

**Problemas de Proximidade e de Caminhos
Mínimos em Superfícies Poliédricas**

Gutemberg Bezerra Guerra Filho

Dissertação de Mestrado

Problemas de Proximidade e de Caminhos Mínimos em Superfícies Poliédricas

Este exemplar corresponde à redação final da
Dissertação devidamente corrigida e defendida
por Gutemberg Bezerra Guerra Filho e apro-
vada pela Banca Examinadora.

Campinas, 13 de julho de 1998.



Prof. Dr. Pedro Jussieu de Rezende
Instituto de Computação - UNICAMP
(Orientador)

Dissertação apresentada ao Instituto de Com-
putação, UNICAMP, como requisito parcial para
a obtenção do título de Mestre em Ciência da
Computação.



Problemas de Proximidade e de Caminhos Mínimos em Superfícies Poliédricas

Gutemberg Bezerra Guerra Filho

Julho de 1998

Banca Examinadora:

- Prof. Dr. Pedro Jussieu de Rezende
Instituto de Computação - UNICAMP (Orientador)
- Prof. Dr. Paulo Cezar Pinto Carvalho
Instituto de Matemática Pura e Aplicada - IMPA
- Prof. Dr. Jorge Stolfi
Instituto de Computação - UNICAMP
- Prof. Dr. João Meidanis (Suplente)
Instituto de Computação - UNICAMP

Tese de Mestrado defendida e aprovada em 08 de julho de 1998
pela Banca Examinadora composta pelos Professores Doutores



Prof. Dr. Paulo César Carvalho



Prof. Dr. Jorge Stolfi



Prof. Dr. Pedro Jussieu de Rezende

© Gutemberg Bezerra Guerra Filho, 1998.
Todos os direitos reservados.

Aos meus pais,
Rosa e Gutemberg.

Agradecimentos

Agradeço a Deus pelos momentos de felicidade que me presenteou nesta vida e por mais uma graça alcançada em meu caminho.

Tenho uma grande dívida de gratidão para com meus pais e minha família. Agradeço o preparo para a vida, o apoio nas decisões mais difíceis, a preocupação e o carinho. À Claudia, Deodato, Marina e Celina, pela linda família que me fizeram compartilhar.

Ao meu Amor, por me ouvir nos momentos de solidão, pelo companheirismo e afeto.

Agradeço aos povos brasileiro, nordestino e paulista que mesmo esgarçados levam o Brasil nas costas e, através de seus impostos, pagaram quase toda minha educação.

Ao Prof. Pedro de Rezende, pela confiança, tempo despendido neste trabalho, ensinamentos e orientação.

Os Profs. Guilherme Ellery, Marcos Negreiros e Fernando Carvalho me ajudaram a dar os primeiros passos na vida acadêmica. A eles, agradeço a sabedoria e incentivo.

Ao amigo Roberto, agradeço a compreensão, a convivência e apoio.

Aos colegas Edicesar, Alessandra e Gláucia pelos momentos agradáveis e conversas descontraídas.

Ao Victor, Elder e Cláudio, pela acolhida em Campinas e conselhos de quem sempre estava passos a frente.

Ao Mário e Guilherme, pelas leituras prévias de meus textos, discussões em Geometria Computacional e, principalmente, irmandade na orientação.

Ao Oliva e à Islene, pelas inúmeras e prestativas ajudas durante a implementação.

Ao CNPq e FAPESP, pelo subsídio financeiro, sem o qual este trabalho não estaria concluído.

Resumo

Planejamento de Caminho Mínimo é a área em *Geometria Computacional* que se preocupa com a determinação dos menores caminhos possíveis de um ponto a outro em um dado ambiente. Abordamos um problema (PGAD) de caminhos mínimos direcional que procura minimizar o trabalho total realizado para se mover um corpo sobre uma superfície poliédrica com coeficientes de atrito e inclinação constantes em cada face. Sua importância se deve ao fato de que este problema generaliza vários outros.

Realizamos a caracterização de caminho geodésico e mínimo segundo as restrições do problema, identificando o critério de otimalidade local correspondente. Para isso, demonstramos a convexidade estrita da *função distância geodésica atritada direcional* (FGAD) utilizando a teoria de funções convexas.

Desenvolvemos um algoritmo, baseado na metodologia *Dijkstra contínuo*, para resolver o problema PGAD. O algoritmo possui algumas particularidades relacionadas ao caráter direcional devido à função distância FGAD depender da direção de movimento e à caracterização de caminhos geodésicos. Realizamos a prova de corretude e a análise de complexidade do algoritmo proposto. Além disso, identificamos alguns detalhes omitidos em algoritmos Dijkstra contínuo encontrados na literatura e os completamos.

Estendemos o problema PGAD obtendo um algoritmo para construir um diagrama de Voronoi (VGAD) de caminhos mínimos sobre uma superfície poliédrica segundo a função distância FGAD. Reduzimos algumas generalizações de problemas de proximidade ao da construção deste diagrama e, dessa forma, o diagrama VGAD resolve estes problemas.

Implementamos um módulo externo ao programa Geomview para visualizar uma árvore de caminhos mínimos e um diagrama de Voronoi em superfície poliédrica para o problema da geodésica discreta (PGD) que é um caso especial do PGAD.

Abstract

Shortest Path Planning is the field of *Computational Geometry* that concerns the determination of feasible shortest paths from a point to another in a given environment. We deal with a directed shortest path problem (DFGP) that minimizes the total work spent to move a body on a polyhedral surface with constant friction coefficient and constant slope in each face. Its importance is due to the fact that it generalizes several others.

In order to characterize geodesic paths and shortest paths according to the constraints of the problem, we identify the corresponding local optimality criterion and we demonstrate the strict convexity of the *directed frictioned geodesic distance function* (DFGF) using convex function theory.

We develop an algorithm, based on the *continuous Dijkstra* methodology, to solve the DFGP problem. The algorithm contains some details related to the directed nature of the paths which is due to the distance function DFGF being dependent on the direction of motion and to the characterization of the geodesic paths. We prove the correctness of the proposed algorithm and analyze its complexity. Furthermore, we identify some details omitted in a few continuous Dijkstra algorithms found in the literature and fill them in.

