Algoritmos de Aproximação

Segundo Semestre de 2012

Floresta de Steiner

Instância:

- grafo $G = (V, E)$
- custo $c_e \geq 0$ para cada aresta e de G
- coleção $\mathcal{R}$ de subconjuntos de V

Floresta de Steiner

Instância:

- grafo $G = (V, E)$
- custo $c_e \geq 0$ para cada aresta e de G
- coleção $\mathcal{R}$ de subconjuntos de V

$S \subseteq V$ é ativo se $R \cap S \neq \emptyset$ e $R \setminus S \neq \emptyset$ para algum R em $\mathcal{R}$.

Floresta de Steiner

Instância:

- grafo $G = (V, E)$
- custo $c_e \geq 0$ para cada aresta e de G
- coleção $\mathcal{R}$ de subconjuntos de V

$S \subseteq V$ é ativo se $R \cap S \neq \emptyset$ e $R \setminus S \neq \emptyset$ para algum R em $\mathcal{R}$.

$\mathcal{A}$: conjunto dos S ativos

Floresta de Steiner

Instância:

- grafo $G = (V, E)$
- custo $c_e \geq 0$ para cada aresta e de G
- coleção $\mathcal{R}$ de subconjuntos de V

$S \subseteq V$ é ativo se $R \cap S \neq \emptyset$ e $R \setminus S \neq \emptyset$ para algum R em $\mathcal{R}$.

$\mathcal{A}$: conjunto dos S ativos

F é $\mathcal{R}$-floresta se $\delta_F(S) \neq \emptyset$ para todo S em $\mathcal{A}$.

Floresta de Steiner

Instância:

- grafo $G = (V, E)$
- custo $c_e \geq 0$ para cada aresta e de G
- coleção $\mathcal{R}$ de subconjuntos de V

$S \subseteq V$ é ativo se $R \cap S \neq \emptyset$ e $R \setminus S \neq \emptyset$ para algum R em $\mathcal{R}$.

$\mathcal{A}$: conjunto dos S ativos

F é $\mathcal{R}$-floresta se $\delta_F(S) \neq \emptyset$ para todo S em $\mathcal{A}$.

Objetivo: encontrar $\mathcal{R}$-floresta F de peso mínimo.

Floresta de Steiner

Instância:

- grafo $G = (V, E)$
- custo $c_e \geq 0$ para cada aresta e de G
- coleção $\mathcal{R}$ de subconjuntos de V

Objetivo: encontrar $\mathcal{R}$-floresta F de peso mínimo.

Formulação como programa linear inteiro:

Dados $G = (V, E)$, c, $\mathcal{R}$, encontrar x que
minimize $\sum_e c_e x_e$
sujeito a $\sum_{e \in \delta(S)} x_e \geq 1$ para cada S em $\mathcal{A}$
$x_e \in \{0, 1\}$ para cada e em E.

Relaxação linear e seu dual

Problema: Dados $G = (V, E)$, c, $\mathcal{R}$, encontrar $\mathcal{R}$-floresta F de custo mínimo.

Relaxação linear (P):
Dados $G = (V, E)$, c, $\mathcal{R}$, encontrar x que
minimize $\sum_e c_e x_e$
sujeito a $\sum_{e \in \delta(S)} x_e \geq 1$ para cada S em $\mathcal{A}$
$x_e \geq 0$ para cada e em E.

Dual (D):
Dados $G = (V, E)$, c, $\mathcal{R}$, encontrar y que
maximize $\sum_{S \in \mathcal{A}} y_S$
sujeito a $\sum_{S : e \in \delta(S)} y_S \leq c_e$ para cada aresta e
$y_S \geq 0$ para cada S em $\mathcal{A}$.

