

UNIVERSIDADE FEDERAL DO ABC
Mestrado em Matemática Aplicada



WENDHEL RAFFA COIMBRA

**Árvores de Steiner: Teoria, Geração
Numérica e Aplicações**

Santo André
16/12/2009

Árvores de Steiner - Teoria, Geração Numérica e Aplicações

Este exemplar corresponde à redação final da dissertação devidamente corrigida e defendida por Wendhel Raffa Coimbra e aprovada pela comissão julgadora.

Santo André, 16 de dezembro de 2009.

Prof. Dr. Valério Ramos Batista
Orientador

Prof. Dr. Marcelo Zanchetta do
Nascimento
Co-orientador

Banca examinadora:

1. Prof. Dr. Valério Ramos Batista (Orientador) - UFABC
2. Prof. Dr. Márcio Fabiano da Silva - UFABC
3. Prof. Dr. Karcus Day Rosario Assis - UFRB

Dissertação apresentada junto ao Centro de Matemática, Computação e Cognição, UFABC, como requisito parcial para obtenção do Título de Mestre em Matemática Aplicada.



Universidade Federal do ABC

Serviço Público Federal
UNIVERSIDADE FEDERAL DO ABC

*Assinaturas dos membros da Banca Examinadora que avaliaram e aprovaram a defesa de Dissertação de Mestrado do candidato **Wendhel Raffa Coimbra**, realizada em 16 de dezembro de 2009:*

Valério Ramos Batista

Dr. Valério Ramos Batista (UFABC)

Márcio Fabiano da Silva

Dr. Márcio Fabiano da Silva (UFABC)

Karcus Day Rosário Assis

Dr. Karcus Day Rosário Assis (UFRB)

Ficha catalográfica produzida pela Biblioteca da Universidade Federal do ABC

COIMBRA, Wendhel Raffa

Árvores de Steiner : Teoria, geração numérica e aplicações / Wendhel Raffa Coimbra
— Santo André : Universidade Federal do ABC, 2009.

76 fls. 27 cm

Orientador: Valério Ramos Batista

Co-orientador: Marcelo Zanchetta do Nascimento

Dissertação (Mestrado) — Universidade Federal do ABC, Centro de Matemática,
Computação e Cognição, Matemática, 2009.

1. Algoritmos numéricos 2. Programação gráfica 3. Árvore de Steiner I. BATISTA,
Valério Ramos . II. NASCIMENTO, Marcelo Zanchetta do. III. Centro de Matemática,
Computação e Cognição, Matemática, 2009, IV. Título.

CDD 510

*Dedico este trabalho a minha mãe e avós.
Valdete Fátima Raffa, Santina de Castro Raffa, Francisco Raffa
por todo apoio durante a vida escolar e acadêmica.*

Agradecimentos

Primeiramente a Deus por tudo.

A minha mãe Valdete F. Raffa e avó Santina Raffa por estarem sempre me apoiando nos momentos mais difíceis, pelas orações mesmo estando distantes e por incentivar a concluir este trabalho.

Ao orientador Valério Ramos Batista e co-orientador Marcelo Zanchetta do Nascimento pela paciência, apoio, dedicação e ajuda durante todo esse tempo de convivência.

Aos meus tios Waldir Raffa e Sérgio Ferreira, minhas tias Patrícia M. Raffa Ferreira e Neuza R. Raffa e meus primos Guilherme R. Raffa, Marcelo H. R. Raffa por sempre me incentivarem.

Meus priminhos Tiago Raffa Ferreira e Maria Eduarda Raffa Ferreira pela alegria.

Aos meus grandes amigos Joel e Leandro da república de Três Lagoas, aos amigos de Três Lagoas e curso de matemática (UFMS) Ana Carla Lelis, Bruno Andrade (Brunão), Bruno Locatelli, Bruno Henrique (Ventuinha), Mateus (Mateusão), Marcelo (Bomba), Luiz Henrique (Moleza), Patrícia (Pati) e aos professores Antônio Carlos Tamarozzi e Sônia Angelina pela ajuda durante todo esse tempo.

A todos amigos, principalmente Mário (Timba), Mateus (João), Marcel, Braga, Diego (Zói), Carlos (Juninho) e Farid pelos momentos de descontração.

Aos colegas da graduação e a turma de mestrado da UFABC, Alexandre, Ana Paula, Anna Lígia, Danilo, Douglas, Jéssica, Jocylaine, Michele, Simone, Márcio, Pablo e ao professor Márcio Silva pelo apoio e ajuda.

À Bolsa Institucional da UFABC, que no período de 01/03/2008 a 28/02/2009 financiou este trabalho de mestrado. Também agradeço à Bolsa FAPESP de Mestrado, que no período de 01/03/2009 a 31/12/2009 financiou este trabalho.

A todos vocês, muito obrigado.

Resumo

Dado um conjunto de pontos no plano, que denominamos terminais, prova-se que sempre existe um grafo minimal que os conecta, chamado “árvore de Steiner” (veja [GP]). Os *terminais* podem representar centros de conexão para rotas, elementos de circuito elétrico, ou servidores principais de rede de computadores. Ou seja, o problema em questão é *otimizar a comunicação* entre os terminais, caso isto seja representado por uma árvore de menor comprimento possível. Nem sempre o “menor comprimento” representa a otimização. O Problema de Steiner possui variações em que as arestas da árvore só podem seguir direções horizontal e vertical, como no caso de circuitos elétricos (veja [Fo]). Outra variação é quando cada ponto Steiner tem custo muito alto, e pretende-se obter uma tal árvore com o menor número de tais pontos. Ela será “mínimo local” para comprimento, mas não necessariamente global (veja [Ch]).

Na área de redes de computadores, aplicam-se árvores de Steiner na distribuição de vídeo, conferências multimídia que utilizam comunicação “multicast” para transmissão de dados (vide [Ji]). Desenvolvemos o trabalho em dois pilares principais: Teoria de Superfícies Mínimas e Implementação de Algoritmos Aproximativos. Uma motivação física para “árvore de Steiner” é que ela pode ser também realizada por películas de sabão, e por isso compartilham propriedades de Superfícies Mínimas. Como exemplo, considere uma solução de sabão. Ao mergulharmos e retirarmos duas placas paralelas ligadas por pinos, uma película irá conectá-los. Esta representa um grafo de comprimento mínimo que interliga os pinos. Como é sabido, as películas de sabão realizam as Superfícies Mínimas.

Para visualizá-la, recorre-se a Algoritmos Numéricos e Programação Gráfica. Os métodos baseiam-se principalmente na implementação dos algoritmos da referência [GP], implementação esta que foi realizada utilizando o “software” MatLab. Os desafios envolvidos se pautam basicamente na programação computacional dos resultados teóricos descritos em [GP]. O problema da árvore de Steiner é bem conhecido, e um ótimo exemplo de interface entre Superfícies Mínimas e Análise Numérica.

Abstract

Given a set of points in the plane, which we call *terminals*, one proves that they are always connected by a minimal graph called “Steiner tree”. The terminals may represent main connection route points, circuit elements or network computer servers. That is, the problem is to optimize traffic among the terminals whenever this is represented by a tree of shortest length. Not always the shortest length corresponds to an optimisation. The Steiner Problem has variations like moving only in horizontal and vertical directions, as happens to circuits (see [Fo]). Another variation is when each Steiner point has a high cost, so one seeks for such a tree with the smallest number of them. In this case, it will be a “local minimum” of length, but not necessarily a global minimum (see [Ch]).

Steiner trees find application in Computer Networks, in video broadcast and conferences which use “multicast” communication for data transmission (see [Ji]). We develop this work by means of two main theories: Minimal Surfaces and Approximation Algorithms. Steiner trees may be physically realised through soap films, and these share many properties with Minimal Surfaces. As an example, consider a soap solution with two parallel plates connected by pins embedded therein. After one takes them out, a soap film will connect the pins, and it represents a minimal graph. Soap films realise Minimal Surfaces in practical experiments.

In order to visualise the Steiner trees one makes use of Numerics and Graphic Programming. The methods used in this work are essentially based upon the algorithms from [GP] and we chose the software MatLab for this purpose. Most of the work consisted in programming the theoretical results from [GP]. The Steiner tree problem is largely known and a good example of interface between Minimal Surfaces and Numerics.

Sumário

1	Introdução	1
2	Preliminares	5
2.1	Algumas demonstrações	5
2.2	Conceitos básicos	7
3	Árvores de Steiner	11
3.1	Introdução	11
3.2	Espécies de árvores	12
4	Propriedades básicas	15
4.1	Introdução	15
4.2	Arestas de um vértice	15
4.3	Números de pontos Steiner	17
4.4	Teorema de Maxwell	18
4.5	Árvore plena de Steiner	19
4.6	Envelope convexo	20
4.7	Unicidade das árvores mínimas relativas	21
4.8	Caso geral (mais de três pontos)	22
4.9	Propriedade da lua	24
4.10	Propriedade da cunha	25
4.11	Distância entre pontos Steiner	26
4.12	Arestas $A_i A_j$	28
4.13	Propriedade do diamante	29
4.14	Região de decisão	34

4.15	Conjectura $\frac{\sqrt{3}}{2}$	36
5	Metodologia	37
5.1	Algoritmo Steiner	37
5.2	Algoritmo Cvxhull	41
5.3	Algoritmo Lune	41
5.4	Algoritmo Remake	42
5.5	Algoritmo Fullt	45
5.6	Algoritmo Lasttry	47
5.7	Algoritmo Mksaw	47
5.8	Algoritmo Hall	48
5.9	Algoritmo de Prim	48
6	Resultados	57
6.1	Dois cenários	57
7	Conclusões	61
7.1	Pesquisas futuras	62
	Bibliografia	63

Lista de Figuras

1.1	(a) Placa com 3 pinos; (b) com 5 pinos.	2
1.2	(a) Árvore de Steiner a partir de Prim; (b) árvore mínima de Steiner.	3
2.1	As aplicações F e \tilde{F}	6
2.2	Grafo da matriz de adjacência.	10
2.3	Tabela da matriz de adjacência.	10
3.1	Um grafo retangular, e outro em “X” de menor comprimento.	11
3.2	Mínimo local.	12
3.3	Mínimo global.	12
3.4	Mínimo local numa topologia.	13
3.5	Variação da Figura 3.2.	13
3.6	Mínimo local em outra topologia.	13
4.1	Ângulo $P\hat{R}Q$ máximo.	15
4.2	Lema dos ângulos de 120°	16
4.3	Figura 3.3 com pontos Steiner.	18
4.4	Representação do Teorema de Maxwell.	19
4.5	Mínimos locais.	21
4.6	Arcos de 120°	22
4.7	Construção de S	24
4.8	(a) Triângulo 1; (b) triângulo 2.	26
4.9	(a) Figura 3.2 (contas); (b) Figura 3.3 (contas).	27
4.10	Propriedade do Diamante.	30
4.11	Leve perturbação de uma árvore, alterando sua topologia.	31
4.12	Ilustração do Caso 1.	31

4.13	Ilustração do Caso 2.	32
4.14	Ilustração do Caso 3.	32
4.15	Aplicação do Teorema de Tales.	33
4.16	Ilustração do Caso 4.	33
4.17	Caso 4 (Parte I).	34
4.18	Caso 4 (Parte II).	34
4.19	A região <i>I</i>	35
5.1	Fluxograma do algoritmo Stree.	38
5.2	Representação da localização do ponto Steiner.	39
5.3	Fluxograma do algoritmo para cálculo do ponto Steiner.	40
5.4	Fluxograma do algoritmo Cvxhull.	42
5.5	Fluxograma do algoritmo Lune.	43
5.6	Fluxograma Remake (Parte I).	45
5.7	Fluxograma Remake (Parte II).	46
5.8	Fluxograma Remake (Parte III).	47
5.9	Fluxograma do algoritmo Fullt (parte I).	50
5.10	Fluxograma do algoritmo Fullt (parte II).	51
5.11	Fluxograma do algoritmo Lasttry (Parte I).	52
5.12	Fluxograma do algoritmo Lasttry (Parte II).	53
5.13	Exemplo da Mksaw.	53
5.14	Fluxograma do algoritmo Mksaw.	54
5.15	Fluxograma do algoritmo Hall.	55
5.16	Fluxograma do algoritmo Prim.	56
6.1	Cenário 1.	58
6.2	Cenário 2 (Película).	59

Capítulo 1

Introdução

Dado um conjunto de pontos no plano, prova-se que sempre existe um grafo minimal que os conecta, chamado “árvore de Steiner”. Um modo de visualizá-la é com o seguinte procedimento: Em uma solução de sabão, ao mergulharmos e retirarmos duas placas paralelas ligadas por pinos, uma película irá conectá-los (vide Figura 1.1). Ao observarmos de cima, vemos um grafo de comprimento mínimo que interliga os pinos. Como é sabido, as películas de sabão realizam as Superfícies Mínimas.

Portanto, compartilham propriedades de Superfícies Mínimas. Na área de redes de computadores, aplicam-se árvores de Steiner na distribuição de vídeo, conferências multimídia que utilizam comunicação “multicast” para transmissão de dados. A tese desenvolve-se em dois pilares principais: Teoria de Superfícies Mínimas e Implementação de Algoritmos Aproximativos.

Para gerá-las computacionalmente, recorre-se a Algoritmos Numéricos e Processamento de Dados.

O presente trabalho consiste em estudar com profundidade a Teoria de Árvores de Steiner e sua Implementação em Linguagens de Programação. Dado um conjunto de N pontos no plano, chamados “terminais”, procura-se um grafo conexo de menor comprimento que interliga todos. Naturalmente, este grafo é uma árvore, e a Teoria de Superfícies Mínimas garante sua existência e demais propriedades: ela poderá ter outros vértices além dos terminais, chamados “pontos Steiner”, em número menor ou igual a $N - 2$, dos quais partem sempre exatamente três arestas formando um ângulo de 120° duas a duas.

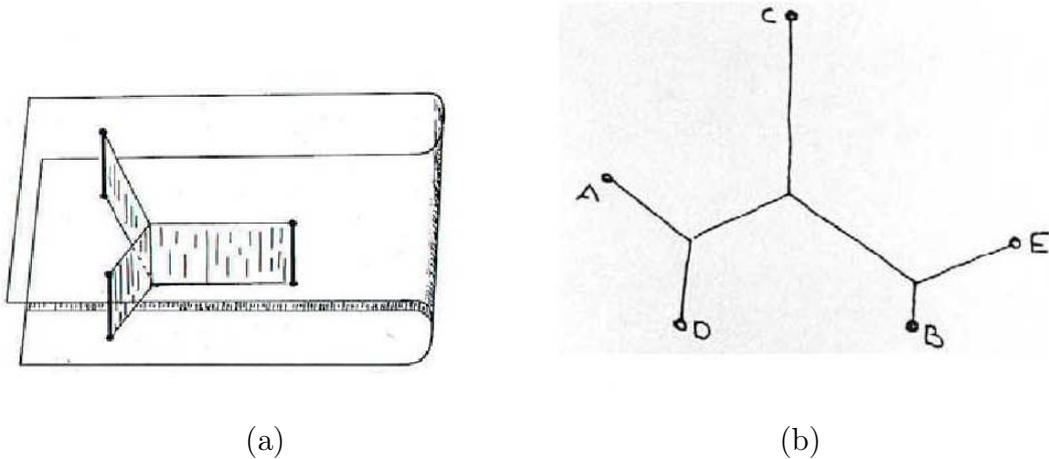


Figura 1.1: (a) Placa com 3 pinos; (b) com 5 pinos.

O problema tem variações em \mathbb{R}^3 e outros espaços, mas já no \mathbb{R}^2 a resolução numérica não é trivial. Diferentes algoritmos procuram reduzir o tempo computacional baseados em critérios como simetrias e distribuição dos terminais.

É um grande desafio implementar multicast em redes Wide Area Networks (WANs) devido à grande complexidade para tal roteamento [Ji,Sa]. Este busca construir uma árvore de roteamento com todas as terminais de destino, permitindo que informações possam ser transmitidas de uma fonte para todos os destinos ao longo da árvore. O principal objetivo do roteamento multicast é minimizar o custo dos caminhos entre o terminal de origem e as conexões de destino. Pesquisas vêm sendo desenvolvidas com o objetivo de obter soluções através de árvores de Steiner para otimização de rotas [Au,Hu].

Sua utilidade prática, aliada à Teoria de Superfícies Mínimas, forma a principal motivação do presente tema de mestrado.

No desenvolvimento do trabalho de mestrado, utilizamos as referências citadas, bem como implementação de algoritmos em MatLab. De fato, já existem programas com pacotes para árvores de Steiner, como o *Geosteiner* e o *Surface Evolver*, disponíveis em <http://www.diku.dk/geosteiner> e <http://www.susqu.edu/brakke>, respectivamente. Entretanto, estes pacotes não são portáteis para linguagens de uso mais difundido. Devido a isso, disponibilizamos em MatLab o programa “stree.p” e alguns *módulos didáti-*

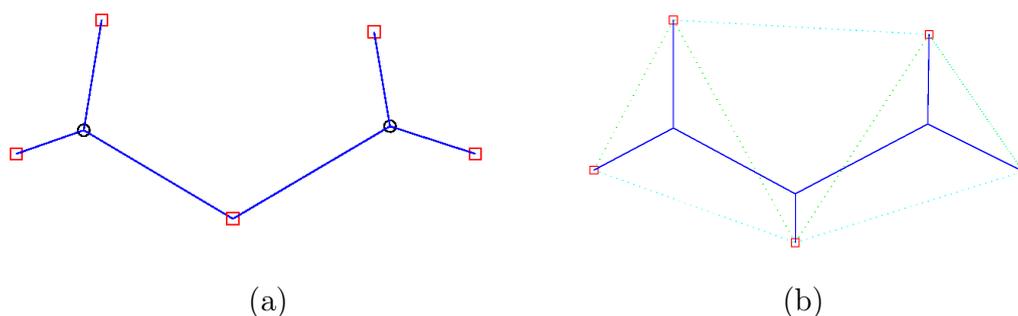


Figura 1.2: (a) Árvore de Steiner a partir de Prim; (b) árvore mínima de Steiner.

cos na página <http://sites.google.com/site/wraffa/arquivos>

A presente tese de mestrado pode ser parcialmente acompanhada estudando o código dos módulos, e não apenas executando-os. Como propósito didático, omitimos o “Help”, justamente para estimular a *leitura* destes códigos, e então o aprendizado da linguagem MatLab.

Além disso, códigos disponíveis como “steiner.c”, de Robert Bell, partem de uma “minimal spanning tree” para construir uma árvore de Steiner, que em geral não é a “árvore mínima de Steiner”. A diferença é que a primeira representa um mínimo local de comprimento, e a segunda, um mínimo global (vide Figuras 2(a) e 2(b)). Pretendemos utilizar [GP] com o intuito de obter este mínimo global.

Capítulo 2

Preliminares

2.1 Algumas demonstrações

No que segue, denotaremos o produto interno de vetores U, V por $\langle U, V \rangle$, $U \cdot V$ ou UV .

Lema 2.1. *Seja $F : \Omega \rightarrow \mathbb{R}^3$ e tome $q \in \Omega, F(q) = p$. Então são equivalentes:*

1. F_x, F_y são linearmente independentes em p ;
2. DF possui posto 2;
3. Existe submatriz 2×2 de DF com $\det \neq 0$;
4. $F_x \times F_y \neq 0$;
5. $\det G \neq 0, G = DF^t \cdot DF$.

Demonstração. [2] \Leftrightarrow [3]. Por definição.

[5] \Leftrightarrow [4]. Sabemos que $\|F_x \times F_y\| = \|F_x\| \|F_y\| \sin \theta$ e $\langle F_x, F_y \rangle = \|F_x\| \|F_y\| \cdot \cos \theta$, então $\|F_x\|^2 \|F_y\|^2 = \|F_x \times F_y\|^2 + \langle F_x, F_y \rangle^2$ (*).

Agora,

$$\det G = \det \begin{bmatrix} F_x F_x & F_x F_y \\ F_x F_y & F_y F_y \end{bmatrix} = \|F_x\|^2 \|F_y\|^2 - \langle F_x, F_y \rangle^2 \stackrel{(*)}{=} \|F_x \times F_y\|^2.$$

[4], [5] \Rightarrow [1]. Como $\|F_x \times F_y\| \neq 0$, temos que $\|F_x\| \|F_y\| \sin \theta \neq 0 \Rightarrow F_x \neq \vec{0}, F_y \neq \vec{0}, \theta \neq 0^\circ$ e $\theta \neq 180^\circ$. Assim, não podemos ter $\mu F_x = \lambda F_y \therefore F_x$ e F_y são *LI*.

