Implementação e Teste de Algoritmos de Aproximação para o Problema de Steiner com Grupos

Fernando Mario de Oliveira Filho
Orientador: Prof. Dr. Carlos Eduardo Ferreira

25 de junho de 2004

1 Introdução

Nesta monografia apresentamos o texto preparado para o exame de qualificação de mestrado do aluno Fernando Mario de Oliveira Filho. O projeto de mestrado consiste de um estudo do problema de Steiner com grupos (que será descrito em detalhes na seção 2). O problema consiste em, dado um grafo G com custos nas arestas e uma coleção $\mathcal{R}$ de subconjuntos de $V(G)$, encontrar um subgrafo de G que conecte pelo menos um vértice de cada conjunto em $\mathcal{R}$ e tenha custo mínimo.

Este problema surge naturalmente em aplicações práticas. Nosso objetivo neste projeto é estudar os algoritmos de aproximação propostos para o problema e implementá-los, assim como desenvolver um algoritmo exato para o problema segundo a estratégia *branch-and-cut*. Isto permitirá uma análise empírica da qualidade das soluções produzidas pelos algoritmos de aproximação estudados. O projeto, portanto, envolve não só o estudo dos algoritmos de aproximação, como também de formulações lineares e heurísticas para o problema.

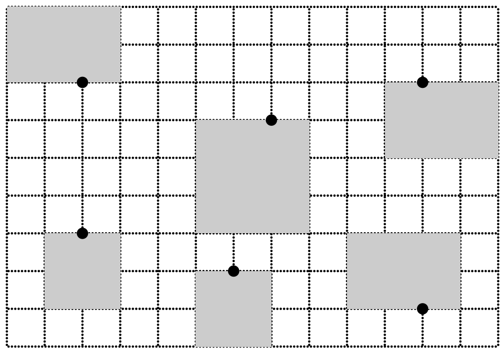
O restante deste texto está organizado da seguinte forma. Na seção 2, apresentamos o problema de Steiner com grupos e abordamos brevemente sua complexidade computacional. Na seção 3, damos uma visão geral sobre os algoritmos de aproximação que foram propostos para o problema e que estudaremos e abordamos em mais detalhe dois destes algoritmos. Na seção 4, descrevemos alguns resultados que obtivemos durante um estudo de possíveis formulações para o problema. Finalmente na seção 5, incluímos um resumo do que já foi feito até o momento e um cronograma para a finalização deste projeto.

2 O problema

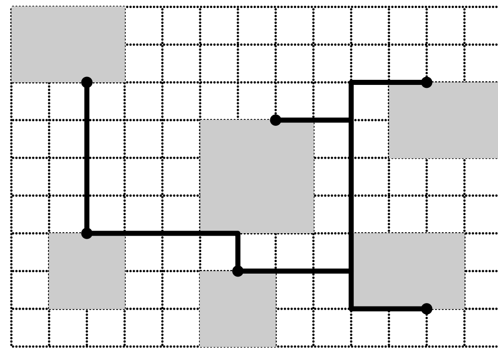
2.1 Roteamento em circuitos VLSI

Há duas fases importantes do desenvolvimento de circuitos VLSI: a fase de *colocação*, em que se decide o lugar de cada componente no circuito, e a fase de *roteamento*, em que se conecta os componentes por fios condutores. A Figura 1a mostra um circuito com seus diversos componentes. Cada componente tem um pino conector; durante a fase de roteamento desejamos ligar os pinos conectores através de uma rede condutora. Podemos usar para tanto os segmentos verticais e horizontais indicados.

A questão é então como obter um roteamento de custo mínimo, sendo o custo definido de alguma forma razoável dependendo da aplicação que se tem em mente. O problema de Steiner em grafos, um dos problemas de otimização combinatória mais estudados, é muito usado para modelar o problema do roteamento em circuitos VLSI. O problema de Steiner em grafos é o seguinte:



(a)



(b)

Figura 1: (a) A representação de um circuito. Os componentes estão representados em cinza e os pinos conectores estão destacados. (b) Uma solução ótima para a instância do problema de Steiner em grafos obtida do circuito em (a). Supomos que cada aresta tenha custo 1. O custo da solução é 24.

Problema PSG (G, Z, c): Dado um grafo G, um conjunto $Z \subseteq V(G)$ e um custo $c(e) \in \mathbb{Q}_+$ para cada $e \in E(G)$, encontar um subgrafo de G que conecte todos os vértices em Z e que tenha custo mínimo.

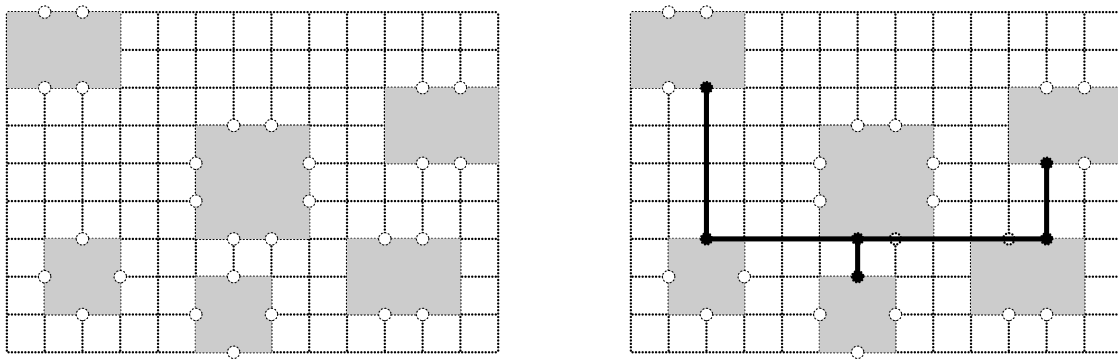
Chamamos os vértices de Z de *terminais*, os demais vértices são chamados de *não-terminais* ou *vértices de Steiner*. Usar o PSG para modelar o problema do roteamento é simples: associamos um grafo ao circuito da forma natural e consideramos os pinos conectores como sendo os terminais. A Figura 1b mostra uma solução ótima para a instância do problema de Steiner associada ao problema de roteamento da Figura 1a.

O problema de Steiner, entretanto, não leva em conta outra operação que podemos fazer com os componentes depois deles já terem sido colocados no circuito. Após a localização de um componente ter sido determinada, o componente ainda pode ser rotacionado ou espelhado. Estas operações podem mudar a posição do pino conector de um dado componente, permitindo que melhores roteamentos sejam encontrados. Por este motivo, faz sentido ao invés de determinar as rotações e espelhamentos de forma arbitrária, encontrar o conjunto de rotações e espelhamentos que minimize o custo do roteamento. Para este propósito Reich e Widmayer [18] propuseram o problema de Steiner com grupos:

Problema PSGrp $(G, \mathcal{R}, c)$: Dado um grafo G, uma coleção $\mathcal{R}$ de subconjuntos de $V(G)$ e um custo $c(e) \in \mathbb{Q}_+$ para cada $e \in E(G)$, encontrar um subgrafo de G que conecte pelo menos um vértice de cada $R \in \mathcal{R}$ e que tenha custo mínimo.

Os elementos de $\mathcal{R}$ são chamados de *grupos*. Um vértice que pertence a algum grupo é um *vértice de grupo*. Os demais vértices são chamados de *não-terminais* ou *vértices de Steiner*. Para modelar o problema do roteamento usando o PSGrp nós associamos a cada componente um grupo de vértices contendo os vértices do grafo que representam as possíveis localizações do pino conector do componente após rotações e espelhamentos. A Figura 2b mostra uma solução ótima para a instância do PSGrp associada ao problema de roteamento da Figura 1a. Note que o roteamento obtido tem custo menor que o roteamento da Figura 1b.

Vale observar que podemos considerar que os grupos são dois a dois disjuntos. De fato, considere uma instância $(G, \mathcal{R}, c)$ do PSGrp e suponha que temos $R_1, R_2 \in \mathcal{R}$ tais que $R_1 \cap R_2 \neq \emptyset$. Tomamos um vértice arbitrário $v \in R_1 \cap R_2$ e inserimos no grafo vértices v_1, v_2, ligados a v por arestas de custo zero. Removemos v de R_1 e R_2 e inserimos v_1 em R_1 e v_2 em R_2. Com isso, diminuímos a cardinalidade de $R_1 \cap R_2$ em 1. Uma solução da instância modificada corresponde a uma solução de mesmo custo da



(a) (b)

Figura 2: (a) A instância do PSGRP associada ao problema do roteamento da Figura 1a. As possíveis localizações dos conectores de um componente estão destacadas. Conectores num mesmo componente estão num mesmo grupo. (b) Uma solução ótima para a instância em (a). O custo da solução é 16.

instância original e vice-versa. Repetindo a operação descrita, podemos então obter uma instância em que os grupos são dois a dois disjuntos e que é equivalente à original. De agora em diante, vamos supor que os grupos são dois a dois disjuntos.

Também é fácil ver que podemos supor que G é conexo e simples. De fato, caso G seja desconexo ou o problema é inviável ou podemos resolver o problema para cada componente de G em separado e escolher a melhor solução. Se G não for simples, podemos remover todas as arestas entre cada par de vértices, exceto a de menor custo. De agora em diante, vamos supor que G é conexo e simples.

2.2 Complexidade computacional

Nesta seção apresentamos rapidamente os resultados de complexidade mais importantes que envolvem o PSGRP.

Primeiramente, observe que o PSG é o caso particular do PSGRP em que cada grupo é unitário. Portanto, como o PSG é $\mathcal{NP}$-difícil [17], o PSGRP também é $\mathcal{NP}$-difícil. A seguir, vamos fornecer uma outra prova deste resultado que envolve o problema da cobertura mínima por conjuntos. Para introduzir este problema, precisamos antes de uma definição simples. Dado um conjunto finito E e uma coleção $\mathcal{C}$ de subconjuntos de E, dizemos que $\mathcal{C}$ *cobre* E se cada elemento de E pertence a algum conjunto de $\mathcal{C}$. Com isso, o problema da cobertura mínima por conjuntos é o seguinte:

> **Problema** MINCC $(E, \mathcal{S}, c)$: Dado um conjunto finito E, uma coleção $\mathcal{S}$ de subconjuntos de E e um custo $c(S) \in \mathbb{Q}_+$ para cada $S \in \mathcal{S}$, encontrar uma subcoleção $\mathcal{S}'$ de $\mathcal{S}$ que cobre E e tenha custo mínimo.