Algoritmo primal-dual

$S \subseteq V$ é ativo se $R \cap S \neq \emptyset$ e $R \subseteq S \neq \emptyset$ para algum R em $\mathcal{R}$

$\mathcal{A}_F$: conjunto dos componentes ativos da floresta F

PRIMALDUAL $(G, c, \mathcal{R})$

para cada S em $\mathcal{S}$ **faça** $y_S \leftarrow 0$
    $F \leftarrow \emptyset$
    enquanto $\mathcal{A}_F \neq \emptyset$ **faça**
        $\epsilon \leftarrow \min\{c_e - \sum_{S' : e \in \delta(S')} y_{S'} : e \in \delta(S)$ e $S \in \mathcal{A}_F\}$
        $f \leftarrow \arg\min\{c_e - \sum_{S' : e \in \delta(S')} y_{S'} : e \in \delta(S)$ e $S \in \mathcal{A}_F\}$
        para cada $S \in \mathcal{A}_F$ **faça** $y_S \leftarrow y_S + \epsilon$
        $F \leftarrow F \cup \{f\}$
    seja F' uma $\mathcal{R}$-floresta minimal em F
    devolva F'

Algoritmo primal-dual

$S \subseteq V$ é ativo se $R \cap S \neq \emptyset$ e $R \subseteq S \neq \emptyset$ para algum R em $\mathcal{R}$

$\mathcal{A}_F$: conjunto dos componentes ativos da floresta F

PRIMALDUAL $(G, c, \mathcal{R})$

1 **para** cada S em $\mathcal{S}$ **faça** $y_S \leftarrow 0$
2 $F \leftarrow \emptyset$
3 **enquanto** $\mathcal{A}_F \neq \emptyset$ **faça**
4 $\epsilon \leftarrow \min\{c_e - \sum_{S':e\in\delta(S')} y_{S'} : e \in \delta(S)$ e $S \in \mathcal{A}_F\}$
5 $f \leftarrow \arg\min\{c_e - \sum_{S':e\in\delta(S')} y_{S'} : e \in \delta(S)$ e $S \in \mathcal{A}_F\}$
6 **para** cada $S \in \mathcal{A}_F$ **faça** $y_S \leftarrow y_S + \epsilon$
7 $F \leftarrow F \cup \{f\}$
8 seja F' uma $\mathcal{R}$-floresta minimal em F
9 **devolva** F'

Teorema: PRIMALDUAL é uma 2-aproximação.

Análise

Aula passada:

Lema: F é uma floresta.

Análise

Aula passada:

Lema: F é uma floresta.

Nesta aula:

Teorema: PrimalDual é uma 2-aproximação.

Análise

Aula passada:

Lema: F é uma floresta.

Nesta aula:

Teorema: PRIMALDUAL é uma 2-aproximação.

Prova: Se provarmos que

$$\sum_{S \in \mathcal{A}} |\delta_{F'}(S)| y_S \ \leq \ 2 \sum_{S \in \mathcal{A}} y_S,$$

Análise

Aula passada:

Lema: F é uma floresta.

Nesta aula:

Teorema: PrimalDual é uma 2-aproximação.

Prova: Se provarmos que

$$\sum_{S \in \mathcal{A}} |\delta_{F'}(S)| y_S \;\leq\; 2 \sum_{S \in \mathcal{A}} y_S,$$

então

$$\begin{aligned}
\sum_{e \in F'} c_e &= \sum_{e \in F'} \sum_{S: e \in \delta(S)} y_S \\
&= \sum_{S \in \mathcal{A}} |\delta_{F'}(S)|\, y_S \\
&\leq 2 \sum_{S \in \mathcal{A}} y_S \;\leq\; 2\, \mathrm{opt}.
\end{aligned}$$

Ideia da análise

Falta então provar que

$$\sum_{S\in\mathcal{A}} |\delta_{F'}(S)|y_S \;\leq\; 2\sum_{S\in\mathcal{A}} y_S,$$

Ideia da análise

Falta então provar que

$$\sum_{S \in \mathcal{A}} |\delta_{F'}(S)| y_S \ \leq \ 2 \sum_{S \in \mathcal{A}} y_S,$$

Para provar isso, antes mostraremos que:

o grau médio em F' das componentes ativas em cada iteração é menor que 2.

Ideia da análise

Falta então provar que

$$\sum_{S\in\mathcal{A}} |\delta_{F'}(S)| y_S \;\leq\; 2 \sum_{S\in\mathcal{A}} y_S,$$

Para provar isso, antes mostraremos que:

o grau médio em F' das componentes ativas em cada iteração é menor que 2.

Lema: No início de cada iteração,

$$\sum_{S\in\mathcal{A}_F} |\delta_{F'}(S)| \;\leq\; 2\,|\mathcal{A}_F|,$$

onde F é a floresta naquela iteração.