We extend the DFGP problem and obtain an algorithm to construct a shortest path Voronoi diagram (DFGV) on a polyhedral surface according to the distance function DFGF. We reduce some generalizations of proximity problems to the construction of this diagram and, therefore, the DFGV diagram solves these proximity problems.

We implement an external module to the *Geomview* program to be able to visualize a shortest path tree and a Voronoi diagram on polyhedral surfaces to the discreet geodesic problem (DGP) that is a special case of DFGF.

Conteúdo

	v
Agradecimentos	vi
Resumo	vii
Abstract	viii
1 Introdução	1
1.1 Motivação e Aplicações	2
1.2 Definições Preliminares	3
1.3 Organização da Dissertação	5
2 Planejamento de Caminho Mínimo	6
2.1 Especificação de um Problema	6
2.1.1 Espaço Livre	7
2.1.2 Função Distância	10
2.1.3 Cardinalidade	12
2.2 Estruturas Geométricas	12
2.2.1 Grade Retilínea	12
2.2.2 Grafo Visibilidade	12
2.2.3 Árvore de Caminhos Mínimos	13
2.2.4 Mapa de Caminhos Mínimos	13
2.2.5 Mapa de Nível de Altura	14
2.2.6 Funil	16
2.2.7 Triangulação Geodésica	16
2.3 Problemas Bidimensionais	17
2.3.1 Interior de um Polígono Simples	18
2.3.2 Plano com Obstáculos Poligonais	19
2.3.3 Superfície de um Poliedro Convexo	22

2.3.4	Problema da Geodésica Discreta	22
2.3.5	Problema da Região Valorada	23
2.3.6	Restrições de Altura	24
2.4	Abordagens Utilizadas	25
2.4.1	Heurísticas	26
2.4.2	Algoritmos Aproximativos	27
2.4.3	Esquemas Aproximativos	29
2.4.4	Metodologias	30
3	Caracterização de Caminhos Mínimos	32
3.1	Definição do Problema	32
3.2	Caminhos Geodésicos e Mínimos	35
3.3	Caminhos Localmente f -Livres	45
3.4	Intervalos de Otimalidade	47
4	O Algoritmo Dijkstra Contínuo Direcional	50
4.1	Estruturas de Dados	51
4.2	Algoritmo	51
4.3	Funções Numéricas	59
4.4	Prova de Corretude	63
4.5	Análise de Complexidade	71
5	Conceitos Básicos de Proximidade	73
5.1	Par Mais Próximo	74
5.2	Todos Vizinhos Mais Próximos	75
5.3	Árvore Geradora Mínima	78
5.4	Proximidade de Caminhos Mínimos	80
5.5	Diagrama de Voronoi Euclideano	81
5.6	Diagramas de Voronoi Generalizados	82
5.6.1	Distância	82
5.6.2	Espaço	84
5.6.3	Objeto Geométrico	84
5.7	Diagrama de Voronoi de Ordem k	86
5.7.1	Diagrama da Seqüência	86
5.7.2	Diagrama do Subconjunto	86
5.8	Diagrama de Voronoi de Caminhos Mínimos	87

6	O Diagrama de Voronoi de Caminhos Mínimos	89
6.1	Diagrama de Voronoi GAD	89
6.2	A Subdivisão de Voronoi	92
6.3	Redução de Problemas	96
7	Implementação	99
7.1	<i>Geomview</i>	100
7.2	<i>FORMS</i>	101
7.3	Módulo Externo	101
7.4	<i>Quad-edge</i>	103
7.5	Fila de Prioridades e Lista de Intervalos	104
7.6	Rotinas Básicas	105
7.7	Melhoramentos no Algoritmo	107
8	Conclusões	110
8.1	Resultados Obtidos	110
8.2	Extensões e Trabalhos Futuros	112
A	O Ângulo β_x	114
B	Funções Convexas	119
C	Critério de Otimalidade Local	123
C.1	A Derivada Primeira	123
C.2	O ponto x_s	124
D	Tabelas de Símbolos	126
D.1	Letras Romanas Minúsculas	126
D.2	Letras Romanas Maiúsculas	127
D.3	Letras Gregas Minúsculas	128
D.4	Letras Gregas Maiúsculas	128
D.5	Notações Gerais	128
	Bibliografia	130

Lista de Tabelas

A.1	Tabela de expressões de β_x em função de θ_x e ψ	116
A.2	Tabela de expressões de $\cos(\beta_x)$ em função de θ_x e ψ	116

Lista de Figuras

1.1	Uma seqüência de faces aresta-adjacentes.	3
1.2	Classes de equivalência de caminho.	4
2.1	O espaço livre de um corpo.	8
2.2	Um caminho possível para um círculo.	8
2.3	Uma superfície poliédrica.	9
2.4	A função distância convexa de Minkowski.	11
2.5	Um grafo visibilidade relativo um conjunto de obstáculos.	13
2.6	Um mapa de caminhos mínimos relativo a s	14
2.7	Uma superfície poliédrica e seu mapa de nível de altura [38].	15
2.8	O grafo direcionado G_S [38].	15
2.9	Um funil para o vértice s	16
2.10	Uma pipa relativa aos vértices v_1, v_2 e v_3	17
2.11	Caminhos mínimos no interior de um polígono simples.	18
2.12	Caminhos mínimos no plano com obstáculos.	19
2.13	Caminhos mínimos para obstáculos com projeções disjuntas.	20
2.14	Um caminho mínimo retilíneo evitando obstáculos retangulares.	21
2.15	O problema da região valorada.	23
2.16	O problema do esconderijo máximo.	24
2.17	Hierarquia de problemas de caminhos mínimos bidimensionais.	25
2.18	Um grafo grade.	26
2.19	Grafo região de uma subdivisão planar.	27
2.20	Regiões acima de z	28
2.21	A imagem do ponto origem s , sua projeção I e sombra.	30
3.1	Diagrama do corpo livre.	33
3.2	Resultante das forças que atuam no corpo.	34
3.3	Superfície poliédrica com coeficientes de atrito cinético atribuídos a cada face.	35
3.4	Subcaminho contido no interior de uma face.	36