O problema da cobertura mínima por conjuntos é $\mathcal{NP}$-difícil [9] mesmo quando os custos são unitários. Também existem resultados fortes de inaproximabilidade sobre o MINCC, como o seguinte resultado de Feige [6]:

Teorema 1: *Para qualquer* $\epsilon > 0$ *não existe uma* $(1-\epsilon)\ln|E|$*-aproximação polinomial para o* MINCC *a não ser que* $\mathcal{NP} \subseteq \mathrm{DTIME}(n^{O(\log\log n)})$.

A seguir, mostramos uma redução do MINCC ao PSGRP. Dada uma instância $(E, \mathcal{S}, c)$ do MINCC, considere a instância $(G, \mathcal{R}, c')$ do PSGRP onde

(i) G contém um vértice r e um vértice v_S adjacente a r para cada $S \in \mathcal{S}$;

(ii) para cada $a \in E$ temos um grupo $R_a \in \mathcal{R}$ onde $R_a := \{v_S \in V(G)\colon a \in S\}$;

(iii) $c'(rv_S) := c(S)$ para cada $S \in \mathcal{S}$.

Suponha que $\mathcal{S}'$ seja uma solução viável do MinCC $(E, \mathcal{S}, c)$. Ponha $X := \{v_S \in V(G)\colon S \in \mathcal{S}'\}$. É fácil ver que $G[X]$ é uma solução viável do PSGrp $(G, \mathcal{R}, c')$ e que $c(\mathcal{S}') = c'(E(G[X]))$. Analogamente, dada uma solução viável do PSGrp $(G, \mathcal{R}, c')$, podemos construir uma solução viável do MinCC $(E, \mathcal{S}, c)$ de mesmo custo. Além disso, a transformação descrita obviamente pode ser executada em tempo polinomial.

Observe também que podemos aplicar o método descrito no final da seção 2.1 e obter uma instância equivalente à instância $(G, \mathcal{R}, c')$ na qual os grupos são dois a dois disjuntos. Mais ainda, a instância resultante desta transformação é uma árvore. Desta forma, qualquer algoritmo que é capaz de resolver o PSGrp restrito a árvores em tempo polinomial pode ser adaptado para resolver o MinCC em tempo polinomial. Acabamos de provar o seguinte resultado:

Teorema 2: *O problema de Steiner com grupos em árvores é $\mathcal{NP}$-difícil.* □

Como nossa transformação preserva o custo das soluções, também é verdade que uma α-aproximação polinomial para o PSGrp restrito a árvores pode ser adaptada de modo a obtermos uma α-aproximação polinomial para o MinCC. Com isso, temos o seguinte:

Teorema 3: *Para qualquer $\epsilon > 0$ não existe uma $(1 - \epsilon)\ln|\mathcal{R}|$-aproximação polinomial para o* PSGrp *a não ser que $\mathcal{NP} \subseteq \mathrm{DTIME}(n^{O(\log\log n)})$.* □

A transformação apresentada acima é devida a Ihler [15]. Neste mesmo artigo, o autor considera vários casos particulares do PSGrp e prova que mesmo quando impomos severas restrições à coleção de grupos, aos custos ou ao grafo, o problema continua $\mathcal{NP}$-difícil. Neste sentido o PSGrp contrasta com o PSG, que pode ser resolvido em tempo polinomial em vários casos particulares (veja [13] para maiores detalhes).

3 Algoritmos de aproximação

3.1 Uma visão geral

Vários algoritmos de aproximação foram propostos para o PSGrp. Em [18], Reich e Widmayer propõem duas heurísticas para o PSGrp mas não fornecem razões de aproximação. Em [14], Ihler considera modificações simples de uma das heurísticas propostas por Reich e Widmayer e prova que as heurísticas modificadas têm razão de aproximação $|\mathcal{R}| - 1$, onde $\mathcal{R}$ é a coleção de grupos. Mais tarde, Helvig, Robins e Zelikovsky [12] forneceram uma $O(|\mathcal{R}|^\epsilon)$-aproximação polinomial para o problema para qualquer $\epsilon > 0$ fixo.

Em sua tese de doutorado, Slavík [20] considera casos particulares do PSGrp e fornece algoritmos de aproximação para estes casos. Ele considera o PSGrp restrito a árvores enraizadas (neste caso, exigimos adicionalmente que a raiz ocorra na solução ótima) e fornece uma $B \cdot H_N$-aproximação polinomial, onde B é o número máximo de vértices de um mesmo grupo numa subárvore da raiz, N é a cardinalidade máxima de um grupo e H_N denota o N-ésimo número harmônico. Para grafos gerais, Slavík fornece uma $2N$-aproximação polinomial.

O algoritmo de aproximação de melhor razão até o momento é devido a Garg, Konjevod e Ravi [10]. Eles consideram o problema restrito a árvores enraizadas e fornecem uma $O((\log n)^2)$-aproximação probabilística polinomial. Usando-se uma aproximação probabilística do grafo de distâncias de um grafo qualquer por árvores devida a Fakcharoenphol *et al.* [5], pode-se então utilizar o algoritmo dos autores também para grafos gerais. Neste caso, a razão de aproximação do algoritmo sobe para $O((\log n)^3)$.

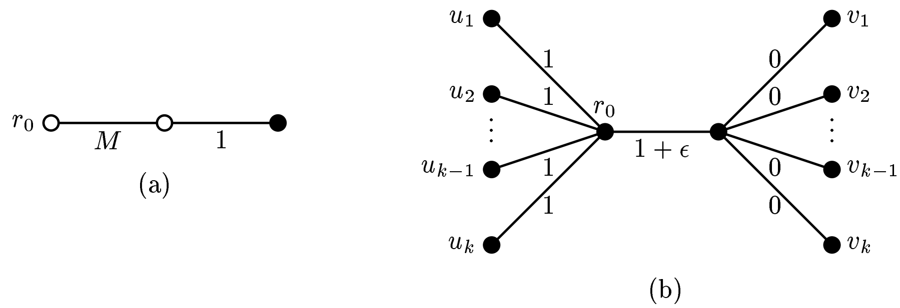


Figura 3: (a) Um exemplo ruim para o algoritmo de Reich e Widmayer. Neste exemplo, os vértices de mesma cor pertencem a um mesmo grupo. Os custos são dados nas arestas. Uma possível solução usa apenas a aresta de custo 1. Outra possível solução usa ambas as arestas. Como podemos escolher M de forma arbitrária, se escolhemos o vértice r_0 no passo (i) do algoritmo, podemos obter uma solução tão distante da ótima quanto desejado. (b) Um exemplo justo para o algoritmo de Reich e Widmayer com a modificação de Ihler. Neste exemplo, temos $k+1$ grupos, dados por $R_0 := \{r_0\}$ e $R_i := \{u_i, v_i\}$, $i = 1, \ldots, k$. Os custos são dados nas arestas, sendo ϵ um número convenientemente pequeno. Se fixamos o grupo R_0 no algoritmo, vamos obter uma solução de custo k, enquanto a solução ótima tem custo $1+\epsilon$.

Vamos apresentar a seguir dois dos algoritmos acima: a adaptação de Ihler da heurística de Reich e Widmayer e a $O((\log n)^2)$-aproximação probabilística de Garg, Konjevod e Ravi para o PSGrp restrito a árvores enraizadas.

3.2 Heurística dos caminhos mínimos

Seja G um grafo e $\mathcal{R}$ uma coleção de grupos de vértices de G. Dado $V' \subseteq V(G)$, dizemos que V' *cobre* um grupo $R \in \mathcal{R}$ se V' contém pelo menos um vértice de R. Dizemos que V' *cobre* $\mathcal{R}$ se V' cobre todos os grupos em $\mathcal{R}$. Se H é um subgrafo de G, dizemos que H *cobre* um grupo R se $V(H)$ cobre R. Analogamente, dizemos que H *cobre* $\mathcal{R}$ se $V(H)$ cobre $\mathcal{R}$.

O algoritmo a seguir, proposto por Reich e Widmayer [18], recebe uma instância $(G, \mathcal{R}, c)$ do PSGrp e devolve uma solução viável para o PSGrp $(G, \mathcal{R}, c)$:

(i) fixe um vértice de grupo r_0 de forma arbitrária;
(ii) seja H o grafo composto apenas por r_0;
(iii) se H cobre $\mathcal{R}$, devolva H;
(iv) se H não cobre $\mathcal{R}$, escolha dentre os vértices dos grupos não cobertos por $V(H)$ aquele que está mais próximo de algum vértice em $V(H)$. Seja r o vértice escolhido e u o vértice em $V(H)$ mais próximo dele. Conecte r a H através de um (u,r)-caminho de custo mínimo em G e vá para o passo (iii).

Está claro que o algoritmo acima devolve uma solução viável para o PSGrp $(G, \mathcal{R}, c)$, já que G é conexo.

Se todos os grupos são unitários, ou seja, se estamos tratando de uma instância do problema de Steiner, o algoritmo acima coincide com o algoritmo proposto por Takahashi e Matsuyama [21] para o PSG. Takahashi e Matsuyama provaram que, neste caso, a razão de aproximação do algoritmo é 2. Para o PSGrp, entretanto, a heurística dos caminhos mínimos pode devolver soluções tão ruins quanto desejado, como mostra o exemplo da Figura 3a.

Uma modificação simples no algoritmo, entretanto, pode transformá-lo numa $|\mathcal{R}|-1$ aproximação para o problema. Basta fixar um grupo $R_0 \in \mathcal{R}$ e repetir o algoritmo para cada vértice $r_0 \in R_0$, fixando r_0 como o vértice inicial no passo (i), e devolver a melhor solução encontrada. Esta alteração foi proposta por Ihler [14].