Análise

Lema: No início de cada iteração,

$$\sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)| \ \leq \ 2\,|\mathcal{A}_F|,$$

onde F é a floresta naquela iteração.

Suponha que o lema está provado.

Análise

Lema: No início de cada iteração,

$$\sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)| \;\leq\; 2\,|\mathcal{A}_F|,$$

onde F é a floresta naquela iteração.

Suponha que o lema está provado.

Vamos provar que

$$\sum_{S \in \mathcal{A}} |\delta_{F'}(S)| y_S \;\leq\; 2 \sum_{S \in \mathcal{A}} y_S.$$

Análise

Lema: No início de cada iteração,

$$\sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)| \;\leq\; 2\,|\mathcal{A}_F|,$$

onde F é a floresta naquela iteração.

Suponha que o lema está provado.

Vamos provar que

$$\sum_{S \in \mathcal{A}} |\delta_{F'}(S)| y_S \;\leq\; 2 \sum_{S \in \mathcal{A}} y_S.$$

Prova: Por indução no número de iterações.

Análise

Lema: No início de cada iteração,

$$\sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)| \;\leq\; 2\,|\mathcal{A}_F|,$$

onde F é a floresta naquela iteração.

Suponha que o lema está provado.

Vamos provar que

$$\sum_{S \in \mathcal{A}} |\delta_{F'}(S)| y_S \;\leq\; 2 \sum_{S \in \mathcal{A}} y_S.$$

Prova: Por indução no número de iterações.

No início da primeira iteração, isso é óbvio pois $y = 0$.

Análise

Lema: No início de cada iteração, $\sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)| \;\leq\; 2\,|\mathcal{A}_F|$, onde F é a floresta naquela iteração.

Vamos provar que $\sum_{S \in \mathcal{A}} |\delta_{F'}(S)| y_S \;\leq\; 2 \sum_{S \in \mathcal{A}} y_S$.

Prova: Por indução no número de iterações, vale que

$$\sum_{S \in \mathcal{A}} |\delta_{F'}(S)| y_S \;\leq\; 2 \sum_{S \in \mathcal{A}} y_S.$$

No início da primeira iteração, isso é óbvio pois $y = 0$.

Análise

Lema: No início de cada iteração, $\sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)| \;\leq\; 2\,|\mathcal{A}_F|$, onde F é a floresta naquela iteração.

Vamos provar que $\sum_{S \in \mathcal{A}} |\delta_{F'}(S)| y_S \;\leq\; 2 \sum_{S \in \mathcal{A}} y_S$.

Prova: Por indução no número de iterações, vale que

$$\sum_{S \in \mathcal{A}} |\delta_{F'}(S)| y_S \;\leq\; 2 \sum_{S \in \mathcal{A}} y_S.$$

No início da primeira iteração, isso é óbvio pois $y = 0$. Suponha que vale no início de uma iteração.

Análise

Lema: No início de cada iteração, $\sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)| \;\leq\; 2\,|\mathcal{A}_F|$, onde F é a floresta naquela iteração.

Vamos provar que $\sum_{S \in \mathcal{A}} |\delta_{F'}(S)| y_S \;\leq\; 2 \sum_{S \in \mathcal{A}} y_S$.

Prova: Por indução no número de iterações, vale que

$$\sum_{S \in \mathcal{A}} |\delta_{F'}(S)| y_S \;\leq\; 2 \sum_{S \in \mathcal{A}} y_S.$$

No início da primeira iteração, isso é óbvio pois $y = 0$.
Suponha que vale no início de uma iteração.

Na iteração, y_S é acrescido de ϵ para cada $S \in \mathcal{A}_F$.

Análise

Lema: No início de cada iteração, $\sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)| \ \leq \ 2\,|\mathcal{A}_F|$, onde F é a floresta naquela iteração.

Vamos provar que $\sum_{S \in \mathcal{A}} |\delta_{F'}(S)| y_S \ \leq \ 2 \sum_{S \in \mathcal{A}} y_S$.

Prova: Por indução no número de iterações, vale que

$$\sum_{S \in \mathcal{A}} |\delta_{F'}(S)| y_S \ \leq \ 2 \sum_{S \in \mathcal{A}} y_S.$$

No início da primeira iteração, isso é óbvio pois $y = 0$. Suponha que vale no início de uma iteração.

