Algoritmos de Aproximação

Segundo Semestre de 2012

Caminho mais curto

Instância:

- grafo $G = (V, E)$
- custo $c_e \geq 0$ para cada aresta e de G
- vértices s e t de G

Caminho mais curto

Instância:

- grafo $G = (V, E)$
- custo $c_e \geq 0$ para cada aresta e de G
- vértices s e t de G

Objetivo: encontrar s-t caminho em G de custo mínimo.

Caminho mais curto

Instância:

- grafo $G = (V, E)$
- custo $c_e \geq 0$ para cada aresta e de G
- vértices s e t de G

Objetivo: encontrar s-t caminho em G de custo mínimo.

$\mathcal{S} := \{S \subseteq V : s \in S,\ t \notin S\}$

Formulação como programa linear inteiro:

dados G, c, s e t, encontrar x que

minimize $\sum_{e \in E} c_e \, x_e$

sujeito a $\sum_{e \in \delta(S)} x_e \geq 1$ para todo S em $\mathcal{S}$

$x_e \in \{0, 1\}$ para todo $e \in E$

Relaxação linear e seu dual

Problema: Dados G, c, s e t, encontrar s-t caminho em G de custo mínimo.

Relaxação linear: Dados G, c, s e t, encontrar x que

minimize $\sum_{e \in E} c_e \, x_e$

sujeito a $\sum_{e \in \delta(S)} x_e \geq 1$ para todo S em $\mathcal{S}$

$x_e \geq 0$ para todo $e \in E$

Relaxação linear e seu dual

Problema: Dados G, c, s e t, encontrar s-t caminho em G de custo mínimo.

Relaxação linear: Dados G, c, s e t, encontrar x que
minimize $\sum_{e \in E} c_e \, x_e$
sujeito a $\sum_{e \in \delta(S)} x_e \geq 1$ para todo S em $\mathcal{S}$
$x_e \geq 0$ para todo $e \in E$

Dual: Dados G, c, s e t, encontrar y que
maximize $\sum_{S \in \mathcal{S}} y_S$
sujeito a $\sum_{S \in \mathcal{S} : e \in \delta(S)} y_S \leq c_e$ para todo $e \in E$
$y_S \geq 0$ para todo $S \in \mathcal{S}$

Algoritmo primal-dual

```
PrimalDual (G, c, s, t)        ▷ G = (V, E)
    𝒮 ← {S ⊆ V : s ∈ S, t ∉ S}
    para cada S em 𝒮 faça y_S ← 0
    F ← ∅
    enquanto não existe s-t caminho em (V, F) faça
        seja C a componente de (V, F) contendo s
        ϵ ← min{c_e − ∑_{S∈𝒮:e∈δ(S)} y_S : e ∈ δ(C)}
        f ← arg min{c_e − ∑_{S∈𝒮:e∈δ(S)} y_S : e ∈ δ(C)}
        y_C ← y_C + ϵ
        F ← F ∪ {f}
    seja P um s-t caminho em F
    devolva P
```

$$\mathcal{S} \leftarrow \{S \subseteq V : s \in S,\ t \notin S\}$$

$$\epsilon \leftarrow \min\{c_e - \textstyle\sum_{S\in\mathcal{S}:e\in\delta(S)} y_S : e \in \delta(C)\}$$

$$f \leftarrow \arg\min\{c_e - \textstyle\sum_{S\in\mathcal{S}:e\in\delta(S)} y_S : e \in \delta(C)\}$$

Algoritmo primal-dual

PRIMALDUAL (G, c, s, t) $\quad \triangleright\ G = (V, E)$

$\mathcal{S} \leftarrow \{S \subseteq V : s \in S,\ t \notin S\}$
para cada S em $\mathcal{S}$ **faça** $y_S \leftarrow 0$
$F \leftarrow \emptyset$
enquanto não existe s-t caminho em (V, F) **faça**
    seja C a componente de (V, F) contendo s
    $\epsilon \leftarrow \min\{c_e - \sum_{S \in \mathcal{S} : e \in \delta(S)} y_S : e \in \delta(C)\}$
    $f \leftarrow \arg\min\{c_e - \sum_{S \in \mathcal{S} : e \in \delta(S)} y_S : e \in \delta(C)\}$
    $y_C \leftarrow y_C + \epsilon$
    $F \leftarrow F \cup \{f\}$
seja P um s-t caminho em F
devolva P

Lema: O tempo todo, F é uma árvore contendo s.