[2], [3] \Leftrightarrow [1]. Temos $\det G \neq 0$ para alguma submatriz $2 \times 2 \Leftrightarrow \nexists \mu, \lambda$ de modo que $\mu F_x = \lambda F_y \Leftrightarrow F_x$ e F_y são *LI*.

[1] \Rightarrow [4]. F_x, F_y são *LI* $\Rightarrow F_x \neq \vec{0}, F_y \neq \vec{0}$ e $\theta \neq \{0^\circ, 180^\circ\} \Rightarrow \|F_x \times F_y\| = \|F_x\| \|F_y\| \sin \theta \neq 0 \Rightarrow F_x \times F_y \neq 0$. \square

Defina S (superfície em \mathbb{R}^3) como sendo (F, Ω) , onde F é diferenciável e Ω é um subconjunto aberto e conexo do \mathbb{R}^2 .

$$F : \Omega \subset \mathbb{R}^2 \longrightarrow \mathbb{R}^3 \\ (x, y) \mapsto F(x, y)$$

Sejam $\Delta \subseteq \Omega$ tal que $\partial\Delta$ é o traço de uma curva simples e fechada φ e \tilde{S} a superfície definida pela restrição de F a Δ . Dizemos que S é mínima se possui a menor área entre todas superfícies \tilde{S}_λ diferenciáveis de modo que:

$$\tilde{F}_\lambda : \Omega \longrightarrow \mathbb{R}^3 \\ (x, y) \mapsto \tilde{F}_\lambda(x, y) \text{ tais que } \tilde{F}_\lambda|_{\{\varphi\}} \equiv F_\lambda|_{\{\varphi\}}$$

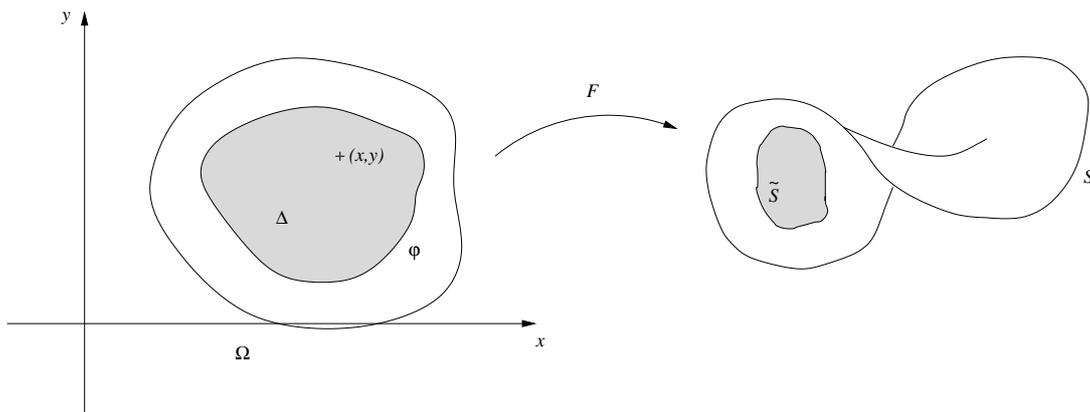


Figura 2.1: As aplicações F e \tilde{F} .

Definição 2.1. Uma superfície S definida por (F, Ω) é **regular** num ponto se F_x e F_y são LI neste ponto, ou ainda, se $\det G > 0$, em que $G = DF^t \cdot DF$, $DF = [F_x \ F_y]_{3 \times 2}$. A superfície S será regular se ela é regular em todos os pontos de Ω .

A área de uma superfície S é dada por:

$$A(S) = \iint_{\Omega} \sqrt{\det G} \, dx dy.$$

Observação 2.1. A área de S independe da parametrização, pois se $(\tilde{F}, \tilde{\Omega})$ é tal que $\tilde{F}(\tilde{\Omega}) = S = F(\Omega)$, então temos o correspondente \tilde{G} com

$$\det \tilde{G} = \det G \left(\frac{\partial (u_1, u_2)}{\partial (\tilde{u}_1, \tilde{u}_2)} \right)^2 > 0.$$

Além disso,

$$\begin{aligned} A(\tilde{S}) &= \iint_{\tilde{\Delta}} \sqrt{\det \tilde{G}} \, d\tilde{x} d\tilde{y} = \iint_{\tilde{\Delta}} \sqrt{\det G \cdot (\det(U))^2} \, d\tilde{x} d\tilde{y} = \\ &= \iint_{\tilde{\Delta}} \sqrt{\det G} \cdot \left| \frac{\partial (x, y)}{\partial (\tilde{x}, \tilde{y})} \right| \, d\tilde{x} d\tilde{y} = \iint_{\Delta} \sqrt{\det G} \, dx dy = A(S). \end{aligned}$$

2.2 Conceitos básicos

Definição 2.2. Um conjunto M não-vazio é dito **espaço métrico** se $\exists d : M \times M \longrightarrow [0, \infty)$ tal que quaisquer $p, q, r \in M$ verificam:

- a. $d(p, q) > 0 \Leftrightarrow p \neq q$;
- b. $d(p, q) = d(q, p)$;
- c. $d(p, q) \leq d(p, r) + d(r, q)$.

Observação 2.2. d é chamada **função distância** (ou **métrica**) em M .

Definição 2.3. Se M é espaço vetorial, então uma **norma** em M é uma função $\| \cdot \| : M \longrightarrow [0, \infty)$ tal que quaisquer $u, v \in M$ e $c \in \mathbb{R}$ verificam:

- a. $\|v\| > 0 \Leftrightarrow v \neq 0$;
- b. $\|cv\| = |c| \|v\|$;
- c. $\|u+v\| \leq \|u\| + \|v\|$.

Observação 2.3. Uma norma $\|\cdot\|$ define $d(u, v) := \|u-v\|$ como métrica, mas a recíproca é falsa. Por exemplo, tome $d : M \times M \rightarrow \{0, 1\}$. Se tivéssemos $\|\cdot\|$ com $d(v, 0) = \|v\| = 1$, então $d(2v, 0) = \|2 \cdot v\| = 2 \cdot 1 \notin \{0, 1\}$.

Definição 2.4. Uma partição P de $[a, b] \subset \mathbb{R}$ é uma $(n+1)$ -upla (t_0, t_1, \dots, t_n) tal que $a = t_0 < t_i < t_{i+1} < t_n = b, \forall i \in \{1, \dots, n-1\}$. Denotamos \wp o conjunto de todas as $P, \forall n \geq 1$.

Definição 2.5. Dada $\alpha : [a, b] \xrightarrow{C^0} M$ onde M é um espaço métrico, definimos o comprimento de α como $l(\alpha) = \sup \Sigma$, onde

$$\Sigma := \left\{ \sum_{i=1}^n d(\alpha(t_i), \alpha(t_{i-1})), P \in \wp \right\} \subset [0, \infty].$$

Observação 2.4. Temos $\alpha([a, b])$ compacto, mas pode ocorrer $l(\alpha) = \infty$.

Exemplo 2.1. Curva de Peano, graf (f) onde $f : [a, b] \xrightarrow{C^0} \mathbb{R}$ e $\nexists f'(x), \forall x \in [a, b]$. Na verdade $l(\alpha) < \infty \Rightarrow \exists \alpha'$ quase sempre (vide [WZ]).

Proposição 2.1. Se $\varphi : [c, d] \xrightarrow{C^0} [a, b]$ é bijetora e $\beta = \alpha \circ \varphi$, então $l(\beta) = l(\alpha)$.

Demonstração. $\forall P \in \wp_{[a,b]}$ temos $\varphi^{-1}(P) \in \wp_{[c,d]}$ e $\forall Q \in \wp_{[c,d]} \Rightarrow \varphi(Q) \in \wp_{[a,b]}$. Além disso, $d(\beta(s_i), \beta(s_{i-1})) = d(\alpha(t_i), \alpha(t_{i-1})), t = \varphi(s)$. Logo, os conjuntos $\Sigma_{\wp_{[a,b]}}$ e $\Sigma_{\wp_{[c,d]}}$ são iguais, e portanto têm o mesmo supremo. \square

A partir de agora trabalharemos somente com curvas α **retificáveis**, isto é, aquelas tais que $l(\alpha) < \infty$. Diremos que φ como na Proposição 2.1 é uma **reparametrização**.

Proposição 2.2. *Se M é espaço vetorial normado e $\alpha \in C^1([a, b])$, então $l(\alpha|_{[a, s]}) = l(s) = \int_a^s \|\alpha'(t)\| dt, \forall s$.*

Demonstração. $\|\alpha(t_i) - \alpha(t_{i-1})\| = \left\| \int_{t_{i-1}}^{t_i} \alpha'(t) dt \right\| \leq \int_{t_{i-1}}^{t_i} \|\alpha'(t)\| dt$.

Logo, para algum $P \in \wp_{[a, s]}$, $\sum_{i=1}^n \|\alpha(t_i) - \alpha(t_{i-1})\| \leq \sum_{i=1}^n \int_{t_{i-1}}^{t_i} \|\alpha'(t)\| dt = \int_a^s \|\alpha'(t)\| dt \Rightarrow l(\alpha|_{[a, s]}) \leq \int_a^s \|\alpha'(t)\| dt$. Agora, para $s > \tau \geq a$ vale $\|\alpha(s) - \alpha(\tau)\| \leq \sup \Sigma_{\wp_{[\tau, s]}} + \sup \Sigma_{\wp_{[a, \tau]}} - \sup \Sigma_{\wp_{[a, \tau]}} = \sup \Sigma_{\wp_{[a, s]}} - \sup \Sigma_{\wp_{[a, \tau]}} = l(s) - l(\tau) \leq \int_{\tau}^s \|\alpha'(t)\| dt = \|\alpha'(t^*)\| \cdot (s - \tau)$, pelo Teorema do Valor Médio, onde $t^* \in [\tau, s]$.

Logo, $\frac{\|\alpha(s) - \alpha(\tau)\|}{s - \tau} \leq \frac{l(s) - l(\tau)}{s - \tau} \leq \|\alpha'(t^*)\| \Rightarrow \exists l'(\tau) = \|\alpha'(\tau)\|$, e

a mesma conclusão vale para $\tau > s \geq a$. Logo $\int_a^s \|\alpha'(\tau)\| d\tau = l(s)$. \square

Definição 2.6. *Dadas $\beta : [0, 1] \xrightarrow{C^0} \mathbb{R}^2$ e $\gamma : [0, 1] \xrightarrow{C^0} \mathbb{R}^2$ com $\beta(1) = \gamma(0)$, a concatenação $\beta * \gamma$ é dada por $\beta * \gamma : [0, 1] \xrightarrow{C^0} \mathbb{R}^2$, $(\beta * \gamma)(t) = \beta(2t)$ se $0 \leq t \leq \frac{1}{2}$, ou $\gamma(2t - 1)$ se $\frac{1}{2} \leq t \leq 1$. A tabela abaixo à direita representa a matriz de adjacência para o grafo da esquerda.*

Definição 2.7. *Um grafo é um conjunto de pontos V_1, \dots, V_n de \mathbb{R}^2 , $n \in \mathbb{N}^*$, chamados vértices, e de curvas $\alpha_i : [0, 1] \rightarrow \mathbb{R}^2$, $1 \leq i \leq m \in \mathbb{N}^*$, tal que $\alpha_i(\{0, 1\})$ são vértices. Além disso, $\forall V_j, V_k$ com $j \neq k$ admitem curvas α_i tal que sua concatenação têm V_j e V_k como extremos. Se $\exists \alpha_1, \dots, \alpha_h$ distintos tal que $\alpha_1 * \dots * \alpha_h$ é fechada, temos um ciclo.*

Definição 2.8. *Uma árvore é um grafo sem ciclos.*

Proposição 2.3. *Seja G grafo com vértices $\{A, B, C, P_1, \dots, P_n\}$, 3 a 3 distintos com n mínimo¹ e curvas α_i . Se G tem comprimento mínimo, isto é,*

¹No sentido que qualquer outro grafo com o mesmo traço $\bigcup_{i=1}^m \alpha_i([0, 1])$ tem no mínimo $n \geq 0$ vértices, além de A, B, C .

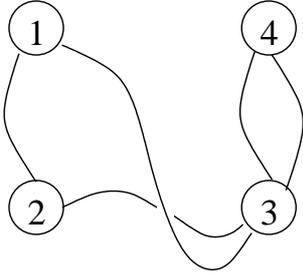


Figura 2.2: Grafo da matriz de adjacência.

	1	2	3	4
1	0	1	1	0
2	1	0	1	0
3	1	1	0	2
4	0	0	2	0

Figura 2.3: Tabela da matriz de adjacência.

$l(\alpha_1) + \dots + l(\alpha_n)$ é mínimo, então G é uma árvore. Além disso, $\{P_1, P_2, \dots\}$ é vazio ou unitário.

Demonstração. Caso $\{P_1, P_2, \dots\} = \emptyset$, a menos de renomear, é claro que $\exists \alpha_i : A \sim B$ e $\alpha_I : B \sim C$. Em particular, $\{\alpha_i\} = AB$ e $\{\alpha_I\} = BC$, pois G tem comprimento mínimo por hipótese.

Se $\{P_1, P_2, \dots\} \neq \emptyset$, a menos de renomear, $\exists \alpha_1 : A \sim P_1$. Como é grafo, todos os vértices estão conectados, e não se tem duas conexões diretas entre A, B, C pois recairíamos em $\{P_1, P_2, \dots\} = \emptyset$ por G ter comprimento mínimo. Logo, sem perda de generalidade B está ligado direta ou indiretamente a P_1 , mas G tem comprimento mínimo, então $\exists \alpha_2 : P_1 \sim B$.

Agora, por exemplo, se $\exists \alpha : C \sim A$, como $\{\alpha_1\}, \{\alpha_2\}$ e $\{\alpha\}$ são AP_1, P_1B e AC , a desigualdade triangular implica que G não é minimal $\not\sim$. Logo $\alpha_3 : C \sim P_1$. \square

Capítulo 3

Árvores de Steiner

3.1 Introdução

Uma árvore mínima para os pontos A_1, \dots, A_n em um plano é uma árvore que tem esses pontos como os seus vértices e tem comprimento mínimo. Contudo, podemos construir árvores ainda mais curtas que unem A_1, \dots, A_n , acrescentando vértices.

Exemplo 3.1. *Se A_1, A_2, A_3, A_4 são vértices de um retângulo, uma árvore mais curta do que o grafo mínimo desses vértices compõe-se de quatro arestas ligando A_1, A_2, A_3, A_4 a um vértice extra S no centro do retângulo.*



Figura 3.1: Um grafo retangular, e outro em “X” de menor comprimento.

Um vértice extra S , quando é adicionado a uma árvore para reduzir seu comprimento, é chamado **ponto Steiner**.

Observação 3.1. *Quando qualquer número de pontos Steiner pode ser acrescentado, a árvore mais curta possível é chamada **árvore mínima de Steiner**. Neste contexto, discutiremos o problema de como achar uma árvore*

mínima de Steiner.

Algumas Aplicações:

1. Redes mínimas para agricultura;
2. Engenharia de planejamento de rotas;
3. Circuitos elétricos.

3.2 Espécies de árvores

Árvores mínimas de Steiner são difíceis de encontrar porque uma árvore de comprimento localmente mínimo nem sempre representa um mínimo global.

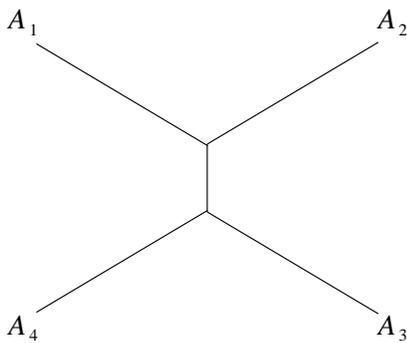


Figura 3.2: Mínimo local.

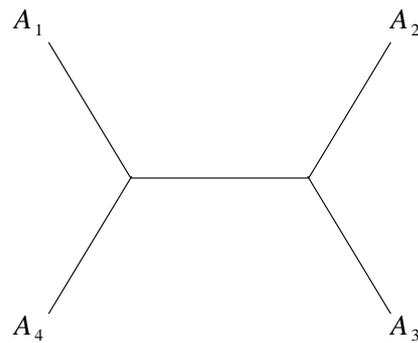


Figura 3.3: Mínimo global.

A Figura 3.2 não pode ser encurtada por pequenos deslocamentos dos pontos Steiner. Todavia, a Figura 3.3 é a mais curta (esta é a árvore mínima de Steiner para A_1, A_2, A_3, A_4), e provaremos isso na Seção 4.11.

A **topologia** de uma árvore é uma matriz de conexão, ou qualquer descrição equivalente, especificando quais pares dentre $A_1, A_2, \dots, A_n, S_1, S_2, \dots, S_s$ formam extremos de aresta. Assim, a topologia especifica uma conexão, mas não as posições de S_1, S_2, \dots, S_s . Cada topologia determina uma matriz de adjacência, e reciprocamente.

Como A_1 e A_2 têm arestas até um ponto comum de Steiner na Figura 3.2, mas não na Figura 3.3, estas duas figuras possuem topologias diferentes. A Figura 3.2 é a mais curta para qualquer outra da mesma topologia (ver

Figura 3.5). Uma tal árvore será chamada **árvore relativamente mínima** porque o mínimo é relativo a uma dada topologia.

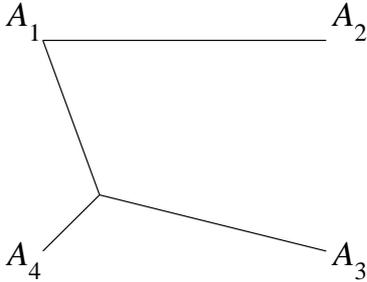


Figura 3.4: Mínimo local numa topologia.

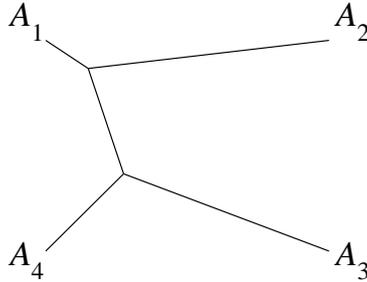


Figura 3.5: Variação da Figura 3.2.

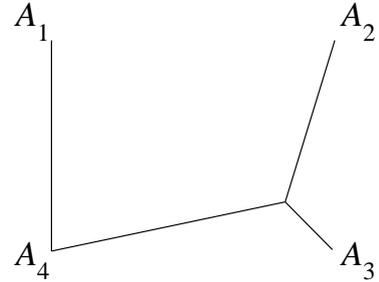


Figura 3.6: Mínimo local em outra topologia.

As Figuras 3.4 e 3.6 são árvores relativamente mínimas para duas topologias distintas. Se a topologia não prescreve nenhum ponto Steiner, então existe somente uma árvore com aquela topologia e conseqüentemente é um mínimo relativo.

Suponha que A_1 fosse muito mais próximo de A_4 , e A_2 de A_3 , na Figura 3.2. Neste caso, alcançamos um mínimo para a topologia da Figura 3.2 com os pontos Steiner arbitrariamente próximos. A árvore-limite será chamada **árvore degenerada**, e é mínima relativa para uma topologia diferente.

Pode-se considerar a Figura 3.5 como uma cópia ligeiramente distinta da Figura 3.4, embora sua topologia seja diferente. A Figura 3.5 é obtida através da Figura 3.4 despreendendo uma duplicação de A_1 (Isto é, 3.4 é uma árvore degenerada de 3.5).

Em geral, para desprender duplicação de um vértice V , é preciso desconectar duas ou mais arestas em V e conectá-las por um ponto Steiner V' , localizado próximo a V e ligado a V por uma aresta adicional. Depois de desprendê-lo por duplicação pode ocorrer, como no caso da Figura 3.5, que $V'V$ encurte a árvore. Se nenhum tal procedimento encurtar a árvore, então esta é chamada de **árvore de Steiner**. As Figuras 3.2 e 3.3 são árvores de Steiner. Uma árvore mínima de Steiner é sempre uma árvore de Steiner, e uma árvore de Steiner é sempre uma árvore relativamente mínima para esta topologia.

Uma árvore mínima de Steiner é uma árvore mínima para todos os vértices

$A_1, \dots, A_n, S_1, S_2, \dots, S_s$, mas o mesmo não é necessariamente verdadeiro para árvores relativamente mínimas (Figura 3.4) nem mesmo para árvores de Steiner. De fato, cada qual pode até representar mínimos locais por meio de árvores em que arestas se cruzam. Uma árvore mínima não pode ter arestas se cruzando devido ao Teorema 4.3 que veremos mais adiante.