Na iteração, y_S é acrescido de ϵ para cada $S \in \mathcal{A}_F$. Logo, o lado esquerdo é acrescido de $\sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)| \epsilon$,

Análise

Lema: No início de cada iteração, $\sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)| \ \leq \ 2\,|\mathcal{A}_F|$, onde F é a floresta naquela iteração.

Vamos provar que $\sum_{S \in \mathcal{A}} |\delta_{F'}(S)| y_S \ \leq \ 2 \sum_{S \in \mathcal{A}} y_S$.

Prova: Por indução no número de iterações, vale que

$$\sum_{S \in \mathcal{A}} |\delta_{F'}(S)| y_S \ \leq \ 2 \sum_{S \in \mathcal{A}} y_S.$$

No início da primeira iteração, isso é óbvio pois $y = 0$.
Suponha que vale no início de uma iteração.

Na iteração, y_S é acrescido de ϵ para cada $S \in \mathcal{A}_F$.
Logo, o lado esquerdo é acrescido de $\sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)| \epsilon$,
e o lado direito, de $2|\mathcal{A}_F|\epsilon$.

Análise

Lema: No início de cada iteração, $\sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)| \;\leq\; 2\,|\mathcal{A}_F|$, onde F é a floresta naquela iteração.

Vamos provar que $\sum_{S \in \mathcal{A}} |\delta_{F'}(S)| y_S \;\leq\; 2 \sum_{S \in \mathcal{A}} y_S$.

Prova: Por indução no número de iterações, vale que

$$\sum_{S \in \mathcal{A}} |\delta_{F'}(S)| y_S \;\leq\; 2 \sum_{S \in \mathcal{A}} y_S.$$

No início da primeira iteração, isso é óbvio pois $y = 0$.
Suponha que vale no início de uma iteração.

Na iteração, y_S é acrescido de ϵ para cada $S \in \mathcal{A}_F$.
Logo, o lado esquerdo é acrescido de $\sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)|\epsilon$,
e o lado direito, de $2|\mathcal{A}_F|\epsilon$.

Pelo lema então, a desigualdade vale no final da iteração.

Análise

Lema: No início de cada iteração, $\sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)| \;\leq\; 2\,|\mathcal{A}_F|$, onde F é a floresta naquela iteração.

Análise

Lema: No início de cada iteração, $\sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)| \;\leq\; 2\,|\mathcal{A}_F|$, onde F é a floresta naquela iteração.

Prova: $|\delta_{F'}(S)|$ é o grau em F' de um componente S de F.

Análise

Lema: No início de cada iteração, $\sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)| \ \leq \ 2\,|\mathcal{A}_F|$, onde F é a floresta naquela iteração.

Prova: $|\delta_{F'}(S)|$ é o grau em F' de um componente S de F.

Os componentes de F de grau 1 em F' são todos ativos: para qualquer componente S de F,

$$\text{se } |\delta_{F'}(S)| = 1 \text{ então } S \in \mathcal{A}_F \,.$$

Análise

Lema: No início de cada iteração, $\sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)| \leq 2\,|\mathcal{A}_F|$, onde F é a floresta naquela iteração.

Prova: $|\delta_{F'}(S)|$ é o grau em F' de um componente S de F.

Os componentes de F de grau 1 em F' são todos ativos: para qualquer componente S de F,

$$\text{se } |\delta_{F'}(S)| = 1 \text{ então } S \in \mathcal{A}_F\,.$$

Grafo H: contração das componentes de F nesta iteração.

Análise

Lema: No início de cada iteração, $\sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)| \;\leq\; 2\,|\mathcal{A}_F|$, onde F é a floresta naquela iteração.

Prova: $|\delta_{F'}(S)|$ é o grau em F' de um componente S de F.

Os componentes de F de grau 1 em F' são todos ativos: para qualquer componente S de F,

$$\text{se } |\delta_{F'}(S)| = 1 \text{ então } S \in \mathcal{A}_F\,.$$

Grafo H: contração das componentes de F nesta iteração.

H é uma floresta, logo $\sum_{S \in V_H} |\delta_H(S)| = 2|E_H| \leq 2(|V_H| - 1)$.