Antes de provarmos a razão de aproximação vamos primeiro verificar que o algoritmo de Reich e Widmayer com a modificação de Ihler é de fato polinomial. Os passos (i) e (ii) consomem tempo constante. O passo (iii) também consome tempo constante: basta mantermos um contador do número de grupos que já foram cobertos. O passo (iv) pode ser executado em tempo $O(m \log n)$ usando-se o algoritmo de Dijkstra implementado com um heap binário (veja [3] para maiores detalhes), onde $n := |V(G)|$ e $m := |E(G)|$. Como o passo (iv) é executado no máximo $|\mathcal{R}| - 1$ vezes, uma execução do algoritmo de Reich e Widmayer consome ao todo tempo $O(|\mathcal{R}| m \log n)$. Como o algoritmo é executado $|R_0|$ vezes quando fixamos um grupo $R_0 \in \mathcal{R}$, o algoritmo de Reich e Widmayer com a modificação proposta por Ihler consome tempo $O(|\mathcal{R}| mn \log n)$ e é portanto polinomial.

Teorema 4: *O algoritmo de Reich e Widmayer, com a alteração proposta por Ihler, é uma* $(|\mathcal{R}| - 1)$*-aproximação polinomial para o* PSGRP.

Prova: Já vimos que o algoritmo é polinomial, resta então provar a razão de aproximação. Seja H^* uma solução ótima para o PSGRP $(G, \mathcal{R}, c)$. Como H^* cobre $\mathcal{R}$, algum vértice $r_0 \in R_0$ deve ocorrer em H^*.

Vejamos o que ocorre quando o vértice r_0 é fixado como sendo o inicial no passo (i) do algoritmo de Reich e Widmayer. Sejam $r_1, \ldots, r_k$ os vértices de grupos não cobertos escolhidos no passo (iv) do algoritmo e $u_1, \ldots, u_k$ os vértices do grafo H sendo construído aos quais eles são ligados, sendo que u_i é ligado a r_i, $i = 1, \ldots, k$. Como H^* cobre $\mathcal{R}$, para cada $i = 1, \ldots, k$, H^* possui um vértice r'_i do mesmo grupo de r_i. Mas então, pela escolha feita no passo (iv) do algoritmo quando r_i foi escolhido temos que $d_G(u_i, r_i) \le d_G(r_0, r'_i) \le d_{H^*}(r_0, r'_i) \le \mathrm{opt}\,(G, \mathcal{R}, c)$.

Observe que o passo (iv) é repetido no máximo $|\mathcal{R}| - 1$ vezes. Mas então, se $\hat{H}$ é a solução devolvida pelo algoritmo temos que

$$\begin{aligned} c(\hat{H}) &\le \sum_{i=1}^{k} d_G(u_i, r_i) \\ &\le \sum_{i=1}^{k} \mathrm{opt}\,(G, \mathcal{R}, c) \\ &\le (|\mathcal{R}| - 1)\mathrm{opt}\,(G, \mathcal{R}, c), \end{aligned}$$

como queríamos. □

A razão de aproximação obtida também é justa, como mostra o exemplo da Figura 3b.

3.3 Algoritmo de Garg, Konjevod e Ravi

Nesta seção, apresentaremos o algoritmo de Garg, Konjevod e Ravi [10] para o PSGRP restrito a árvores. Vamos também mostrar como usar o algoritmo com grafos quaisquer.

O algoritmo para árvores enraizadas

O problema de Steiner com grupos para árvores enraizadas é o seguinte:

> **Problema** PSGRAIZ $(T, r, \mathcal{R}, c)$: Dada uma árvore $T = (V, E)$ com raiz r, uma coleção $\mathcal{R}$ de subconjuntos de $V(T)$ e um custo $c(e) \in \mathbb{Q}_+$ para cada $e \in E(T)$, encontrar uma subárvore de T que contenha r, cubra $\mathcal{R}$ e tenha custo mínimo.

Dada uma árvore T de raiz r e uma aresta $e \in E(T)$, denotaremos por e^+ sua ponta mais profunda (em relação a r) e por e^- sua outra ponta. Se $v \in V(T)$, $v \neq r$, denotaremos por $\pi(v)$ o *pai* de v, i.e., o vizinho de v no (r, v)-caminho em T. Denotaremos por $\delta^\uparrow(v)$ a aresta que liga v a $\pi(v)$. Dada uma aresta $e \in E(T)$ que não incide na raiz, denotaremos por $\pi(e)$ a aresta $\delta^\uparrow(e^-)$.

Qualquer α-aproximação polinomial para o PSGRAIZ pode ser transformada numa α-aproximação polinomial para o PSGRP restrito a árvores. De fato, suponha que temos uma instância $(T, \mathcal{R}, c)$ do PSGRP onde T é uma árvore. Basta fixarmos um grupo $R_0 \in \mathcal{R}$ e executar nossa α-aproximação para as instâncias $(T, r, \mathcal{R} \setminus \{R_0\}, c)$, $r \in R_0$, do PSGRAIZ, e devolver a melhor solução encontrada.

Considere então uma instância $(T, r, \mathcal{R}, c)$ do PSGRAIZ. Suponha, sem perda de generalidade, que todos os vértices de grupo são folhas de T.[1] Dado um conjunto $S \subseteq V(T)$, dizemos que $\delta(S)$ é um $(r, \mathcal{R})$-*corte* se $r \in S$ e $R \cap S = \emptyset$ para algum $R \in \mathcal{R}$. Com isso, o seguinte problema de programação linear é uma relaxação do PSGRAIZ:

$$
(1) \qquad \begin{array}{rrcll}
\text{minimizar} & cx & & & \\
\text{sujeito a} & x(\delta(S)) & \geq & 1, & \text{para todo } (r, \mathcal{R})\text{-corte } \delta(S), \\
& x & \geq & 0. &
\end{array}
$$

O problema da separação das inequações correspondentes a $(r, \mathcal{R})$-cortes pode ser resolvido em tempo polinomial. Ele pode ser reduzido ao problema de se encontrar um corte mínimo num grafo derivado de T (veja [10]). Portanto, o problema da separação para (1) pode ser resolvido em tempo polinomial e segue da equivalência entre separação e otimização [11] que (1) pode ser resolvido em tempo polinomial.

Dizemos que um vetor $x \in \mathbb{R}^E$ é r-*decrescente* se $0 \leq x \leq \mathbf{1}$ e $x(e) \leq x(\pi(e))$ para toda aresta $e \in E(T)$ não incidente em r. Obviamente, toda solução básica $\hat{x}$ de (1) é tal que $\hat{x} \leq \mathbf{1}$. Mais do que isso, temos o seguinte resultado:

Proposição 5: *Toda solução básica de* (1) *é r-decrescente.*

Prova: Seja $\hat{x}$ uma solução básica de (1) e suponha que a aresta $e \in E(T)$ seja tal que $\hat{x}(e) > \hat{x}(\pi(e))$. Sejam $u, v, w \in V(T)$ tais que $\pi(e) = uv$ e $e = vw$. Considere o vetor $x' \in \mathbb{R}^E$ tal que $x'(f) := \hat{x}(f)$ para toda aresta $f \in E(T)$, $f \neq e$, e $x'(e) := \hat{x}(\pi(e))$. Afirmamos que x' é uma solução viável de (1).

De fato, seja $S \subseteq V(T)$ tal que $\delta(S)$ é um $(r, \mathcal{R})$-corte. Se $e \notin \delta(S)$, $x'(\delta(S)) = \hat{x}(\delta(S)) \geq 1$ e nada temos a fazer. Suponha então que $e \in \delta(S)$. Temos dois casos a considerar. Se $v \in S$, denote por X o conjunto dos vértices da subárvore de T enraizada em v e ponha $S' := S \setminus X$. Note que $\delta(S')$ é um $(r, \mathcal{R})$-corte e que $\delta(S') \subseteq (\delta(S) \setminus \{e\}) \cup \{\pi(e)\}$. Mas então $x'(\delta(S)) \geq \hat{x}(\delta(S')) \geq 1$.

Suponha agora que $w \in S$. Denote por X o conjunto dos vértices da subárvore de T enraizada em w e ponha $S' := S \setminus X$. Note que $\delta(S')$ é um $(r, \mathcal{R})$-corte e que $\delta(S') \subseteq \delta(S) \setminus \{e\}$. Mas então $x'(\delta(S)) \geq \hat{x}(\delta(S')) \geq 1$, provando a afirmação.

Considere então o vetor $x'' \in \mathbb{R}^E$ tal que $x''(f) := \hat{x}(f)$ para toda aresta $f \in E(T)$, $f \neq e$, e $x''(e) := 2\hat{x}(e)$. Como $x'(e) < \hat{x}(e) < x''(e)$, segue que $\hat{x}$ é combinação convexa não-trivial de x' e x'', uma contradição. $\square$

Seja $x \in \mathbb{R}^E$ um vetor r-decrescente. A seguir, descrevemos um experimento aleatório em função de T e x, mas ele pode ser executado sobre qualquer árvore $T' = (V', E')$ de raiz r' e qualquer vetor r'-decrescente $x' \in \mathbb{R}^{E'}$. O experimento é o seguinte:

(i) para cada aresta $e \in E(T)$, se e é incidente a r, escolha e com probabilidade $x(e)$. Caso contrário, escolha e com probabilidade $x(e)/x(\pi(e))$. Denote por F a floresta resultante;

(ii) remova todos os componentes de F que não contêm r. Denote por T_x a árvore resultante.

[1] De fato, não perdemos generalidade ao fazê-lo, já que se $u \in R$, $R \in \mathcal{R}$, não é folha de T, podemos inserir um novo vértice v em T, ligá-lo a u por uma aresta de custo 0, remover u de R e inserir v em R.