3.5	Lugar geométrico dos pontos equidistantes da origem em uma face.	38
3.6	Ângulos de incidência e refração.	39
3.7	Crítério de otimalidade local.	39
3.8	O ponto x^* deve estar no intervalo $[x_s, x_t]$	40
3.9	Caminhos geodésicos sobre uma aresta.	42
3.10	Um caminho geodésico que intercepta a seqüência de arestas E	44
3.11	Caminhos mínimos que se interceptam podem ser encurtados.	46
3.12	Intervalos de otimalidade são conjuntos conexos.	48
3.13	Partição da face f' em canais de acesso.	49
4.1	A frente de onda sobre uma superfície poliédrica.	50
4.2	A função $Difundir(c, e_i, f)$	53
4.3	A função $Projetar(I, e_i)$	56
4.4	A função $Usar-Criticamente(I, e_i, a)$	58
4.5	Casos distintos entre os intervalos I e I_1	64
4.6	Caso (*).	65
4.7	Caso (a).	65
4.8	Caso (b).	66
4.9	Caso (d).	67
4.10	Caso (e).	69
4.11	Prova do lema.	70
5.1	O problema de escalamento.	75
5.2	O problema de todos vizinhos mais próximos.	76
5.3	Varredura planar por um reta vertical sobre o diagrama de Voronoi.	77
5.4	Uma árvore geradora mínima.	78
5.5	O problema da árvore geradora mínima generalizado.	79
5.6	Uma árvore geradora mínima retilínea com obstáculos retangulares isotéticos.	80
5.7	Uma região e um diagrama de Voronoi euclideano.	81
5.8	Diagrama de Voronoi retilíneo.	82
5.9	Diagrama de Voronoi fluvial.	83
5.10	Diagramas de Voronoi valorados.	83
5.11	Diagrama de Voronoi de segmentos.	85
5.12	Diagrama de Voronoi de ordem $k = 2$	86
5.13	Diagrama de Voronoi em um polígono simples.	87
6.1	Um ponto evento para a subdivisão.	92
6.2	Células em uma subdivisão de uma face.	95
7.1	A apresentação inicial do <i>Geomview</i>	100

7.2	Um painel de interação <i>FORMS</i>	102
7.3	O painel de controle do módulo externo.	102
7.4	Os objetos geométricos apresentados no <i>Geomview</i> pelo módulo externo.	103
7.5	As primitivas <i>Onext</i> e <i>Lnext</i> da <i>quad-edge</i>	104
7.6	A função <i>Ponto-Empate</i> (I, I').	106
7.7	A função <i>Canal-Acesso</i> (I).	107
7.8	A função <i>Projetar</i> (I, e).	108
7.9	Um contra-exemplo para o Lema 6.6 [95] no PGD.	109
A.1	Casos distintos para a disposição da face f	114
A.2	Casos para a disposição da face f segundo os pontos q, q' e r	115
A.3	Cosseno e seno do ângulo de incidência θ_x	117
A.4	Gráficos em função de x	117
B.1	A função distância FGAD.	120

Capítulo 1

Introdução

A *Geometria Computacional* é uma área da Ciência da Computação que utiliza uma abordagem algorítmica para resolução de problemas geométricos. Os problemas geométricos avaliados são geralmente restritos ao plano, podendo generalizar-se para dimensões maiores ou métricas diferentes.

Contudo, seus objetos geométricos continuam imersos em espaços métricos bem comportados como, por exemplo, superfícies de curvatura constante. Algumas áreas da geometria computacional realizaram esforços para resolução de problemas específicos envolvendo superfícies poliédricas.

O *Planejamento de Movimentos (Motion Planning)* [104] é um campo da geometria computacional que se preocupa em determinar um caminho contínuo para a movimentação de um corpo entre uma origem e um destino em um ambiente que impõe restrições aos movimentos do corpo.

Os caminhos não precisam ser apenas qualquer caminho viável, podendo ter uma certa qualidade mensurada por uma métrica ou função distância que especifica um custo de movimento a ser minimizado. O *Planejamento de Caminho Mínimo (Shortest Path Planning)* consiste em encontrar um menor caminho viável [96]. Um problema de caminhos mínimos em um domínio geométrico é um problema fundamental em geometria computacional.

Muitos problemas de planejamento de caminho mínimo são resolvidos por técnicas heurísticas ou aproximativas. Contudo, nos concentramos no campo do planejamento de caminho mínimo que se preocupa em formalizar modelos e analisar complexidades de algoritmos.

Abordamos um problema de caminhos mínimos direcional onde um veículo autônomo (robô) usa algum conhecimento (mapa de um terreno) sobre o ambiente no qual está imerso para planejar um melhor caminho possível pelo terreno de uma posição a outra. Neste problema, o trabalho total realizado por uma força aplicada pelo veículo para se

mover de um ponto a outro por um caminho deve ser minimizado. O *trabalho* realizado por uma força durante um deslocamento elementar é a quantidade escalar correspondente ao deslocamento multiplicado pela força. O veículo pode se mover em diferentes tipos de terreno e o mapa divide o terreno em regiões segundo um índice de resistência ao movimento (coeficiente de atrito cinético) e uma inclinação constante relativa ao plano horizontal. Dessa forma, consideramos o terreno como uma superfície poliédrica.

A definição de uma função distância de caminhos mínimos implica na necessidade de resolver o problema de caminhos mínimos em superfície poliédrica. Os caminhos mínimos representam a noção de geodésicas em uma superfície poliédrica. Dessa forma, estendemos problemas de caminhos mínimos [95, 97] para considerar atrito, inclinação e direção de movimento através da metodologia Dijkstra contínuo.

Um problema em geometria computacional cuja função objetivo usa uma função distância pode ser generalizado para considerar o caso no qual as distâncias devem ser medidas de acordo com o comprimento dos caminhos mínimos entre pontos. Geralmente, os problemas de proximidade assumem esta característica. Generalizamos alguns destes problemas e determinamos estruturas geométricas imersas em uma superfície poliédrica.

