Capítulo 5

Método Primal-Dual

Este capítulo apresenta o método de aproximação primal-dual. Trata-se de uma generalização do método primal-dual clássico que tem sido aplicada com bastante sucesso ao projeto de algoritmos de aproximação.

Neste capítulo, mostramos os algoritmo obtidos da aplicação do método de aproximação primal-dual ao problema da transversal mínima, ao problema do caminho mínimo e ao problema da floresta de Steiner.

5.1 Seção vazia [era primal-dual clássico]

5.2 Método de aproximação primal-dual

Sejam M e N os conjuntos de índices de linhas e colunas de uma matriz A. Seja b um vetor indexado por M e c um vetor indexado por N. Considere o seguinte problema de programação linear: encontrar um vetor x indexado por N que

$$\begin{array}{rrcll} \text{minimize} & c\,x & & & \\ \text{sob as restrições} & (Ax)_i & \geq & b_i & \text{para cada } i \text{ em } M\,, \\ & x_j & \geq & 0 & \text{para cada } j \text{ em } N\,. \end{array} \tag{5.1}$$

Diremos que esse é **o problema primal**. Um vetor x é **viável** nesse problema (ou é uma **solução viável** do problema) se satisfaz as restrições, ou seja, se $x \geq$ e $Ax \geq b$.

O dual do problema (5.1) consiste em encontrar um vetor y indexado por M que

$$\begin{array}{rrcll} \text{maximize} & y\,b & & & \\ \text{sob as restrições} & (yA)_j & \leq & c_j & \text{para cada } j \text{ em } N\,, \\ & y_i & \geq & 0 & \text{para cada } i \text{ em } M\,. \end{array} \tag{5.2}$$

Diremos que esse é **o problema dual**. Um vetor y é **viável** nesse problema se $y \geq 0$ e $yA \leq c$. A relação fundamental entre os dois problemas é resumida no lema da dualidade (lema C.1, apêndice C): se x é viável no primal e y é viável no dual então $cx \geq yb$. Segue daí que se $cx = yb$ então x é solução ótima do primal e y é solução ótima do dual.

Dois vetores x e y, indexados por M e N respectivamente, têm **folgas primais complementares** se, para cada índice j em N, tem-se que

$$x_j = 0 \quad \text{ou} \quad (yA)_j = c_j\,.$$

Analogamente, x e y têm **folgas duais complementares** se, para cada índice i em M, tem-se

$$y_i = 0 \quad \text{ou} \quad (Ax)_i = b_i\,.$$

Se x e y têm folgas complementares então é fácil verificar (veja lema C.2, apêndice C) que $cx = yb$. Portanto, se os vetores x e y são viáveis (no primal e dual respectivamente) e têm folgas primais e duais complementares então x é solução ótima de (5.1) e y é solução ótima de (5.2).

Para discutir o método de aproximação primal-dual é preciso relaxar a definição de folgas complementares. Antes de fazer isso, convém retringir a atenção aos problemas em que

$$A \geq 0\,, \quad b \geq 0 \quad \text{e} \quad c \geq 0\,. \tag{5.3}$$

Com essa restrição, o vetor nulo, $y = 0$, é solução viável do problema dual e o problema primal também tem soluções viáveis. Agora suponha dados números α e β tais que α, β

$$0 < \alpha \leq 1 \leq \beta\,.$$

Dizemos que dois vetores x e y têm **folgas primais α-complementares** se, para cada índice j em N,

$$x_j = 0 \quad \text{ou} \quad \alpha c_j \leq (yA)_j \leq c_j\,.$$

Analogamente, x e y têm **folgas duais β-complementares** se, para cada índice i em M,

$$y_i = 0 \quad \text{ou} \quad b_i \leq (Ax)_i \leq \beta b_i\,.$$

Diremos que essas condições definem folgas complementares **aproximadas**. O lema seguinte generaliza a primeira parte do lema C.2 (apêndice C):