Note que, devido à Proposição 5, podemos executar o experimento sobre toda solução básica de (1). Uma propriedade interessante do experimento é então a seguinte:

Proposição 6: *Seja $\hat{x}$ uma solução viável r-decrescente de* (1). *O custo esperado de $T_{\hat{x}}$ é $c\hat{x}$.*

Prova: Seja e uma aresta de T. O experimento inclui e em $T_{\hat{x}}$ se e somente se todas as arestas num (r, e^+)-caminho forem incluidas em $T_{\hat{x}}$.

Sejam $e_0, \ldots, e_k = e$ as arestas num (r, e^+)-caminho em T, na ordem em que ocorrem quando o caminho é percorrido de r a e^+. Como incluimos as arestas de forma independente, a probabilidade de incluirmos e em $T_{\hat{x}}$ é então $\hat{x}(e_0) \cdot \prod_{i=1}^{k} (\hat{x}(e_i)/\hat{x}(e_{i-1})) = \hat{x}(e)$. Segue então da linearidade da esperança que o custo esperado de $T_{\hat{x}}$ é $c\hat{x}$. □

O seguinte resultado, juntamente com a Proposição 6, vai nos permitir criar um algoritmo de aproximação para o PSGraiz a partir do experimento aleatório descrito.

Teorema 7: *Seja $\hat{x}$ uma solução viável r-decrescente de* (1) *e fixe $R_0 \in \mathcal{R}$. A probabilidade de $T_{\hat{x}}$ conter um vértice de R_0 é pelo menos $1/128 \lg N$, onde N é o tamanho do maior grupo.*

A demostração do Teorema 7 será feita mais adiante. Ponha $t := |\mathcal{R}|$. No que segue, ignoramos chãos e tetos por uma questão de estética, mas todos os cálculos podem ser feitos com maior precisão levando-os em conta. Um importante corolário é o seguinte:

Corolário 8: *Seja $\hat{x}$ uma solução viável r-decrescente de* (1). *Suponha que repetimos o experimento aleatório descrito $128 \lg N \ln 4t$ vezes. A união das árvores resultantes de cada execução do experimento cobre $\mathcal{R}$ com probabilidade pelo menos* $3/4$.

Prova: Fixe um grupo $R_0 \in \mathcal{R}$. A probabilidade de nenhuma árvore gerada conter um vértice de R_0 é no máximo

$$\begin{aligned} \left(1 - \frac{1}{128 \lg N}\right)^{128 \lg N \ln 4t} &\leq (e^{-1/128 \lg N})^{128 \lg N \ln 4t} \\ &= e^{-\ln 4t} \\ &= 1/4t, \end{aligned}$$

onde usamos que $1 - \lambda \leq e^{-\lambda}$ para todo λ real. Mas então, a probabilidade da união de todas as árvores geradas não conter um vértice de algum grupo é no máximo $1/4$ e portanto a probabilidade da união das árvores cobrir $\mathcal{R}$ é pelo menos $3/4$, com queríamos. □

Com isso podemos desenvolver um algoritmo de aproximação para o PSGraiz. O algoritmo recebe uma instância $(T, r, \mathcal{R}, c)$ do PSGraiz e um número ϵ, $0 < \epsilon < 1$, e devolve uma solução viável $\hat{T}$ do PSGraiz $(T, r, \mathcal{R}, c)$ que, com probabilidade pelo menos $1 - \epsilon$, é tal que $c(\hat{T}) \leq (512 \lg N \ln 4t)\mathrm{opt}\,(T, r, \mathcal{R}, c)$, onde N é o tamanho do maior grupo:

(i) seja x^* uma solução ótima básica de (1);
(ii) para $i = 1, \ldots, 4 \ln 1/\epsilon$, repita o experimento aleatório $128 \lg N \ln 4t$ vezes e denote por T_i a união das árvores geradas;
(iii) dentre as árvores T_i que cobrem $\mathcal{R}$, devolva a de menor custo. Se nenhuma tal árvore cobre $\mathcal{R}$, devolva T.

Obviamente, o algoritmo devolve uma solução viável para o problema e é polinomial. Resta provar sua razão de aproximação.

Teorema 9: *Se $\hat{T}$ é a árvore devolvida pelo algoritmo, $c(\hat{T}) \leq (512 \lg N \ln 4t)\mathrm{opt}\,(T, r, \mathcal{R}, c)$ com probabilidade pelo menos $1 - \epsilon$.*

Prova: Fixe $i = 1, \ldots, 4 \ln 1/\epsilon$. O custo esperado de T_i é no máximo a soma dos custos esperados das $128 \lg N \ln 4t$ árvores cuja união a compõe. Pela Proposição 6, o custo esperado é então no máximo $(128 \lg N \ln 4t) cx^*$. Segue então da desigualdade de Markov que $\Pr[c(T_i) \geq 4 \cdot (128 \lg N \ln 4t) cx^*] \leq 1/4$.

Mas então a árvore T_i tem custo baixo com probabilidade pelo menos $3/4$. Pelo Corolário 8, ela cobre $\mathcal{R}$ com probabilidade pelo menos $3/4$. Mas então ela tem custo baixo e cobre $\mathcal{R}$ com probabilidade pelo menos $1/4$.

Com isso, a probabilidade de nenhuma das árvores T_i ter custo baixo e cobrir $\mathcal{R}$ é no máximo $(1 - 1/4)^{4 \ln 1/\epsilon} \leq \epsilon$. Logo, alguma das árvores T_i cobre $\mathcal{R}$ e tem custo baixo com probabilidade pelo menos $1 - \epsilon$. Como $cx^* \leq \text{opt}\,(T, r, \mathcal{R}, c)$, segue o resultado. $\square$

Análise do experimento

Seja $T' = (V', E')$ uma árvore de raiz r'. Denotaremos por $L(T')$ o conjunto das folhas de T'. Seja $X \subseteq L(T')$ e $x' \in \mathbb{R}^{E'}$ um vetor r'-decrescente. Nosso objetivo é limitar inferiormente a probabilidade de $T'_{x'}$ conter um vértice de X (no caso do Teorema 7, temos que $T' := T$, $X := R_0$ e $x' := \hat{x}$).

Para cada $e \in \delta(X)$, seja A_e o conjunto das arestas pertencentes ao (r, e^+)-caminho em T'. Note que incluímos e em $T'_{x'}$ se e somente se $A_e \subseteq E(T'_{x'})$. Seja B_e o evento $A_e \subseteq E(T'_{x'})$. Queremos um limitante inferior para

$$(2) \quad \Pr\left[\bigvee_{e \in \delta(X)} B_e\right] = 1 - \Pr\left[\bigwedge_{e \in \delta(X)} \overline{B_e}\right].$$

Se os eventos B_e, $e \in \delta(X)$, fossem independentes, teríamos que

$$\Pr\left[\bigwedge_{e \in \delta(X)} \overline{B_e}\right] = \prod_{e \in \delta(X)} \Pr[\overline{B_e}].$$

Entretanto, este não precisa ser o caso. De fato, se $e, f \in \delta(X)$, B_e, B_f são independentes se e somente se $A_e \cap A_f = \emptyset$. Para encontrarmos um limitante para (2), vamos fazer uso da Desigualdade de Janson [16] (veja também o capítulo 8 de [1]).

Dados $e, f \in \delta(X)$, escreveremos $e \sim f$ se $A_e \cap A_f \neq \emptyset$, i.e., se os eventos B_e, B_f forem dependentes. Ponha $\Delta := \sum_{e \sim f} \Pr[B_e \wedge B_f]$ e $\mu := \sum_{e \in \delta(X)} \Pr[B_e]$. A Desigualdade de Janson, já adaptada para o nosso caso, é a seguinte:

Teorema 10: *Sejam B_e, $e \in \delta(X)$, Δ e μ como acima e suponha que $\Pr[B_e] \leq \epsilon$ para todo $e \in \delta(X)$. Então*

$$(3) \quad \Pr\left[\bigwedge_{e \in \delta(X)} \overline{B_e}\right] \leq \exp\left(-\mu + \frac{1}{1 - \epsilon} \frac{\Delta}{2}\right).$$

Se, adicionalmente, $\Delta \geq \mu(1 - \epsilon)$ então

$$(4) \quad \Pr\left[\bigwedge_{e \in \delta(X)} \overline{B_e}\right] \leq \exp\left(-\frac{\mu^2 (1 - \epsilon)}{2\Delta}\right).$$

Antes de provarmos o Teorema 7, precisamos de dois resultados simples. Dizemos que $\delta(S)$, $S \subseteq V(T')$, é um (r', X)-*corte* se $r' \in S$ e $X \cap S = \emptyset$. Dado um vértice $v \in V(T')$ que não é folha de T', denotamos por $\delta^{\downarrow}(v)$ o conjunto das arestas que ligam v a seus filhos em T'.

Lema 11: *Seja $T' = (V', E')$ uma árvore de raiz r', X um subconjunto das folhas de T' e $x_0 \in \mathbb{R}^{E'}$ um vetor r'-decrescente. Suponha que para algum $\beta \geq 0$, $x_0(\delta(S)) \geq \beta$ para todo (r', X)-corte $\delta(S)$. Existe um vetor $\tilde{x}_0$, $0 \leq \tilde{x}_0 \leq x_0$, tal que*

(i) $\tilde{x}_0(\delta^{\downarrow}(r')) = \beta$;

(ii) *para todo $v \in V(T')$ que não seja nem uma folha de T' nem a raiz,* $\tilde{x}_0(\delta^{\uparrow}(v)) = \tilde{x}_0(\delta^{\downarrow}(v))$;

(iii) *para toda subárvore T'' de T' com raiz v tal que $L(T'') \subseteq L(T')$,* $\tilde{x}_0(\delta^{\downarrow}_{T''}(v)) = \tilde{x}_0(\delta_{T''}(L(T'')))$.

Prova: Construa um grafo dirigido D a partir de T' da seguinte forma:

(i) $V(D) := V(T') \cup \{s\}$;

(ii) para cada aresta e de T', D possui um arco e^-e^+;

(iii) D possui um arco us para cada $u \in X$.

A cada arco uv de D que corresponde a uma aresta $e = uv$ de T', associe capacidade $x_0(e)$. A cada arco us de D, associe capacidade infinita. Denote por w o vetor de capacidades resultante. Associe demanda $-\beta$ ao vértice r' e β ao vértice s, aos demais vértices associe demanda 0. Denote por b o vetor de demandas resultante.