Resolvemos problemas usuais da geometria computacional (par mais próximo, todos vizinhos mais próximos, árvore geradora mínima) considerando uma superfície poliédrica como espaço e uma *função distância de caminhos mínimos*. Nesta função distância, as distâncias são medidas segundo o comprimento de um caminho mínimo em uma superfície poliédrica que considera o atrito cinético e a inclinação de cada face e aresta. Para isso, estudamos problemas de caminhos mínimos em superfícies poliédricas e suas extensões para problemas de proximidade através do diagrama de Voronoi de caminhos mínimos.

1.1 Motivação e Aplicações

Uma razão teórica importante para considerar o problema direcional é que este problema generaliza vários problemas de caminhos mínimos encontrados na literatura. Dessa forma, estendemos a teoria de caminhos mínimos para incluir este caso. Além disso, a versão de múltiplas origens deste problema implica em um diagrama de Voronoi generalizado que resolve vários problemas de proximidade.

Existem muitas aplicações motivando problemas de planejamento de caminho mínimo no campo da navegação em terrenos (*terrain navigation*), robótica (*robot motion*), automação industrial, em sistemas de informação geográfica (SIG) e no planejamento otimizado de projetos em engenharia.

Um usuário de um SIG pode estar interessado em projetar uma nova rodovia entre duas cidades em uma região montanhosa. No planejamento de rodovias em montanhas, o custo de um caminho depende mais do trabalho total realizado do que do comprimento

euclideano.

1.2 Definições Preliminares

Um polígono é *simples* se duas arestas não se interceptam, exceto quando adjacentes. Um polígono simples é *monotônico* se existe uma direção tal que qualquer linha perpendicular a esta direção intercepta o polígono em no máximo dois pontos. Um polígono é *estrelado* se existe um ponto não exterior ao polígono tal que toda reta passando por este ponto intercepta o polígono em exatamente dois pontos.

Uma *subdivisão planar* conexa S é uma partição do plano em polígonos simples de interiores disjuntos denominados *regiões* de S . Uma subdivisão planar S tem uma região ilimitada, chamada *região externa*. Uma subdivisão S é representada por um grafo planar imerso no plano tal que as arestas são segmentos de reta.

Uma *triangulação* é uma subdivisão planar onde todas as regiões limitadas, denominadas *faces*, são triângulos. Uma triangulação de um conjunto finito Q de pontos é um grafo planar em Q com o número máximo de arestas. Uma triangulação de um polígono simples é uma decomposição do polígono em triângulos tal que dois triângulos quaisquer se interceptam apenas em um vértice ou uma aresta comum. Uma triangulação de um polígono simples pode ser computada em tempo linear [22]. A *envoltória convexa* de um conjunto Q de pontos é o menor conjunto convexo que o contém.

Duas faces f e f' são *aresta-adjacentes* se compartilham uma aresta e . Uma *seqüência de faces aresta-adjacentes* é uma seqüência $(f_1, f_2, \dots, f_{k+1})$, de duas ou mais faces, tal que a face f_i é aresta-adjacente à face f_{i+1} compartilhando a aresta e_i (veja Fig. 1.1). A *seqüência de arestas* associada a esta seqüência de faces aresta-adjacentes é a seqüência, $E = (e_1, e_2, \dots, e_k)$, das arestas $e_i = f_i \cap f_{i+1}$. A *raiz* r de E é o vértice da face f_1 que não pertence a e_1 . Uma seqüência de faces (arestas) é chamada *simples* se nenhuma face (aresta) aparece mais que uma vez na seqüência.

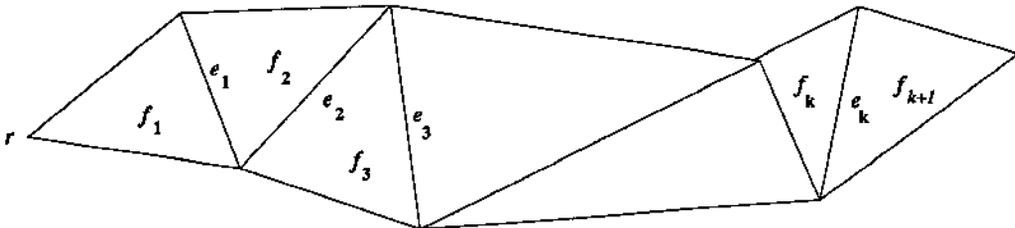


Figura 1.1: Uma seqüência de faces aresta-adjacentes.

Formalmente, um *caminho* de um ponto origem s a um ponto destino t é a imagem de

uma função Φ contínua do intervalo real $[0, 1]$ em um espaço F , com $\Phi(0) = s$ e $\Phi(1) = t$. Um *caminho mínimo* consiste em um caminho que tem comprimento mínimo dentre todos os possíveis caminhos de s a t . Podem existir muitos caminhos mínimos de um ponto a outro, mas o comprimento destes caminhos é único. A *distância geodésica* de s a t é o comprimento, denotado por $d_F(s, t)$, de um caminho mínimo no espaço F do ponto s ao ponto t .

Dois caminhos são *equivalentes* se as seqüências de arestas que estes caminhos interceptam são iguais. Esta relação particiona o conjunto de caminhos em classes de equivalência. Na figura 1.2, os caminhos p_1 e p_2 são equivalentes, mas p_3 não é equivalente a p_1 ou p_2 . A seqüência de arestas interceptadas por um caminho é uma estrutura combinatória do caminho e pode ser vista como uma descrição discreta do caminho.

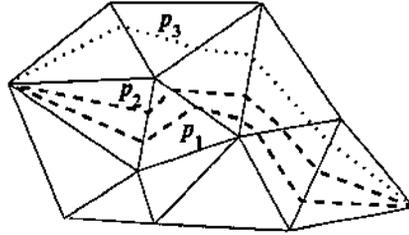


Figura 1.2: Classes de equivalência de caminho.

Um *caminho geodésico* é um caminho entre dois pontos que é localmente ótimo, ou seja, não pode ser encurtado por *pequenas perturbações* e preserva sua estrutura combinatória ao sofrer pequenas mudanças. Um *caminho mínimo* é um caminho geodésico que é globalmente ótimo. Caminhos geodésicos não são necessariamente caminhos mínimos, enquanto caminhos mínimos são sempre geodésicos. Caminhos mínimos não se interceptam, mas caminhos geodésicos e mínimos podem interceptar uma face várias vezes.