Árvores mínimas relativas têm duas propriedades úteis. Primeiramente, a Seção 4.7 mostrará que existe no máximo uma árvore relativamente mínima para um dado conjunto A_1, A_2, \dots, A_n , e uma dada topologia. Depois, a Seção 4.8 descreve uma construção para uma árvore relativamente mínima, caso ela exista naquela topologia.

Assim, uma árvore mínima de Steiner pode ser obtida construindo uma árvore relativamente mínima para cada uma das topologias.

Infelizmente, há milhares de topologias a considerar, mesmo quando $n = 6$. A maior parte delas são obviamente descartáveis. A partir do Capítulo 4, teremos critérios que excluem imediatamente algumas topologias.

Capítulo 4

Propriedades básicas

4.1 Introdução

Este capítulo resume alguns fatos elementares sobre o problema da árvore mínima de Steiner. Muitos desses resultados são conhecidos, mas não se encontram reunidos em uma mesma literatura.

4.2 Arestas de um vértice

Teorema 4.1. *Seja $\alpha : [0, 1] \rightarrow \mathbb{R}^2$ retificável, $\alpha(0) = P, \alpha(1) = Q$, ambos do mesmo lado de uma reta r , com $\overleftrightarrow{PQ} \not\perp r$ e $\{\alpha\} \cap r \neq \emptyset$. Se $l(\alpha)$ é mínimo, então $\{\alpha\} \cap r = \{R\}$ tal que \widehat{PRQ} é máximo.*

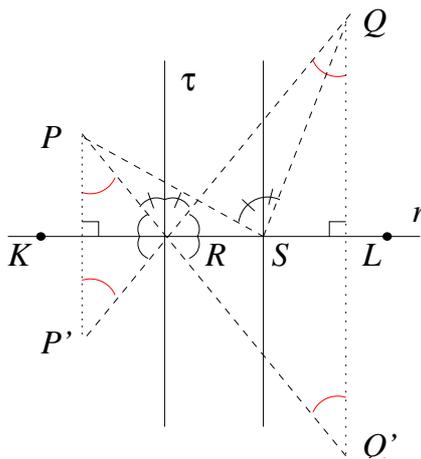


Figura 4.1: Ângulo \widehat{PRQ} máximo.

Demonstração. Seja $\overleftrightarrow{Q'Q} \perp r$ com $dist(Q', r) = dist(Q, r)$ e $\alpha(s) = R \in \{\alpha\} \cap r$. Sem perda de generalidade, r é o eixo real de $\mathbb{R}^2 = \mathbb{C}$, donde $\alpha|_{[0,s]} * \bar{\alpha}|_{[s,1]}$ têm mesmo $l(\alpha)$ mínimo \therefore é o segmento PQ' tal que $PQ' \cap r = \{R\}$. Considere a circunferência $\mathcal{C}(\mathcal{O}, \delta)$ determinada por P, Q, R , o arco capaz $\widehat{PQ} \ni R$ e $r \perp \tau \ni R$. Então τ é bissetriz de $P\hat{R}Q$ por argumentos geométricos como ângulos opostos pelo vértice (opv), casos de congruência e propriedades de triângulo retângulo. Logo $\widehat{PQ} \cap r = \{R\}$, e vejamos que $P\hat{R}Q$ é máximo. Suponha que $\exists S \neq R$ nas mesmas condições que R , isto é, uma outra curva $\beta : [0, 1] \rightarrow \mathbb{R}^2$ retificável, $\beta(0) = P$, $\beta(1) = Q$ com $\{\beta\} \cap r \neq \emptyset$ tal que $l(\beta)$ seja mínimo e que $\{\beta\} \cap r = \{S\}$. Pelo Teorema do Ângulo Externo, temos que $K\hat{R}Q > R\hat{S}Q \Rightarrow 90^\circ + P\hat{R}Q/2 > 90^\circ + P\hat{S}Q/2 \Rightarrow P\hat{R}Q > P\hat{S}Q$. Portanto segue-se a tese. \square

Teorema 4.2. *Seja $\alpha : [0, 1] \rightarrow \mathbb{R}^2$, $\alpha(0) = P$, $\alpha(1) = Q$, $\overleftrightarrow{PQ} \subset Ext \tau$, $\{\alpha\} \cap \tau \neq \emptyset$, onde τ é uma curva C^1 convexa, $\overleftrightarrow{PQ} \not\subset \tau$. Se $l(\alpha)$ é mínimo, então $\{\alpha\} \cap \tau = \{R\}$ tal que $P\hat{R}Q$ é máximo.*

Demonstração. Seja o conjunto de elipses com focos P e Q , e seja ξ a elipse que primeiro tangencia τ . Seja $r \perp \xi$ e $r \perp \tau$ em $\{R\} = \tau \cap \xi$. Então $\{\alpha\} \cap r \neq \emptyset$ e do resultado anterior temos a tese. \square

Lema 4.1. *Seja $f(x) = 1 - x + 2\sqrt{x^2 + 1}$, que representa o comprimento da árvore abaixo descrita, com vértices $(0, 0)$, $(1, \pm 1)$ e $(1 - x, 0)$. Então ela é mínima se, e somente se $\beta = 60^\circ$.*

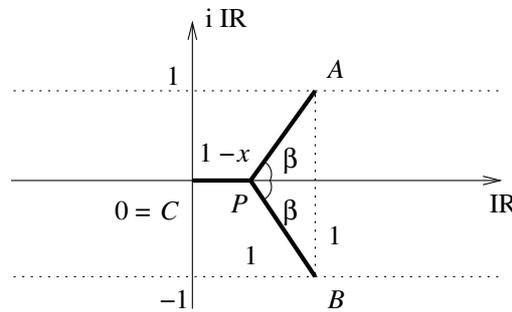


Figura 4.2: Lema dos ângulos de 120° .

Demonstração. Sendo $f(x) = 1 - x + 2 \cdot \sqrt{x^2 + 1}$, $f' = 0 \Leftrightarrow -1 + \frac{2x}{\sqrt{x^2 + 1}} = 0 \Leftrightarrow 0 < \sqrt{x^2 + 1} = 2x \Leftrightarrow x^2 + 1 = 4x^2 \Leftrightarrow x = \frac{1}{\sqrt{3}} \Leftrightarrow \beta = 60^\circ$. \square

Teorema 4.3. *Sejam A, B, C não colineares. Então $\exists!$ P tal que a árvore com vértices $\{A, B, C, P\}$ é mínima. Além disso, se $P \notin \{A, B, C\}$ então $A\hat{P}B = B\hat{P}C = C\hat{P}A = 120^\circ$.*

Demonstração. Da Proposição 2.3, caso o conjunto $\{P_1, P_2, \dots, P_n\} = \emptyset$, tome sem perda de generalidade $A\hat{C}B$ como o maior ângulo do $\triangle ABC$. Então é claro que $AC \cup CB$ realiza a árvore mínima, pois $|AB| \geq \max\{|AC|, |CB|\}$, donde $\exists! P = C$. Caso $\{P_1, P_2, \dots, P_n\} = \{P\} \neq \emptyset$ da Proposição 2.3, seja $c = |CP| > 0$ e $\mathcal{C}(C, c)$. Vejamos que $AB \cap \mathcal{C} = \emptyset$. Como $P \notin \{A, B, C\}$, então $P \in AB \perp \mathcal{C}$ ou sem perda de generalidade $A \in \text{int } \mathcal{C}$, donde $|\underbrace{AP}_a| + |\underbrace{PB}_b| \geq |AB|$ e como $|AC| < c$ teríamos $a + b + c > |AB| + |AC| \not\leq 1$. Também não pode $P \in AB \perp \mathcal{C}$ pelo Lema 4.1, pois sem perda de generalidade $|\text{proj}_{i\mathbb{R}} AP| = |\text{proj}_{i\mathbb{R}} PB| = 1$.

Assim, $AB \cap \mathcal{C} = \emptyset$, donde o Teorema 4.2 implica que $A\hat{P}B$ é máximo com \overleftrightarrow{CP} bissetriz. Analogamente, \overleftrightarrow{AP} bissetriz de $B\hat{P}C$ e \overleftrightarrow{BP} bissetriz de $A\hat{P}C$, donde todos valem 120° por opv. \square

Corolário 4.1. *Se $P = C$, $A\hat{C}B$ maior ângulo do $\triangle ABC$, então $A\hat{C}B \geq 120^\circ$.*

Demonstração. Se $A\hat{C}B < 120^\circ$, do Lema 4.1 mais o Teorema 4.3 teríamos uma contradição. \square

4.3 Números de pontos Steiner

Toda árvore tem um vértice a mais do que o número de arestas. Uma árvore com terminais A_1, \dots, A_n e s pontos Steiner deve ter $n + s - 1$ arestas. Como cada aresta tem dois extremos, obtemos um total de $2 \cdot (n + s - 1)$ arestas incidentes ao somar percorrendo todos os vértices.

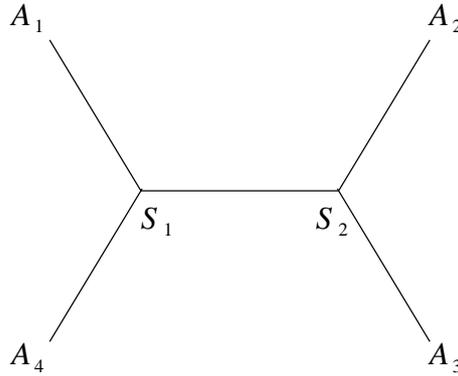


Figura 4.3: Figura 3.3 com pontos Steiner.

Se n_k dos vértices A_i tem k arestas incidentes ($n_4 = n_5 = \dots = 0$ para árvore de Steiner), a soma em questão é

$$\begin{aligned} 3s + \sum_{k=1}^3 kn_k &= 3s + n_1 + 2n_2 + 3n_3 = \\ &= 2 \cdot (n + s - 1) = 2s - 2 + 2 \cdot (n_1 + n_2 + n_3). \end{aligned}$$

Então

$$s = n_1 - 2 - n_3 \leq n_1 - 2 \leq n - 2.$$

Caso $s = n - 2$, então $s = n_1 - 2$, e portanto $n = n_1$. Agora, $s = n_1 - 2 = n_1 - 2 - n_3 \Rightarrow n_3 = 0$ e $n_1 - 2 = n - 2 \Rightarrow n_2 = 0$. Logo, vale a igualdade em $s \leq n - 2 \Leftrightarrow$ de cada A_i parte só uma aresta.

4.4 Teorema de Maxwell

Considere vetores unitários externos a uma árvore relativamente mínima, nas direções das arestas incidentes em A_1, \dots, A_n . Seja F_i a soma destes vetores unitários em A_i . O comprimento L da árvore tem a fórmula simples

$$L = \sum_{i=1}^n A_i \cdot F_i. \quad (4.1)$$

Note que qualquer origem conveniente pode ser usada para determinar as posições dos vetores A_i .

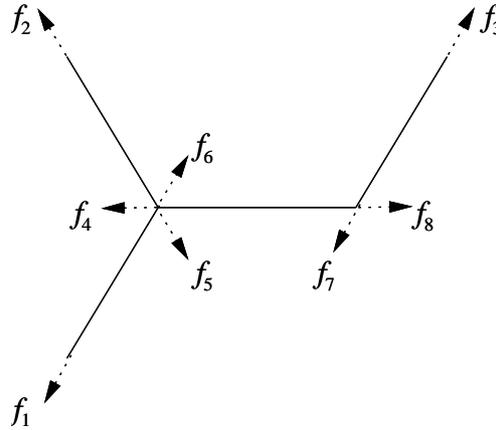


Figura 4.4: Representação do Teorema de Maxwell.

Para provar (4.1) considere um vetor unitário f_k externo a cada extremo P_k de cada aresta. Como temos $n + s - 1$ arestas, há $2 \cdot (n + s - 1)$ vetores assim. Agora considere a soma $\Sigma = \sum P_k \cdot f_k$, em que percorrem-se todos os $2 \cdot (n + s - 1)$ extremos.

Os termos de Σ podem ser agrupados de dois modos. Primeiro combine os termos para os quais P_k é o mesmo ponto. Cada ponto Steiner aparece em três dos termos de Σ . A contribuição de um ponto Steiner S para Σ é zero. A contribuição para um dado ponto A_i em Σ é $A_i \cdot F_i$. Então Σ é o lado direito de (4.1). O segundo agrupamento combina os dois termos associados com os dois extremos de cada aresta. Se P_1P_2 é uma aresta com comprimento L_{12} , então $P_2 = P_1 + L_{12} \cdot u$, onde u é um vetor unitário. Os termos para tal aresta contribuem com $P_1 \cdot (-u) + P_2 \cdot (u) = L_{12}$. Então Σ é também o lado esquerdo de (4.1).

4.5 Árvore plena de Steiner

A topologia com $s = n - 2$ será chamada **topologia plena**. A árvore relativamente mínima será chamada **árvore plena de Steiner**. De acordo com o número de pontos Steiner (visto na Seção 4.3), uma topologia é plena se e somente se parte somente uma aresta de cada A_i . Se $n \geq 3$, a única aresta em A_i deve levar a um ponto Steiner. Uma árvore plena de Steiner é de fato uma árvore de Steiner porque não há como desprender uma duplicação

de $A_i, \forall i$.

Movendo-se ao longo de qualquer caminho por uma árvore plena de Steiner, modifica-se a direção ao passar por um ponto Steiner, mas somente de $\pm 60^\circ$. Então cada F_k é um dos seis números complexos $\pm 1, \pm \omega, \pm \omega^2$, onde $\omega = \exp(2\pi i/3)$, considerando um sistema de coordenadas adequado. Cada árvore de Steiner que não é plena pode ser decomposta em uma união de árvores plenas de Steiner. Substitua cada vértice A_i que tenha $k \geq 2$ arestas por k novos vértices A_{i1}, \dots, A_{ik} . Deste modo, a árvore original é dissecada em várias árvores plenas menores que chamamos de **componentes plenas** da árvore original. Reciprocamente, quando a topologia especificada não é plena, obtêm-se as topologias das componentes plenas. Pode-se então construir as componentes plenas separadamente, e depois juntá-las para obter a árvore de Steiner procurada.

4.6 Envelope convexo

Definição 4.1. *Um conjunto é dito **simplexo** se for o menor convexo que contenha todos pontos.*

Teorema 4.4. *Seja $S = \{A_1, \dots, A_n\}$ pontos do plano tal que três deles não são colineares. Então seu **simplexo** τ é um polígono cujos vértices estão em S .*

Demonstração. Fixe um sistema de coordenadas ortogonais Oxy no plano, donde $A_i = (x_i, y_i)$. Sem perda de generalidade $x_1 := \min \{x_i : 1 \leq i \leq n\} = 0$ e $y_1 := \max \{y_j : x_j = 0\} = 0$.

Dado $z = (x, y) \in \mathbb{C}$, definimos $Arg : \mathbb{C}^* \rightarrow (-\pi, \pi]$ por $Arg(x, y) = \arctan(y/x)$, se $x > 0$, $Arg(x, y) = \text{sgn}(y)\pi/2 - \arctan(x/y)$, se $y \neq 0$, e $Arg(\mathbb{R}_+^*) = \{0\}$.

Tome $\alpha = \min \{Arg(A_i) ; 2 \leq i \leq n\} \in [-\pi/2, \pi/2)$. Sem perda de generalidade, temos

$$Arg(A_2) = \alpha \text{ e } |A_2| = \max \{|A_i| : Arg(A_i) = \alpha\}.$$

Tome agora um novo sistema $Ox'y'$ com a mesma orientação de Oxy e $Oy' = A_2A_1$. Nele, seja $\beta = \min \{Arg(A_i) ; 3 \leq i \leq n\} \in (-\pi/2, \pi/2]$; sem perda de generalidade $Arg(A_3) = \beta$ e $|A_3| = \max \{|A_i| ; Arg(A_i) = \beta\}$.

Repita o processo com $Ox''y''$ com a mesma orientação, $Oy'' = \overrightarrow{A_3A_2}$ e assim por diante, até reencontrar A_1 . Teremos então um polígono convexo \mathcal{P} com vértices A_1, A_2, \dots, A_m , $3 \leq m \leq n$. Como $\tau \supset \partial\mathcal{P}$ e τ é convexo, então $\tau \supset \mathcal{P}$ donde são iguais. \square

4.7 Unicidade das árvores mínimas relativas

Sejam $A_1, \dots, A_n, S_1, \dots, S_s$ terminais e pontos Steiner de uma certa topologia. Defina $V_k = A_k$ e $V_{n+h} = S_h$, $1 \leq k \leq n$, $1 \leq h \leq s$. Como existe apenas um número finito de matrizes de adjacência, então o número de topologias é finito. Seja $(a_{ij})_{(n+s) \times (n+s)}$ uma delas $\therefore a_{ij} = 1 \Leftrightarrow V_i V_j$ é aresta.

Assim, $L = \sum_{i < j} a_{ij} \cdot |V_i - V_j|$ varia com s pontos. Em outras palavras, é função contínua de $(X_h, Y_h) = S_h$, derivável exceto quando a topologia se degenera. Sejam T, T' e T'' árvores nesta topologia, com comprimento L, L' e L'' , tais que $V_i'' = pV_i + qV_i'$ com $V_i = V_i', 1 \leq i \leq n$ e $p, q \geq 0$ com $p + q = 1$.

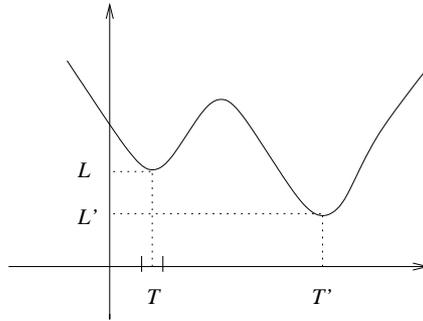


Figura 4.5: Mínimos locais.

Assim,

$$\begin{aligned} L'' &= \sum a_{ij} \cdot |p(V_i - V_j) + q(V_i' - V_j')| \\ &\leq \sum a_{ij} \cdot (p|V_i - V_j| + q|V_i' - V_j'|) \\ L'' &\leq pL + qL'. \end{aligned}$$

Se T e T' fossem mínimos locais com $L' < L$, então T'' com $q \cong 0$ estaria arbitrariamente próxima de T e $L'' \leq pL + qL' < pL + qL = L$. Portanto, $L'' < L \Rightarrow T$ não é mínimo local $\not\Leftarrow$.

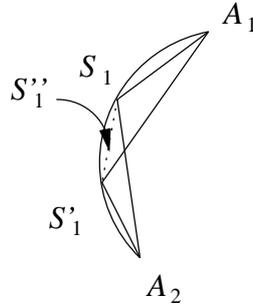


Figura 4.6: Arcos de 120° .

É claro que para $s = 0$ os mínimos locais são assumidos uma única vez. Suponha que isto é verdade para $s - 1$ pontos, $s \geq 1$, e com s pontos haja $T \neq T'$ assumindo o mínimo local $L = L'$. Então T'' também assume este mesmo mínimo local L . Sem perda de generalidade A_1S_1 e S_1A_2 são arestas donde $S''_1 = pS'_1 + qS_1$ move-se no segmento S'_1S_1 (se ocorrer $S_1 \neq S'_1$), uma contradição pois $A_1\hat{S}''_1A_2 = 120^\circ$. Então $S_1 = S'_1 = S''_1$, todos fixos (vide Figura 4.6).

Como para $s - 1$ pontos temos unicidade, então também ocorre unicidade para s pontos.

4.8 Caso geral (mais de três pontos)

Suponha que os terminais A_1, A_2, \dots, A_n são dados ao mesmo tempo com s pontos Steiner e uma topologia. A árvore relativamente mínima é construída por uma indução sobre s na qual a construção da Seção 4.2 é usada várias vezes para localizar pontos Steiner. Como é trivial construir uma árvore com $s = 0$, podemos considerar o caso $s \geq 1$ e mostrar que pode ser reduzido a uma ou mais árvores, cada qual com menos pontos Steiner.

O primeiro passo é encontrar um ponto Steiner S com a topologia unindo dois terminais A_i, A_j . Tal S existe, pois considerando qualquer componente plena (Seção 4.5) com $s' \geq 1$ pontos Steiner e n' terminais, de cada um parte

só uma aresta, que conduz a um ponto Steiner. Já que $n' = s' + 2$, pelo menos um ponto Steiner S deve conectar-se diretamente a dois terminais (existem dois ou mais tais pontos Steiner se $s' \geq 2$).