Considere um conjunto $S \subseteq V(D)$. Se $b(S) \geq 0$ então trivialmente $b(S) + w(\delta^+_D(S)) \geq 0$. Senão, note que devemos ter $r' \in S$ e $s \notin S$. Se $X \cap S \neq \emptyset$, $w(\delta^+_D(S)) = \infty$ e, trivialmente, $b(S) + w(\delta^+_D(S)) \geq 0$. Senão, $\delta_{T'}(S)$ é um (r', X)-corte e segue que $b(S) + w(\delta^+_D(S)) = -\beta + x_0(\delta_{T'}(S)) \geq 0$. Segue então da caracterização de Gale [8] de fluxos viáveis em redes que existe um fluxo viável em D com capacidades dadas por w e demandas dadas por b.

Seja então $\bar{x}$ um tal fluxo viável e $\tilde{x}_0 \in \mathbb{R}^{E'}$ tal que $\tilde{x}_0(e) := \bar{x}(e^-e^+)$ para todo $e \in E(T')$. Como $0 \leq \bar{x} \leq w$ segue que $0 \leq \tilde{x}_0 \leq x_0$. Note também que, como $\delta^-_D(r') = \emptyset$, $\tilde{x}_0(\delta^{\downarrow}(r')) = \bar{x}(\delta^+_D(r')) = \beta$, provando (i).

O item (ii) segue trivialmente do fato de que $\bar{x}$ é um fluxo viável. Para ver o item (iii), considere uma subárvore T'' de T' como em (iii) e ponha $S := V(T'')$. Todo arco $a \in \delta^+_D(S)$ é incidente a uma folha u de T''. Logo, como $\bar{x}$ é um fluxo viável, $\tilde{x}_0(L(T'')) = \bar{x}(\delta^+_D(S))$. Mas então $\tilde{x}_0(\delta^{\downarrow}_{T''}(v)) = \bar{x}(\delta^-_D(S)) - b(S) = \bar{x}(\delta^+_D(S)) = \tilde{x}_0(L(T''))$, provando (iii). □

Note que, no lema acima, o vetor $\tilde{x}_0$ também é r'-decrescente. O segundo resultado de que precisamos é o seguinte:

Lema 12: *Seja $T' = (V', E')$ uma árvore de raiz r' e X um subconjunto das folhas de T'. Sejam $x', x'' \in \mathbb{R}^{E'}$ vetores r'-decrescentes e suponha que x', x'' diferem apenas na componente e de modo que $x'(e) \geq x''(e)$. A probabilidade de $T'_{x'}$ conter um vértice de X é maior ou igual à probabilidade de $T'_{x''}$ conter um vértice de X.*

Prova: Seja A o evento em que $T'_{x'}$ possui um vértice de X na subárvore abaixo da aresta e. Então $\Pr[\overline{A}] = (1 - x'(e)) + x'(e) \Pr[\overline{A} \mid e \text{ é escolhida}]$. Note que o coeficiente de $x'(e)$ nesta expressão é sempre negativo. Logo, se $x'(e)$ é decrescido, $\Pr[\overline{A}]$ aumenta e $\Pr[A]$ decresce. □

Com isso estamos prontos para provar o Teorema 7.

Prova do Teorema 7: Se $|R_0| = 1$, a probabilidade de incluirmos algum vértice de R_0 em $T_{\hat{x}}$ é 1. Suponha então que $|R_0| \geq 2$.

Atribua 0 a cada componente de $\hat{x}$ que corresponde a uma aresta de T que não ocorre num caminho entre r e algum vértice de R_0 e arredonde os valores resultantes para baixo para a potência de 2 mais próxima. Segundo o Lema 12, a probabilidade de $T_{\hat{x}}$ conter um vértice de R_0 só decresceu. Note também que agora temos $\hat{x}(\delta(S)) \geq 1/2$ para cada (r, R_0)-corte $\delta(S)$ de T.

Ponha $d := \lceil \lg |R_0| \rceil$ e atribua 0 a cada componente de $\hat{x}$ cujo valor é menor ou igual a $1/2^{d+2}$. Novamente, a probabilidade de sucesso só diminui. Além disso, o valor de cada (r, R_0)-corte diminuiu no máximo $|R_0|/2^{d+2} \leq 1/4$, logo $\hat{x}(\delta(S)) \geq 1/4$ para cada (r, R_0)-corte $\delta(S)$.

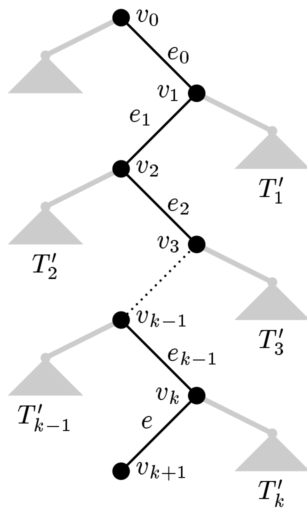


Figura 4: A figura ilustra as definições dadas na demonstração do Teorema 7. Note que o vértice v_i também faz parte da árvore T'_i, marcada em cinza na figura, e que mais de uma aresta pode ligar v_i a vértices de T'_i.

Finalmente, contraia cada aresta e de T tal que $e^- \neq r$ e $\hat{x}(e) = \hat{x}(\pi(e))$, a não ser que e seja incidente a uma folha. Denote por T' a árvore resultante e por x' a restrição de $\hat{x}$ às arestas de T'. Note que x' é r-decrescente para T' e que a altura de T' é no máximo $d+3$. Além disso, a probabilidade de $T'_{x'}$ conter um vértice de R_0 é igual a probabilidade de $T_{\hat{x}}$ conter um vértice de R_0. Aplicando o Lema 11 a T' e x' com $\beta := 1/4$ e $X := R_0$, obtemos um vetor r-decrescente x'', $0 \leq x'' \leq x'$, satisfazendo (i)–(iii) do Lema 11.

Vamos calcular Δ para T' e x''. Dadas arestas $e, f \in \delta_{T'}(R_0)$ tais que $A_e \cap A_f \neq \emptyset$, denote por $\pi(e,f)$ o ancestral comum mais próximo de e^+ e f^+. Dada uma aresta $e \in \delta_{T'}(R_0)$, ponha

$$\Delta_e := \sum_{f \sim e} \Pr[B_e \wedge B_f] = \sum_{f \sim e} \frac{x''(e)x''(f)}{x''(\delta^{\uparrow}_{T'}(\pi(e,f)))}.$$

É fácil ver que Δ_e é a contribuição de e a Δ, de modo que $\Delta = \sum_{e \in \delta(R_0)} \Delta_e$. Vamos estimar Δ_e. Seja $r = v_0, e_0, v_1, e_1, \ldots, v_k, e_k = e, v_{k+1}$ o (r, e^+)-caminho em T'. Para cada $i = 1, \ldots, k$, denote por H_i a subárvore de T' enraizada em v_i e denote por T'_i o componente de $H_i - e_i$ que contém v_i. Ponha $f_i := x''(\delta^{\downarrow}_{T'_i}(v_i))$. A Figura 4 ilustra estas definições.

Como x'' satisfaz a propriedade (ii) do Lema 11, $f_i \leq x''(e_{i-1})$ para $i = 1, \ldots, k$. Além disso, como x'' satisfaz (iii), temos que

$$\begin{aligned}
\Delta_e &= \sum_{i=1}^{k} \sum_{f \in L(T'_i)} \frac{x''(e)x''(f)}{x''(e_{i-1})} \\
&= \sum_{i=1}^{k} \frac{x''(e)}{x''(e_{i-1})} \sum_{f \in L(T'_i)} x''(f) \\
&= \sum_{i=1}^{k} \frac{x''(e) f_i}{x''(e_{i-1})} \\
&\leq x''(e)(d+2).
\end{aligned}$$

Mas então

$$\begin{aligned}
\Delta &= \sum_{e \in \delta_{T'}(R_0)} \Delta_e \\
&\le x''(\delta_{T'}(R_0))(d+2) \\
&= (d+2)/4 \\
&\le \lg |R_0|,
\end{aligned}$$

pois $|R_0| \ge 2$. Note que temos $\mu = 1/4$ e podemos fazer $\epsilon := 1/4$. Se $\Delta \le \mu(1-\epsilon) = 3/16$, vem de (3) que

$$\Pr\left[\bigwedge_{e \in \delta(X)} \overline{B_e}\right] \le e^{-1/8} \le .89.$$

Por outro lado, se $\Delta \ge \mu(1-\epsilon)$, vem de (4) que

$$\begin{aligned}
\Pr\left[\bigwedge_{e \in \delta(X)} \overline{B_e}\right] &\le e^{-3/2 \cdot 64 \lg |R_0|} \\
&\le e^{-1/64 \lg |R_0|} \\
&\le 1 - 1/128 \lg |R_0|,
\end{aligned}$$

onde usamos que $e^{-\lambda} \le 1-\lambda/2$ para todo real λ no intervalo $(0,1/2)$. Mas então, como $1-1/128 \lg |R_0| \ge e^{-1/8}$, a probabilidade de $T_{\hat{x}}$ incluir um vértice de R_0 é pelo menos $1/128 \lg |R_0| \le 1/128 \lg N$, como queríamos. □

Grafos gerais

O algoritmo de Garg, Konjevod e Ravi pode ser adaptado para o PSGRP em grafos gerais. O processo consiste em aproximar as distâncias do grafo original probabilisticamente por uma árvore e resolver o problema nesta árvore.

Considere uma instância $(G, \mathcal{R}, c)$ do PSGRP. Nesta seção, suporemos que existe um grupo unitário e denotaremos por r o vértice deste grupo. Não perdemos generalidade ao fazê-lo pelo mesmo motivo que não perdemos generalidade ao considerar o PSGRP restrito a árvores enraizadas.