Análise

Lema: No início de cada iteração, $\sum_{S\in\mathcal{A}_F} |\delta_{F'}(S)| \ \le\ 2\,|\mathcal{A}_F|$, onde F é a floresta naquela iteração.

Prova: $|\delta_{F'}(S)|$ é o grau em F' de um componente S de F.

Os componentes de F de grau 1 em F' são todos ativos: para qualquer componente S de F,

$$\text{se } |\delta_{F'}(S)| = 1 \text{ então } S \in \mathcal{A}_F\,.$$

Grafo H: contração das componentes de F nesta iteração.

H é uma floresta, logo $\sum_{S\in V_H} |\delta_H(S)| = 2|E_H| \le 2(|V_H| - 1)$.

$$\sum_{S\in\mathcal{A}_F\cup\,\mathcal{I}_F} |\delta_{F'}(S)| \le 2(|\mathcal{A}_F| + |\mathcal{I}_F| - 1),$$

onde $\mathcal{I}_F$ são os componentes inativos de F na iteração, de grau não nulo.

Conclusão da análise

Lema: No início de cada iteração, $\sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)| \;\leq\; 2\,|\mathcal{A}_F|$, onde F é a floresta naquela iteração.

Prova: Se $|\delta_{F'}(S)| = 1$ então $S \in \mathcal{A}_F$.

Grafo H: contração das componentes de F nesta iteração.

Como H é uma floresta,

$$\sum_{S \in \mathcal{A}_F \cup \, \mathcal{I}_F} |\delta_{F'}(S)| \leq 2(|\mathcal{A}_F| + |\mathcal{I}_F| - 1).$$

$\mathcal{I}_F$: componentes inativos de F na iteração de grau > 0.

Conclusão da análise

Lema: No início de cada iteração, $\sum_{S\in\mathcal{A}_F} |\delta_{F'}(S)| \;\leq\; 2\,|\mathcal{A}_F|$, onde F é a floresta naquela iteração.

Prova: Se $|\delta_{F'}(S)| = 1$ então $S \in \mathcal{A}_F$.

Grafo H: contração das componentes de F nesta iteração.

Como H é uma floresta,

$$\sum_{S\in\mathcal{A}_F\cup\,\mathcal{I}_F} |\delta_{F'}(S)| \leq 2(|\mathcal{A}_F| + |\mathcal{I}_F| - 1).$$

$\mathcal{I}_F$: componentes inativos de F na iteração de grau > 0.

Como $|\delta_{F'}(S)| \geq 2$ para todo S em $\mathcal{I}_{F'}$, temos que

$$\sum_{S\in\mathcal{A}_F} |\delta_{F'}(S)| \;=\; \sum_{S\in\mathcal{A}_F\cup\mathcal{I}_F} |\delta_{F'}(S)| - \sum_{S\in\mathcal{I}_F} |\delta_{F'}(S)|$$

Conclusão da análise

Lema: No início de cada iteração, $\sum_{S\in\mathcal{A}_F} |\delta_{F'}(S)| \;\leq\; 2\,|\mathcal{A}_F|$, onde F é a floresta naquela iteração.

Prova: Se $|\delta_{F'}(S)| = 1$ então $S \in \mathcal{A}_F$.

Grafo H: contração das componentes de F nesta iteração.

Como H é uma floresta,

$$\sum_{S\in\mathcal{A}_F\cup\,\mathcal{I}_F} |\delta_{F'}(S)| \leq 2(|\mathcal{A}_F| + |\mathcal{I}_F| - 1).$$

$\mathcal{I}_F$: componentes inativos de F na iteração de grau > 0.

Como $|\delta_{F'}(S)| \geq 2$ para todo S em $\mathcal{I}_{F'}$, temos que

$$\begin{aligned}
\sum_{S\in\mathcal{A}_F} |\delta_{F'}(S)| &= \sum_{S\in\mathcal{A}_F\cup\mathcal{I}_F} |\delta_{F'}(S)| - \sum_{S\in\mathcal{I}_F} |\delta_{F'}(S)| \\
&\leq 2(|\mathcal{A}_F| + |\mathcal{I}_F| - 1) - 2|\mathcal{I}_F|
\end{aligned}$$

Conclusão da análise

Lema: No início de cada iteração, $\sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)| \;\leq\; 2\,|\mathcal{A}_F|$, onde F é a floresta naquela iteração.

