$ pode ser particionado em conjuntos $F_1, F_2, \ldots$ onde $F_k = N(j_k)$ para algum cliente j_k, então abrindo a facilidade i_k mais barata de $N(j_k)$ tem-se

$$\sum_k f_{i_k} \leq \sum_{i \in F} f_i \, y_i^* .$$

Para um conjunto S de facilidades escolhidas assim, talvez nem todo cliente seja vizinho de uma facilidade em S.

Problema...

Se um conjunto $F' \subseteq F$ pode ser particionado em conjuntos $F_1, F_2, \ldots$ onde $F_k = N(j_k)$ para algum cliente j_k, então abrindo a facilidade i_k mais barata de $N(j_k)$ tem-se

$$\sum_k f_{i_k} \leq \sum_{i \in F} f_i \, y_i^* .$$

Para um conjunto S de facilidades escolhidas assim, talvez nem todo cliente seja vizinho de uma facilidade em S.

Como garantir custo baixo das facilidades abertas e custo baixo de conexão dos clientes?

Problema...

Se um conjunto $F' \subseteq F$ pode ser particionado em conjuntos $F_1, F_2, \ldots$ onde $F_k = N(j_k)$ para algum cliente j_k, então abrindo a facilidade i_k mais barata de $N(j_k)$ tem-se

$$\sum_k f_{i_k} \leq \sum_{i \in F} f_i \, y_i^* .$$

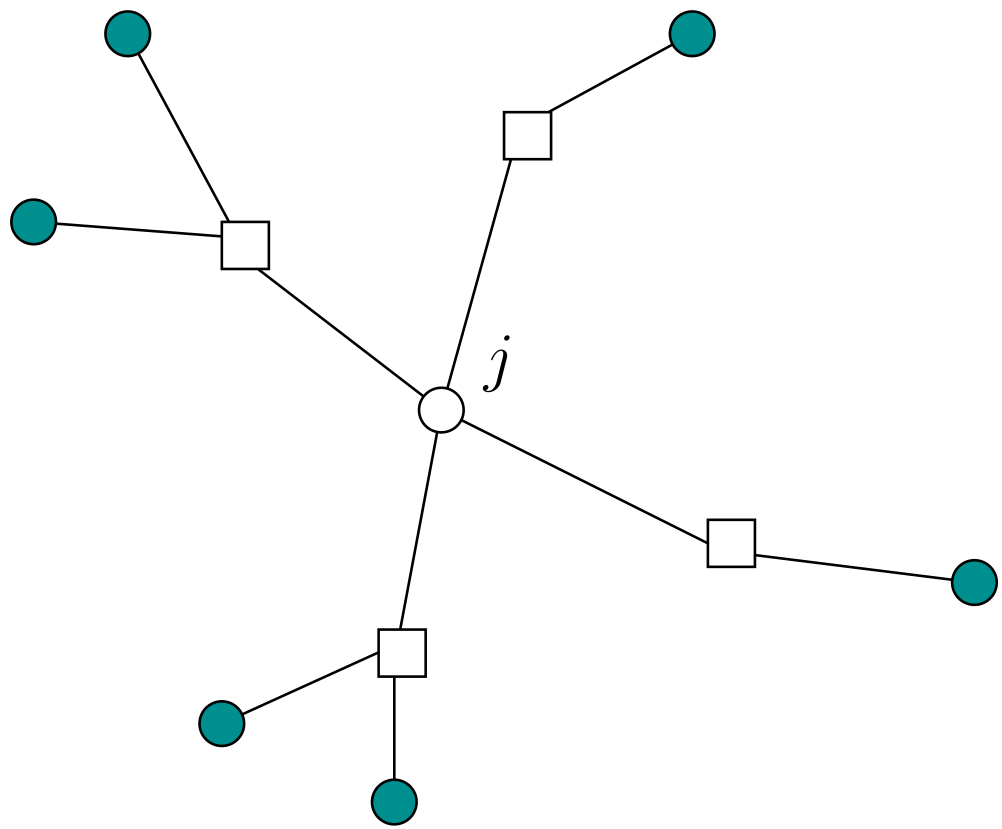
Para um conjunto S de facilidades escolhidas assim, talvez nem todo cliente seja vizinho de uma facilidade em S.

Como garantir custo baixo das facilidades abertas
e custo baixo de conexão dos clientes?