Lema 5.1 (das folgas complementares aproximadas): *Se vetores não-negativos x e y têm folgas primais α-complementares e folgas duais β-complementares então*

$$y\,b \le c\,x \le \tfrac{\beta}{\alpha}\,y\,b\,.$$

Demonstração: Em virtude do lema da dualidade (lema C.1, apêndice C), temos $yb \le cx$. Em virtude das folgas complementares aproximadas,

$$\alpha\,c\,x \;\le\; (y A)\,x \;=\; y\,(A x) \;\le\; \beta\,y\,b\,,$$

onde a primeira desigualdade segue da não-negatividade de x e a segunda da não-negatividade de y. ∎

Corolário 5.2: *Se x é solução viável do problema primal e tem folgas primais α-complementares e folgas duais β-complementares com uma solução viável y do problema dual então*

$$c\,\hat{x} \le c\,x \le \tfrac{\beta}{\alpha}\,c\,\hat{x}\,,$$

onde $\hat{x}$ é solução ótima do problema primal.

Demonstração: Como x é viável, temos $c\hat{x} \le cx$. As demais desigualdades seguem do lema acima e do lema da dualidade (lema C.1, apêndice C). ∎

Esse corolário é a base do **método de aproximação primal-dual**. O método procura encontrar uma solução aproximada do problema primal, ou seja, uma solução viável cujo valor esteja próximo do valor ótimo. Vamos adotar $\alpha = 1$ na descrição a seguir, pois esse é o valor de α nas aplicações que veremos nas demais seções do capítulo. O método tem duas fases, que podem ser descritas (de maneira necessariamente vaga) como segue:

> Cada iteração da primeira fase começa com um vetor y viável no dual. (A primeira iteração pode começar com o vetor $y = 0$.) Cada iteração procura encontrar um vetor x que seja viável no primal e tenha folgas primais complementares com y. Em outras palavras, procura-se um vetor x tal que
>
> x é viável no primal,
> $x_k = 0$ para cada k em $N \setminus J$,
>
> sendo $J := \{j \in N : (yA)_j = c_j\}$. Se um tal x for encontrado, podemos J

começar a segunda fase do método. Suponha agora que um tal x não existe. Procuramos então por um vetor y' tal que

$$y + \theta y' \text{ é viável no dual}$$

para algum número θ estritamente positivo. Uma nova iteração começa com $y + \theta y'$ no papel de y.

A segunda fase do método começa com vetores x e y que têm folgas primais complementares, sendo x viável no primal e y viável no dual. Procura-se então por um vetor $\tilde{x} \leq x$ que seja minimal com relação às seguintes propriedades: $\tilde{x}$ é viável no primal e não existe vetor $\tilde{x}' < \tilde{x}$ que seja viável no primal. É claro que $\tilde{x}$ e y têm folgas primais complementares. Resta apenas determinar β tal que $\tilde{x}$ e y têm folgas duais β-complementares e anunciar que $\tilde{x}$ é solução aproximada do problema primal.

Nas aplicações do método a problemas de otimização combinatória, o vetor característico de J é um candidato natural ao papel de x. Teremos então, no início de cada iteração da primeira fase, um vetor x tal que $x_k = 0$ para cada k em $N \setminus J$ e $(Ax)_h \geq b_h$ para cada h em $M \setminus I$ sendo $I := \{i \in M : y_i = 0\}$. Essa última condição pode ser vista como uma parte das condições de folgas β-complementares entre x e y. Além disso, teremos $y' \geq 0$ e $y'_h = 0$ para cada h em $M \setminus I$. Com isso, o conjunto J aumentará e I diminuirá a cada iteração da primeira fase.

I

5.3 Transversal mínima

Seja $\mathcal{S}$ uma coleção finita de subconjuntos de um conjunto finito E. Um subconjunto T de E é uma **transversal** de $\mathcal{S}$ se $T \cap S \neq \emptyset$ para cada S em $\mathcal{S}$. O **problema da transversal mínima** (*hitting set problem*) consiste no seguinte:

> **Problema** MINTC $(E, \mathcal{S}, c)$: *Dados um conjunto finito E, uma coleção finita $\mathcal{S}$ de subconjuntos de E e um custo c_e em $\mathbb{Q}_{\geq}$ para cada e em E, encontrar uma transversal T de $\mathcal{S}$ que minimize $c(T)$.*

Esse problema é equivalente ao MINCC. Diversos problemas combinatórios são casos particulares do MINTC; veja os exercícios no fim do capítulo. Às vezes, dizemos que $c(T)$ é o **custo** da transversal T. Com isso, o problema consiste em encontrar uma transversal de $\mathcal{S}$ de custo mínimo.