Algoritmo primal-dual

PRIMALDUAL (G, c, s, t) $\qquad \triangleright\ G = (V, E)$

$F \leftarrow \emptyset \qquad \mathcal{S} \leftarrow \{S \subseteq V : s \in S, t \notin S\}$
para cada S em $\mathcal{S}$ **faça** $y_S \leftarrow 0$
enquanto não existe s-t caminho em (V, F) **faça**
    seja C a componente de (V, F) contendo s
    $\epsilon \leftarrow \min\{c_e - \sum_{S \in \mathcal{S}: e \in \delta(S)} y_S : e \in \delta(C)\}$
    $f \leftarrow \arg\min\{c_e - \sum_{S \in \mathcal{S}: e \in \delta(S)} y_S : e \in \delta(C)\}$
    $y_C \leftarrow y_C + \epsilon \qquad F \leftarrow F \cup \{f\}$
seja P um s-t caminho em F
devolva P

Algoritmo primal-dual

PRIMALDUAL (G, c, s, t) $\quad\triangleright\ G = (V, E)$

$F \leftarrow \emptyset \quad \mathcal{S} \leftarrow \{S \subseteq V : s \in S, t \notin S\}$
para cada S em $\mathcal{S}$ **faça** $y_S \leftarrow 0$
enquanto não existe s-t caminho em (V, F) **faça**
    seja C a componente de (V, F) contendo s
    $\epsilon \leftarrow \min\{c_e - \sum_{S \in \mathcal{S}: e \in \delta(S)} y_S : e \in \delta(C)\}$
    $f \leftarrow \arg\min\{c_e - \sum_{S \in \mathcal{S}: e \in \delta(S)} y_S : e \in \delta(C)\}$
    $y_C \leftarrow y_C + \epsilon \quad F \leftarrow F \cup \{f\}$
seja P um s-t caminho em F
devolva P

Teorema: P é um s-t caminho de custo mínimo.

Algoritmo primal-dual

```
PRIMALDUAL (G, c, s, t)        ▷ G = (V, E)
    F ← ∅     𝒮 ← {S ⊆ V : s ∈ S, t ∉ S}
    para cada S em 𝒮 faça y_S ← 0
    enquanto não existe s-t caminho em (V, F) faça
        seja C a componente de (V, F) contendo s
        ε ← min{c_e − ∑_{S∈𝒮: e∈δ(S)} y_S : e ∈ δ(C)}
        f ← arg min{c_e − ∑_{S∈𝒮: e∈δ(S)} y_S : e ∈ δ(C)}
        y_C ← y_C + ε     F ← F ∪ {f}
    seja P um s-t caminho em F
    devolva P
```

Teorema: P é um s-t caminho de custo mínimo.

Prova:

$$
\begin{aligned}
\sum_{e\in P} c_e &= \sum_{e\in P} \sum_{S: e\in\delta(S)} y_S \\
&= \sum_{S\in\mathcal{S}} y_S \, |P \cap \delta(S)| \\
&= \sum_{S\in\mathcal{S}} y_S \ \leq \ \mathrm{opt}.
\end{aligned}
$$

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.

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.

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 $\mathcal{A}$.

Primeira tentativa

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

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

Primeira tentativa

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

$\mathcal{A}_F$: 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**
    seja S um elemento de $\mathcal{A}_F$
    $\epsilon \leftarrow \min\{c_e - \sum_{S' : e \in \delta(S')} y_{S'} : e \in \delta(S)\}$
    $f \leftarrow \arg\min\{c_e - \sum_{S' : e \in \delta(S')} y_{S'} : e \in \delta(S)\}$
    $y_S \leftarrow y_S + \epsilon$
    $F \leftarrow F \cup \{f\}$
seja F' uma $\mathcal{R}$-floresta minimal em F
devolva F'

Primeira tentativa

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**
    seja S um elemento de $\mathcal{A}_F$
    $\epsilon \leftarrow \min\{c_e - \sum_{S' : e \in \delta(S')} y_{S'} : e \in \delta(S)\}$
    $f \leftarrow \arg\min\{c_e - \sum_{S' : e \in \delta(S')} y_{S'} : e \in \delta(S)\}$
    $y_S \leftarrow y_S + \epsilon$
    $F \leftarrow F \cup \{f\}$
seja F' uma $\mathcal{R}$-floresta minimal em F
devolva F'

Primeira tentativa

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**
    seja S um elemento de $\mathcal{A}_F$
    $\epsilon \leftarrow \min\{c_e - \sum_{S': e \in \delta(S')} y_{S'} : e \in \delta(S)\}$
    $f \leftarrow \arg\min\{c_e - \sum_{S': e \in \delta(S')} y_{S'} : e \in \delta(S)\}$
    $y_S \leftarrow y_S + \epsilon$
    $F \leftarrow F \cup \{f\}$
seja F' uma $\mathcal{R}$-floresta minimal em F
devolva F'

Será que PRIMALDUAL é uma ???-aproximação??