Não usamos a desigualdade triangular até agora...

Vizinhança aumentada

$N^2(j)$: clientes vizinhos de alguma facilidade de $N(j)$



Algoritmo

Arred-Det (F, C, f, c)

sejam (x^*, y^*) e (v^*, w^*) soluções ótimas de (P) e (D)
$D \leftarrow C \qquad k \leftarrow 0$
enquanto $D \neq \emptyset$ **faça**
    $k \leftarrow k + 1$
    $j_k \leftarrow \arg\min\{v_j^* : j \in D\}$
    $i_k \leftarrow \arg\min\{f_i : i \in N(j_k)\}$
    abra i_k e conecte $\{j_k\} \cup (N^2(j_k) \cap D)$ a i_k
    $D \leftarrow D \setminus (\{j_k\} \cup N^2(j_k))$

Algoritmo

Arred-Det (F, C, f, c)

sejam (x^*, y^*) e (v^*, w^*) soluções ótimas de (P) e (D)
$D \leftarrow C$ $\qquad k \leftarrow 0$
enquanto $D \neq \emptyset$ **faça**
$\quad k \leftarrow k + 1$
$\quad j_k \leftarrow \arg\min\{v_j^* : j \in D\}$
$\quad i_k \leftarrow \arg\min\{f_i : i \in N(j_k)\}$
$\quad$ abra i_k e conecte $\{j_k\} \cup (N^2(j_k) \cap D)$ a i_k
$\quad D \leftarrow D \setminus (\{j_k\} \cup N^2(j_k))$

Teorema: Arred-Det é uma 4-aproximação.

Algoritmo

Arred-Det (F, C, f, c)

sejam (x^*, y^*) e (v^*, w^*) soluções ótimas de (P) e (D)
$D \leftarrow C \qquad\qquad k \leftarrow 0$
enquanto $D \neq \emptyset$ **faça**
  $k \leftarrow k + 1$
  $j_k \leftarrow \arg\min\{v_j^* : j \in D\}$
  $i_k \leftarrow \arg\min\{f_i : i \in N(j_k)\}$
  abra i_k e conecte $\{j_k\} \cup (N^2(j_k) \cap D)$ a i_k
  $D \leftarrow D \setminus (\{j_k\} \cup N^2(j_k))$

Teorema: Arred-Det é uma 4-aproximação.

Prova: Primeiro note que os $N(j_k)$ são 2-a-2 disjuntos.

Algoritmo

Arred-Det (F, C, f, c)

sejam (x^*, y^*) e (v^*, w^*) soluções ótimas de (P) e (D)
$D \leftarrow C \qquad\qquad k \leftarrow 0$
enquanto $D \neq \emptyset$ **faça**
  $k \leftarrow k + 1$
  $j_k \leftarrow \arg\min\{v_j^* : j \in D\}$
  $i_k \leftarrow \arg\min\{f_i : i \in N(j_k)\}$
  abra i_k e conecte $\{j_k\} \cup (N^2(j_k) \cap D)$ a i_k
  $D \leftarrow D \setminus (\{j_k\} \cup N^2(j_k))$

Teorema: Arred-Det é uma 4-aproximação.

Prova: Primeiro note que os $N(j_k)$ são 2-a-2 disjuntos.
Logo $\sum_k f_{i_k} \leq \sum_{i \in F} f_i \, y_i^* \leq \mathrm{OPT}$.

Algoritmo

Arred-Det (F, C, f, c)

sejam (x^*, y^*) e (v^*, w^*) soluções ótimas de (P) e (D)
$D \leftarrow C \qquad k \leftarrow 0$
enquanto $D \neq \emptyset$ **faça**
  $k \leftarrow k + 1$
  $j_k \leftarrow \arg\min\{v_j^* : j \in D\}$
  $i_k \leftarrow \arg\min\{f_i : i \in N(j_k)\}$
  abra i_k e conecte $\{j_k\} \cup (N^2(j_k) \cap D)$ a i_k
  $D \leftarrow D \setminus (\{j_k\} \cup N^2(j_k))$

Teorema: Arred-Det é uma 4-aproximação.

Prova: Primeiro note que os $N(j_k)$ são 2-a-2 disjuntos.
Logo $\sum_k f_{i_k} \leq \sum_{i \in F} f_i \, y_i^* \leq \mathrm{OPT}$.
E os custos de conexão?