O seguinte programa linear é uma relaxação de MinTC $(E, \mathcal{S}, c)$: encontrar um vetor x indexado por E que

$$\begin{array}{rlll} \text{minimize} & c\,x & & \\ \text{sob as restrições} & x(S) \geq 1 & \text{para cada } S \text{ em } \mathcal{S}\,, & \\ & x_e \geq 0 & \text{para cada } e \text{ em } E\,. & \end{array} \tag{5.5}$$

De fato, o vetor característico x de qualquer transversal T é um vetor viável de (5.5) de custo $c(T)$. O correspondente programa dual consiste em encontrar um vetor y indexado por $\mathcal{S}$ que

$$\begin{array}{rlll} \text{maximize} & y(\mathcal{S}) & & \\ \text{sob as restrições} & y(\mathcal{S}_e) \leq c_e & \text{para cada } e \text{ em } E\,, & \\ & y_S \geq 0 & \text{para cada } S \text{ em } \mathcal{S}\,, & \end{array} \tag{5.6}$$

onde $\mathcal{S}_e := \{S \in \mathcal{S} : S \ni e\}$. $\mathcal{S}_e$

Eis as condições de folgas primais complementares para dois vetores x e y: para cada e em E, se $y(\mathcal{S}_e) \neq c_e$ então $x_e = 0$. Eis as condições de folgas duais complementares: para cada S em $\mathcal{S}$, se $y_S \neq 0$ então $x(S) = 1$. Se adotarmos a notação $J := \{e \in E : y(\mathcal{S}_e) = c_e\}$, as condições de folgas primais complementares e as condições de folgas duais β-complementares podem ser escritas assim:

$$\text{para cada } e \text{ em } E \setminus J \text{ tem-se } x_e = 0\,, \tag{5.7}$$

$$\text{para cada } S \text{ em } \mathcal{S}, \text{ se } y_S \neq 0 \text{ então } 1 \leq |J \cap S| \leq \beta\,. \tag{5.8}$$

O algoritmo que discutiremos a seguir resulta da aplicação do método de aproximação primal-dual ao problema MinTC. A descrição abaixo supõe que $\emptyset \notin \mathcal{S}$ e portanto o problema MinTC$(E, \mathcal{S}, c)$ é viável.

Algoritmo MinTC-BE $(E, \mathcal{S}, c)$

```
para cada S em 𝒮 faça y_S ← 0
J ← ∅
enquanto J não é uma transversal faça
    escolha Z em 𝒮 tal que J ∩ Z = ∅
    seja y' o vetor característico de {Z} em 𝒮
    escolha o maior θ tal que
        y(𝒮_e) + θy'_Z ≤ c_e para cada e em Z
    y ← y + θy'
    seja f um elemento de Z tal que y(𝒮_f) = c_f
    J ← J ∪ {f}
devolva J
```

O algoritmo é devido a Bar-Yehuda e Even [BYE81] e foi originalmente concebido para o problema da cobertura mínima por vértices.

Teorema 5.3: *O algoritmo* MinTC-BE *é uma β-aproximação polinomial para o* MinTC $(E, \mathcal{S}, c)$, *onde* $\beta := \max_{S \in \mathcal{S}} |S|$. β

Demonstração: No início de cada iteração do algoritmo, y é viável em (5.6). Ademais, se denotarmos por x o vetor característico de J em E (isto é, $x_e = 1$ se $e \in J$ e $x_e = 0$ em caso contrário), podemos dizer que x e y têm folgas primais complementares e folgas duais β-complementares. Em suma, no início de cada iteração,

(i1) y é viável em (5.6),

(i2) x e y satisfazem (5.7),

(i3) x e y satisfazem (5.8).