Denote por K_G o grafo completo com o mesmo conjunto de vértices de G e considere o vetor $c_K \in \mathbb{R}^{E(K_G)}$ dado por $c_K(uv) := d_G(u,v)$ para todo par de vértices $u, v \in V(G)$.

Proposição 13: *Temos que* $\operatorname{opt}(G, \mathcal{R}, c) = \operatorname{opt}(K_G, \mathcal{R}, c_K)$. *Toda solução ótima do* PSGRP $(G, \mathcal{R}, c)$ *pode ser transformada numa solução ótima do* PSGRP $(K_G, \mathcal{R}, c_K)$ *em tempo polinomial e vice-versa.*

Prova: Seja T^* uma solução ótima do PSGRP $(G, \mathcal{R}, c)$. Note que T^* também é uma solução do PSGRP $(K_G, \mathcal{R}, c_K)$ e que, como T^* é ótima, $c(uv) = d_G(u,v)$ para toda aresta $uv \in E(T^*)$. Logo $\operatorname{opt}(K_G, \mathcal{R}, c_K) \le \operatorname{opt}(G, \mathcal{R}, c)$.

Tome agora uma solução ótima T^* do PSGRP $(K_G, \mathcal{R}, c_K)$. Seja H o grafo obtido de T^* substituindo-se cada aresta uv de T^* por um (u,v)-caminho mínimo em G. Note que H é uma solução viável para o PSGRP $(G, \mathcal{R}, c)$ e que $c(H) \le c_K(T^*)$. Logo $\operatorname{opt}(G, \mathcal{R}, c) \le \operatorname{opt}(K_G, \mathcal{R}, c_K)$. Ademais, é fácil ver através da demonstração como transformar uma solução ótima de um problema numa solução ótima do outro problema em tempo polinomial. □

Em outras palavras, é equivalente resolver o problema em G ou no grafo de distâncias de G. Vamos agora ver como aproximar as distâncias em G através de uma árvore. Dizemos que uma árvore T com pesos nas arestas *domina* as distâncias em G se $V(G) \subseteq V(T)$ e $d_T(u,v) \geq d_G(u,v)$ para todo par de vértices $u, v \in V(G)$.

Considere agora uma coleção $\mathcal{S}$ de árvores que dominam as distâncias em G e uma distribuição $\mathcal{D}$ de probabilidades sobre as árvores em $\mathcal{S}$. Suponha que façamos o seguinte experimento: sorteamos ao acaso uma árvore de $\mathcal{S}$ segundo a distribuição $\mathcal{D}$ e denotamos por $X(u,v)$ a variável aleatória que representa a distância entre $u, v \in V(G)$ na árvore sorteada. Dizemos que $(\mathcal{S}, \mathcal{D})$ é uma α-aproximação probabilística para as distâncias em G se $\mathbb{E}[X(u,v)] \leq \alpha d_G(u,v)$ para todo par de vértices $u, v \in V(G)$.

O problema de se encontrar α-aproximações probabilísticas para as distâncias em um grafo foi estudado por muitos autores (veja, por exemplo, [2] para um apanhado geral sobre o problema). Em [5], Fakcharoenphol *et al.* mostram que para todo grafo G existe uma família $\mathcal{S}$ de árvores que dominam as distâncias em G e uma distribuição $\mathcal{D}$ de probabilidades sobre as árvores em $\mathcal{S}$ tais que $(\mathcal{S}, \mathcal{D})$ é uma $(16 \lg n)$-aproximação probabilística para as distâncias em G, onde $n := |V(G)|$. Além disso, os autores fornecem um algoritmo polinomial capaz de sortear uma árvore desta distribuição. As árvores em $\mathcal{S}$ são tais que os vértices em $V(G)$ são folhas.

Com isso, podemos estender o algoritmo de Garg, Konjevod e Ravi para o PSGrp em grafos gerais. O algoritmo recebe uma instância $(G, \mathcal{R}, c)$ do PSGrp e um número ϵ, $0 < \epsilon < 1$, e devolve uma solução viável $\hat{T}$ do problema que, com probabilidade pelo menos $1 - \epsilon$, é tal que $c(\hat{T}) \leq (2^{17} \lg n \lg N \ln 4t) \mathrm{opt}\,(G, \mathcal{R}, c)$:

(i) para $i = 1, \ldots, 8 \ln 1/\epsilon$, sorteie uma árvore T_i de $\mathcal{S}$ segundo a distribuição $\mathcal{D}$. Suponha que T_i tem pesos nas arestas dados pelo vetor c_i;

(ii) para cada árvore T_i, seja x_i^* uma solução ótima básica do problema (1) adaptado para a instância $(T_i, r, \mathcal{R}, c_i)$ do PSGraiz. Repita o experimento aleatório $128 \lg N \ln 4t$ vezes para T_i e x_i^* e denote por T_i' a união das árvores geradas;

(iii) se nenhuma árvore T_i' cobre $\mathcal{R}$, devolva G. Senão, dentre as árvores T_i' que cobrem $\mathcal{R}$, escolha a de menor custo. Denote esta árvore por $\dot{T}$;

(iv) considere uma trilha fechada W em $\dot{T}$ que passa exatamente duas vezes por cada aresta de $\dot{T}$. Note que cada vértice de G nesta trilha tem grau 2. Substitua cada trecho da trilha entre dois vértices $u, v \in V(G)$ consecutivos por um (u,v)-caminho mínimo em G e denote por H o grafo resultante. Devolva uma árvore geradora mínima $\hat{T}$ de H.

O algoritmo certamente devolve uma solução viável e é claramente polinomial. A seguir, provaremos sua razão de aproximação.

Teorema 14: *Se $\hat{T}$ é a solução devolvida pelo algoritmo, $c(\hat{T}) \leq (2^{17} \lg n \lg N \ln 4t) \mathrm{opt}\,(G, \mathcal{R}, c)$ com probabilidade pelo menos $1 - \epsilon$.*

Prova: Fixe $i = 1, \ldots, 8 \ln 1/\epsilon$. O custo esperado de T_i' é no máximo a soma dos custos das $128 \lg N \ln 4t$ árvores que a compõe. Pela Proposição 6, o custo esperado é então no máximo $(128 \lg N \ln 4t) c_i x_i^* \leq (128 \lg N \ln 4t) \mathrm{opt}\,(T_i, r, \mathcal{R}, c_i)$. Segue então da desigualdade de Markov que

$$\Pr[c_i(T_i') \geq 4 \cdot (128 \lg N \ln 4t) \mathrm{opt}\,(T_i, r, \mathcal{R}, c_i)] \leq 1/4.$$

Mas então a árvore T_i' tem custo baixo com probabilidade pelo menos $3/4$. Pelo Corolário 8, ela cobre $\mathcal{R}$ com probabilidade pelo menos $3/4$. Mas então ela tem custo baixo e cobre $\mathcal{R}$ com probabilidade pelo menos $1/4$.

Pela Proposição 13 e pela propriedade da distribuição $\mathcal{D}$, o custo esperado de uma solução ótima para o PSGraiz $(T_i, r, \mathcal{R}, c_i)$ é no máximo $(16 \lg n) \mathrm{opt}\,(G, \mathcal{R}, c)$. Logo, pela Desigualdade de Markov,

$$\Pr[\mathrm{opt}\,(T_i, r, \mathcal{R}, c_i) \geq 8 \cdot (16 \lg n) \mathrm{opt}\,(G, \mathcal{R}, c)] \leq 1/8.$$

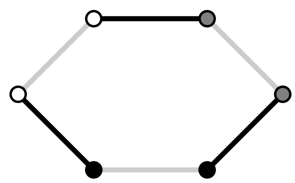


Figura 5: No exemplo acima, vértices de mesma cor estão num mesmo grupo. Considere o vetor x_0 indexado pelas arestas do grafo acima que atribui 1 às arestas pretas e 0 às arestas cinzas. Note que o vetor x_0 não pertence a $\hat{P}$ mas satisfaz todas as inequações de $\mathcal{R}$-cortes.

Portanto, a árvore T_i' cobre $\mathcal{R}$ e tem custo no máximo $(2^{16} \lg n \lg N \ln 4t)\mathrm{opt}\,(G, \mathcal{R}, c)$ com probabilidade pelo menos $1/8$. Mas então, a probabilidade de nenhuma das árvores T_i' cobrir $\mathcal{R}$ e ter custo baixo é no máximo $(1 - 1/8)^{8 \ln 1/\epsilon} \leq \epsilon$, logo alguma das árvores T_i' cobre $\mathcal{R}$ e tem custo baixo com probabilidade pelo menos $1 - \epsilon$.

Com isso, com probabilidade pelo menos $1-\epsilon$ vamos executar o passo (iv) do algoritmo e a árvore $\dot{T}$ considerada no passo (iv) será tal que $\dot{c}(\dot{T}) \leq (2^{16} \lg n \lg N \ln 4t)\mathrm{opt}\,(G, \mathcal{R}, c)$, onde $\dot{c}$ é o vetor de pesos para $\dot{T}$. Como $\dot{T}$ domina as distâncias em G, o custo do grafo H é no máximo $2\dot{c}(\dot{T})$. Mas então, com probabilidade pelo menos $1 - \epsilon$, temos $c(\hat{T}) \leq (2^{17} \lg n \lg N \ln 4t)\mathrm{opt}\,(G, \mathcal{R}, c)$, como queríamos. $\square$

4 Uma investigação poliédrica

Nesta seção apresentaremos brevemente alguns resultados obtidos durante o estudo de possíveis formulações lineares para o PSGrp. Vamos começar definindo o objeto de nosso estudo.

Considere um grafo $G = (V, E)$ e uma coleção de grupos $\mathcal{R}$. Dizemos que um subgrafo H de G é um $\mathcal{R}$-*subgrafo* de G se H conecta pelo menos um vértice de cada grupo em $\mathcal{R}$. Denote por $\mathcal{H}$ o conjunto de todos os $\mathcal{R}$-subgrafos de G e considere o politopo

$$P(G, \mathcal{R}) := \mathrm{conv.hull}\,\{\chi^{E(H)} \in \mathbb{R}^E : H \in \mathcal{H}\}.$$

O politopo $P(G, \mathcal{R})$ é chamado de *politopo do problema de Steiner com grupos*. Ele foi estudado por Salazar [19], que fornece uma formulação estendida e usa procedimentos de *lifting* para obter facetas de $P(G, \mathcal{R})$ a partir de facetas do politopo da árvore geradora mínima. Daqui em diante, quando não houver ambigüidade, vamos escrever P ao invés de $P(G, \mathcal{R})$.