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

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 𝒮 faça y_S ← 0
F ← ∅
enquanto 𝒜_F ≠ ∅ faça
    ε ← min{c_e − Σ_{S′:e∈δ(S′)} y_{S′} : e ∈ δ(S) e S ∈ 𝒜_F}
    f ← arg min{c_e − Σ_{S′:e∈δ(S′)} y_{S′} : e ∈ δ(S) e S ∈ 𝒜_F}
    para cada S ∈ 𝒜_F faça y_S ← y_S + ε
    F ← F ∪ {f}
seja F′ uma 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

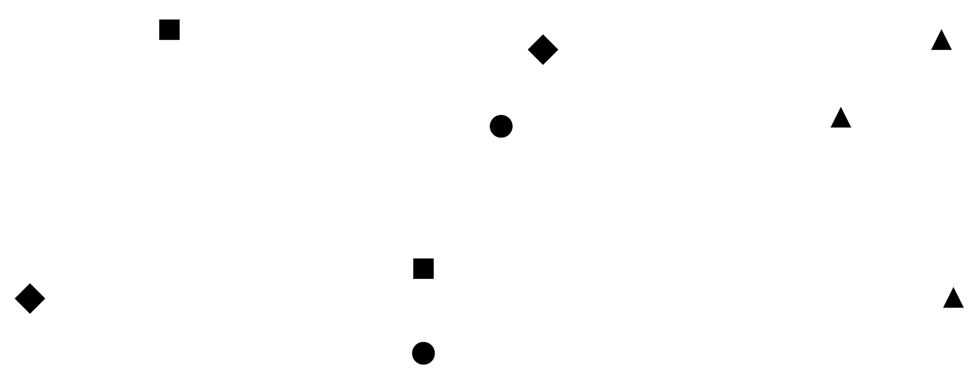
PRIMALDUAL $(G, c, \mathcal{R})$
para cada S em $\mathcal{S}$ **faça** $y_S \leftarrow 0$
$F \leftarrow \emptyset$
enquanto $\mathcal{A}_F \neq \emptyset$ **faça**
  $\epsilon \leftarrow \min\{c_e - \sum_{S':e\in\delta(S')} y_{S'} : e \in \delta(S)$ e $S \in \mathcal{A}_F\}$
  $f \leftarrow \arg\min\{c_e - \sum_{S':e\in\delta(S')} y_{S'} : e \in \delta(S)$ 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'

Teorema: PRIMALDUAL é uma 2-aproximação.

Simulação

Instância: $(G, c, \{R_1, R_2, R_3\})$

G: completo com 9 vértices c: distância euclidiana
R_1, R_2 e R_3: triângulos, losangos e quadrados

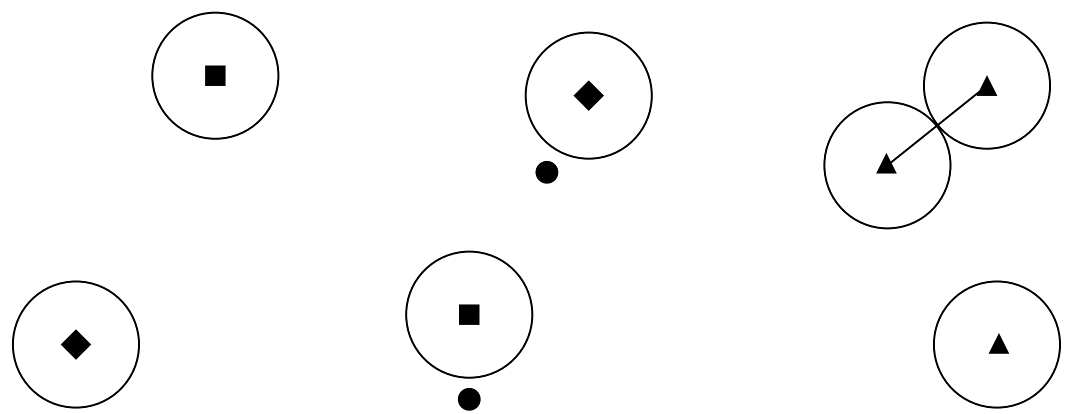


Simulação

Instância: $(G, c, \{R_1, R_2, R_3\})$

G: completo com 9 vértices c: distância euclidiana
R_1, R_2 e R_3: triângulos, losangos e quadrados

Início da segunda iteração do algoritmo:



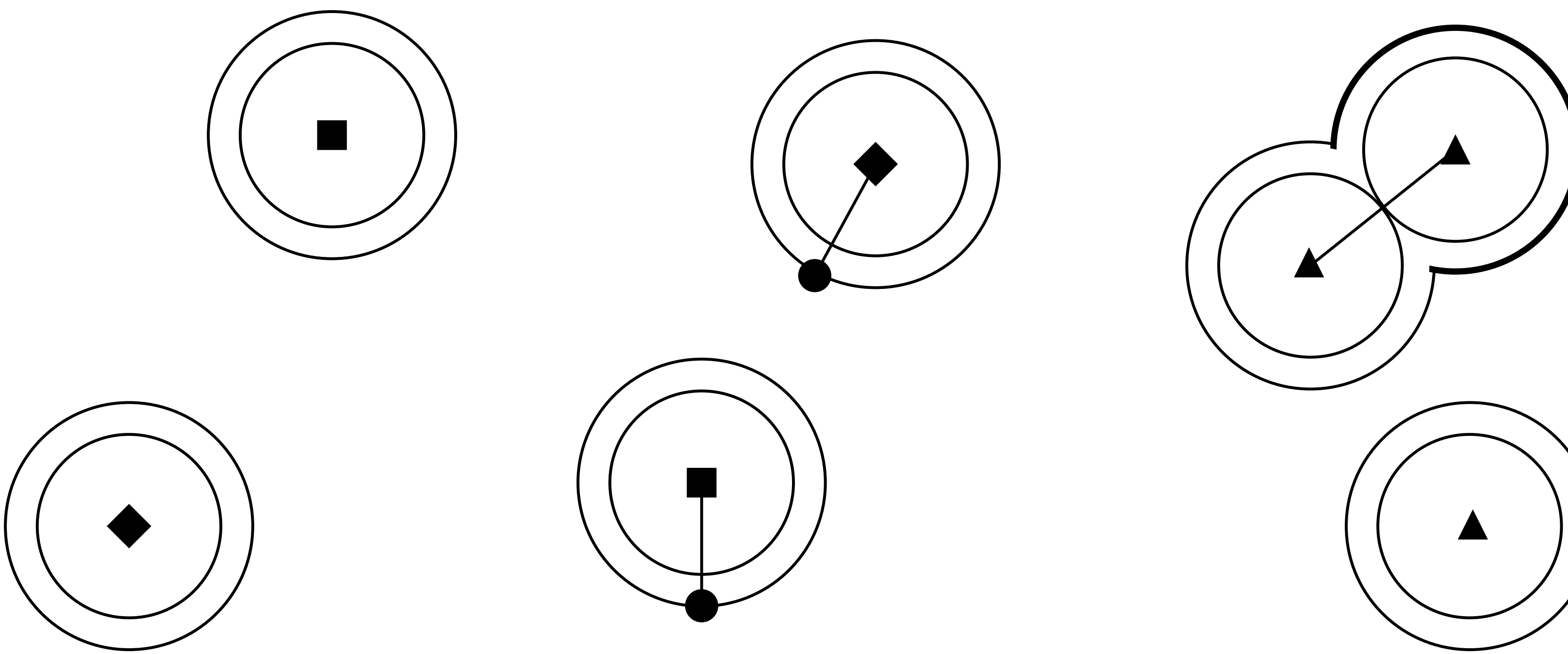
Os círculos representam componentes ativos.
Raio de cada círculo S proporcional ao valor de y_S.

Simulação

Instância: $(G, c, \{R_1, R_2, R_3\})$

G: completo com 9 vértices c: distância euclidiana
R_1, R_2 e R_3: triângulos, losangos e quadrados

Início da quarta iteração:

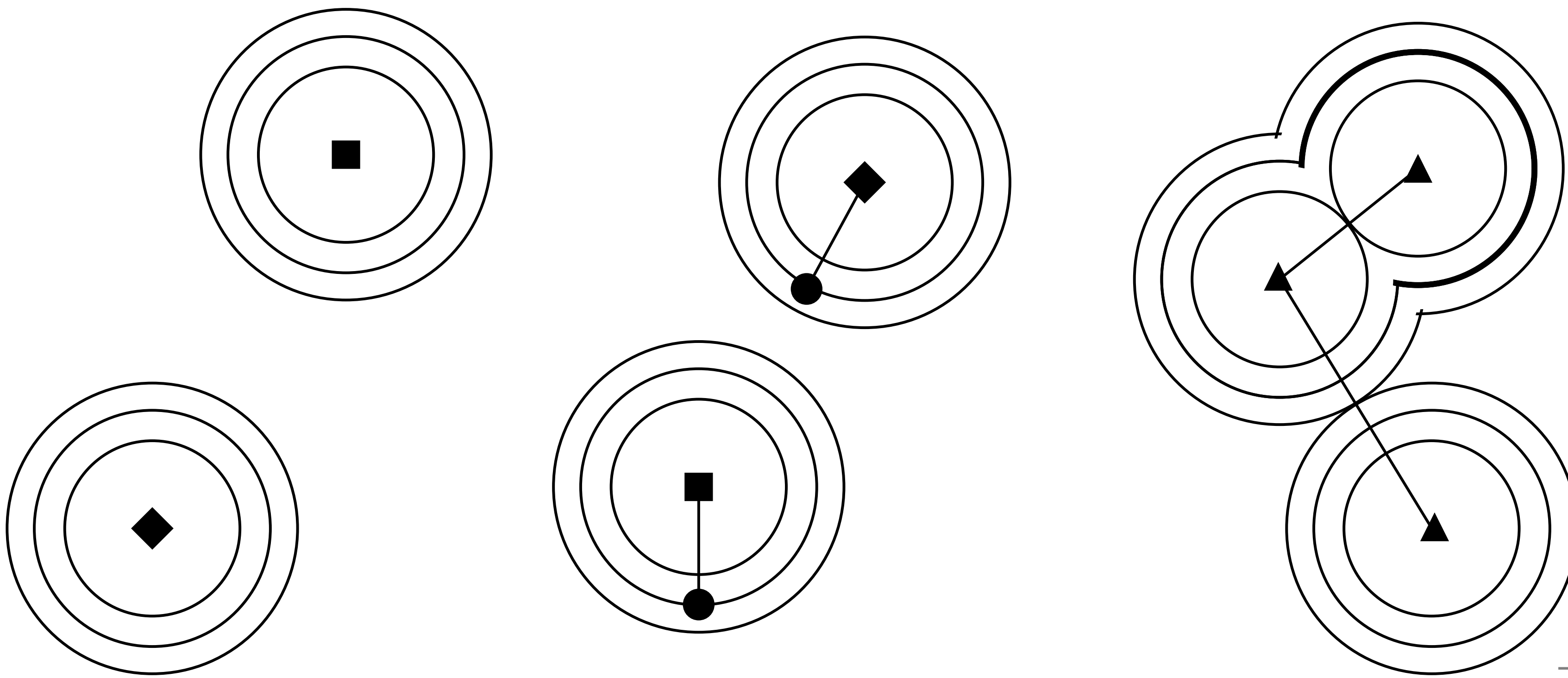


Simulação

Instância: $(G, c, \{R_1, R_2, R_3\})$

G: completo com 9 vértices c: distância euclidiana
R_1, R_2 e R_3: triângulos, losangos e quadrados

Início da quinta iteração:

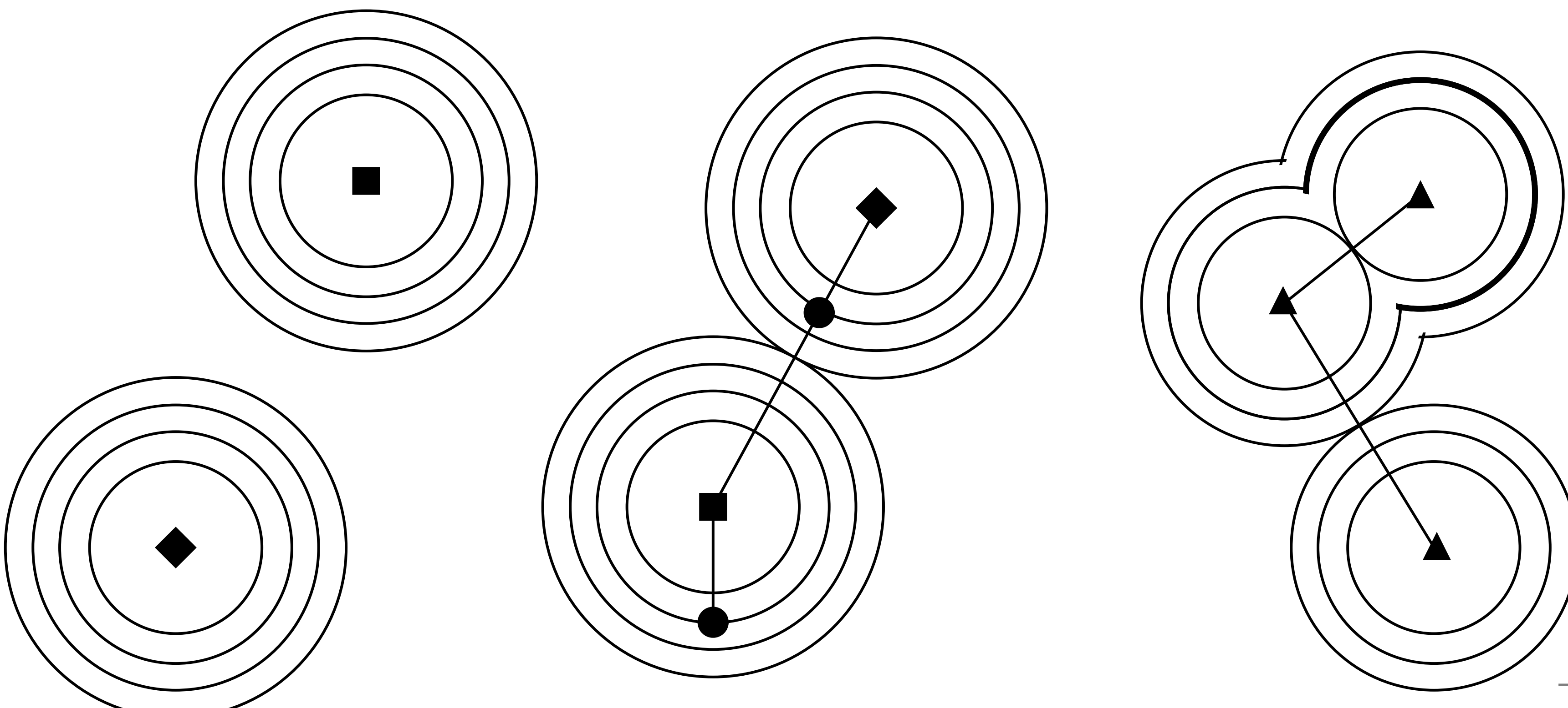


Simulação

Instância: $(G, c, \{R_1, R_2, R_3\})$

G: completo com 9 vértices c: distância euclidiana
R_1, R_2 e R_3: triângulos, losangos e quadrados