(A desigualdade $|J \cap S| \le \beta$ em (5.8) é satisfeita trivialmente pois ela vale para qualquer subconjunto J de E.) No início da última iteração, J é uma transversal de $\mathcal{S}$ e portanto x é viável em (5.5). Assim, de acordo com o corolário 5.2 (com $\alpha = 1$), temos $c\hat{x} \le cx \le \beta c\hat{x}$ no fim da execução do algoritmo, sendo $\hat{x}$ uma solução ótima de (5.5). Portanto,

$$c(J) \;\le\; \beta \operatorname{opt}(E, \mathcal{S}, c)\,.$$

A cardinalidade de J cresce a cada iteração, donde o número de execuções das linhas 3 a 10 do algoritmo é no máximo $|E|$. Ademais, a execução de cada linha consome tempo $O(|E||\mathcal{S}|)$. ∎

Uma segunda fase do algoritmo poderia extrair de J uma transversal minimal e esta poderia ter custo menor que $c(J)$. Infelizmente, essa manobra não chega a melhorar a razão de aproximação do algoritmo.

A próxima seção trata de um caso particular do MinTC. O método de aproximação primal-dual aplicado a esse caso particular resultará em um algoritmo que tem uma razão de aproximação substancialmente melhor que a de MinTC-BE. Esta melhora se deve a dois fatores: (1) a uma escolha cuidadosa do vetor y' e (2) uma segunda fase que extrai uma transversal minimal da transversal produzida pela primeira fase.

5.3a Caminho mínimo

Esta seção examina um caso particular do algoritmo MinTC-BE. Essa discussão é uma boa preparação para o algoritmo MinFS-GW.

Sejam s e t dos vértices de um grafo G. Suponha que cada aresta e de G tem um custo racional $c_e \geq 0$. Seja $\mathcal{S}$ a coleção

$$\{S \subseteq V_G : s \in S,\ t \notin S\}\,.$$

Estamos interessados nas transversais de $\{\delta(S) : S \in \mathcal{S}\}$. Para simplificar, diremos simplesmente "transversal de $\mathcal{S}$". As transversais de $\mathcal{S}$ são objetos bastante familiares. De fato, se P é um caminho de s a t então E_P é uma transversal de $\mathcal{S}$ e, reciprocamente, se T é uma transversal minimal[1] de $\mathcal{S}$ então T é o conjunto de arestas de um caminho de s a t. Portanto, o problema de determinar uma transversal de $\mathcal{S}$ de custo mínimo é idêntico ao seguinte

> **Problema** MinPath (G, s, t, c): *Dado um grafo G e um custo $c_e \geq 0$ para cada aresta e, encontrar um caminho P de s a t que minimize $c(P)$.*

O programa linear primal que corresponde ao problema consiste em encontrar um vetor racional $x \geq 0$ indexado por E_G que minimize $c\,x$ sob as restrições $x(\delta(S)) \geq 1$ para cada S em $\mathcal{S}$. O correspondente programa dual consiste em encontar um vetor racional $y \geq 0$ indexado por $\mathcal{S}$ que maximize $y(\mathcal{S})$ sob as restrições $y(\mathcal{S}_e) \leq c_e$ para cada e em E_G, onde

$$\mathcal{S}_e := \{S \in \mathcal{S} : \delta(S) \ni e\}\,.$$

Eis as condições de folgas primais complementares: para cada e em E_G, se $y(\mathcal{S}_e) \neq c_e$ então $x_e = 0$. Eis as condições de folgas duais complementares: para cada S em $\mathcal{S}$, se $y_S \neq 0$ então $x(\delta(S)) = 1$.

No algoritmo abaixo, x terá valores em $\{0, 1\}$ e será representado pelo conjunto $X := \{e \in E_G : x_e = 1\}$. Assim, x é viável se e somente se X é uma transversal. Nesse contexto, as condições de folgas complementares podem ser reescritas assim:

(P) para cada e, se $y(\mathcal{S}_e) \neq c_e$ então $e \notin X$ e

(D) para cada S, se $y_S \neq 0$ então $|X \cap \delta(S)| = 1$.