Prova: Se $|\delta_{F'}(S)| = 1$ então $S \in \mathcal{A}_F$.

Grafo H: contração das componentes de F nesta iteração.

Como H é uma floresta,

$$\sum_{S \in \mathcal{A}_F \cup\, \mathcal{I}_F} |\delta_{F'}(S)| \leq 2(|\mathcal{A}_F| + |\mathcal{I}_F| - 1).$$

$\mathcal{I}_F$: componentes inativos de F na iteração de grau > 0.

Como $|\delta_{F'}(S)| \geq 2$ para todo S em $\mathcal{I}_{F'}$, temos que

$$\begin{aligned} \sum_{S \in \mathcal{A}_F} |\delta_{F'}(S)| &= \sum_{S \in \mathcal{A}_F \cup \mathcal{I}_F} |\delta_{F'}(S)| - \sum_{S \in \mathcal{I}_F} |\delta_{F'}(S)| \\ &\leq 2(|\mathcal{A}_F| + |\mathcal{I}_F| - 1) - 2|\mathcal{I}_F| \;<\; 2|\mathcal{A}_F|\,. \end{aligned}$$

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

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.

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

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

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

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

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 (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.

Relaxação linear

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.

Relaxação linear

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.

Soluções maximais do dual

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.

Soluções maximais do dual

Dual (D):

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

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

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.

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

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

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

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

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

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

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

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

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

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 $v_j^* \geq c_{ij}$.

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 $w_{ij}^* > 0$.

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

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

Facilidades justas

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

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 $w_{ij}^* > 0$.

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

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

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

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 $j\}$.

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

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

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

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 $T = \emptyset$ $\quad$ ($v^* = w^* = 0$).

Aumente v^*_j uniformemente para $j \in S$ até...

Ideia do algoritmo primal-dual

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

Aumente v_j^* uniformemente para $j \in S$ até...
que algum $v_j^* = c_{ij}$,

Ideia do algoritmo primal-dual

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

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

Ideia do algoritmo primal-dual

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

Aumente v_j^* uniformemente para $j \in S$ até...
que algum $v_j^* = c_{ij}$,
quando começamos 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 $T = \emptyset$ $\quad (v^* = w^* = 0)$.

Aumente v_j^* uniformemente para $j \in S$ até...
que algum $v_j^* = c_{ij}$,
quando começamos 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 $T = \emptyset$ $\quad (v^* = w^* = 0)$.

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 $T = \emptyset$ $\quad$ ($v^* = w^* = 0$).

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 (com i's tq os $N(i)$ são disjuntos).

Algoritmo

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

$v \leftarrow 0$ $\qquad w \leftarrow 0$

$S \leftarrow C$ $\qquad T \leftarrow \emptyset$

enquanto $S \neq \emptyset$ **faça**

  aumente v_j para $j \in S$ e w_{ij} para $i \in N(j)$ e $j \in S$.

  se $j \in S$ é vizinho de $i \in T$ **então** $S \leftarrow S \setminus \{j\}$

  se a desigualdade do dual para $i \notin T$ é justa

    então $T \leftarrow T \cup \{i\}$ $\qquad S \leftarrow S \setminus N(i)$

Algoritmo

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

$v \leftarrow 0 \qquad w \leftarrow 0$

$S \leftarrow C \qquad T \leftarrow \emptyset$

enquanto $S \neq \emptyset$ **faça**

  aumente v_j para $j \in S$ e w_{ij} para $i \in N(j)$ e $j \in S$.

  se $j \in S$ é vizinho de $i \in T$ **então** $S \leftarrow S \setminus \{j\}$

  se a desigualdade do dual para $i \notin T$ é justa

    então $T \leftarrow T \cup \{i\} \qquad S \leftarrow S \setminus N(i)$

$T' \leftarrow \emptyset$

enquanto $T \neq \emptyset$ **faça**

  seja i um elemento de T

  $T' \leftarrow T' \cup \{i\}$

  $T \leftarrow T \setminus \{h \in T : \exists j \in C, w_{ij} > 0, w_{hj} > 0\}$

devolva T'