E os custos de conexão?

$j_k \leftarrow \arg\min\{v_j^* : j \in D\}$

$i_k \leftarrow \arg\min\{f_i : i \in N(j_k)\}$

abra i_k e conecte $\{j_k\} \cup (N^2(j_k) \cap D)$ a i_k

E os custos de conexão?

$j_k \leftarrow \arg\min\{v_j^* : j \in D\}$

$i_k \leftarrow \arg\min\{f_i : i \in N(j_k)\}$

abra i_k e conecte $\{j_k\} \cup (N^2(j_k) \cap D)$ a i_k

Lembre-se que $c_{ij} \leq v_j^*$ sempre que $i \in N(j)$.

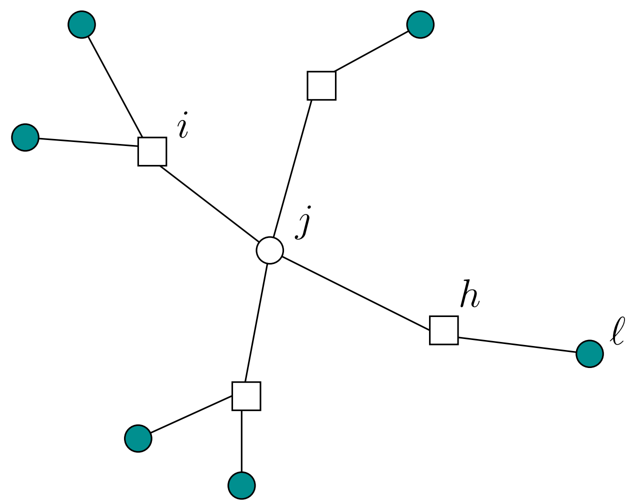
E os custos de conexão?

$j_k \leftarrow \arg\min\{v_j^* : j \in D\}$

$i_k \leftarrow \arg\min\{f_i : i \in N(j_k)\}$

abra i_k e conecte $\{j_k\} \cup (N^2(j_k) \cap D)$ a i_k

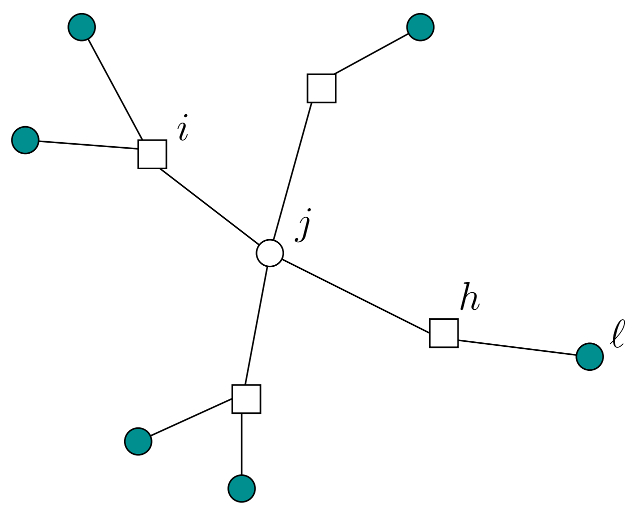
Lembre-se que $c_{ij} \leq v_j^*$ sempre que $i \in N(j)$.



E os custos de conexão?

$j_k \leftarrow \arg\min\{v_j^* : j \in D\}$
$i_k \leftarrow \arg\min\{f_i : i \in N(j_k)\}$
abra i_k e conecte $\{j_k\} \cup (N^2(j_k) \cap D)$ a i_k

Lembre-se que $c_{ij} \leq v_j^*$ sempre que $i \in N(j)$.