O algoritmo abaixo tem essas condições de folgas complementares por base. Ele é uma adaptação do MinTC-BE e um caso particular do MinFS-GW. Vamos supor que G tem um caminho de s a t para evitar as instâncias inviáveis.

Algoritmo MinPath-Primal-Dual (G, s, t, c)

$\mathcal{S} \leftarrow \{S \subseteq V_G : s \in S, t \notin S\}$
$X \leftarrow \emptyset$
para cada S em $\mathcal{S}$ faça $y_S \leftarrow 0$
enquanto X não é transversal de $\mathcal{S}$ faça
    seja Z um elemento minimal de $\{S \in \mathcal{S} : X \cap \delta(S) = \emptyset\}$

[1] Uma transversal é *minimal* se, para toda e em T, o conjunto $T \setminus \{e\}$ não é uma transversal.

06 seja y' o vetor característico de Z
07 determine o maior número θ tal que
08 $y(\mathcal{S}_e) + \theta y'_Z \leq c_e$ para cada e em $\delta(Z)$
09 $y \leftarrow y + \theta y'$
10 seja f uma aresta em $\delta(Z)$ tal que $y(\mathcal{S}_f) = c_f$
11 $X \leftarrow X \cup \{f\}$
12 seja P um caminho de s a t no grafo (V_G, X)
13 devolva E_P

No começo de cada iteração do algoritmo, valem as seguintes propriedades invariantes, com $\mathcal{S}_> := \{S \in \mathcal{S} : y_S \neq 0\}$:

(i1) y é viável;

(i2) X e y satisfazem as condições de folgas primais complementares;

(i3) o grafo (V_G, X) é uma floresta; essa floresta tem no máximo um componente não-trivial e esse componente contém s;

(i4) para cada S em $\mathcal{S}_>$ e cada v em S, o caminho que liga s a v em (V_G, X) está todo em S.

Portanto, X é uma transversal, o vetor y é viável e as condições de folgas primais complementares estão satisfeitas no início da linha 12. No fim da linha 12, é evidente que E_P é uma transversal e o par E_P, y satisfaz as condições de folgas primais complementares. Além disso, diante da invariante (i4), podemos afirmar que

$$|E_P \cap \delta(S)| \leq 1 \text{ para cada } S \text{ em } \mathcal{S}_>, \tag{5.9}$$

donde se conclui imediatamente que $|E_P \cap \delta(S)| = 1$ para cada S em $\mathcal{S}_>$, ou seja, temos folgas duais complementares.

A validade das folgas duais complementares tem duas origens: a primeira é a escolha cuidadosa de Z na linha 05 (o MinTC-BE escolheria *qualquer* Z tal que $X \cap \delta(Z) = \emptyset$); e a segunda é o descarte de uma parte de X na linha 12.

Assim, no fim da execução do algoritmo temos

$$\begin{aligned}
c(P) &= \textstyle\sum_{e \in E_P} c_e \\
&= \textstyle\sum_{S \in \mathcal{S}_>} |E_P \cap \delta(S)|\, y_S \\
&= \textstyle\sum_{S \in \mathcal{S}_>} y_S \\
&= y(\mathcal{S}) \\
&\leq \hat{y}(\mathcal{S}) \\
&= c\,\hat{x} \\
&\leq c(P_*) \\
&= \mathrm{opt}(G, s, t, c)\,,
\end{aligned}$$

sendo $\hat{x}$ e $\hat{y}$ soluções ótimas dos programas primal e dual definidos no início da seção e sendo P_* uma solução ótima de MINPATH(G, s, t, c).

Portanto, nosso algoritmo é uma 1-aproximação — ou seja, um algoritmo exato — para o problema MINPATH.