Um *caminho retilíneo* é sempre paralelo aos eixos x ou y e um polígono é *retilíneo* quando suas arestas são paralelas aos eixos x ou y . Caminhos mínimos envolvendo orientações fixas são denominados caminhos O -orientados [129]. Um *caminho O -orientado* está restrito a seguir as orientações (direções) de $|O|$ vetores de comprimento unitário.

Um *caminho $(1 + \varepsilon)$ -ótimo* é um caminho cujo comprimento é menor ou igual a $(1 + \varepsilon)$ vezes o comprimento do caminho mínimo (ótimo), onde $\varepsilon > 0$ é um grau de precisão real denominado *fator de erro relativo*.

Se s é um ponto e δ um real positivo, a *bola* aberta (fechada) com centro em s e raio δ é o conjunto de pontos t cuja distância de s a t é menor que (menor ou igual a) δ .

1.3 Organização da Dissertação

O capítulo 2 realiza um estudo sobre planejamento de caminho mínimo. Neste capítulo, discutimos a especificação de um problema de caminhos mínimos qualquer em termos do espaço livre, função distância e cardinalidade do conjunto de pontos origem e destino. Apresentamos estruturas geométricas usadas em algoritmos para resolver este problema. Introduzimos problemas de caminhos mínimos em espaços bidimensionais e descrevemos as abordagens utilizadas para resolvê-los, incluindo a metodologia Dijkstra contínuo.

O capítulo 3 aborda a caracterização de caminhos mínimos e geodésicos em uma superfície poliédrica segundo uma função distância que considera direção de movimento. Neste capítulo, especificamos o problema da geodésica atritada direcional (PGAD) que resolvemos; definimos a função distância atritada direcional (FGAD) e o critério de otimalidade local para caminhos geodésicos. O capítulo introduz os conceitos de caminho localmente f -livre e intervalo de otimalidade.

O capítulo 4 descreve o algoritmo Dijkstra contínuo direcional que resolve o problema PGAD. Neste capítulo, apresentamos as estruturas de dados, rotinas e funções numéricas usadas pelo algoritmo. O capítulo inclui a prova de corretude e a análise de complexidade do algoritmo.

O capítulo 5 apresenta conceitos básicos de problemas de proximidade. Neste capítulo, introduzimos o problema do par mais próximo, o problema de todos vizinhos mais próximos e o problema da árvore geradora mínima. Discutimos o diagrama de Voronoi euclideano e suas generalizações, incluindo o diagrama de Voronoi de caminhos mínimos.

O capítulo 6 considera o diagrama de Voronoi de caminhos mínimos em uma superfície poliédrica segundo a função distância FGAD. Neste capítulo, descrevemos o diagrama de Voronoi específico para esta função distância, desenvolvemos um algoritmo para sua construção e realizamos a redução dos problemas de proximidade ao da construção deste diagrama de Voronoi.

O capítulo 7 discute a implementação de um módulo externo ao visualizador *Geomview* que resolve o problema da geodésica discreta [95]. Neste capítulo, descrevemos o módulo externo e as ferramentas utilizadas em seu desenvolvimento. Apresentamos alguns melhoramentos ao algoritmo relacionados à sua corretude.

O capítulo 8 apresenta as conclusões e enumera os resultados obtidos. Neste capítulo, discutimos extensões e trabalhos futuros.

Capítulo 2

Planejamento de Caminho Mínimo

Um problema de planejamento de caminho mínimo aparece quando se deseja mover um corpo (robô autônomo ou articulado) em um espaço. O corpo deve se mover para o seu destino sem colidir com qualquer obstáculo através de um caminho que minimiza alguma noção conveniente de custo deste movimento. Um *obstáculo* é uma região conexa do espaço cujo interior é proibido para os caminhos.

Na seção 2.1, discutimos a especificação de um problema de caminhos mínimos qualquer. Introduzimos o conceito de espaço livre, apresentamos algumas representações para superfícies e definimos uma superfície poliédrica. Abordamos algumas métricas encontradas na literatura e a função distância de Minkowski. Além disso, esta seção considera as versões caso único, origem única e todos os pares de um problema de caminhos mínimos. Na seção 2.2, apresentamos estruturas geométricas usadas por algoritmos que resolvem problemas de caminhos mínimos. Introduzimos, dentre outras estruturas, o mapa de caminhos mínimos. A seção 2.3 discute problemas de caminhos mínimos no plano ou em uma superfície poliédrica. Nesta seção, introduzimos o problema da geodésica discreta e o problema da região valorada. Na seção 2.4, consideramos as abordagens usadas para resolver problemas de caminhos mínimos. Apresentamos heurísticas; algoritmos e esquemas aproximativos; e metodologias aplicadas em vários problemas. Introduzimos a metodologia Dijkstra contínuo utilizada para resolver o problema que abordamos.

2.1 Especificação de um Problema

Um problema de caminhos mínimos é especificado, basicamente, pelo espaço ambiente, pela função distância, e pela cardinalidade do conjunto de pontos origem e destino. A forma geométrica do corpo, os tipos de movimento (rotação, translação) e a variação temporal do ambiente (estático ou dinâmico) também são relevantes, mas tratamos apenas de corpos pontuais em ambientes estáticos.

A suposição de um objeto pontual é bastante razoável quando um corpo é pequeno demais comparado às dimensões do espaço. Além disso, muitos problemas que consideram outros objetos geométricos são reduzidos ao caso pontual quando aplicamos a seguinte abordagem do espaço de configuração [28, 59, 84, 106]: diminuimos o corpo a um *ponto de referência* e aumentamos os obstáculos (soma de Minkowski) até que a fronteira de cada obstáculo seja o conjunto de todas as posições do ponto de referência quando o corpo tangencia o obstáculo. Assim, o complemento dos obstáculos modela a região que contém as posições do ponto de referência para as quais não existe colisão entre o corpo e um obstáculo. Dessa forma, podemos resolver um problema de caminhos mínimos que evita obstáculos considerando um disco (objeto circular ou esférico) ou um corpo que pode transladar mas não pode rotacionar.