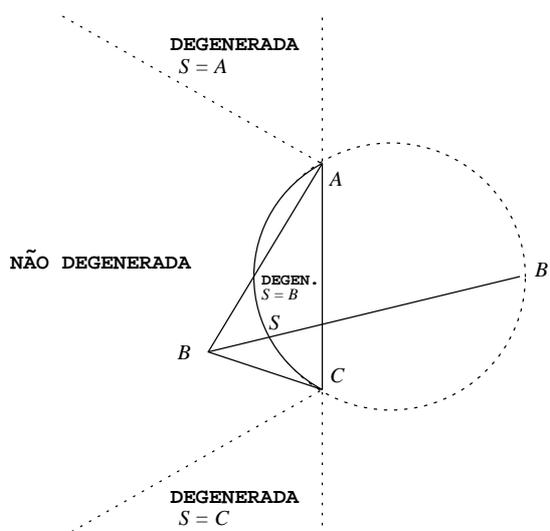
Sejam A_i, A_j e B três pontos conectados a S . Se B é um ponto A_k , a construção da Seção 4.5 localiza S ou mostra que não existe solução naquela topologia. Suponha que S tenha sido localizado. Se S e as arestas A_iS, A_jS, A_kS são removidas a partir da árvore relativamente mínima, teremos três novas árvores relativamente mínimas, cada qual contendo A_i, A_j, A_k . Cada uma tem menos que s pontos Steiner e pode ser construída por processo indutivo. Se B é ponto Steiner, sua localização é desconhecida. Todavia, o modo da construção da Figura 4.7 pode proceder parcialmente. O ponto B' tem somente duas localizações (em lados opostos de A_iA_j). Tentam-se ambas possibilidades, a não ser que propriedades como as que veremos entre as Seções 4.9 e 4.13 eliminem alguma delas.

Se uma árvore relativamente mínima T existe, esta pode ser construída como segue. Primeiro resolva o problema mais simples em que os vértices A_i, A_j e S são removidos e B' é um terminal. Conexões com pontos exceto S permanecem as mesmas, mas requer-se uma aresta adicional BB' . As novas árvores mínimas têm o mesmo comprimento que a anterior porque $|B'B| = |A_iS| + |A_jS| + |BS|$. De fato, pela Figura 4.7, tomando $|AS| = a, |CS| = b, |BS| = c$ e $|SB'| = d$, temos que $b^2 + d^2 - 2bd \cdot \cos 60^\circ = 1 = a^2 + d^2 - 2ad \cdot \cos 60^\circ \Rightarrow a^2 - b^2 = (a - b)d \Rightarrow (a + b)(a - b) = (a - b)d$. Se $a \neq b \Rightarrow d = a + b$. Agora, se $a = b \Rightarrow d = a + b$ por continuidade. Assim segue-se a tese. Quando uma nova árvore é encontrada, $B'B$ pode ser trocado por A_iS, A_jS e BS para obter a árvore desejada.

Nem sempre podemos excluir de imediato uma de duas escolhas incorretas para B' . Para um dado B' , o cálculo pode colocar B fora da região "não-degenerada" da Figura 4.7, ou pode concluir que a árvore relativamente mínima para a nova topologia (com B' substituindo S, A_i, A_j) não existe. Tenta-se então a outra escolha de B' . Se esta ainda falhar, a topologia dada não tem árvore relativamente mínima.

A indução substitui o problema com s pontos Steiner por um com $s - 1$. Este é então substituído por $s - 2$ pontos, e assim por diante.

A árvore relativamente mínima, em cada um desses problemas mais sim-

Figura 4.7: Construção de S .

ples, tem o mesmo comprimento que a árvore relativamente mínima procurada. Dessa maneira, quando o problema for simplificado até que a solução seja evidente, o comprimento é conhecido de imediato. Por exemplo, se a árvore original é uma árvore plena, então cada árvore mais simples também o é. No último passo, recai-se na Figura 4.7, cujo comprimento é $|B'B|$.

Se existem s pontos Steiner, cada um podendo estar em dois lados distintos, devem-se a princípio calcular 2^s árvores para encontrar a relativamente mínima. Em cálculos manuais a situação não é tão ruim pois geralmente podemos intuir uma escolha correta para cada B' . Se a intuição for de fato correta e uma árvore mínima é encontrada, a unicidade da Seção 4.7 mostra que não precisa tentar a outra possibilidade. Quando o cálculo é feito por máquina, é bem difícil de se intuir. Alguns dos resultados adiante poderão ajudar neste procedimento.

4.9 Propriedade da lua

Seja AB uma aresta de uma árvore mínima. Seja $L(A, B)$ a região consistindo de todos os pontos X que satisfazem $|XA| < |AB|$ e $|XB| < |AB|$.

$L(A, B)$ é a lua, interseção dos círculos de raio $|AB|$ centrados em A e B . Nenhum outro vértice da árvore pode estar em $L(A, B)$. De fato, se X_0 fosse

tal vértice, a árvore conteria um caminho de X_0 a A que não passa por B , ou um caminho de X_0 a B não passando por A . No último caso, por exemplo, pode-se encurtar a árvore removendo AB e acrescentando X_0A .

Em uma árvore mínima de Steiner, $L(A, B)$ também não deve conter ponto interior X_0 de outra aresta da árvore. O mesmo argumento aplica-se novamente com X_0 adicionando um ponto Steiner de uma outra topologia, cuja árvore mínima relativa seria ainda mais curta, o que é absurdo.

4.10 Propriedade da cunha

Teorema 4.5. *Seja W uma região aberta qualquer em forma de cunha que tem ângulo de 120° ou mais e que não contém nenhum dos A_1, \dots, A_n . Então W não contém pontos Steiner de uma árvore relativamente mínima.*

Demonstração. A prova é por contradição. Suponha que W contém um ponto Steiner S . Se existem vários pontos Steiner em W , podemos introduzir um sistema de coordenadas cartesianas (x, y) com eixo x positivo ao longo do ângulo bissetor de W e escolher S como o ponto Steiner com a maior coordenada x . Das três arestas de S , uma parte numa direção entre $\pm 60^\circ$ com o eixo x positivo. Esta não pode deixar W e assim não pode terminar em nenhum dos A_1, \dots, A_n .

Além disso, o ponto final tem uma coordenada x maior do que S . \square

Corolário 4.2 (Envelope Convexo). *Em uma árvore relativamente mínima todos os pontos Steiner encontram-se na casca convexa de A_1, \dots, A_n .*

A união de todas cunhas W não contém pontos Steiner. Assim, todos os pontos Steiner encontram-se no complemento K desta união. Em alguns casos, K é um conjunto bem pequeno.

Exemplo 4.1. *Se A_1, A_2, A_3 são três terminais de um triângulo equilátero, o correspondente K contém um ponto simples no centro do triângulo. Mais geralmente, K consiste de uma ou mais regiões fechadas limitadas por arcos de círculos, e é sempre um subconjunto da casca convexa de A_1, \dots, A_n ,*

porque cada aresta da casca convexa corresponde a uma cunha W de 180° , que não contém nenhum dos A_1, \dots, A_n .

4.11 Distância entre pontos Steiner

Lema 4.2. *Seja $P_1P_2P_3P_4$ um retângulo, S_1 o ponto Steiner mais próximo de P_1 e $L_o = |P_1S_1|$. Então os lados do retângulo medem $|P_1P_2| = |P_3P_4| = L_o\sqrt{3}$ e $|P_2P_3| = |P_4P_1| = L_o + |S_1S_2|$.*

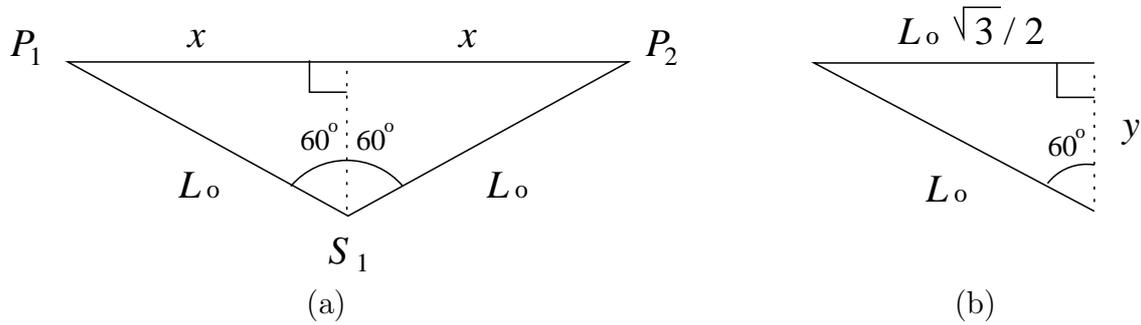


Figura 4.8: (a) Triângulo 1; (b) triângulo 2.

Demonstração. Pela Figura 4.8(a) temos $\sin 60^\circ = \frac{x}{L_o} \Rightarrow \frac{\sqrt{3}}{2} = \frac{x}{L_o} \Rightarrow x = \frac{\sqrt{3}}{2}L_o$. Como $\ell = |P_1P_2| = 2x$, então $\ell = 2 \cdot \frac{\sqrt{3}}{2}L_o \Rightarrow \ell = \sqrt{3}L_o$. Agora, na Figura 4.8(b) temos $\cos 60^\circ = \frac{y}{L_o} \Rightarrow y = \frac{L_o}{2}$. Logo, segue-se a tese. \square

Lema 4.3. *Se $|P_1P_2| > |P_1P_4|$, na Figura 3.2 então a árvore mais curta corresponde à da Figura 3.3.*

Demonstração.

Temos $L_o \frac{\sqrt{3}}{2} = \frac{\ell}{2} \Rightarrow \ell = L_o\sqrt{3}$. Observando o $\triangle S_2P_4M$, $\operatorname{tg} 30^\circ = \frac{x}{\ell/2} \Rightarrow \frac{\sqrt{3}}{3} = \frac{2x}{\ell} \Rightarrow x = \frac{\ell\sqrt{3}}{6}$. Então $z = 1 - \frac{2\ell\sqrt{3}}{6} \Rightarrow z = \frac{3 - \ell\sqrt{3}}{3}$ e $L_o = \frac{\ell\sqrt{3}}{3}$.

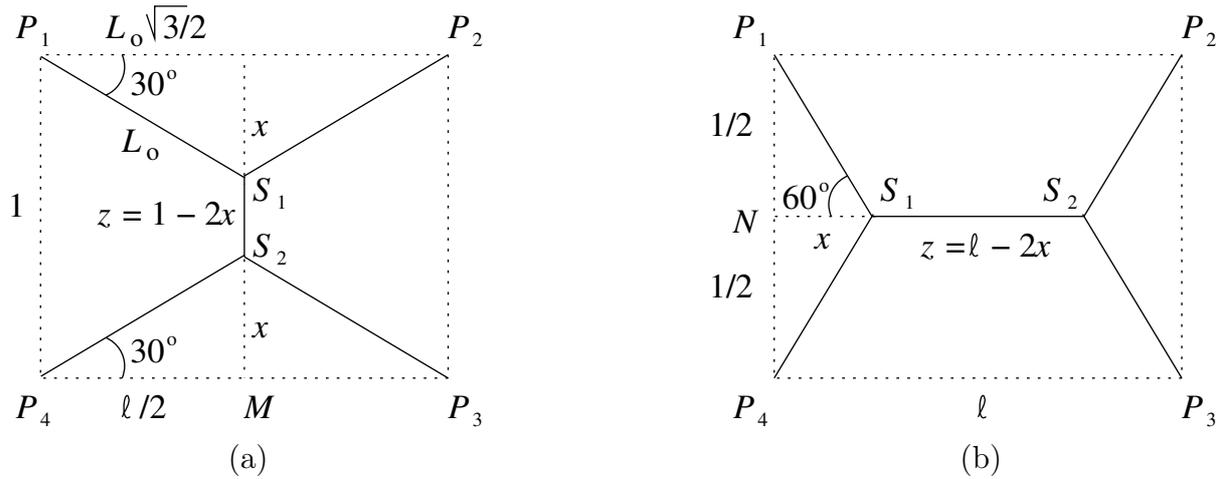


Figura 4.9: (a) Figura 3.2 (contas); (b) Figura 3.3 (contas).

Logo, o comprimento desta árvore (Figura 4.9(a)) é $4L_o + z = \frac{4\ell\sqrt{3}}{3} + \frac{3 - \ell\sqrt{3}}{3} = 1 + \ell\sqrt{3}$.

Observando o $\triangle NS_1P_1$ da Figura 4.9(b) obtemos $\text{tg}60^\circ = \frac{1/2}{x} \Rightarrow \sqrt{3} = \frac{1}{2x} \Rightarrow x = \frac{\sqrt{3}}{6}$ e $\text{sen}60^\circ = \frac{1/2}{L_o} \Rightarrow \frac{\sqrt{3}}{2} = \frac{1}{2L_o} \Rightarrow L_o = \frac{\sqrt{3}}{3}$.

Então, $z = \ell - 2x = \ell - \frac{\sqrt{3}}{3} = \frac{3\ell - \sqrt{3}}{3}$. Assim, $4L_o + z = \frac{4\sqrt{3}}{3} + \frac{3\ell - \sqrt{3}}{3} = \sqrt{3} + \ell$ é o comprimento desta árvore (Figura 4.9(b)). Logo, $1 + \ell\sqrt{3} > \sqrt{3} + \ell \Leftrightarrow (\sqrt{3} - 1) \cdot \ell > \sqrt{3} - 1 \Leftrightarrow \ell > 1 \therefore$ a Figura 3.3 tem menor comprimento. \square

Teorema 4.6. *Suponha S_1 e S_2 pontos Steiner conectados por uma aresta em uma árvore mínima de Steiner. De S_1 e S_2 partem duas outras arestas. Seja L_o o comprimento mais curto entre essas quatro. Então $|S_1S_2| \geq (\sqrt{3} - 1)L_o$.*

Demonstração. Sejam P_1 e P_2 dois pontos os quais são extremos de duas arestas de S_1 , ambas medindo L_o . Do mesmo modo, sejam P_3 e P_4 dois pontos que são extremos de S_2 em suas duas arestas de comprimento L_o . Os

pontos P_1, P_2, P_3, P_4 são vértices de um retângulo, e a árvore que conecta P_1, P_2, P_3, P_4 está representada na Figura 3.2 ou 3.3, nas quais os A_i são renomeados P_i .

Pelos Lemas 4.2 e 4.3 concluímos a tese, pois vale $z = \ell - L_o \geq 1 - L_o = L_o\sqrt{3} - L_o = (\sqrt{3} - 1) L_o$. \square

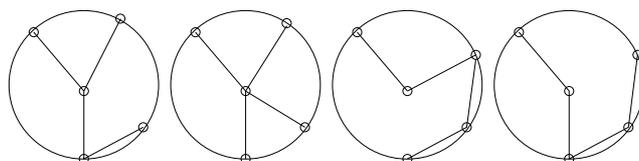
A Figura 3.3 é a árvore mínima de Steiner pois existem apenas duas árvores plenas de Steiner e assim, dois tipos de matriz de adjacência.

Veja uma delas a seguir:

	P_1	P_2	P_3	P_4	S_1	S_2
P_1	0	0	0	0	1	0
P_2	0	0	0	0	0	1
P_3	0	0	0	0	0	1
P_4	0	0	0	0	1	0
S_1	1	0	0	1	0	1
S_2	0	1	1	0	1	0

4.12 Arestas $A_i A_j$

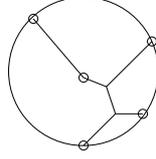
Uma árvore de menor comprimento para as topologias com $s = 0$ é dita **minimal spanning tree**, ou seja, **árvore mínima**. Note que ela pode não ser única, e pode diferir das "árvores caixeiro-viajante".



Quanto à árvore mínima, podemos obtê-la pelo **algoritmo de Prim**.

Algoritmo de Prim : Escolhe-se um terminal qualquer (sem perda de generalidade A_1), toma-se o que menos dista de A_1 (sem perda de generalidade A_2), e então o que menos dista de $\{A_1, A_2\}$ (sem perda de generalidade A_3) etc. Em suma, o que menos dista de $\{A_1, A_2, \dots, A_k\}$, e vão se compondo as arestas em cada passo.

Agora, dados $A_1, A_2, \dots, A_n, S_1, S_2, \dots, S_s$, se $A_i A_j$ é uma aresta da árvore mínima de Steiner, então $A_i A_j$ é também aresta da "minimal spanning tree".



Demonstração. Seja \mathcal{C}_k o conjunto de vértices que se conectam a A_k sem passar por $A_{(i+j)-k}$, $k = i, j$. Qualquer vértice está em algum \mathcal{C}_k . Temos que $\forall P_k \in \mathcal{C}_k \Rightarrow |P_i P_j| \geq |A_i A_j|$, do contrário, a árvore ficaria mais curta com $P_i P_j$ substituindo $A_i A_j$. A menos de reindexar, $i = 1$.

Agora, usando o algoritmo de Prim teremos que j é o primeiro índice para o qual $A_j \notin \mathcal{C}_1$, donde $A_1 A_j$ pertence à "minimal spanning tree". \square

4.13 Propriedade do diamante

Numa aresta AB , o diamante definido por $\mathcal{D}(A, B)$ é o conjunto dos pontos P tal que $\max\{P\hat{A}B, P\hat{B}A\} < 30^\circ \therefore \mathcal{D}(A, B)$ é interior do losango com ângulos internos 60° e 120° , tendo AB como diagonal maior. Considerando $\mathcal{D}(V_i, V_j)$ e $\mathcal{D}(V_k, V_h)$ numa árvore mínima de Steiner, temos que $\{i, j\} \neq \{k, h\} \Rightarrow \mathcal{D}_{ij} \cap \mathcal{D}_{kh} = \emptyset$, o que provaremos mais adiante.

Seja $R \supset \{A_1, \dots, A_n, S_1, \dots, S_s\}$, $U = \bigcup_{P, Q \in R} \mathcal{D}(P, Q)$. Se N é o número de vértices, $N = n + s$, então há $N - 1$ diamantes disjuntos. Também, $s \leq n - 2 \Rightarrow 2s \leq N - 2 \Rightarrow 2(s + 1) \leq N$ e $s \leq n - 2 \Rightarrow N \leq 2(n - 1)$.

Cada diamante tem a seguinte área

$$\text{Área}(\mathcal{D}) = L_i \cdot \frac{L_i}{2\sqrt{3}} = \frac{L_i^2}{2\sqrt{3}} \text{ donde } \sum_{i=1}^{N-1} \frac{L_i^2}{2\sqrt{3}} \leq \text{área}(U), \text{ então}$$

$$\sum_{i=1}^{N-1} L_i \cdot 1 \leq \left\{ \sum_{i=1}^{N-1} L_i^2 \right\}^{1/2} \cdot \left\{ \sum_{i=1}^{N-1} 1 \right\}^{1/2} \leq \left\{ 2\sqrt{3} \text{ área}(U) \right\}^{1/2} \cdot \sqrt{N-1} =$$

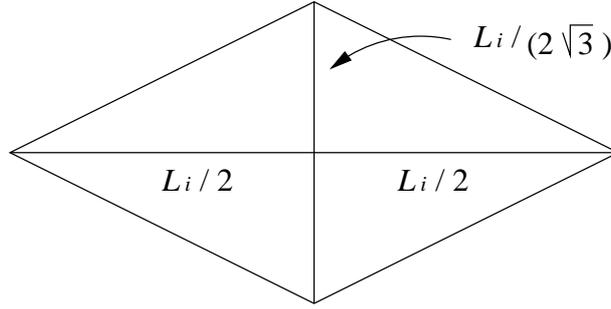


Figura 4.10: Propriedade do Diamante.

$$= \sqrt{2} \sqrt[4]{3} \sqrt{N-1} \cdot \sqrt{\text{área}(U)}.$$

Exemplo 4.2. Se R for uma circunferência unitária, U é o círculo de raio $2/\sqrt{3}$. Então o comprimento da árvore mínima de Steiner é $\leq 3,81\sqrt{N-1}$. Mas como depende de N , só serve para valores $3,81\sqrt{N-1} \leq 2\pi \Rightarrow N \leq 3,7$. Em outro artigo de Gilbert (vide [G]), o valor esperado é menor ou igual a $1,25\sqrt{N-1} \Rightarrow N \leq 26,3$ (distribuição uniforme).