Para implementar efetivamente o algoritmo, convém trocar $\mathcal{S}$ por $\mathcal{S}_>$ (o conjunto começa vazio e é acrescido de Z imediatamente depois da linha 05). A cardinalidade de $\mathcal{S}_>$ é limitada por $|V_G| - 1$ pois $\mathcal{S}_>$ é "encaixada", ou seja, para cada par S, S' de elementos de $\mathcal{S}_>$, tem-se $S \subset S'$ ou $S \supset S'$.

Uma implementação efetiva pode simplificar as coisas ainda mais trocando as variáveis y por um vetor z indexado por V_G: para cada v em V_G, $z_v := \sum_{S \ni v} y_S$. Com isso, o algoritmo fica muito semelhante ao clássico algoritmo de Dijkstra.

Versão "homogênea"

Refaça o algoritmo de modo que ele seja simétrico em relação a s e t. Adote $\mathcal{S} := \{S \subseteq V_G : |S \cap \{s, t\}| = 1\}$.

5.4 Floresta de Steiner

Seja G um grafo e $\mathcal{R}$ uma coleção de partes de V_G. Um conjunto X de arestas é uma $\mathcal{R}$-**floresta** se o grafo (V_G, X) é uma floresta e todo elemento de $\mathcal{R}$ está contido em algum componente dessa floresta.

> **Problema** MINFS $(G, c, \mathcal{R})$: *Dado um custo $c_e \geq 0$ para cada aresta e de G, encontrar uma $\mathcal{R}$-floresta de custo mínimo.*

Um conjunto de vértices S é **ativo**[2] se $R \cap S \neq \emptyset$ e $R \setminus S \neq \emptyset$ para algum R em $\mathcal{R}$. Seja

$$\mathcal{S} := \{S \subseteq V_G : S \text{ é ativo}\} .$$

Uma $\mathcal{R}$-floresta é essencialmente o mesmo que uma transversal da coleção de cortes $\{\delta(S) : S \in \mathcal{S}\}$.

O programa linear que corresponde ao problema consiste em encontrar um vetor racional $x \geq 0$ indexado por E_G que minimize $c\,x$ sob as restrições $x(\delta(S)) \geq 1$ para cada S em $\mathcal{S}$. O correspondente programa dual consiste em encontar um vetor racional $y \geq 0$ indexado por $\mathcal{S}$ que maximize $y(\mathcal{S})$ sob as restrições $y(\mathcal{S}_e) \leq c_e$ para

[2] Teria sido melhor usar o termo *separador*.

cada e em E_G, onde

$$\mathcal{S}_e := \{S \in \mathcal{S} : \delta(S) \ni e\}\,.$$

Eis as condições de folgas primais complementares: para cada e em E_G, se $y(\mathcal{S}_e) \neq c_e$ então $x_e = 0$. Eis as condições de folgas duais 2-complementares: para cada S em $\mathcal{S}$, se $y_S \neq 0$ então $1 \leq x(\delta(S)) \leq 2$.

No algoritmo abaixo, x terá valores em $\{0,1\}$ e será representado pelo conjunto $X := \{e \in E_G : x_e = 1\}$. Assim, x é viável se e somente se X é uma $\mathcal{R}$-floresta. Nesse contexto, as condições de folgas complementares podem ser reescritas assim:

(PA) se $y(\mathcal{S}_e) \neq c_e$ então $e \notin X$ e

(DA) para cada S, se $y_S \neq 0$ então $1 \leq |X \cap \delta(S)| \leq 2$.

O algoritmo abaixo tem essas condições de folgas complementares por base. Para evitar as instâncias inviáveis, vamos supor que G é conexo.

Algoritmo MINFS-GW $(G, c, \mathcal{R})$

$\mathcal{S} \leftarrow \{S \subseteq V_G : S$ é ativo$\}$
para cada S em $\mathcal{S}$ faça $y_S \leftarrow 0$
$X \leftarrow \emptyset$
enquanto X não é uma $\mathcal{R}$-floresta faça
    seja $\mathcal{S}_X$ a coleção dos elementos
        minimais de $\{S \in \mathcal{S} : X \cap \delta(S) = \emptyset\}$
    seja y' o vetor característico de $\mathcal{S}_X$
    escolha o maior θ tal que $y(\mathcal{S}_e) + \theta y'(\mathcal{S}_X) \leq c_e$
        para cada aresta externa e
    $y \leftarrow y + \theta y'$
    seja f uma aresta externa tal que $y(\mathcal{S}_f) = c_f$
    $X \leftarrow X \cup \{f\}$
seja M uma $\mathcal{R}$-floresta minimal de (V_G, X)
devolva M