Neste trabalho estamos interessados no dominante de $P(G, \mathcal{R})$, mais especificamente no poliedro

$$(5) \qquad \hat{P}(G, \mathcal{R}) := P(G, \mathcal{R}) + \mathbb{R}^E_+.$$

O poliedro $\hat{P}(G, \mathcal{R})$, ou simplesmente $\hat{P}$, claramente tem dimensão plena. Nas seções seguintes vamos apresentar duas possíveis formulações para $\hat{P}$ e explorar suas facetas.

4.1 Uma formulação inicial

Nesta seção apresentamos uma formulação inicial para $\hat{P}$. Considere um conjunto $S \subseteq V(G)$. Dizemos que $\delta(S)$ é um $\mathcal{R}$-*corte* se existem grupos $R_1, R_2 \in \mathcal{R}$ tais que $R_1 \subseteq S$ e $R_2 \cap S = \emptyset$. Note que, se H é um $\mathcal{R}$-subgrafo de G, H deve conter pelo menos uma aresta de cada $\mathcal{R}$-corte de G. Por analogia com o problema de Steiner, poderíamos ser levados a pensar que $\hat{P}$ é simplesmente o conjunto dos vetores inteiros $x \in \mathbb{R}^E_+$ que satisfazem $x(\delta(S)) \geq 1$ para todo $\mathcal{R}$-corte $\delta(S)$, mas como mostra a Figura 5, este não é o caso.

Portanto, para fornecermos uma formulação inteira para $\hat{P}$, precisamos de mais inequações. Uma partição $\mathcal{S} := (V_1, \ldots, V_p)$ de $V(G)$ é uma $\mathcal{R}$-*partição* se para cada V_i, $i = 1, \ldots, p$, existe um grupo

$R_i \in \mathcal{R}$ tal que $R_i \cap V_i = \emptyset$. Dada uma $\mathcal{R}$-partição $\mathcal{S}$, denotamos por $\Delta(\mathcal{S})$ o conjunto das arestas de G que têm pontas em partes distintas de $\mathcal{S}$. Obviamente, se $x \in \hat{P}$ então

(6) $\quad x(\Delta(\mathcal{S})) \geq 1$

para cada $\mathcal{R}$-partição $\mathcal{S}$. As desigualdades (6) são chamadas de *inequações de $\mathcal{R}$-partições.* Seja $P_1(G, \mathcal{R})$ o conjunto de todos os vetores $x \in \mathbb{R}_+^E$ que satisfazem todas as inequações de $\mathcal{R}$-partições. Obviamente, P_1 é um poliedro. Uma propriedade simples de P_1 é a seguinte:

Proposição 15: *Tome $x_0 \in P_1$. Existe um componente conexo de $G[\operatorname{supp} x_0]$ cujo conjunto de vértices cobre $\mathcal{R}$.*

Prova: Suponha que nenhum componente de $G[\operatorname{supp} x_0]$ cobre $\mathcal{R}$ e sejam $V_1, \ldots, V_p$ os conjuntos de vértices dos componentes de $G[\operatorname{supp} x_0]$. Então $\mathcal{S} := (V_1, \ldots, V_p)$ é uma $\mathcal{R}$-partição mas $x_0(\Delta(\mathcal{S})) = 0$, uma contradição. $\square$

Usando isso, podemos provar o seguinte resultado:

Corolário 16: $\hat{P} = \text{int.hull}\, P_1$.

Prova: Segue da Proposição 15 e do fato de P_1 ser do tipo bloqueador. $\square$

Vamos agora tratar do problema de separação para as inequações de $\mathcal{R}$-partições. O problema da separação é $\mathcal{NP}$-difícil em geral, já que podemos reduzir o problema do corte multiterminal mínimo [4] a ele. De fato, podemos provar que (veja [7]):

Teorema 17: *O problema da separação para as inequações de $\mathcal{R}$-partições é $\mathcal{NP}$-difícil mesmo se temos exatamente três grupos dois a dois disjuntos com dois vértices cada.*

O resultado abaixo segue então da equivalência entre separação e otimização [11]:

Corolário 18: *O problema de otimização sobre P_1 é $\mathcal{NP}$-difícil.* $\square$

Entretanto, observe que se existe um grupo unitário então toda $\mathcal{R}$-partição é também um $\mathcal{R}$-corte. Mas então o problema da separação pode ser resolvido em tempo polinomial como uma seqüência de $|\mathcal{R}| - 1$ problemas de fluxo máximo. Em outras palavras, temos o seguinte:

Proposição 19: *Se há um grupo unitário então a separação das inequações de $\mathcal{R}$-partições pode ser feita em tempo polinomial.* $\square$

Vale observar que se G é uma árvore, o problema da separação das inequações de $\mathcal{R}$-partições pode ser resolvido em tempo polinimial por um simples algoritmo de programação dinâmica. Logo, se G é uma árvore, podemos otimizar sobre P_1 em tempo polinimial.

Vamos agora apresentar rapidamente resultados sobre inequações que induzem facetas de $\hat{P}$. Estes resultados podem ser encontrados em [7]. Neste mesmo artigo, encontram-se procedimentos a partir dos quais pode-se obter inequações que induzem facetas de $\hat{P}$ a partir de inequações que induzem facetas do poliedro do problema de Steiner ou do problema da cobertura mínima por conjuntos. Apresentaremos apenas um dos resultados, que diz respeito às inequações de $\mathcal{R}$-partições.

Dizemos que uma $\mathcal{R}$-partição $\mathcal{S}$ é *minimal* se não existe uma $\mathcal{R}$-partição $\mathcal{S}'$ tal que $\Delta(\mathcal{S}')$ é um subconjunto próprio de $\Delta(\mathcal{S})$. É fácil ver que a $\mathcal{R}$-partição $\mathcal{S} := (V_1, \ldots, V_p)$ é minimal se e somente se toda aresta $uv \in \Delta(\mathcal{S})$, $u \in V_i$, $v \in V_j$, é tal que uv conecta componentes de $G[V_i]$ e $G[V_j]$ que juntos cobrem $\mathcal{R}$. Temos o seguinte resultado:

Teorema 20: *Seja $\mathcal{S} := (V_1, \ldots, V_p)$ uma $\mathcal{R}$-partição. A inequação*

(7) $\quad x(\Delta(\mathcal{S})) \geq 1$

induz uma faceta de $\hat{P}$ se e somente se $\mathcal{S}$ é minimal.

Prova: Primeiros mostramos a necessidade. Se $\mathcal{S}$ não é minimal então existe uma $\mathcal{R}$-partição $\mathcal{S}'$ tal que $\Delta(\mathcal{S}')$ é um subconjunto próprio de $\Delta(\mathcal{S})$. Mas então (7) é a soma das inequações válidas não-triviais $x(\Delta(\mathcal{S}')) \geq 1$ e $x(\Delta(\mathcal{S}) \setminus \Delta(\mathcal{S}')) \geq 1$, não sendo portanto uma inequação que induz faceta de $\hat{P}$.

Para ver a suficiência, seja $ax \geq \beta$ uma inequação que induz uma faceta de $\hat{P}$ tal que

$$\{x \in \hat{P}: x(\Delta(\mathcal{S})) = 1\} \subseteq \{x \in \hat{P}: ax = \beta\}.$$

Afirmamos que $ax \geq \beta$ é um múltiplo não-negativo de (7). Para ver isso, ponha $F := E(V_1) \cup \cdots \cup E(V_p)$. Seja $f \in \Delta(\mathcal{S})$ e $x_0 := \chi^{F \cup \{f\}}$. Como $\mathcal{S}$ é minimal, $x_0 \in \hat{P}$. Como $\hat{P}$ é do tipo bloqueador, para qualquer aresta $e \notin \Delta(\mathcal{S})$ temos que $x_0 + \chi^e \in \hat{P}$. Mas como x_0 e $x_0 + \chi^e$ ambos satisfazem (7) com igualdade, temos que $ax_0 = \beta = a(x_0 + \chi^e)$. Segue que $a(e) = 0$ para todo $e \notin \Delta(\mathcal{S})$.

Considere agora $e, f \in \Delta(\mathcal{S})$ e ponha $x_1 := \chi^{F \cup \{e\}}$ e $x_2 := \chi^{F \cup \{f\}}$. Ambos x_1 e x_2 pertencem a $\hat{P}$ e satisfazem (7) com igualdade, logo $ax_1 = \beta = ax_2$. Segue que $a(e) = a(f)$. Como e, f são arbitrárias, existe um número α tal que $a(e) = \alpha$ para cada aresta $e \in \Delta(\mathcal{S})$, provando a afirmação. Como $\hat{P}$ tem dimensão plena, segue que (7) induz uma faceta de $\hat{P}$. □

4.2 Formulações estendidas

Nesta seção consideraremos brevemente uma formulação estendida para $\hat{P}$ sobre a qual podemos otimizar em tempo polinomial. Mais informações podem ser encontradas em [7].

Novamente, considere um grafo $G = (V, E)$ e uma coleção $\mathcal{R}$ de grupos de vértices de G. Fixe um grupo $R_0 \in \mathcal{R}$ que será chamado de *grupo raiz*. Seja $P_2(G, \mathcal{R})$ o conjunto de todos os vetores $(x, y) \in \mathbb{R}_+^E \times \mathbb{R}_+^{R_0}$ satisfazendo

$$\begin{aligned} x(\delta(S)) &\geq y(S \cap R_0), \quad \text{para cada } S \subseteq V(G) \text{ que não cobre } \mathcal{R}, \\ y(R_0) &\geq 1. \end{aligned}$$

Denote por $P_2[E]$ a projeção de P_2 em $\mathbb{R}^E$. O seguinte resultado pode ser provado de forma análoga à Proposição 15.