Pelo nossos propósitos, não iremos nos aprofundar nesses cálculos, uma vez que têm abordagem estatística e muito particular quanto à figura geométrica (círculo, quadrado, etc).

A propriedade do diamante é a melhor no sentido em que os ângulos de 60° não podem ser aumentados. De fato, assumindo a propriedade, que logo será demonstrada, considere uma árvore mínima, donde os diamantes não se intersectam (vide Figura 4.11). Ao sofrer pequena alteração nos vértices V_3 e V_4 , se passarmos a ter intersecção dos diamantes, certamente haverá uma árvore mais curta em outra topologia.

Para uma prova mais eficaz, dividiremos a mesma em 4 casos.

Explicação da figura acima: Quando os bordos dos diamantes se tocam, há duas árvores com mínimo absoluto. Deslocar V_4 de modo que os diamantes se interceptem, como são dois abertos, continua árvore de Steiner, mas passa a não ser mínima.

Sejam AB e XY duas arestas de uma árvore mínima de Steiner, C_A e C_B círculos abertos de raio $|AB|$ centrados em A e B , respectivamente. Assim, a lua $L(A, B) = C_A \cap C_B$ não intercepta XY em nenhum ponto.

Caso 1. $\{X, Y\} \cap C_A \neq \emptyset \neq \{X, Y\} \cap C_B$. Como $L(A, B) \cap \{X, Y\} = \emptyset$,

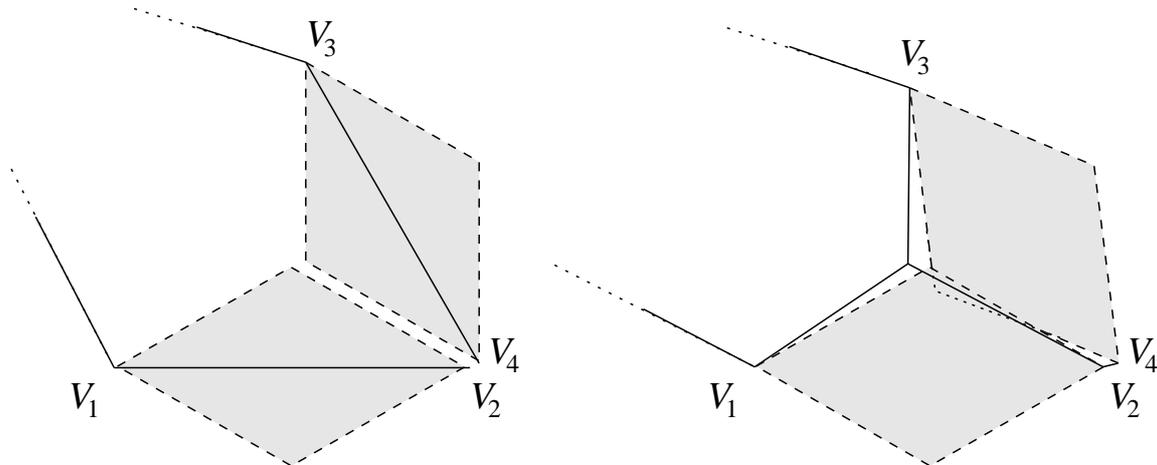


Figura 4.11: Leve perturbação de uma árvore, alterando sua topologia.

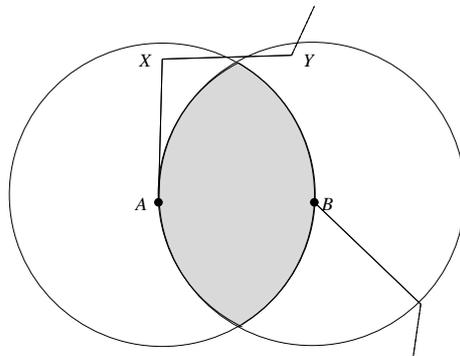


Figura 4.12: Ilustração do Caso 1.

então sem perda de generalidade $X \in C_A$ e $Y \in C_B$. Removendo AB , teremos duas árvores disjuntas, e uma delas conterá tanto XY quanto algum dos vértices A ou B , sendo que $\max\{|XA|, |YB|\} < |AB|$.

Se for A , adicione a aresta BY , e se for B , AX . Voltamos a ter uma só árvore, mais curta que a mínima de Steiner $\not\prec$.

Então, $\{X, Y\} \cap C_A = \emptyset$ ou $\{X, Y\} \cap C_B = \emptyset$, e sem perda de generalidade supomos que vale este último. Então $\{X, Y\} \subset C_B^c$.

Do mesmo modo, $\{A, B\} \cap C_X = \emptyset$ ou $\{A, B\} \cap C_Y = \emptyset$, de forma que podemos supor $|XY| \leq |AB| = 1$.

Caso 2. $\{X, Y\} \subset C_A^c \cap C_B^c$. Neste caso, $\min\{|XP|, |YP|\} \geq \sqrt{3}/2 - \frac{1}{2\sqrt{3}} = \frac{3\sqrt{3} - \sqrt{3}}{6} = \frac{1}{\sqrt{3}}$.

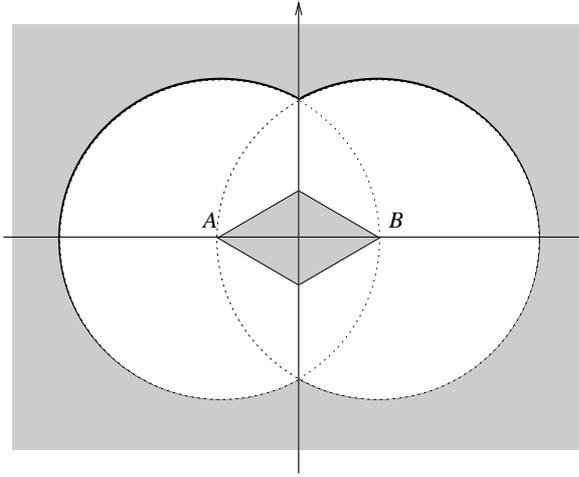


Figura 4.13: Ilustração do Caso 2.

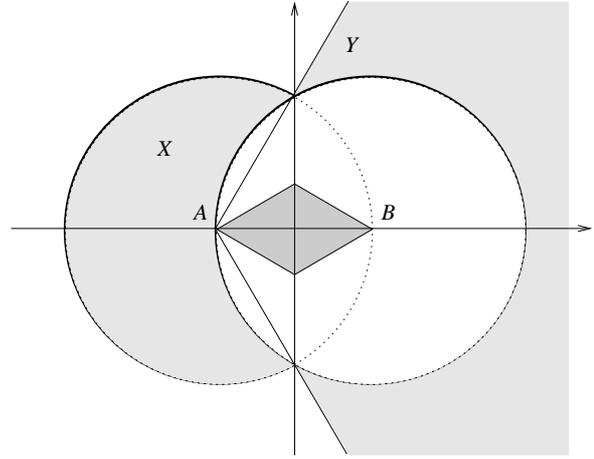


Figura 4.14: Ilustração do Caso 3.

Se $\exists P \in \mathcal{D}(A, B) \cap \mathcal{D}(X, Y)$, então $X\hat{P}Y > 120^\circ$, donde $|XY|^2 = |XP|^2 + |YP|^2 - 2|XP||YP| \cdot \cos(X\hat{P}Y) \geq \frac{1}{3} + \frac{1}{3} - \frac{2}{3} \cdot \cos(X\hat{P}Y) > \frac{2}{3} + \frac{1}{3} = 1 \checkmark$ pois $|XY| \leq 1$.

Então, como $\{X, Y\} \subset C_B^c$, $|XY| \leq |AB| = 1$ e não ocorre o Caso 2, devemos também ter X ou Y em C_A (sem perda de generalidade $X \in C_A \therefore X \in C_A \cap C_B^c$). Podemos ainda ter $Y \in C_A$ ou $Y \in C_A^c$.

Caso 3. $Y\hat{A}B \leq 60^\circ$. Isso implica $Y \in C_A^c$. A propriedade da lua implica $XY \cap L(A, B) = \emptyset$, e também temos $|XY| \leq |AB| = 1$. Um ponto $P \in \mathcal{D}(A, B) \cap \mathcal{D}(X, Y)$ é tal que $P \in L(A, B)$.

Um vértice de 120° do $\mathcal{D}(X, Y)$ terá aproximação máxima de B quando XY for corda de C_B e $|XY| = 1$, donde $|PB| > \frac{1}{\sqrt{3}}$. Por outro lado, quando Y se aproxima de $\partial C_A \cap \partial C_B$, temos que os pontos de $\mathcal{D}(X, Y)$ estão a uma altura maior ou igual a $\frac{\sqrt{3}-1}{2}$ de AB , devido ao Teorema de Tales (vide Figura 4.15).

Como $\frac{1}{2\sqrt{3}} < \frac{\sqrt{3}-1}{2}$, segue-se que não ocorre o Caso 3.

Caso 4. $Y\hat{A}B > 60^\circ$. É fácil ver que duas curvas convexas α, β tal que $\{\alpha\} \cap \{\beta\} = \emptyset$ e $\{\alpha\} \not\subset \text{int}(\beta), \{\beta\} \not\subset \text{int}(\alpha) \Rightarrow \{\alpha\} \subset \text{Ext}(\beta)$ e $\{\beta\} \subset$

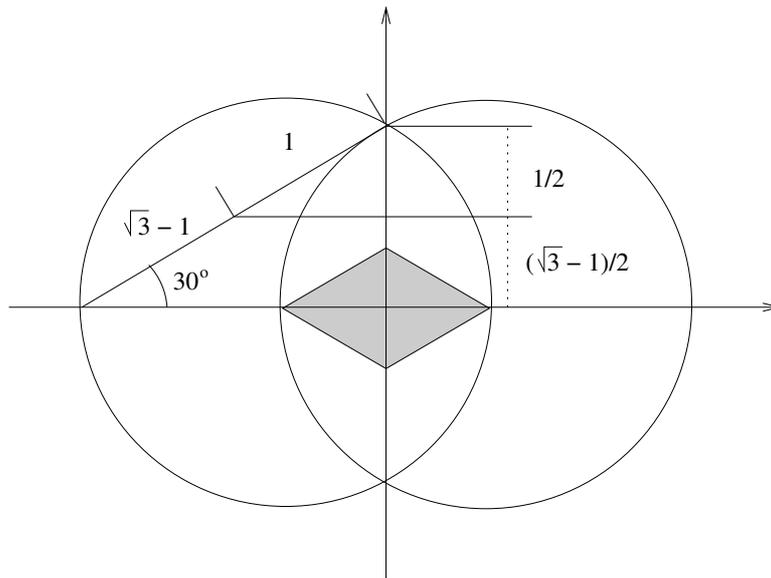


Figura 4.15: Aplicação do Teorema de Tales.

$Ext(\alpha)$. Agora, se $\exists P \in int(\alpha) \cap int(\beta)$ então $PQ \cap \{\alpha\} \ni R, Q \in \{\beta\}$ donde $R \in int(\beta)$ (\nearrow , pois $\{\alpha\} \subset Ext(\beta)$).

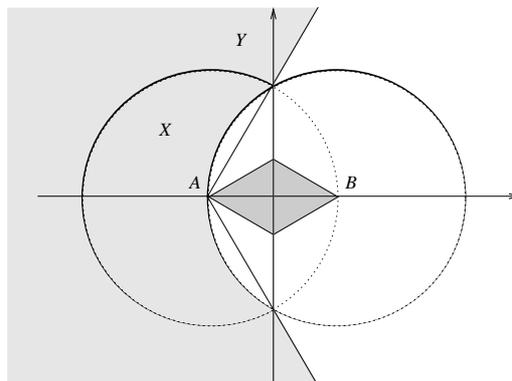


Figura 4.16: Ilustração do Caso 4.

Note que $\forall P \in \mathcal{D}(A, B) \cap \mathcal{D}(X, Y) \Rightarrow |PB| \geq 1 - \frac{1}{2\sqrt{3}} > \frac{1}{2}$, pois no pior caso teríamos $|XY| = 1$ tangente à C_B . Suponha que $\exists P \in \mathcal{D}(A, B) \cap \mathcal{D}(X, Y)$. Portanto $\partial\mathcal{D}(A, B) \cap \partial\mathcal{D}(X, Y) \neq \emptyset$ (*).

Ocorre que $|PB| > \frac{1}{2}$, donde (*) acontece nas arestas que partem de A. A(s) aresta(s) de $\partial\mathcal{D}(X, Y)$ que corta(m) $\partial\mathcal{D}(A, B)$, ou é uma só (donde

$A \in \mathcal{D}(X, Y) \not\subset$ ou são duas.

Seja Q o vértice de $\partial\mathcal{D}(X, Y)$ em $\mathcal{D}(A, B)$, como na Figura 4.17. Podemos então prolongar os lados de $\partial\mathcal{D}(X, Y)$ até $|XY| = 1$, deslocar $Q \in \mathcal{D}(A, B)$ até que se torne vértice de $\partial\mathcal{D}(A, B)$, e rotacionar $X\hat{Q}Y$ no sentido horário até que \overrightarrow{YQ} seja vertical (note que ela faz inicialmente ângulo maior que 90° à direita com \overrightarrow{AB}). Mas $|QY| = 1/\sqrt{3} \Rightarrow Y = C$ e $X = A \not\subset$. Então, para que $\mathcal{D}(X, Y) \cap \mathcal{D}(A, B) \neq \emptyset$ devemos ter X, Y em lados opostos de \overleftrightarrow{AB} .

Seja \vec{s} com origem em X e mesmo sentido e direção de \overrightarrow{AB} , $\vec{s} = \overrightarrow{XZ}$ como na Figura 4.18. Assim $Y\hat{X}Z < 60^\circ$ e $\overrightarrow{YA} \cap \overrightarrow{XZ} = A'$, mas $A' \in \overrightarrow{XZ}$ pelo Teorema do Ângulo Externo (por hipótese, $Y\hat{A}'Z > 60^\circ$). Mas a ordenada de A' é menor que a ordenada de $A = 0 \therefore A \in \text{int}(Y\hat{X}Z)$.

Também, $\overline{YQ} \cap \partial\mathcal{D}(A, B) \neq \emptyset$ e sem perda de generalidade $\overrightarrow{YQ} \cap \vec{s} = Z$. Então A, X estão do mesmo lado que $\overrightarrow{YQ} = \overrightarrow{YZ}$.

Agora, $Y\hat{A}B > 60^\circ$ e $Y\hat{X}Z < 60^\circ \therefore X\hat{Y}A > 0$. Também, $X\hat{Y}Z = 30^\circ > 0 \Rightarrow A, Z$ estão do mesmo lado que \overrightarrow{XY} , o que implica $A \in X\hat{Y}Z$ por definição $\therefore A \in \triangle XYZ \subset L(X, Y) \not\subset$

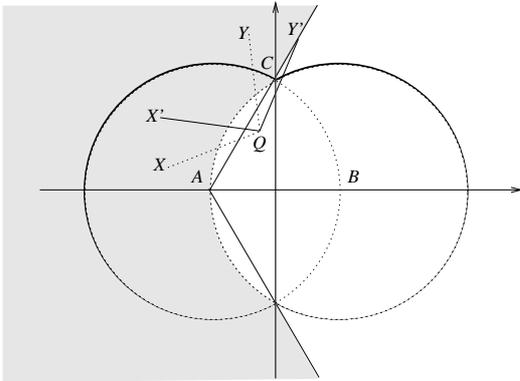


Figura 4.17: Caso 4 (Parte I).

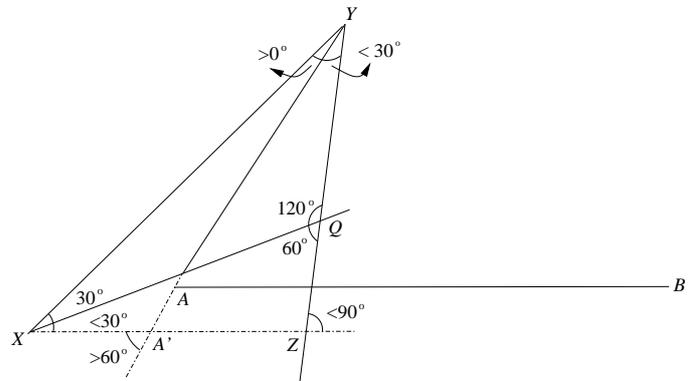


Figura 4.18: Caso 4 (Parte II).

4.14 Região de decisão

Em cada passo do algoritmo da Seção 4.8, temos a situação mostrada na Figura 4.7. Pode ocorrer ou não que A e C estejam conectados por um ponto Steiner S , e no primeiro caso B ser outro ponto Steiner cuja localização é

desconhecida.

Considere todas as árvores mínimas de Steiner com ao menos três terminais dados $A_1 = A, A_2 = C, A_3, A_4, \dots$, onde A_3, A_4, \dots podem variar em todo plano, ha hipótese de que A e C estão conectados por um ponto Steiner S . Um ponto P no plano é *ponto de decisão* se não for vértice de nenhuma destas árvores. O conjunto de todos os pontos de decisão é a *região de decisão*, que iremos determinar agora.

Ao determinar esta região, caso algum terminal A_k esteja nela, conclui-se que a hipótese estava errada, ou seja, A e C não estão conectados por nenhum ponto Steiner.

Voltando à hipótese, a Seção 4.9 garante que $(L(S, A) \cup L(S, C)) \cap \{A_1, A_2, A_3, \dots\} = \emptyset$. Portanto, a região de decisão contém o seguinte conjunto

$$I = \bigcap_{S \in \widehat{AC}} L(S, A) \cup L(S, C).$$

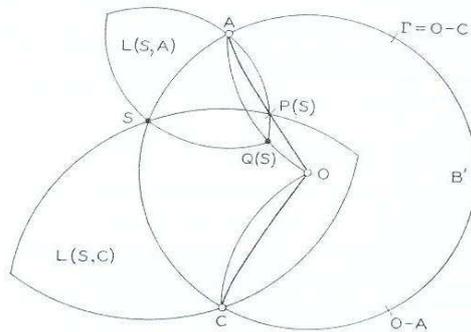


Figura 4.19: A região I .

A Figura 4.19 mostra I . Ela é limitada por \widehat{AC} e dois outros arcos, que não são de circunferência. Estes são determinados pelos pontos $P(S)$ que estão simultaneamente no contorno de $L(S, A)$ e $L(S, C)$, ou seja

$$|P(S) - S| = |A - S|, \quad |P(S) - C| = |C - S|. \quad (4.2)$$

O arco entre A e O é dado por (4.2) para S variando em \widehat{AC} , com $\widehat{SA} \leq 60^\circ$. Note que (4.2) recai numa equação de grau quatro. O artigo [GP] trata a região I de modo aproximado por arcos de circunferência, uma limitação

própria da época em que foi escrito. Atualmente, podemos trabalhar com um arco gráfico padrão, que rapidamente através de rotações, translações e homotetias determina I na tela do computador.

Implementaremos esta Seção futuramente do seguinte modo: após obtermos uma árvore de Steiner, traçamos o conjunto de todas as regiões I como descritas acima. Caso algum terminal recaia no interior deste conjunto, então conclui-se que a árvore obtida não é mínima. Senão, nada se conclui.

4.15 Conjectura $\frac{\sqrt{3}}{2}$

O comprimento de uma árvore de Prim L_P é obviamente limite superior do comprimento da árvore de Steiner L_S . Em [DH] prova-se que $\rho = \sqrt{3}/2$, como conjecturado em [GP]. Para cada conjunto de terminais A_1, A_2, \dots , considere a razão L_S/L_P . Percorrendo todos os possíveis A_1, A_2, \dots , teremos um valor $\rho > 0$ tal que $L_S/L_P \leq \rho$ e $\forall \epsilon > 0 \exists A_1, A_2, \dots, A_n$ para os quais $L_S/L_P > \rho - \epsilon$. Ou seja, ao obtermos numericamente uma árvore de Steiner para um certo conjunto de terminais A_1, A_2, \dots, A_m e computarmos L_S/L_P , caso este seja maior que $\sqrt{3}/2$, então com certeza não temos a árvore mínima de Steiner. Caso for menor ou igual a $\sqrt{3}/2$, não podemos ainda concluir que a árvore de Steiner é mínima.

Capítulo 5

Metodologia

Este capítulo apresenta os principais algoritmos desenvolvidos para representar seus respectivos fluxogramas. O programa principal “stree.m” simplesmente chama os demais, como mostra o Fluxograma 5.1.