2.1.1 Espaço Livre

O espaço onde o corpo realmente se move é chamado *espaço de trabalho*. Assim, seja C um corpo se movendo em um espaço de trabalho na presença de um conjunto de obstáculos $O = \{o_1, \dots, o_h\}$. Um posicionamento do corpo C é denominado *configuração do corpo* C e pode ser especificado por um vetor de translação ou pelas coordenadas do ponto de referência do corpo. Dessa forma, o corpo transladado por um vetor V , denotado por $C(V)$, especifica que o corpo C está com seu ponto de referência em V . Geralmente, o ponto de referência não precisa estar dentro do corpo.

Se o corpo pode mudar sua direção por rotação ao redor de seu ponto de referência, um outro parâmetro, ω , especifica a orientação do corpo. Denotamos por $C(V, \omega)$ o corpo com ponto de referência em V e rotacionado por um ângulo ω . O espaço induzido pelo conjunto de parâmetros que especifica a configuração de um corpo C é chamado *espaço de configuração* $\mathcal{EC}(C)$. Um ponto q no espaço de configuração corresponde a um certo posicionamento $C(q)$ do corpo C no espaço de trabalho.

Quando um corpo pode transladar e rotacionar no plano, o espaço de configuração é tridimensional. Um ponto (x, y, ω) neste espaço corresponde ao posicionamento $C(x, y, \omega)$ no espaço de trabalho. O espaço de configuração não é o espaço euclidiano tridimensional, mas o espaço $\mathbb{R}^2 \times [0, 2\pi)$.

O espaço de configuração de um corpo que pode apenas transladar no plano é o plano euclidiano bidimensional e idêntico ao espaço de trabalho. Um corpo no espaço de trabalho é representado por um ponto no espaço de configuração e qualquer ponto no espaço de configuração corresponde a algum posicionamento de um corpo no espaço de trabalho.

Podemos especificar um posicionamento de um corpo através dos parâmetros, ou melhor, especificando um ponto no espaço de configuração. Contudo, pontos no espaço de configuração correspondendo a posicionamentos onde o corpo intercepta um dos obstácu-

los em O são proibidos. A parte do espaço de configuração consistindo nestes pontos é o *espaço proibido* $\mathcal{EC}_P(C, O)$. O *espaço livre* $\mathcal{EC}_L(C, O)$ consiste nos pontos onde o corpo não intercepta qualquer obstáculo, denominados *posicionamentos livres* (veja Fig. 2.1).

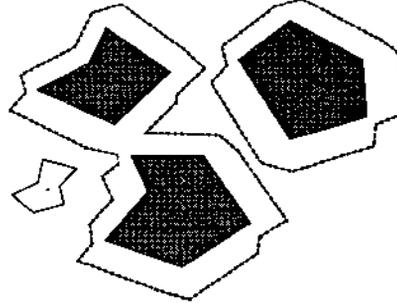


Figura 2.1: O espaço livre de um corpo.

Um caminho para o corpo corresponde a uma curva no espaço de configuração e vice-versa: cada posicionamento sobre o caminho corresponde a um ponto no espaço de configuração. Um caminho livre de colisões corresponde a uma curva no espaço livre (veja Fig. 2.2).

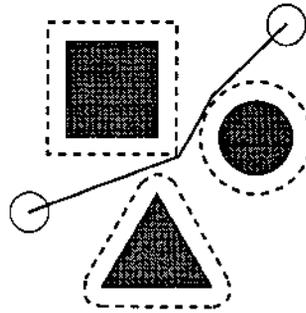


Figura 2.2: Um caminho possível para um círculo.

Um obstáculo o_i corresponde ao conjunto de pontos q no espaço de configuração tais que $C(q)$ intercepta o_i . O conjunto resultante é chamado *EC-obstáculo* de o_i . *EC-obstáculos* podem sobrepor-se mesmo quando os obstáculos no espaço de trabalho são disjuntos. Isto ocorre quando existem posicionamentos onde o corpo intercepta mais que um obstáculo.

Os espaços mais abordados na literatura são, dentre outros, o interior de um polígono simples, o plano com obstáculos poligonais simples disjuntos, a superfície de um poliedro convexo, a superfície de um poliedro geral e o espaço tridimensional com obstáculos poliédricos.

Superfície Poliédrica

Uma superfície pode ser representada por um conjunto de *curvas de nível* como em mapas geológicos ou por uma *matriz de elevação* que consiste em uma matriz de números que representam a altitude de cada ponto em uma grade regular.

Um *modelo de elevação digital* (MED) é uma forma de representar uma função real contínua linear por partes em duas variáveis definida sobre uma região limitada e conexa do plano. MEDs podem ser usados para representar densidades demográficas ou uma parte da superfície terrestre.

O exemplo mais comum de um MED é a *rede irregular triangulada* [107], onde um conjunto de pontos com coordenadas x e y é dado com uma triangulação destes pontos. Cada ponto tem uma altura ou elevação que é representada pela coordenada z . A altura dos pontos que não são dados na entrada é obtida quando localizamos o triângulo que o contém e computamos a interpolação linear das alturas dos vértices deste triângulo.

Uma *superfície poliédrica* S é um objeto geométrico conexo tridimensional que consiste em um conjunto de faces conectadas por arestas tal que uma reta ortogonal a um certo plano intercepta S em apenas um ponto (veja Fig. 2.3). Assumimos que este plano é um plano horizontal. Os aspectos adimensionais de uma superfície poliédrica são os *vértices*, seus aspectos unidimensionais são as *arestas* e os bidimensionais são as *faces*. Os vértices de S correspondentes a pontos na fronteira da pré-imagem de S sobre um plano horizontal são chamados *vértices fronteira*; os vértices restantes são chamados *vértices interiores*. A altura com relação ao plano horizontal de um ponto q em S é denotada por q_z . De forma semelhante, denotamos a abscissa (ordenada) do ponto q por q_x (q_y).

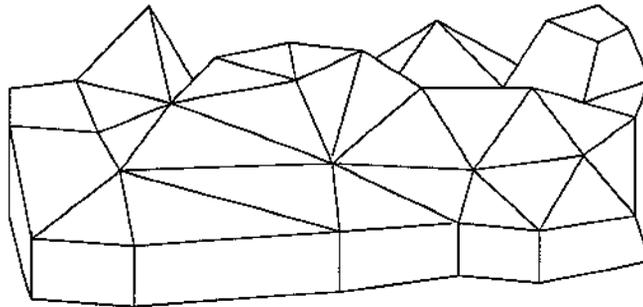


Figura 2.3: Uma superfície poliédrica.