Início da sexta iteração:

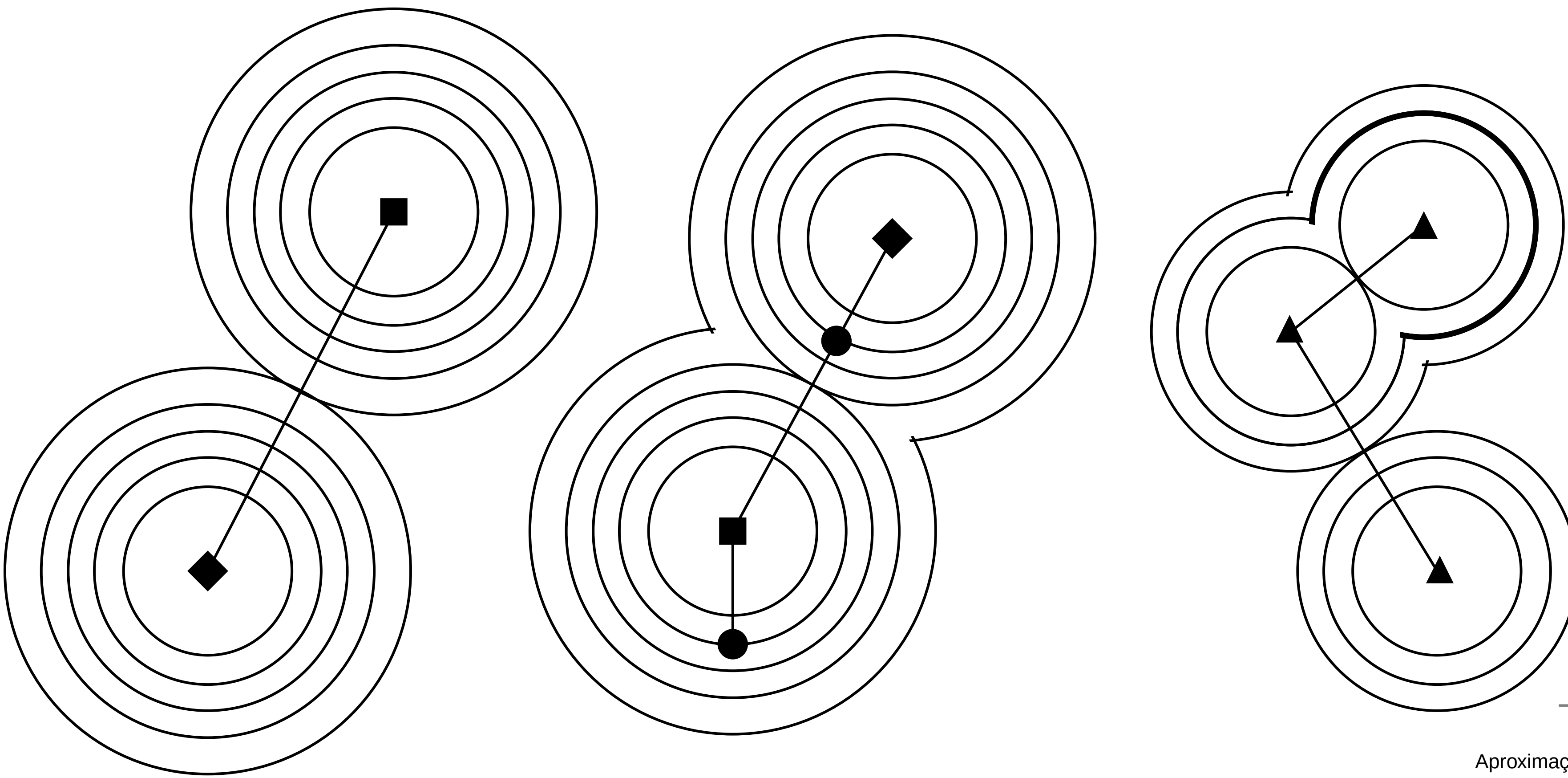


Simulação

Instância: $(G, c, \{R_1, R_2, R_3\})$

G: completo com 9 vértices c: distância euclidiana
R_1, R_2 e R_3: triângulos, losangos e quadrados

Início da sétima iteração:

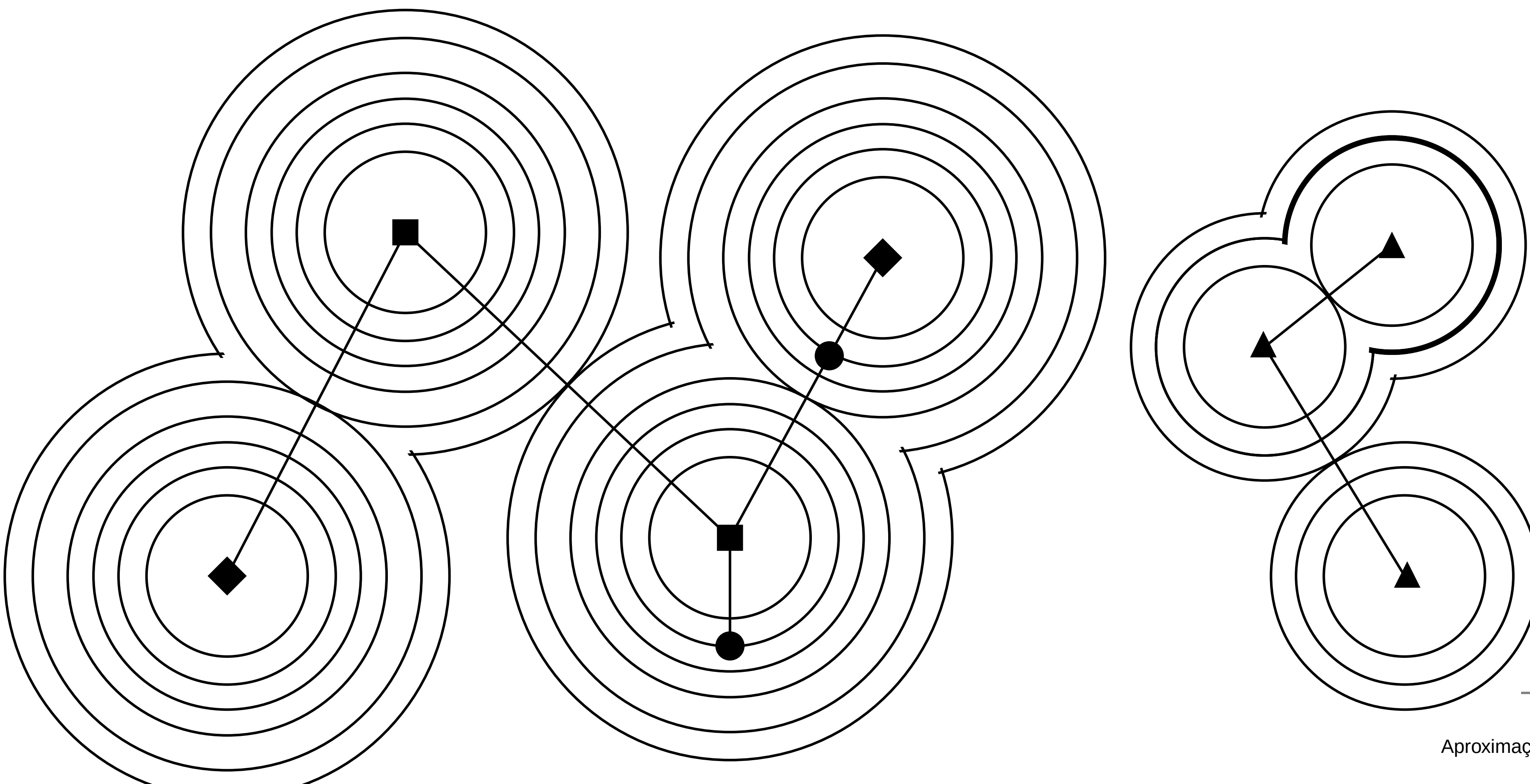


Simulação

Instância: $(G, c, \{R_1, R_2, R_3\})$

G: completo com 9 vértices c: distância euclidiana
R_1, R_2 e R_3: triângulos, losangos e quadrados

Última iteração:

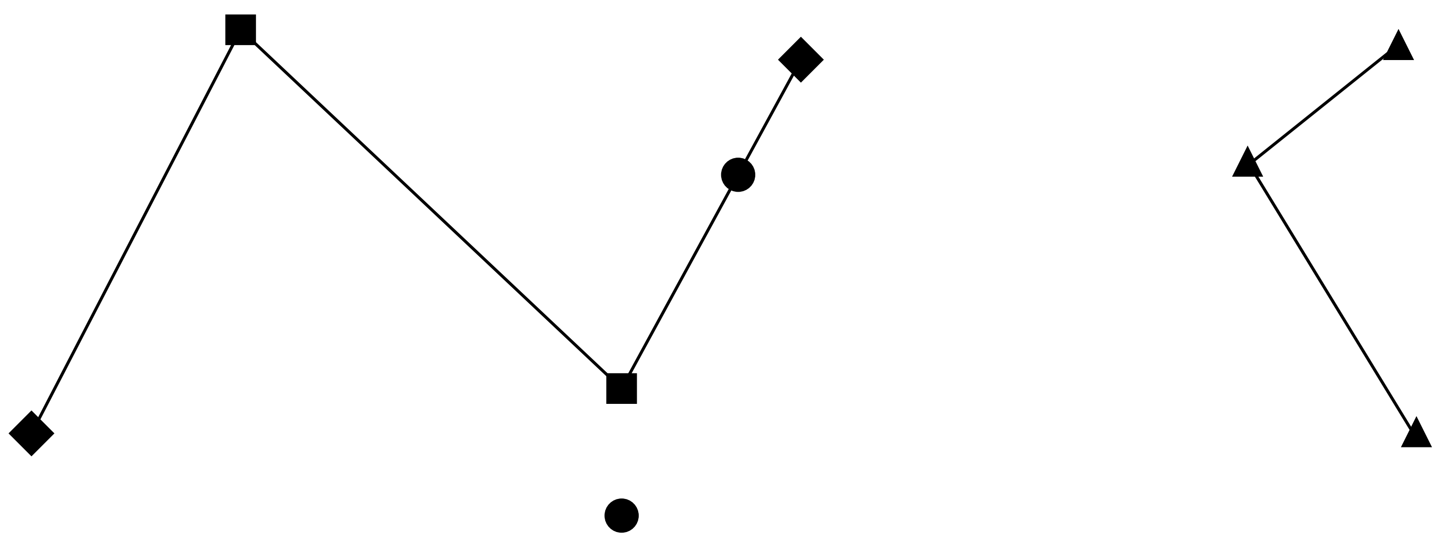


Simulação

Instância: $(G, c, \{R_1, R_2, R_3\})$

G: completo com 9 vértices $\quad$ c: distância euclidiana
R_1, R_2 e R_3: triângulos, losangos e quadrados

Floresta F' final:



Análise

- F é uma floresta.

Análise

- F é uma floresta.

- No início de cada iteração,

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

Análise

- F é uma floresta.

- No início de cada iteração,

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

- PrimalDual é uma 2-aproximação.

Análise

- F é uma floresta.

- No início de cada iteração,

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

- PrimalDual é uma 2-aproximação.

- Há uma implementação $\mathrm{O}(n^2 \lg n)$ de PrimalDual, onde n é o número de vértices do grafo.

Análise

Lema: F é uma floresta.

Análise

Lema: F é uma floresta.

Prova: Toda vez inserimos uma aresta que tem exatamente uma ponta em uma componente de F.

Análise

Lema: F é uma floresta.

Prova: Toda vez inserimos uma aresta que tem exatamente uma ponta em uma componente de F.

Implementação:

Como implementar o algoritmo PRIMALDUAL de modo que consuma tempo $\mathrm{O}(n^2 \lg n)$, onde n é o número de vértices do grafo?

Análise

Lema: F é uma floresta.

Prova: Toda vez inserimos uma aresta que tem exatamente uma ponta em uma componente de F.

Implementação:

Como implementar o algoritmo PRIMALDUAL de modo que consuma tempo $\mathrm{O}(n^2 \lg n)$, onde n é o número de vértices do grafo?

Implementação

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'

Implementação

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'

Para cada vértice v, armazene $d(v) = \sum_{S:v\in S} y_S$.

Implementação

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

```
para cada S em 𝒮 faça y_S ← 0
F ← ∅
enquanto 𝒜_F ≠ ∅ faça
        ε ← min{c_e − Σ_{S':e∈δ(S')} y_{S'} : e ∈ δ(S) e S ∈ 𝒜_F}
        f ← arg min{c_e − Σ_{S':e∈δ(S')} y_{S'} : e ∈ δ(S) e S ∈ 𝒜_F}
        para cada S ∈ 𝒜_F faça y_S ← y_S + ε
        F ← F ∪ {f}
seja F' uma ℛ-floresta minimal em F
devolva F'
```

Para cada vértice v, armazene $d(v) = \sum_{S:v\in S} y_S$.
Para cada aresta uv externa a F,
$\theta_{uv} := \infty$ se uv liga dois componentes inativos de F,
$\theta_{uv} := c_e - d(u) - d(v)$ se uv liga componente ativo a inativo,
e $\theta_{uv} := \frac{1}{2}(c_e - d(u) - d(v))$ se uv liga dois ativos.

Implementação

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

```
para cada S em 𝒮 faça y_S ← 0
  F ← ∅
  enquanto 𝒜_F ≠ ∅ faça
      ε ← min{c_e − Σ_{S′:e∈δ(S′)} y_{S′} : e ∈ δ(S) e S ∈ 𝒜_F}
      f ← arg min{c_e − Σ_{S′:e∈δ(S′)} y_{S′} : e ∈ δ(S) e S ∈ 𝒜_F}
      para cada S ∈ 𝒜_F faça y_S ← y_S + ε
      F ← F ∪ {f}
  seja F′ uma 𝓡-floresta minimal em F
  devolva F′
```

Para cada vértice v, armazene $d(v) = \sum_{S: v \in S} y_S$.

Para cada aresta uv externa a F,

$\theta_{uv} := \infty$ se uv liga dois componentes inativos de F,

$\theta_{uv} := c_e - d(u) - d(v)$ se uv liga componente ativo a inativo,

e $\theta_{uv} := \frac{1}{2}(c_e - d(u) - d(v))$ se uv liga dois ativos.