5.1 Algoritmo Steiner

O algoritmo Steiner tem como propósito minimizar o comprimento da árvore com três terminais. Para um dado conjunto de pontos (três), inseridos pelo usuário, os coeficientes 1, 2 e 3 definidos por cf_1, cf_2 e cf_3 , são calculados pela Equação 5.4.

Após o cálculo dos coeficientes, obtém-se $r = |pts(1) - S| < 1$, onde S é o ponto Steiner. Então S é calculado aplicando-se a equação 5.1:

$$S = pts(1) + w \cdot ((1 - r) \cdot z/w + r \cdot \exp(-i \cdot \pi/3)). \quad (5.1)$$

$$S = w\tilde{S} + pts(1) \quad (5.2)$$

Para $\tilde{z} = \frac{z}{w}$, temos:

$$\frac{\tilde{S} - \tilde{z}}{P - \tilde{z}} = r \Leftrightarrow \tilde{S} = \tilde{z} + r \cdot (P - \tilde{z}) \Leftrightarrow$$

$$\tilde{S} = (1 - r) \cdot \tilde{z} + r \cdot P. \quad (5.3)$$

Para resolver (5.3), deve-se encontrar o raio(ρ) e o centro da circunferência (C).

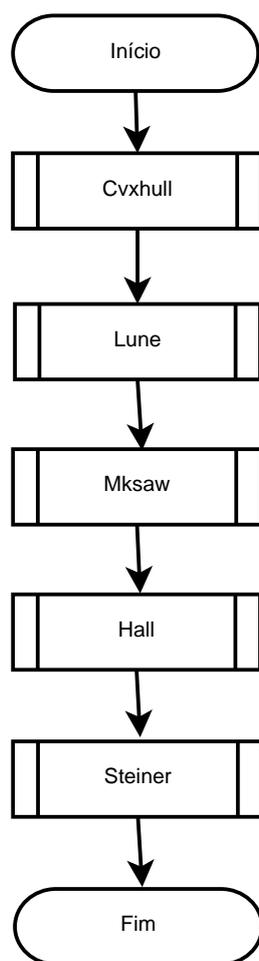


Figura 5.1: Fluxograma do algoritmo Stree.

$$\text{Temos } \cos 30^\circ = \frac{1/2}{\rho} \Leftrightarrow \rho = \frac{1}{\sqrt{3}}.$$

Como a altura de um triângulo equilátero de lado 1 é $h = \frac{\sqrt{3}}{2}$, então $y = \frac{1}{3} \cdot h \Rightarrow y = \frac{\sqrt{3}}{6} \therefore C = (1/2, -\sqrt{3}/6)$ e $\rho = 1/\sqrt{3}$.

Pela equação da circunferência definida por $\gamma(t) = C + \rho \cdot e^{it}$, temos:

$$S = \gamma(t) = \frac{1}{2} - \frac{\sqrt{3}}{6}i + \frac{1}{\sqrt{3}}e^{it} = z + r \cdot \left(\frac{1}{2} - \frac{\sqrt{3}}{2}i - z \right) \Leftrightarrow$$

$$\cdot \left[\left(\sqrt{3}z - \frac{\sqrt{3}}{2} + \frac{1}{2}i \right) \right] \cdot r + 3 \cdot \left| z - \frac{1}{2} + \frac{\sqrt{3}}{6}i \right|^2 - 1 = 0.$$

Logo, os coeficientes que determinam o ponto Steiner são:

$$\begin{cases} cf_1 : 3 \cdot \left| z - e^{-\frac{i\pi}{3}} \right|^2 \\ cf_2 : 2\sqrt{3} \cdot \text{Re} \left[\left(\sqrt{3}z - e^{-\frac{i\pi}{6}} \right) \cdot \left(e^{\frac{i\pi}{3}} - \bar{z} \right) \right] \\ cf_3 : \left| \sqrt{3}z - e^{-\frac{i\pi}{6}} \right|^2 - 1 \end{cases} \quad (5.4)$$

O Fluxograma 5.3 representa o algoritmo para calcular o ponto Steiner.

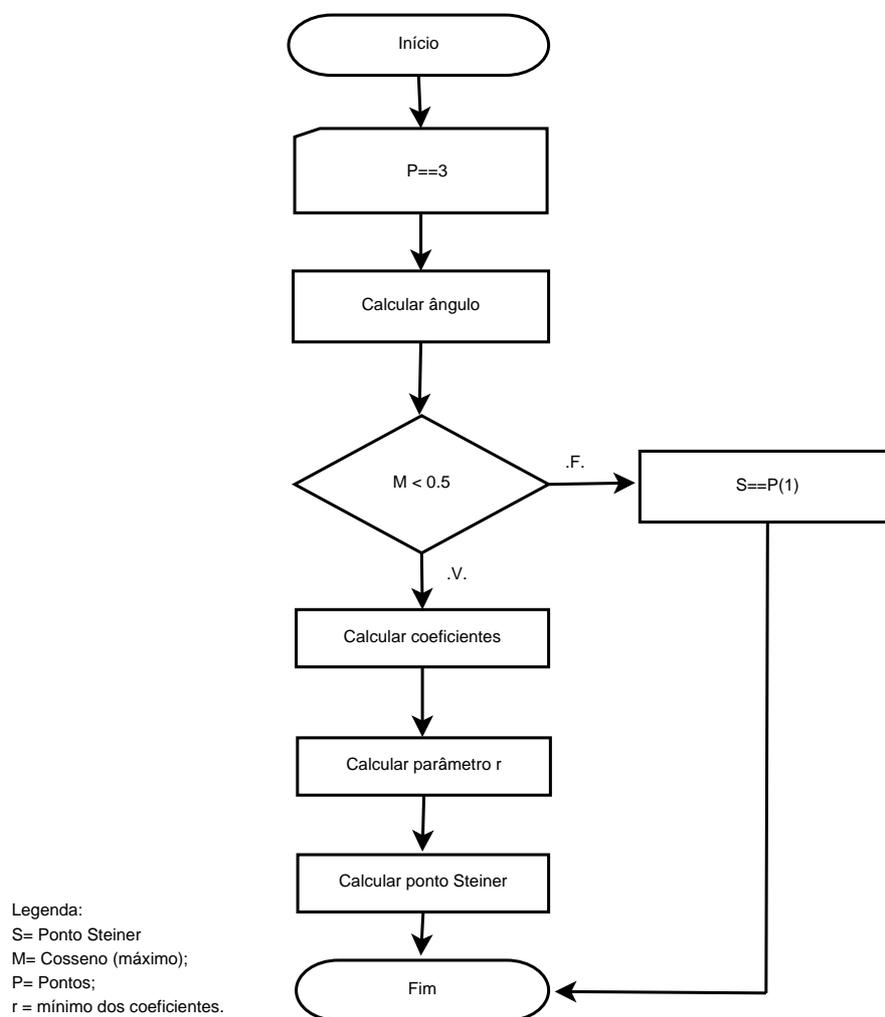


Figura 5.3: Fluxograma do algoritmo para cálculo do ponto Steiner.

5.2 Algoritmo Cvxhull

O algoritmo Cvxhull têm por objetivo construir o menor polígono convexo que contenha um dado conjunto de pontos. Para isso, o mesmo recebe um conjunto de pontos no sistema de eixos. Em seguida, uma nova origem do sistema de eixos é determinada a partir do ponto com menor abscissa, e depois com a maior ordenada dentre os que tem aquela abscissa. Com a obtenção desse sistema, verifica-se o ponto que realiza o menor ângulo com a origem e, dentre todos eles, seleciona o de maior distância para plotar a primeira aresta do polígono. Com isso, o método determina o novo sistema de eixos (conforme visto na Seção 4.6). Então, de forma semelhante obtém a segunda aresta do polígono com um novo sistema de eixos. Assim sucessivamente, esse procedimento é aplicado até quando o ponto, que possui a maior distância com a origem do respectivo sistema de eixos, coincidir com o ponto da origem do primeiro sistema. Com isso o algoritmo permite determinar o polígono convexo. O fluxograma apresentado na Figura 5.4 mostra as principais etapas desse algoritmo.

5.3 Algoritmo Lune

O algoritmo Lune busca minimizar o comprimento da árvore (maiores detalhes podem ser obtidos na Seção 4.9). Para aplicação desse algoritmo, inicialmente, o método Cvxhull é aplicado sobre o conjunto de pontos para traçar o polígono convexo. Diante dessas informações, aplica-se a propriedade da lua sobre cada aresta do polígono. Esse procedimento verifica se há pontos no interior da região demarcada em cada aresta. Caso isso ocorra, calcula-se a distância entre eles e o ponto que menor dista da aresta é selecionado. Com isso, essa região é dividida em duas novas arestas, uma entre o ponto anterior e o novo ponto, e a outra desse ponto até o ponto seguinte. Novamente, a propriedade da lua é aplicada com o propósito de verificar se há novos pontos no interior da nova região demarcada. Caso encontre novos pontos, esse procedimento é repetido enquanto houver pontos no interior da nova região. Esse procedimento é aplicado sobre todas as arestas do polígono traçado pelo Cvxhull. A Figura 5.5 representa o fluxograma do algoritmo.

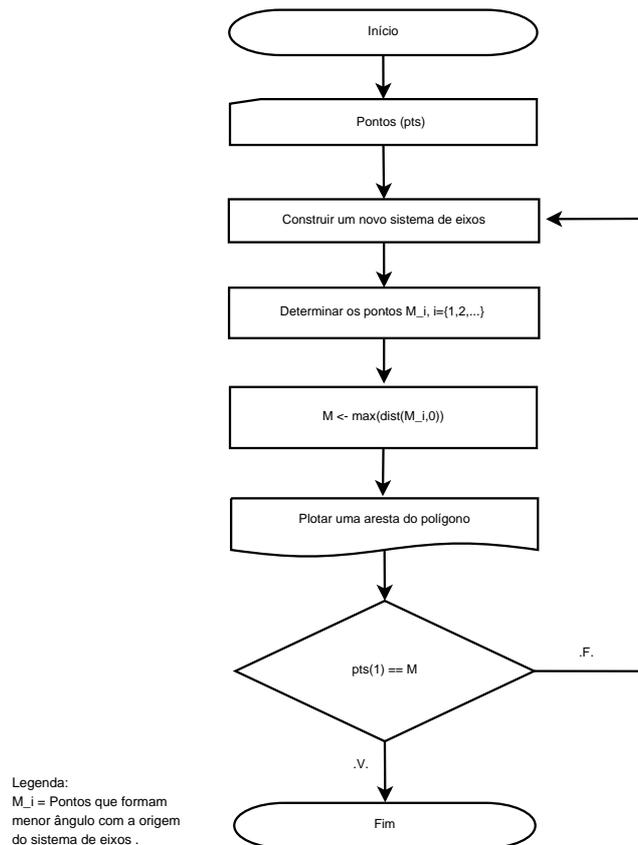


Figura 5.4: Fluxograma do algoritmo Cvxhull.

5.4 Algoritmo Remake

O algoritmo Remake é um submódulo do método Fullt que, a princípio tenta traçar de uma árvore plena. Caso não encontre-a, retorna as informações para Fullt. Para o algoritmo proposto, localiza-se os pontos plotados pelo usuário no sistema de coordenadas. Em seguida, irá construir três vetores 0, 1 e 2 e aplica-se um processo de verificação entre eles, ou seja, caso o vetor 1 não esteja entre os vetores 0 e 2, após percorrer todo arco, o algoritmo é finalizado. Caso contrário, um contador será incrementado e o vetor 1 será novamente construído. No entanto, se o vetor 1 estiver entre os vetores 0 e 2, plota-se o primeiro ponto Steiner e assim, calcula a distância t definida pela Equação 5.5. Diante deste resultado determina-se o segundo ponto Steiner. Construindo novos vetores 0, 1 e 2, verifica se o vetor 1 está à esquerda do vetor 2. Caso afirmativo, volta ao início e modifica o vetor 1 construído para

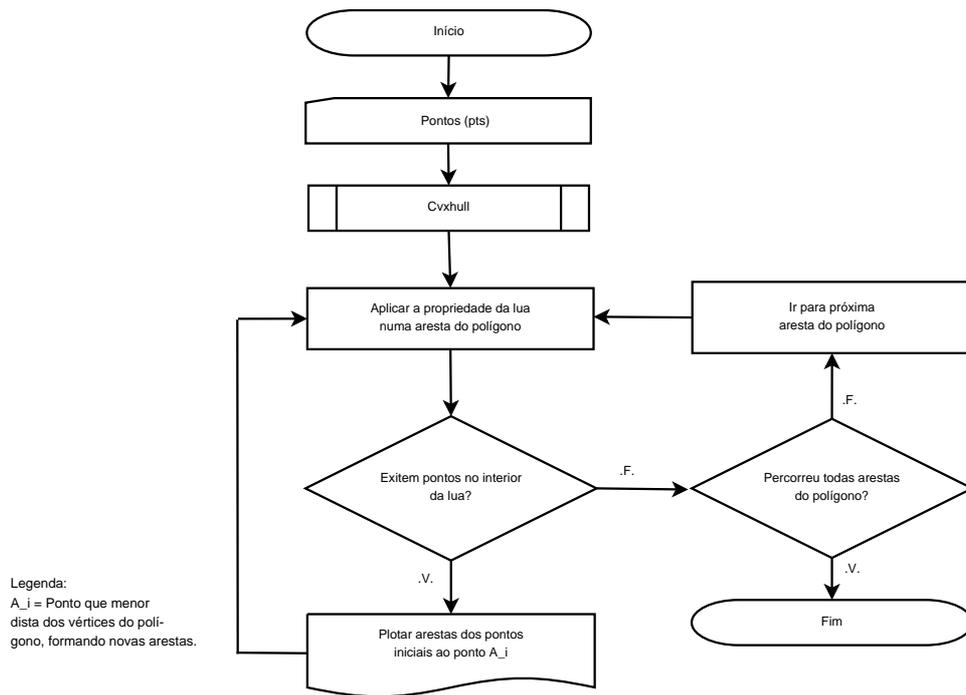


Figura 5.5: Fluxograma do algoritmo Lune.

efetuar novamente o traçado. Porém, se o vetor 2 não estiver à direita do vetor 1, verifica se o vetor 1 percorreu todo o arco e, em caso afirmativo, o algoritmo é finalizado, caso contrário, se o vetor 1 não estiver à esquerda do vetor 0, chama o algoritmo Lasttry para realizar uma última tentativa. Se encontrar a árvore, plota a mesma e conclui. Senão, concluir o algoritmo. Entretanto, se o vetor 1 estiver à direita do vetor 0, calcula-se a distância t pela Equação 5.5 e plota o terceiro ponto Steiner. Em seguida, se $s < n - 2$ e o vetor 1 estiver entre os vetores 0 e 2, calcular t (Equação 5.5) e plotar os pontos Steiner até que ou $s = n - 2$ ou o vetor 1 não esteja entre os vetores 0 e 2. Por fim, Stn recebe todos os valores de $stn(1:end)$, onde conclui-se o algoritmo.

Problema 1. *Sejam dois pontos $A = (A_1, A_2)$ e $B = (B_1, B_2)$, donde partem dois vetores não paralelos $\vec{v} = (v_1, v_2)$ e $\vec{w} = (w_1, w_2)$ respectivamente. Encontre o ponto de encontro das retas (A, \vec{v}) e (B, \vec{w}) .*

Solução. Considere as retas r_1 e r_2 de modo que $r_1 : A + t \cdot \vec{v}$ e $r_2 : B + s \cdot \vec{w}$.

O ponto de encontro entre elas é $A + t \cdot \vec{v} = B + s \cdot \vec{w} \Rightarrow (A_1, A_2) + (tv_1, tv_2) = (B_1, B_2) + (sw_1, sw_2)$. Assim,

$$\begin{aligned} \begin{cases} A_1 + tv_1 = B_1 + sw_1 \\ A_2 + tv_2 = B_2 + sw_2 \end{cases} &\Rightarrow \begin{cases} tv_1 - sw_1 = B_1 - A_1 \\ tv_2 - sw_2 = B_2 - A_2 \end{cases} \Rightarrow \\ &\Rightarrow \begin{bmatrix} v_1 & -w_1 \\ v_2 & -w_2 \end{bmatrix} \cdot \begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} B_1 - A_1 \\ B_2 - A_2 \end{bmatrix}. \end{aligned}$$

Assim,

$$t = \frac{\text{Im}\{w \cdot \bar{A} + B \cdot \bar{w}\}}{\text{Im}\{v \cdot \bar{w}\}} \quad (5.5)$$

◇

Problema 2. Verificaremos quando dois vetores $vec0$ e $vec2$ estão do mesmo lado em relação ao vetor $vec1$.

Solução. Sejam

$$\begin{cases} vec0 = a + bi & (a, b) \\ vec1 = c + di & (c, d) \\ vec2 = e + fi & (e, f) \end{cases}$$

$$\begin{cases} vec1 \cdot \overline{vec0} = (a, b) \cdot (c, -d) = (ac + bd, bc - ad) \Rightarrow \text{Im} = bc - ad \\ vec1 \cdot \overline{vec2} = (a, b) \cdot (e, -f) = (ae + bf, be - af) \Rightarrow \text{Im} = be - af \end{cases}$$

Assim, $vec1 \times vec0 = (0, 0, ad - bc)$ e $vec1 \times vec2 = (0, 0, af - be)$.

Logo, invertendo os sinais do produto vetorial acima, teremos que um vetor qualquer estará do lado direito de $vec1$ se seu produto for positivo e a esquerda, negativo. ◇

A representação do algoritmo segue-se do Fluxograma 5.6.

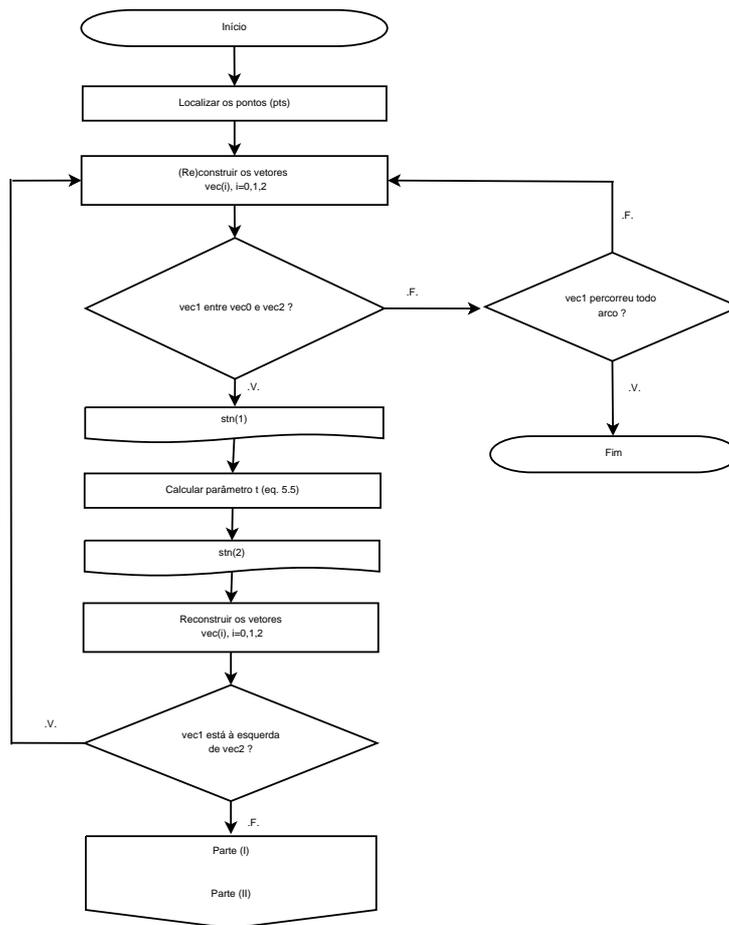


Figura 5.6: Fluxograma Remake (Parte I).