Proposição 21: $\hat{P} = \text{int.hull}\, P_2[E]$. □

Ao contrário do problema de separação para P_1, o problema de separação para P_2 pode ser resolvido em tempo polinomial. Para ver isso, considere um vetor $(x_0, y_0) \in \mathbb{R}_+^E \times \mathbb{R}_+^{R_0}$ tal que $y_0(R_0) \geq 1$. Fixe um grupo $R \in \mathcal{R}$, $R \neq R_0$. Construa um grafo dirigido $D = (V', A)$ da seguinte forma:

(i) $V(D) := V(G) \cup \{s\}$;
(ii) para cada aresta uv de G, D tem arcos uv e vu;
(iii) D tem um arco rs para cada $r \in R$.

Considere agora um vetor de capacidades $w \in \mathbb{R}^A$ tal que $w(uv) := x_0(uv)$, $w(vu) := x_0(uv)$ para cada aresta uv de G e $w(rs) := \infty$ para cada $r \in R$. Considere também um vetor de demandas $b \in \mathbb{R}^V$ tal que $b(r) := -y_0(r)$ para cada $r \in R_0$, $b(s) := y_0(R_0)$ e $b(v) := 0$ para todos os demais $v \in V(D)$.

Proposição 22: *Existe um conjunto $S \subseteq V(G)$ tal que $R \cap S = \emptyset$ e $x_0(\delta(S)) < y_0(S \cap R_0)$ se e somente se existe um fluxo viável em D com capacidades dadas por w e demandas dadas por b.*

Prova: Suponha primeiro que existe um tal conjunto S. Então $w(\delta^+(S)) = x_0(\delta(S)) < y_0(S \cap R_0) = -b(S)$ e segue da caracterização de Gale de fluxos viáveis em redes [8] que não existe um fluxo viável em D com capacidades dadas por w e demandas dadas por b.

Suponha agora que não exista um tal fluxo viável. Mas então deve existir um conjunto $S \subseteq V(D)$ tal que $w(\delta^+(S)) < -b(S)$. Como $b(s) = y_0(R_0)$, não podemos ter $s \in S$. Mas então, como todos os arcos que saem de vértices de R têm capacidade infinita, devemos ter $S \cap R = \emptyset$. Segue então que $x_0(\delta(S)) = w(\delta^+(S)) < -b(S) = y_0(S \cap R_0)$, como queríamos. □

Como podemos decidir se há um tal fluxo viável em tempo polinomial e, se não há fluxo viável, podemos encontrar um conjunto S com as propriedades acima também em tempo polinomial, podemos resolver o problema da separação para as inequações $x(\delta(S)) \geq y(S \cap R_0)$ tais que $S \cap R = \emptyset$ em tempo polinomial. Portanto, o problema da separação para P_2 pode ser resolvido por uma seqüência de $|\mathcal{R}| - 1$ problemas de viabilidade de fluxo. Segue então da equivalência entre separação e otimização [11] que:

Corolário 23: *O problema de otimização sobre P_2 pode ser resolvido em tempo polinomial.* □

Não é verdade que $P_2[E] \subseteq P_1$. Mesmo se há um grupo unitário, se este grupo não for escolhido como sendo o grupo raiz, ainda há casos em que $P_2[E] \subseteq P_1$ é falso. Entretanto, como mostra o seguinte resultado, o valor ótimo de P_2 aproxima o valor ótimo de P_1.

Proposição 24: *Seja $c \in \mathbb{R}^E_+$. Então* $\min\{cx : x \in P_2[E]\} \geq (1/2) \min\{cx : x \in P_1\}$.

Para provar a Proposição 24, precisamos de um lema simples. Para introduzí-lo vamos primeiro introduzir uma definição. Seja ϵ tal que $0 < \epsilon \leq 1$ e denote por P_1^ϵ o conjunto de todos os vetores $x \in \mathbb{R}^E_+$ que satisfazem

$$x(\Delta(\mathcal{S})) \geq \epsilon$$

para toda $\mathcal{R}$-partição $\mathcal{S}$.

Lema 25: *Seja $c \in \mathbb{R}^E_+$ e ϵ tal que $0 < \epsilon \leq 1$. Então* $\min\{cx : x \in P_1^\epsilon\} = \epsilon \min\{cx : x \in P_1\}$.

Prova: Seja x_1 uma solução ótima de $\min\{cx : x \in P_1^\epsilon\}$ e x_2 uma solução ótima de $\min\{cx : x \in P_1\}$. Como $\epsilon^{-1} x_1 \in P_1$, $cx_2 \leq \epsilon^{-1} cx_1$. Analogamente, como $\epsilon x_2 \in P_1^\epsilon$, $cx_1 \leq \epsilon cx_2$. Segue que $cx_1 \leq \epsilon cx_2 \leq \epsilon\epsilon^{-1} cx_1$, como queríamos. □

Com isso, podemos provar a Proposição 24.

Prova da Proposição 24: Seja $x_0 \in P_2[E]$ e y_0 tal que $(x_0, y_0) \in P_2$. Considere uma $\mathcal{R}$-partição $\mathcal{S} := (V_1, \ldots, V_p)$. Como $\mathcal{S}$ é uma $\mathcal{R}$-partição, para cada $i = 1, \ldots, p$ existe $R \in \mathcal{R}$ tal que $R \cap V_i = \emptyset$. Mas então, como $(x_0, y_0) \in P_2$, $x_0(\delta(V_i)) \geq y_0(V_i \cap R_0)$ para cada $i = 1, \ldots, p$. Logo

$$\begin{aligned} y(R_0) &= \sum_{i=1}^{p} y_0(V_i \cap R_0) \\ &\leq \sum_{i=1}^{p} x_0(\delta(V_i)) \\ &= 2x_0(\Delta(\mathcal{S})), \end{aligned}$$

e, como $y_0(R_0) \geq 1$, segue que $x_0 \in P_1^{1/2}$. Mas então segue do Lema 25 que

$$\min\{cx : x \in P_2[E]\} \geq \min\{cx : x \in P_1^{1/2}\} = (1/2) \min\{cx : x \in P_1\},$$

como queríamos. □

5 Conclusão

Neste texto, descrevemos o problema de Steiner com grupos, alguns algoritmos de aproximação que foram propostos para ele e alguns resultados poliédricos que o envolvem.

Nosso projeto consiste em implementar estes algoritmos de aproximação, bem como implementar um algoritmo exato baseado na estratégia *branch-and-cut*. Isto possibilitará uma análise empírica da qualidade das soluções produzidas pelos algoritmos de aproximação estudados.

Até o momento, fizemos um estudo detalhado da maior parte dos algoritmos de aproximação citados na seção 3 e também um estudo preliminar de formulações inteiras para o problema. Temos também resultados sobre métodos de redução de instâncias, que têm se mostrado muito úteis na solução de problemas que surgem em aplicações.

As atividades envolvidas na finalização deste projeto são as seguintes:

Núm.	**Atividade**
1	Cumprimento dos créditos
2	Estudo da bibliografia
3	Implementação
4	Testes
5	Escrita da dissertação
6	Defesa

O cronograma que propomos para a realização destas atividades encontra-se abaixo:

	2004												**2005**					
Núm.	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6
1		×	×	×	×	×												
2	×	×	×	×	×	×	•	•	•	•								
3								•	•	•	•	•	•					
4												•	•					
5												•	•	•	•	•	•	
6																		•

Acima, um "×" indica uma atividade que já foi realizada.

Referências

[1] N. Alon and J. Spencer. *The Probabilistic Method.* Wiley Interscience, 1992.

[2] Y. Bartal. On approximating arbitrary metrics by tree metrics. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 161–168, 1998.

[3] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms.* MIT Press, Cambridge, MA, second edition, 2001.

[4] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23:864–894, 1994.

[5] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *FOCS*, 2003.

[6] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45:634–652, 1998.

[7] C. E. Ferreira and F. M. de Oliveira Filho. Some formulations for the group Steiner tree problem. Em preparação, 2004.

[8] D. Gale. A theorem on flows in networks. *Pacific Journal of Mathematics*, 7:1073–1082, 1957.

[9] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, San Francisco, 1979.

[10] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. *Journal of Algorithms*, 37(1):66–84, 2000.

[11] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, 1988.

[12] C. H. Helving, R. Robins, and A. Zelikovsky. An improved approximation scheme for the group Steiner tree problem. *Networks*, 37(1):8–20, 2001.

[13] F. K. Hwang, D. S. Richards, and P. Winter. *The Steiner Tree Problem*, volume 53 of *Annals of Discrete Mathematics*. North-Holland, 1992.

[14] E. Ihler. Bounds on the quality of approximate solutions to the group Steiner problem. In *Graph-Theoretic Concepts in Computer Science WG90*, volume 484 of *Lecture Notes in Computer Science*, pages 109–118. Springer, 1991.

[15] E. Ihler. The complexity of approximating the class Steiner tree problem. In *Graph-Theoretic Concepts in Computer Science WG91*, volume 570 of *Lecture Notes in Computer Science*, pages 85–96. Springer, 1992.

[16] S. Janson. Poisson approximations for large deviations. *Random Structures and Applications*, 1:221–230, 1990.

[17] R. M. Karp. Reducibility among combinatorial problems. In Miller and Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.

[18] G. Reich and P. Widmayer. Beyond Steiner's problem: A VLSI oriented generalization. In *Graph-Theoretic Concepts in Computer Science WG89*, volume 411 of *Lecture Notes in Computer Science*, pages 196–210. Springer, 1990.

[19] J. J. Salazar. A note on the generalized Steiner tree polytope. *Discrete Applied Mathematics*, 100:137–144, 2000.

[20] P. Slavík. *Approximation Algorithms for Set Cover and Related Problems*. Tese de doutorado, University of New York at Buffalo, 1998.

[21] H. Takahashi and A. Matsuyama. An approximate solution for the Steiner problem in graphs. *Math. Japonica*, 24:573–577, 1980.