Se $S \in \mathcal{S}_X$ então S é um componente do grafo (V_G, X). Reciprocamente, se um componente S de (V_G, X) é ativo então $S \in \mathcal{S}_X$. É claro que os elementos de $\mathcal{S}_X$ são disjuntos dois a dois. Uma aresta é **externa** se tem pontas em dois elementos distintos de $\mathcal{S}_X$.

Eis as invariantes do algoritmo: no início de cada iteração,

(i1) y é viável,

(i2) X e y satisfazem as condições (PA),

(i3) o grafo (V_G, X) é uma floresta,

(i4) $\sum_{S\in\mathcal{S}} |M \cap \delta(S)|\, y_S \le 2 \sum_{S\in\mathcal{S}} y_S$, sendo M o conjunto devolvido na linha 14.

(A invariante (i4) é um bocado estranha pois envolve o conjunto M, que será conhecido apenas no fim do processo iterativo. Mas isso deve ser entendido como uma abreviatura da seguinte afirmação: a desigualdade vale para qualquer $\mathcal{R}$-floresta minimal M que esteja contida em qualquer $\mathcal{R}$-floresta que inclua X.)

A propósito, a determinação do conjunto M na linha 13 pode ser detalhada pelo seguinte processo iterativo. Cada iteração começa com um conjunto M de arestas, sendo $M = X$ no início da primeira iteração. Em cada iteração, seja F a floresta (V_G, M). Para cada e em M, tome a componente T de F que contém e e verifique se existe R em $\mathcal{R}$ que intercepta cada um dos dois componentes de $T - e$. Se tal R não existe, retire e de M.

Prova de (i4): A prova seria fácil se tivéssemos $|M \cap \delta(S)| \le 2$ para todo S, ou seja, se estivessem satisfeitas as folgas duais complementares. Infelizmente, isso só vale "em média":

$$\frac{1}{|\mathcal{S}_X|} \sum_{S\in\mathcal{S}_X} |M \cap \delta(S)| \le 2\,. \tag{5.10}$$

Eis a prova dessa desigualdade. . . [veja lema 5.4].

Uma vez demonstrada a relação (5.10), a invariante (i4) segue por indução. No fim de cada iteração,

$$\begin{aligned} \sum_{S\in\mathcal{S}} |M \cap \delta(S)|(y+\theta y')_S &= \sum_{S\in\mathcal{S}} |M \cap \delta(S)|\, y_S + \sum_{S\in\mathcal{S}_X} |M \cap \delta(S)|\, \theta \\ &\le 2\, y(\mathcal{S}) + 2\theta|\mathcal{S}_X| \\ &= 2\, y(\mathcal{S}) + 2\theta y'(\mathcal{S}) \\ &= 2\,(y+\theta y')(\mathcal{S})\,. \end{aligned}$$

Isso prova (i4).

Teorema 5.5: MinFS-GW é uma 2-aproximação.

PROVA: De (i4) segue que

$$\begin{aligned}
c(M) &= \sum_{e \in M} c_e \\
&= \sum_{e \in M} y(\mathcal{S}_e) \\
&= \sum_{e \in M} \sum_{\delta(S) \ni e} y_S \\
&= \sum_{S \in \mathcal{S}} |M \cap \delta(S)|\, y_S \\
&\leq 2\, y(\mathcal{S}) \\
&\leq 2\, \hat{y}(\mathcal{S}) \\
&= 2\, c\, \hat{x} \\
&\leq 2\, \mathrm{opt}(G, c, \mathcal{R}) ,
\end{aligned}$$

sendo $\hat{x}$ uma solução ótima da relaxação linear do problema e $\hat{y}$ uma solução ótima do dual. Isso prova o teorema.