5.5 Algoritmo Fullt

O propósito do algoritmo Fullt proposto é construir uma árvore plena. No primeiro momento, o método executa o submódulo Remake. Caso encontre uma falha (bug), conclui-se o algoritmo (pois nem sempre existe a árvore plena). Senão, terá uma tolerância de três graus e o número de iterações será no máximo cem. O vetor 1 estando à direita do vetor 2, o arco é dividido em 6 partes iguais e, enquanto este vetor 1 estiver à direita do vetor 2, não houver falha e o número de iterações for menor ou igual a cem, chama-se o algoritmo Remake. Isso é feito enquanto verifica-se o cosseno entre os vetores 1 e 2. Se o valor do cosseno for menor que a tolerância, não houver falha e o número de iterações não ultrapassar o limite, conclui-se o programa. Senão,

retorna ao início do programa para fazer a verificação do vetor 1 estar à direita ou não do vetor 2. Agora, se o vetor 1 não estiver a direita do vetor 2, novamente o algoritmo divide o arco em 6 partes iguais. Caso o vetor 1 estiver à esquerda do vetor 2, não tiver falha e o número de iterações for menor ou igual a cem, chama o algoritmo Remake até que alguma dessas hipóteses não se verifique. Em seguida, faz a verificação do cosseno como descrito acima. Caso o número de iterações esteja superior a cem, conclui-se o algoritmo e, caso contrário, volta ao início para efetuar a verificação dos lados dos vetores 1 e 2. O Fluxograma 5.9 representa o algoritmo descrito.

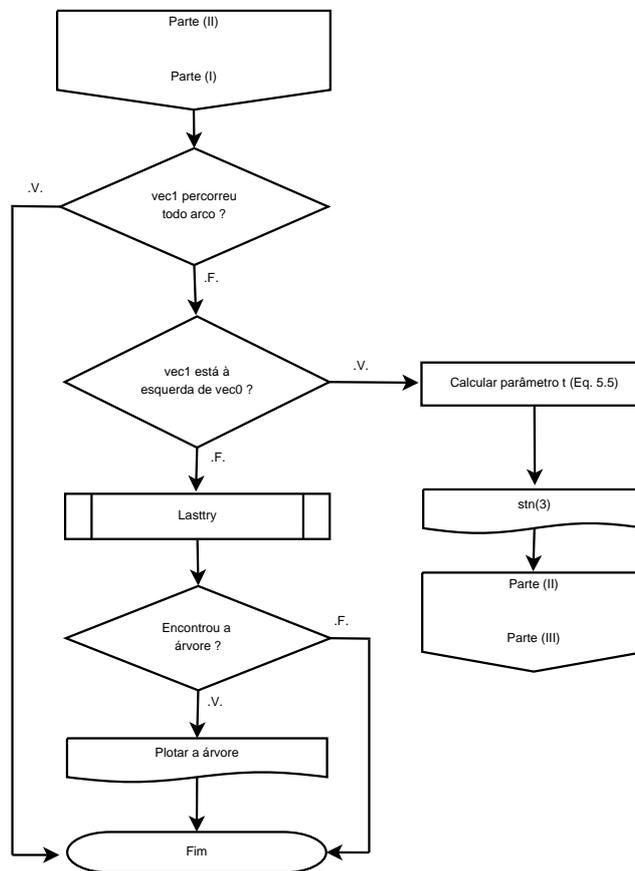


Figura 5.7: Fluxograma Remake (Parte II).

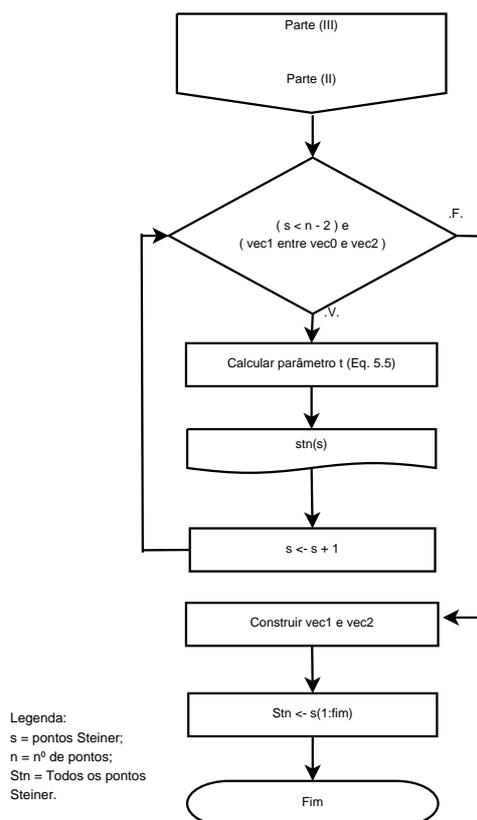


Figura 5.8: Fluxograma Remake (Parte III).

5.6 Algoritmo Lasttry

O algoritmo faz uma última tentativa com objetivo de encontrar a árvore plena (do algoritmo Fullt). As Figuras 5.11 e 5.12 representam o fluxograma do algoritmo Lasttry.

5.7 Algoritmo Mksaw

O propósito desse algoritmo é verificar se a estrutura topológica analisada tem representação semelhante a uma “serra”. Caso isso não ocorra, a representação atual irá sofrer mudança para que obtenha a forma desejada. O algoritmo trabalha da seguinte forma, analisa a quantidade de pontos presentes no sistema de coordenadas. Caso esse número de pontos esteja entre três e cinco, o método calcula os vetores para os pontos X e Y conforme a

Figura 5.13. Então, os pontos da presente região são re-enumerados (ocorre a troca do ponto 3 e ponto 4 somente quando não apresentar a topologia semelhante a uma “serra”). O Fluxograma 5.14 representa esse algoritmo.

5.8 Algoritmo Hall

O objetivo desse algoritmo é traçar uma árvore plena para quatro pontos. Inicialmente, verifica se a quantidade de pontos é igual a 4. Caso afirmativo, o método Fullt é executado e se retornar uma falha, renomeia os pontos e aplica-se novamente o algoritmo Fullt, determinando os pontos Steiner. Em seguida, desenha a árvore desejada. Caso negativo, recebe pontos Steiner e traça uma sub-árvore não plena. Se a quantidade de pontos for maior que 3, volta ao início para verificação de 4 pontos. Porém, se tiver uma quantidade maior que 4, não havendo falha, chama o algoritmo Fullt e recebe os pontos Steiner. Se obtiver falha, renomeia os pontos e chama Fullt, que determina os pontos Steiner. Finalmente, o algoritmo desenha a árvore encontrada e, novamente verifica-se a quantidade de pontos. Caso tenha uma quantidade maior que 4 pontos, volta para o início. Se o número de pontos for igual a 3, chama o algoritmo Steiner e traça a árvore. Com apenas 2, o método traça a aresta e finaliza o algoritmo. O Fluxograma 5.15 representa esse algoritmo.

5.9 Algoritmo de Prim

Esse algoritmo têm como objetivo encontrar uma árvore que contenha todos os vértices do grafo e cuja soma das arestas seja mínima. Inicia-se localizando os pontos marcados pelo usuário no sistema de coordenadas. Partindo do primeiro ponto, verifica-se dentre todos os outros, o que menor dista deste e traça a aresta. Renomeando este ponto para ponto 2, verifica-se os pontos que menor distam dos pontos 1 e 2, e traça a aresta de menor comprimento entre as duas encontradas. Renomeando o último ponto para 3, novamente verificam-se as arestas que possuem menor comprimento em cada um dos três pontos. A aresta com menor comprimento será traçada, renomeando o novo ponto 4. Esse procedimento será realizado até conectar todos os pontos

livres do sistema de coordenadas. A Figura 5.16 representa o fluxograma desse algoritmo de Prim.

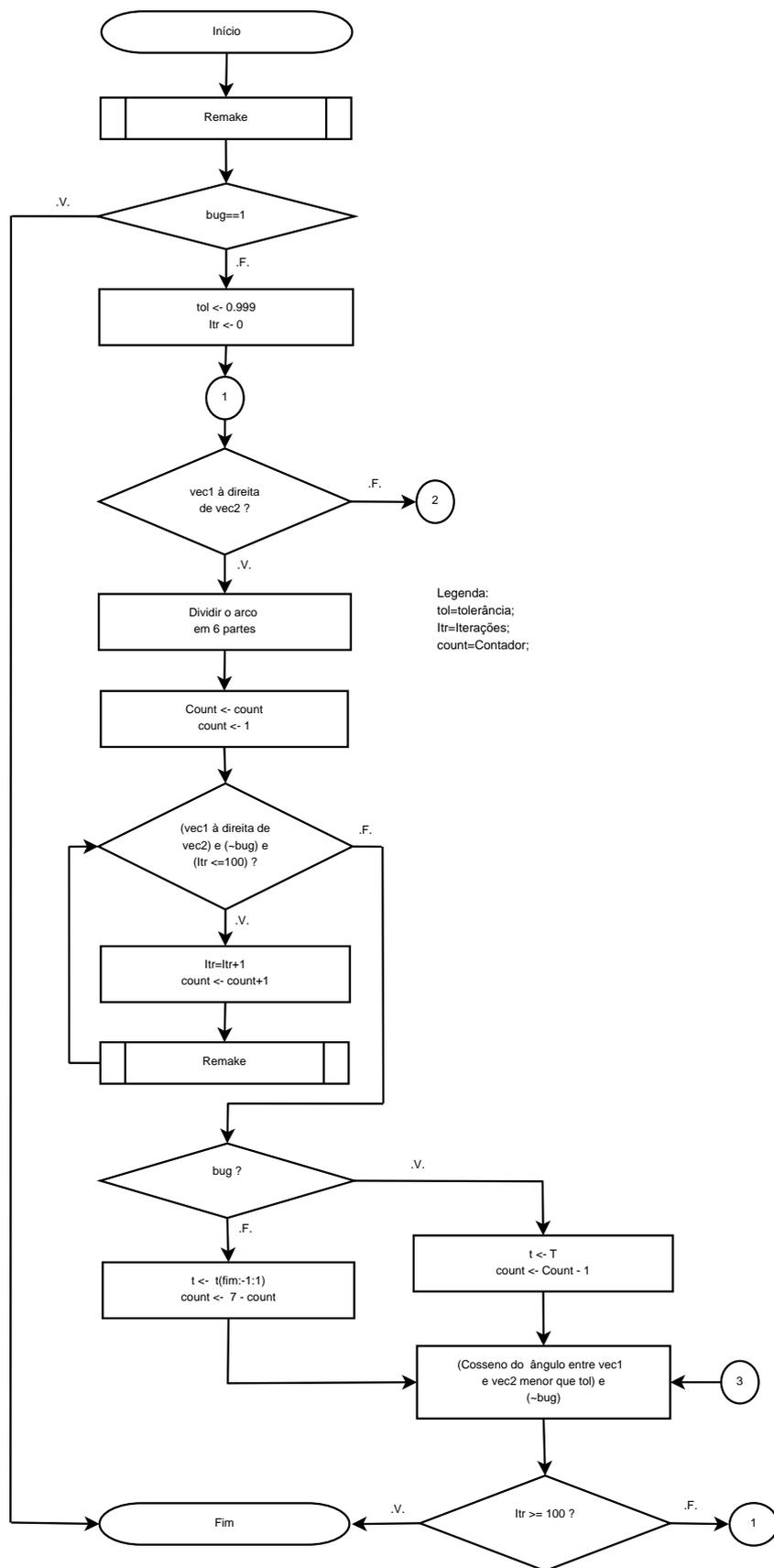


Figura 5.9: Fluxograma do algoritmo Fullt (parte I).

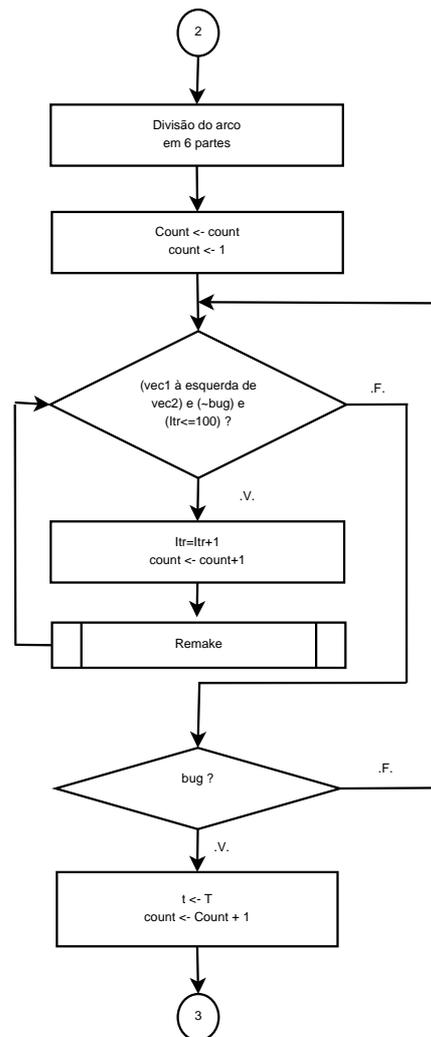


Figura 5.10: Fluxograma do algoritmo Fullt (parte II).

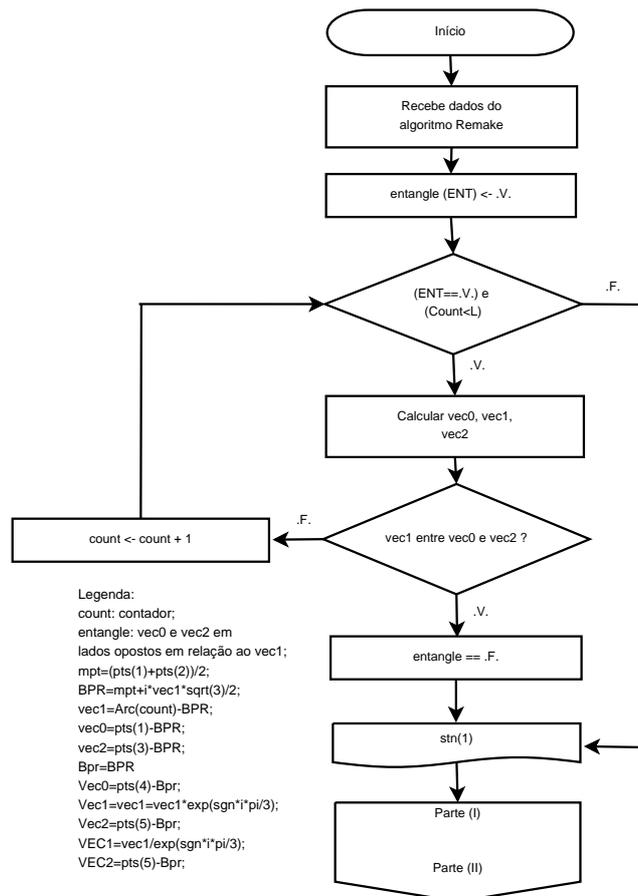


Figura 5.11: Fluxograma do algoritmo Lasstry (Parte I).

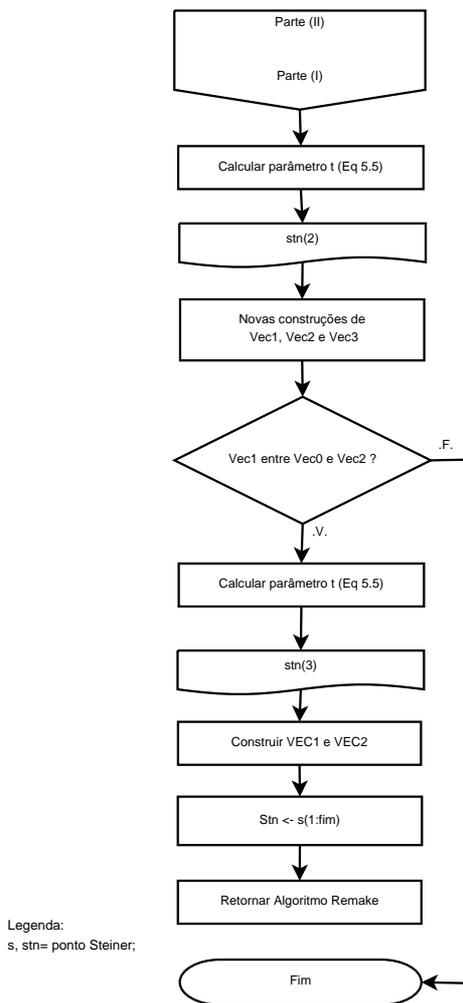


Figura 5.12: Fluxograma do algoritmo Lasttry (Parte II).

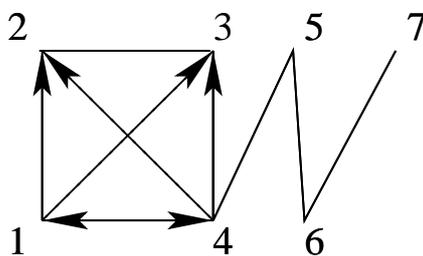


Figura 5.13: Exemplo da Mksaw.

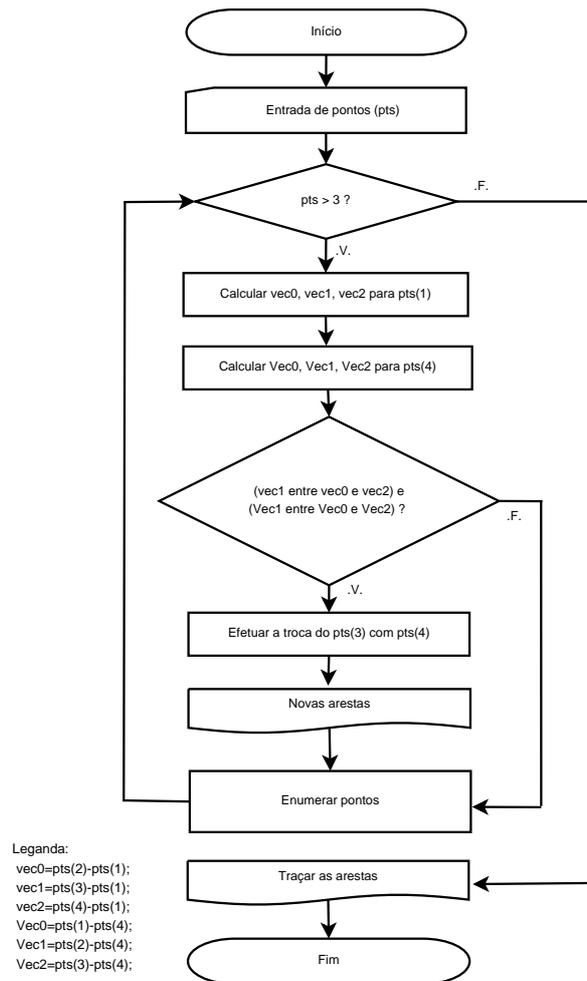


Figura 5.14: Fluxograma do algoritmo Mksaw.

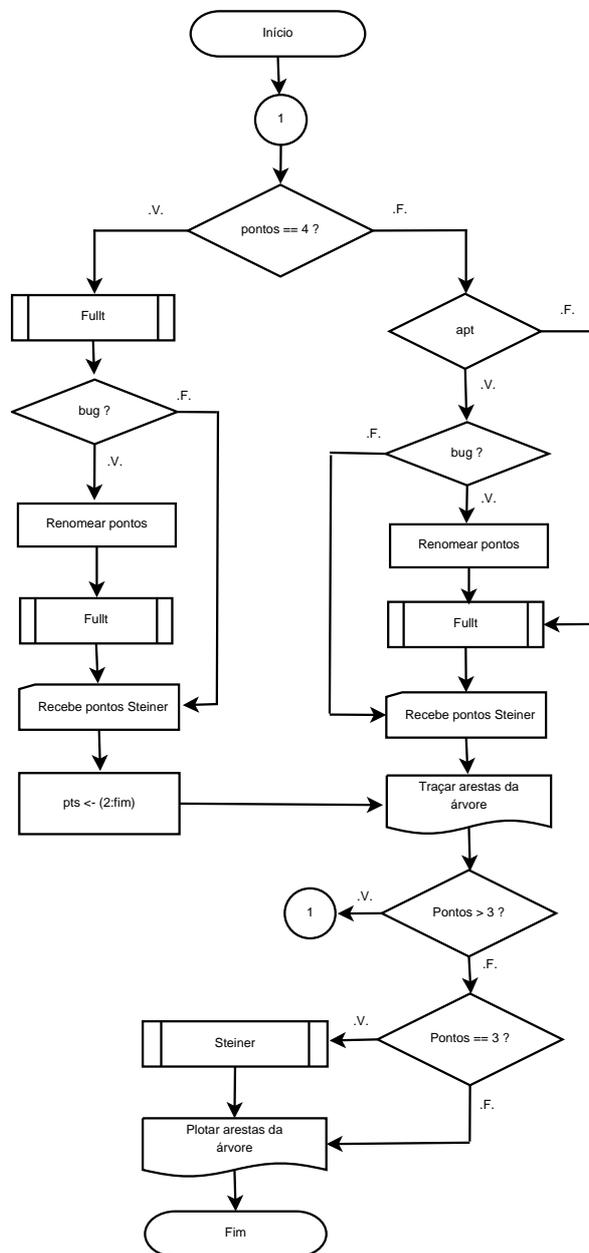


Figura 5.15: Fluxograma do algoritmo Hall.