Um vértice v é um *máximo local* (pico) se todos os vizinhos de v (vértices em arestas incidentes a v) possuem altura menor que v_z . Um vértice v é um *mínimo local* (fosso) se todos os vizinhos de v possuem altura maior que v_z . Um vértice que é um máximo ou mínimo local é um *extremo local*. Um vértice v interior à superfície poliédrica S será um *vértice sela* se e somente se existem quatro arestas incidentes a v cujos outros vértices destas arestas possuem altura alternadamente menor e maior que v_z quando visitadas em

ordem cíclica. Um vértice que é um extremo local ou um vértice sela é chamado *vértice especial*.

A *inclinação* de uma aresta é a diferença em altura de suas extremidades dividida pelo comprimento da projeção da aresta sobre o plano horizontal. A inclinação ou gradiente de uma face é a inclinação máxima de qualquer segmento de reta nesta face.

Assumimos que uma superfície poliédrica S é representada por uma estrutura topológica como a *winged-edge* [11], a *quad edge* [57], a *half-edge* [86], ou a DCEL [109]. Estas representações permitem o percurso da superfície com a utilização de tempo constante por aresta.

2.1.2 Função Distância

Uma *métrica* sobre um conjunto de pontos Q é uma função $d : Q \times Q \mapsto \mathbb{R}_+$ que, para quaisquer pontos $q_1, q_2, q_3 \in Q$, satisfaz as seguintes condições:

1. $d(q_1, q_2) = 0 \Leftrightarrow q_1 = q_2$;
2. simetria, $d(q_1, q_2) = d(q_2, q_1)$;
3. desigualdade triangular, $d(q_1, q_2) \leq d(q_1, q_3) + d(q_3, q_2)$.

Um par (Q, d) , onde d é uma métrica sobre Q , chama-se *espaço métrico*. Uma função que satisfaz as condições acima, exceto a de simetria, é denominada *função distância*. As métricas mais comuns em planejamento de caminho mínimo, dentre outras, são a euclideana, a retilínea, das dobras [56] e de orientações fixas.

A *métrica retilínea* entre dois pontos $q_1 = (q_{1x}, q_{1y})$ e $q_2 = (q_{2x}, q_{2y})$ no plano é definida como $d_R(q_1, q_2) = |q_{1x} - q_{2x}| + |q_{1y} - q_{2y}|$. As métricas retilínea e euclideana são generalizadas pela *métrica Minkowski* (L_p) com a L_p -distância no espaço d -dimensional entre dois pontos q_1 e q_2 sendo $d_p(q_1, q_2) = (\sum_{k=1}^d |q_{1k} - q_{2k}|^p)^{\frac{1}{p}}$, onde $p \geq 1$ é um real qualquer e q_{i1}, \dots, q_{ik} são as coordenadas do ponto q_i para $i = 1, 2$. Quando p tende ao infinito, a distância L_p tende a maior das diferenças das coordenadas ($L_\infty = \max_{k=1}^d |q_{1k} - q_{2k}|$).

Uma *isometria* é uma função de transformação que preserva distâncias e ângulos entre pontos. Quando resolvemos certos problemas que consideram a métrica L_1 , resolvemos simultaneamente estes problemas segundo a métrica L_∞ , pois existe isometria entre estas métricas [83]. Esta isometria consiste no seguinte mapeamento linear g do plano com métrica L_1 para o plano com métrica L_∞ : $g(x, y) = \left(\frac{x+y}{2}, \frac{y-x}{2}\right)$. Como g e g^{-1} são isometrias, os problemas na métrica L_∞ podem ser transformados em problemas na métrica L_1 pela função g^{-1} .

A *métrica das dobras* se refere ao número de dobras ou segmentos em um caminho entre a origem e o destino. Se os caminhos estão restritos a serem retilíneos ou O -orientados,

temos a métrica das dobras retilínea ou O -orientada [1], respectivamente. Em muitos problemas, a métrica das dobras proporciona uma medida mais natural para o caminho com aplicações na *simplificação de curvas* [56, 69, 99].

A *métrica de orientações fixas* define a distância entre dois pontos como o comprimento do caminho mínimo que percorre apenas sobre um dado conjunto O de orientações fixas.

Seja A um conjunto convexo cujo interior contém a origem dos eixos. O comprimento de um vetor V relativo a A é a quantidade que A deve ser escalado para incluir V : $\|V\|_A = \liminf\{\lambda \geq 0 | V \in \lambda A\}$. A *função distância convexa de Minkowski* [18] do ponto q_1 a q_2 é $\|q_2 \ominus q_1\|_A$ (veja Fig. 2.4). A função não precisa ser simétrica ($\|V\|_A$ pode não ser igual a $\|-V\|_A$), mas satisfaz a desigualdade de triângulo: se $V_1 = V_2 \oplus V_3$ então $\|V_1\|_A \leq \|V_2\|_A + \|V_3\|_A$.

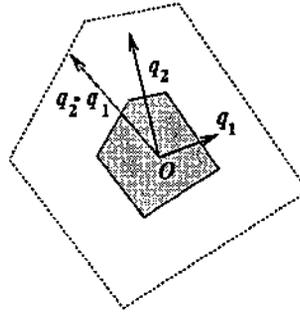


Figura 2.4: A função distância convexa de Minkowski.

Os vértices na fronteira de A são os vetores unitários da função distância $\|\cdot\|_A$. Quando A é o círculo unitário, temos a métrica euclideana; quando A é definido pelos quatro vetores unitários nas direções dos eixos, temos a métrica L_1 .

Para um caminho O -orientado, A é a envoltória convexa dos vetores unitários. Neste caso, a envoltória convexa é um polígono de $|O|$ lados que contém a origem em seu interior. Assim, qualquer vetor V pode ser expresso como uma combinação linear dos vetores unitários com coeficientes positivos. O comprimento $\|V\|_A$ é a soma mínima de coeficientes dentre todas as possíveis combinações lineares que são iguais a V de um ou dois vetores unitários adjacentes a V em ordem circular.