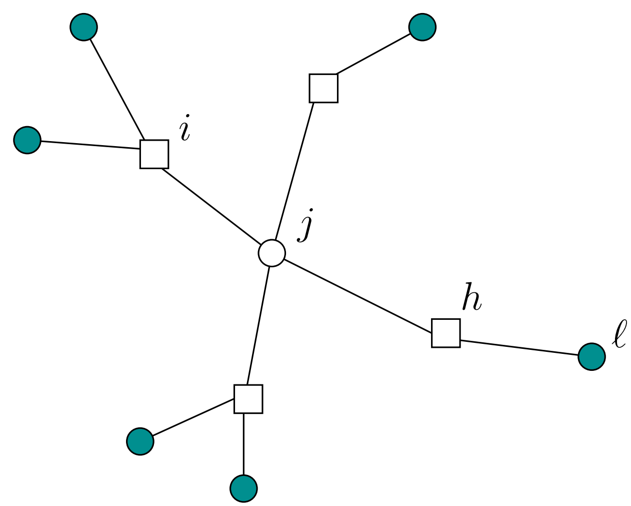


$$c_{i\ell} \leq c_{ij} + c_{hj} + c_{h\ell} \leq v_j^* + v_j^* + v_\ell^* \leq 3\, v_\ell^*.$$

E os custos de conexão?

$j_k \leftarrow \arg\min\{v_j^* : j \in D\}$
$i_k \leftarrow \arg\min\{f_i : i \in N(j_k)\}$
abra i_k e conecte $\{j_k\} \cup (N^2(j_k) \cap D)$ a i_k

Lembre-se que $c_{ij} \leq v_j^*$ sempre que $i \in N(j)$.



$$c_{i\ell} \leq c_{ij} + c_{hj} + c_{h\ell} \leq v_j^* + v_j^* + v_\ell^* \leq 3\, v_\ell^*.$$

Portanto o custo de conexão é $\leq 3 \sum_{\ell \in C} v_\ell^* \leq 3\, \mathrm{OPT}$.

Algoritmo

Arred-Det (F, C, f, c)

```
sejam (x*, y*) e (v*, w*) soluções ótimas de (P) e (D)
D ← C        k ← 0
enquanto D ≠ ∅ faça
    k ← k + 1
    j_k ← arg min{v*_j : j ∈ D}
    i_k ← arg min{f_i : i ∈ N(j_k)}
    abra i_k e conecte {j_k} ∪ (N²(j_k) ∩ D) a i_k
    D ← D \ ({j_k} ∪ N²(j_k))
```

Teorema: Arred-Det é uma 4-aproximação.

Prova: Primeiro note que os $N(j_k)$ são 2-a-2 disjuntos.
Logo $\sum_k f_{i_k} \leq \sum_{i \in F} f_i \, y_i^* \leq \mathrm{OPT}$.

Algoritmo

Arred-Det (F, C, f, c)
1 sejam (x^*, y^*) e (v^*, w^*) soluções ótimas de (P) e (D)
2 $D \leftarrow C$ $\quad k \leftarrow 0$
3 **enquanto** $D \neq \emptyset$ **faça**
4 $\quad k \leftarrow k + 1$
5 $\quad j_k \leftarrow \arg\min\{v_j^* : j \in D\}$
6 $\quad i_k \leftarrow \arg\min\{f_i : i \in N(j_k)\}$
7 $\quad$ abra i_k e conecte $\{j_k\} \cup (N^2(j_k) \cap D)$ a i_k
8 $\quad D \leftarrow D \setminus (\{j_k\} \cup N^2(j_k))$

Teorema: Arred-Det é uma 4-aproximação.

Prova: Primeiro note que os $N(j_k)$ são 2-a-2 disjuntos.
Logo $\sum_k f_{i_k} \leq \sum_{i \in F} f_i \, y_i^* \leq \mathrm{OPT}$.
E os custos de conexão somados são $\leq 3\,\mathrm{OPT}$. ∎

Aleatoriedade

Na próxima aula, começamos o cap 5.

Vamos começar por dois problemas.

Aleatoriedade

Na próxima aula, começamos o cap 5.

Vamos começar por dois problemas.

MAX CUT: Dado um grafo G e pesos $w_e \geq 0$ para cada aresta e, encontrar um corte em G de peso máximo.

Aleatoriedade

Na próxima aula, começamos o cap 5.

Vamos começar por dois problemas.

MAX CUT: Dado um grafo G e pesos $w_e \geq 0$ para cada aresta e, encontrar um corte em G de peso máximo.

Versão sem pesos: $w_e = 1$ para toda e.

Aleatoriedade

Na próxima aula, começamos o cap 5.

Vamos começar por dois problemas.

MAX CUT: Dado um grafo G e pesos $w_e \geq 0$ para cada aresta e, encontrar um corte em G de peso máximo.

Versão sem pesos: $w_e = 1$ para toda e.

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