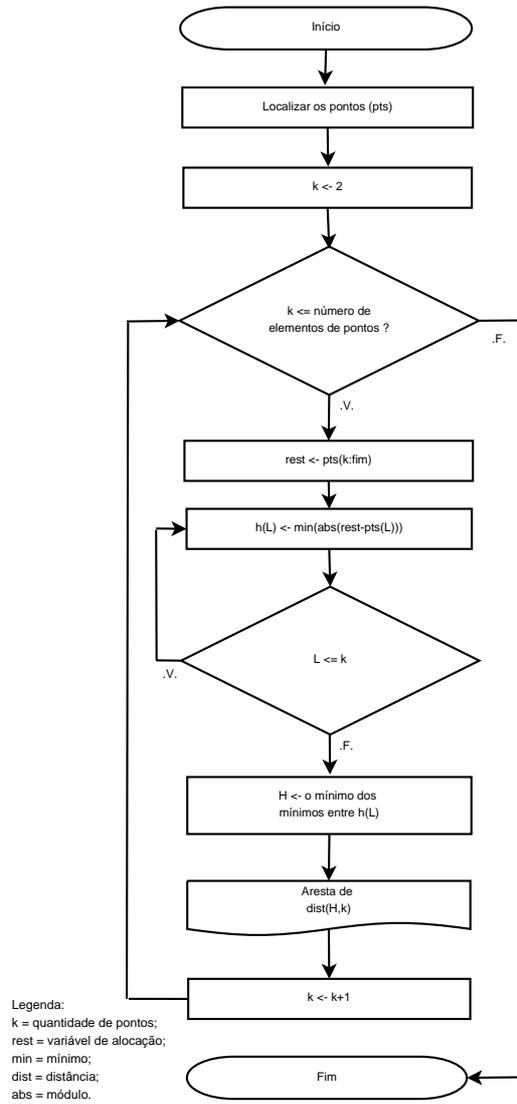


Figura 5.16: Fluxograma do algoritmo Prim.

Capítulo 6

Resultados

Este capítulo apresenta alguns resultados obtidos pelo desenvolvimento dos algoritmos, que propõem-se a encontrar a árvore plena.

6.1 Dois cenários

A Figura 6.1 apresenta o Cenário 1. Observe que para esta quantidade de pontos, o programa Stree chama vários subprogramas. Em certas localizações, percebe-se que duas arestas possuem um ângulo menor que 120° . Este fato ocorre porque antes de efetuar o traçado, tínhamos um ângulo “gordo” (maior que ou igual a 120°) e, por ter traçado um ponto Steiner, o ângulo se torna “magro” (menor que 120°). Para a resolução disto, o programa deve localizar a região dos pontos em que houve trocas de ângulos. Apenas nessas regiões de pontos que tenham este tipo de problema, deve-se efetuar o traçado novamente para poder encontrar a árvore mínima.

Para o Cenário 2 apresentado pela Figura 6.2, percebe-se que a árvore resultante do programa nos fornece é uma película. Para resolução desta, de início o programa tentaria eliminar um ponto Steiner ligado a três terminais. O terminal que ficasse isolado teria que conectar-se a um outro de modo que tenha o menor comprimento possível, preservando a propriedade do 120° . A questão seria descobrir qual a escolha para exclusão do ponto Steiner. Esta juntamente com o Cenário 1 são as próximas correções a serem feitas.

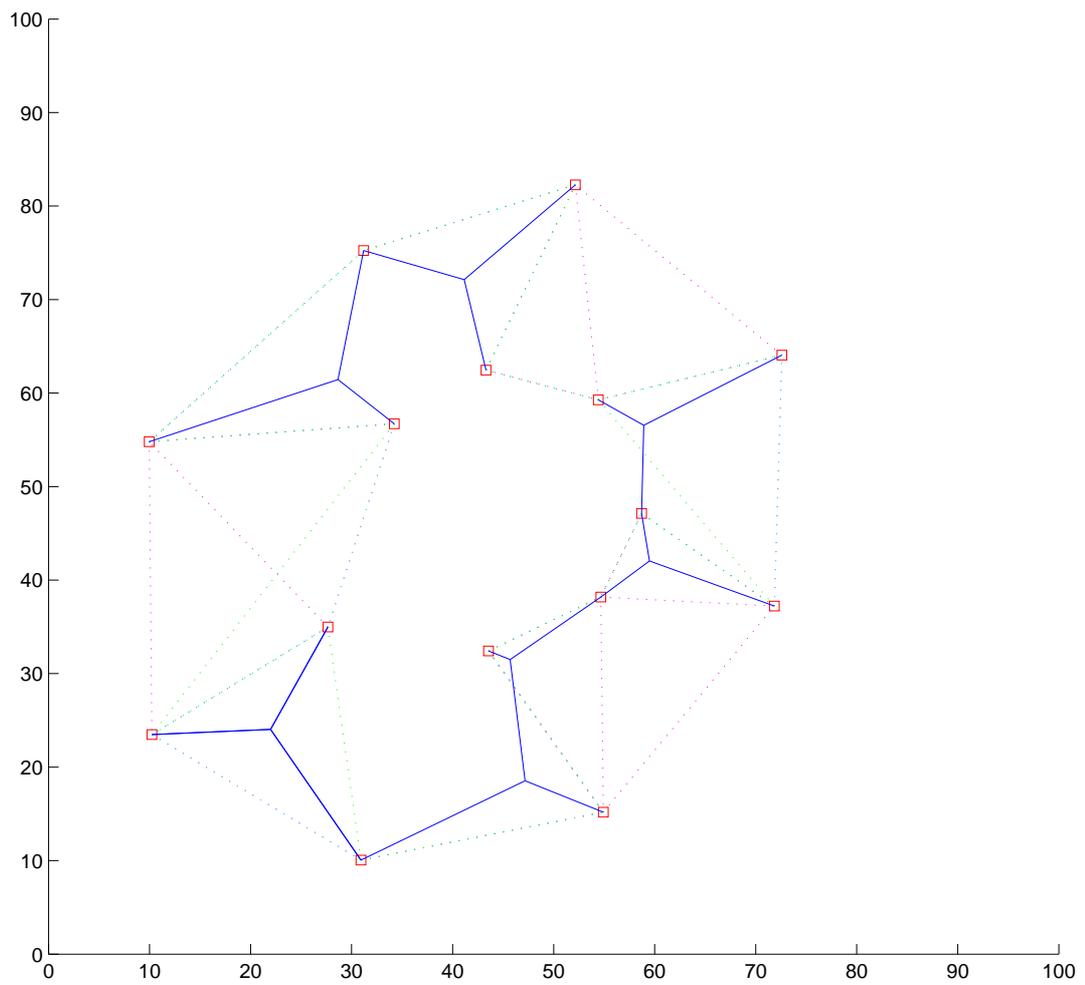


Figura 6.1: Cenário 1.

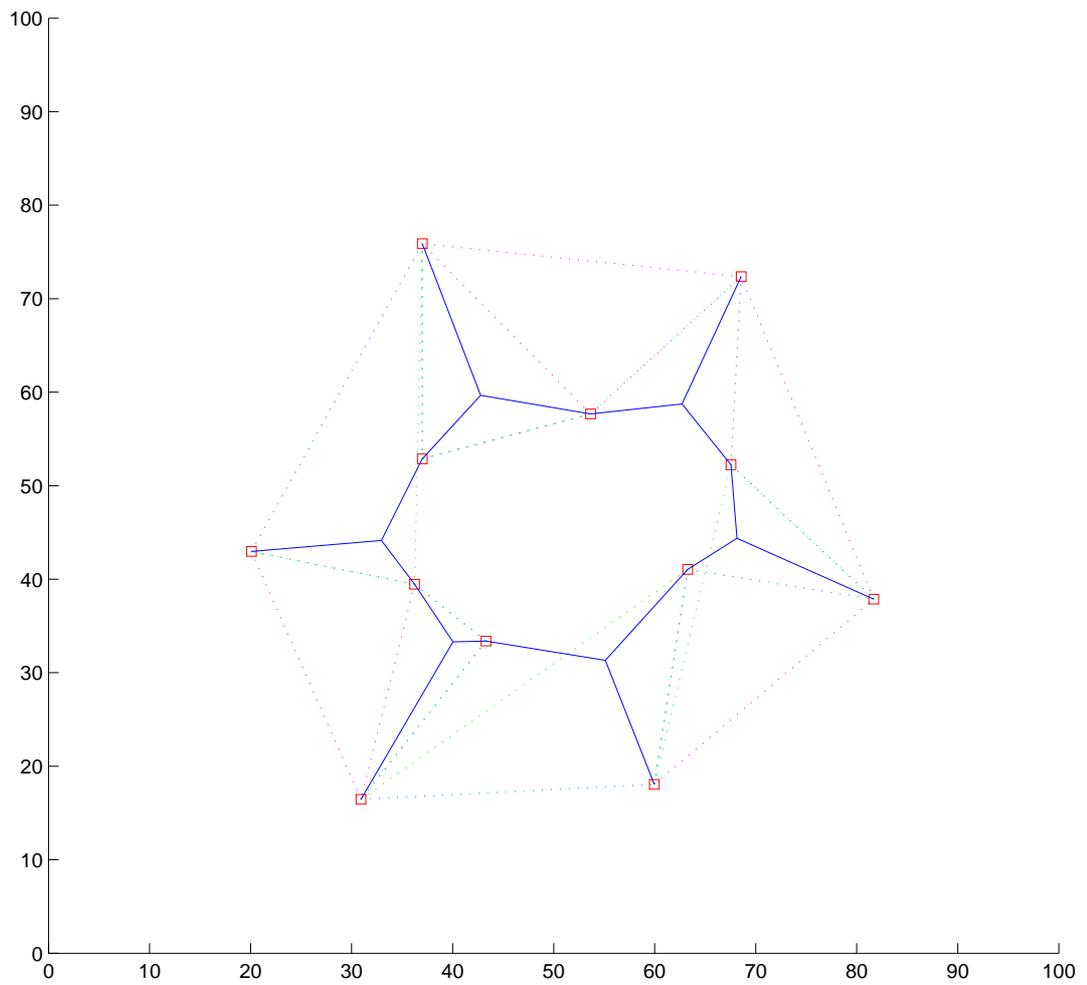


Figura 6.2: Cenário 2 (Película).

Capítulo 7

Conclusões

Apesar das pequenas falhas que, por vezes, o programa “stree.m” apresenta, julgamos que é vantajoso investir em seu desenvolvimento, a despeito de já termos outros “softwares” disponíveis para o traçado de árvores de Steiner. Retomando o que foi explicado na Introdução deste trabalho, programas como *Geosteiner* e o *Surface Evolver* não se encontram numa linguagem tão acessível como a do MatLab. De fato, Java ou C++ destinam-se a especialistas em programação, o que não é exigido pelo MatLab.

Certo, códigos mais acessíveis em C, como o programa “steiner.c” de Robert Bell, pautam-se por extremas simplificações, no caso partindo da “minimal spanning tree” gerada pelo algoritmo de Prim (vide Figuras 1.2(a), 1.2(b) e seções 4.12 e 5.15). Mas com isso perde-se o objetivo principal, que é o de obter uma árvore que tenha todas as chances de ser o mínimo global de comprimento, a árvore mínima de Steiner.

Dizemos “todas as chances”, pois as propriedades que foram estudadas neste trabalho, como a “Lua”, o “Diamante” e a “Conjectura $\sqrt{3}/2$ ”, são testes que indicam quando uma árvore de Steiner *não* é a mínima. Mesmo com sucesso em *todos* os testes, ainda não se pode afirmar que encontramos o mínimo global, pois eles trazem condições necessárias, mas não suficientes.

De qualquer modo, o objetivo de obter uma árvore com máxima chance de ser a mínima de Steiner, é justamente o que motiva desenvolvermos a programação do “stree.m”. Portanto, abrimos mão de códigos já disponíveis e retomamos [GP], o clássico artigo de Gilbert e Pollak, desde o seu início. Claro, o programa “stree.m” ainda não possui todos os testes implementados,

o que será realizado futuramente (vide a próxima seção).

Agora, com o intuito de garantir nossa propriedade intelectual, ainda não disponibilizamos *todos* os códigos MatLab. Isso será feito futuramente, após obtermos uma publicação científica de nosso trabalho, pois ele é original devido ao seu *propósito didático*. De fato, como tese de mestrado, seu maior propósito é tornar acessível a teoria e o aprendizado desta por meio de um “software” tão prático quanto o MatLab. Por causa disso, disponibilizamos o programa “stree.p” (executável mas com proteção de código), e alguns *Módulos Didáticos*, disponíveis em

<http://sites.google.com/site/wraffa/arquivos>

Os módulos didáticos são ao todo cinco: Cvxhull.m, Lune.m, Mksaw.m, Prim.m e Steiner.m. Eventualmente, chamam sub-programas como sread.p, swrite.p (todos os módulos de “stree.m” são em minúsculas). Sugerimos que o leitor execute estes programas em MatLab acompanhando as respectivas seções dos Capítulos 4 e 5.

Além do objetivo didático, também pretendemos não patentear o programa. Defendemos os “softwares” livres, e inclusive faremos adaptações do “stree.m” também para o Octave, versão livre do MatLab.

7.1 Pesquisas futuras

Alguns algoritmos serão desenvolvidos futuramente, como explicamos na seção anterior. Seguem-se os principais:

- Propriedade do Diamante;
- A Conjectura $\sqrt{3}/2$;
- Teorema de Maxwell;
- Região de Decisão.

Além destes, há outros como o da Seção 4.11, e de outras literaturas que não trataremos aqui, como a Propriedade do “Probe”. Um ótimo resumo de tais propriedades pode ser apreciado em

<http://www.archive.org/details/RonaldLG1988>

Referências Bibliográficas

- [Au] - G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. Complexity and approximation: combinatorial optimization problems and their approximability properties. Springer-Verlag, Berlin and Heidelberg, 1999.
- [Ca] - M. Carvalho, M. Cerioli, et al. Uma introdução sucinta a algoritmos de aproximação. 23º Colóquio Brasileiro de Matemática, IMPA, Rio de Janeiro, 2001.
- [Ch] - D. Chen, D.-Z. Du, X.-D. Hu, G.-H. Lin, L. Wang, G. Xu, Approximations for Steiner trees with minimum number of Steiner points, *Theoretical Computer Science*, 262 (2001) 83-99.
- [CR] - R. Courant & H. Robbins. O que é matemática? Ed. Ciência Moderna Ltda., Rio de Janeiro, 2000.
- [DH] - D.Z. Du, F.K. Hwang. A proof of the Gilbert-Pollak Conjecture on the Steiner Ratio. *Algorithmica* 7 (1992) 121-135.
- [Fo] - U. Fößmeier, M. Kaufmann. Solving rectilinear Steiner tree problems exactly in theory and practice. Proc. 5th European Symp. on Algorithms (1997), Springer-Verlag LNCS 1284, pp. 171-185.
- [G] - E. N. Gilbert, Random Steiner Minimal Trees. *SIAM J. Appl. Math.* 13 (1965), 376-387.
- [GP] - E. N. Gilbert, H. O. Pollak. Steiner Minimal Trees. *SIAM J. Appl. Math.* 16(1): 1-29, 1968.
- [Hu] - X.-D. Hu, T.-P. Shuai, Xiaohua Jia and Mu-Hong Zhang. Multicast routing and wavelength assignment in WDM networks with limited drop-offs. IEEE INFOCOM, 2004, url = cite-

seer.ist.psu.edu/hu04multicast.html.

[Ji] - X. Jia, X.-D. Hu, M. Lee, D.-Z. Du, and J. Gu. Optimization of wavelength assignment for QoS multicast in WDM networks. *IEEE Transactions on Communications* 49 (2001), 341-350.

[Ra] - S. Ravada, A. Sherman. Experimental evaluation of a partitioning algorithm for the Steiner tree problem in \mathbb{R}^2 and \mathbb{R}^3 . Technical Report TR~CS-93-03, University of Maryland, Baltimore County, 1993.

[Sa] - L. H. Sahasrabuddhe and B. Mukherjee. Multicast routing algorithms and protocols: a tutorial. *IEEE Network* 14 (2000), 90-102.

[Wa] - D. Warne, P. Winter, M. Zachariasen. Exact algorithms for plane Steiner tree problems: a computational study. Technical Report DIKU-TR-98/11, University of Copenhagen, 1998, <http://www.diku.dk>.

[Wi] - P. Winter, M. Zachariasen. Large euclidean Steiner minimum trees in an hour. Technical report DIKU-TR-96/34, University of Copenhagen, 1996, <http://www.diku.dk>.

[WZ] - R.L. Wheeden & A. Zygmund. Measure and integral. Marcel Dekker ed., New York, 1977.

[Za] - M. Zachariasen. Rectilinear full Steiner tree generation. Technical Report DIKU-TR-97/29, University of Copenhagen, 1997, <http://www.diku.dk>.

DADOS PESSOAIS

Nome: Wendhel Raffa Coimbra

Local e data de nascimento: Guararapes, 18/06/1985

EDUCAÇÃO

Colégio Lumen (Anglo), Guararapes, 2000-2002

Universidade Federal de Mato Grosso do Sul, Três Lagoas, 2003-2007

Matemática (Licenciatura)

FORMAÇÃO COMPLEMENTAR

2008-2008 Curso de verão em Análise na Reta (Carga horária: 48 h).

Universidade Federal do ABC, UFABC, Santo André, Brasil.

ATIVIDADES ACADÊMICAS

Bolsa Mestrado: UFABC Período:01\2008 – 03\2009

Bolsa Mestrado: FAPESP Período:03\2009 – 01\2010

PUBLICAÇÕES

1. RAFFA, W. C

ÁRVORES DE STEINER - TEORIA, GRAFOS E GERAÇÃO NUMÉRICA, 2009.

2. RAFFA, W. C, SANTOS, A. C. L. S., SANTOS, E. P., BECKER, J. M., SOUZA, M. C. A., MODESTO, S. A. G.

A Matemática Integrando A Universidade E A Escola Pública, 2007.

3. RAFFA, W. C

SOFTWARE LATEX COMO APOIO NA PRODUÇÃO ACADÊMICO-CIENTÍFICA DO CÂMPUS DE TRÊS LAGOAS, 2007.

4. LAGOAS-MS, C. P. T., SANTOS, B. R. L., FARIA, R. F., RAFFA, W. C

Integração e Recuperação, 2006.

5. RAFFA, W. C
Equações Diofantinas, 2005.

Participação em eventos

1. Apresentação de Poster / Painel no(a) **I Semana do CMCC - UFABC**, 2009.
(Congresso)
ÁRVORES DE STEINER - TEORIA, GERAÇÃO NUMÉRICA E APLICAÇÕES.

2. Apresentação de Poster / Painel no(a) **II ENCONTRO PAULISTA DE GEOMETRIA**,
2009. (Encontro)
ÁRVORES DE STEINER - TEORIA, GERAÇÃO NUMÉRICA E APLICAÇÕES.

3. **Semana do CMCC**, 2009. (Congresso)

4. **XIV EREMatSul - Encontro Regional de Estudantes de Matemática da Região Sul**,
2008. (Encontro)

5. **IV BIENAL DA SOCIEDADE BRASILEIRA DE MATEMÁTICA**, 2008. (Congresso)

6. **Matemática Financeira no Excel**, 2007. (Simpósio)

7. Apresentação Oral no(a) **I SEMANA CULTURAL DA FUNEPE**, 2006. (Simpósio)
INTRODUÇÃO AO EDITOR LÁTEX.

8. **SOFTWARE LATEX COMO APOIO NA PRODUÇÃO ACADÊMICO-CIENTÍFICA DO
CAMPUS DE TRÊS LAGOAS**, 2005. (Outra)

9. **V SEMANA DA MATEMÁTICA**, 2005. (Simpósio)

10. **V SEMANA DE MATEMÁTICA**, 2004. (Simpósio)

11. **IV SEMANA DE MATEMÁTICA**, 2003. (Simpósio)

Organização de evento

1. RAFFA, W. C
Matemática Financeira no Excel, 2007.

2. DEX/CPTL, P., RAFFA, W. C
DIGITAÇÃO DAS NOTAS DE AULAS DOS PROFESSORES DO DEX/CPTL, 2006.

3. DEX/CPTL, P., RAFFA, W. C
V SEMANA DA MATEMÁTICA, 2005.