Um problema de caminhos mínimos sobre uma superfície poliédrica pode assumir a existência de forças peso, normal e atrito constantes em uma mesma face. Assim, o problema de caminhos mínimos que abordamos utiliza uma função distância que varia segundo o ângulo de direção do movimento. Esta função distância direcional representa a quantidade total de trabalho que deve ser realizado para se mover o corpo através do caminho.

2.1.3 Cardinalidade

Um problema de caminhos mínimos pode ter versões diferentes segundo a cardinalidade do conjunto de pontos origem e destino. A versão *caso único* (*single shot*) encontra um caminho mínimo entre dois pontos dados, uma origem e um destino fixos. A versão *origem única* (*single source*) constrói uma subdivisão do espaço tal que um caminho mínimo entre um ponto origem fixo e qualquer ponto destino possa ser reportado eficientemente. A versão *todos os pares* (*all pairs*) constrói uma subdivisão tal que o caminho mínimo entre qualquer par de pontos no espaço possa ser obtido eficientemente.

O algoritmo que propomos resolve a versão origem única do problema que estudamos. Contudo, utilizamos a versão todos os pares como uma extensão com múltiplas origens. Assim, construímos um diagrama de Voronoi generalizado que resolve alguns problemas de proximidade em superfície poliédrica.

2.2 Estruturas Geométricas

A teoria de problemas de caminhos mínimos utiliza algumas estruturas geométricas na resolução destes problemas. Nesta seção, apresentamos a grade retilínea, o grafo visibilidade, a árvore e o mapa de caminhos mínimos, o mapa de nível de altura, os conceitos de funil e triangulação geodésica.

2.2.1 Grade Retilínea

Uma *grade retilínea* em um conjunto finito Q de pontos no plano é o grafo obtido por todas as retas passando pelos pontos de Q que são paralelas aos eixos x ou y . Os nós deste grafo são posicionados nos pontos de interseção de duas retas e as arestas são os segmentos entre nós consecutivos em uma reta.

2.2.2 Grafo Visibilidade

Um *grafo visibilidade* para um conjunto finito Q de pontos relativo a um conjunto de obstáculos consiste em um grafo cujos nós são os pontos de Q e cujas arestas ligam um par de pontos visíveis e têm comprimento igual à distância entre os dois pontos (veja Fig. 2.5). Um par de pontos é visível relativo a métrica L_1 (L_2) quando um L_1 -segmento (L_2 -segmento) entre os dois pontos não intercepta o interior de qualquer obstáculo. Um L_1 -segmento será um caminho retilíneo cujo comprimento é igual à distância retilínea entre os dois pontos, enquanto um L_2 -segmento é um segmento de reta. Existem casos onde o número de arestas em um grafo visibilidade é $O(n^2)$, onde n é o número total

de pontos e vértices dos obstáculos. Portanto, um grafo visibilidade pode ocupar espaço quadrático.

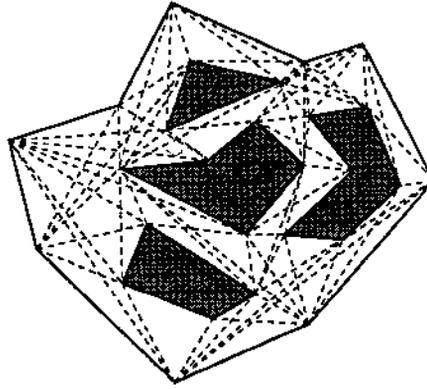


Figura 2.5: Um grafo visibilidade relativo um conjunto de obstáculos.

2.2.3 Árvore de Caminhos Mínimos

Uma *árvore de caminhos mínimos* T_s relativa a um ponto s no espaço livre F é a união dos caminhos mínimos de s a todos os vértices do espaço livre. Os nós correspondem aos vértices e as arestas correspondem a caminhos entre vértices, tais que o caminho poligonal, obtido pelo encadeamento dos nós (vértices) ao longo do caminho do nó s a v em T_s é um caminho mínimo. Sejam v_1 e v_2 dois vértices arbitrários em F , denotamos por $\tau_s(v_1, v_2)$ o *ancestral comum mais baixo* de v_1 e v_2 em T_s . Árvores de caminhos mínimos com origem em s não são únicas, pois podem existir mais de um caminho mínimo de um ponto a outro.

2.2.4 Mapa de Caminhos Mínimos

Um mapa de caminhos mínimos M_s relativo a um ponto s no espaço livre F é uma generalização da árvore de caminhos mínimos que permite responder consultas de caminhos mínimos de s a qualquer ponto t no espaço livre quando localizamos t no mapa [82].

Um *mapa de caminhos mínimos* M_s é um particionamento do espaço em células C_r , onde $r \in F$ é a raiz da célula C_r , segundo a estrutura combinatoria dos caminhos mínimos (veja Fig. 2.6). Assim, cada célula tem a seguinte propriedade: se $t \in C_r$, então um caminho mínimo de s a t é obtido por um caminho mínimo de s a r seguido por um caminho geodésico de r a t em C_r . Se a célula C_r é adjacente à célula $C_{r'}$, então a fronteira compartilhada por C_r e $C_{r'}$ é um subconjunto do bissetor geodésico $B(r, r')$ entre r e r' . O *bissetor geodésico* $B(r, r')$ é o lugar geométrico dos pontos a igual distância geodésica de r e r' .

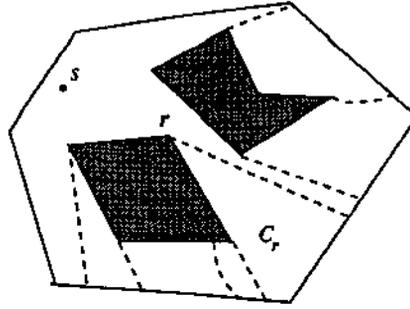


Figura 2.6: Um mapa de caminhos mínimos relativo a s .

O comprimento de um caminho mínimo de s a t pode ser encontrado em tempo $O(\log n)$ quando resolvemos um problema de localização [45, 72, 108] para determinar a célula C_r contendo t , onde n é o número de vértices do espaço livre. Um caminho mínimo de s a t pode ser traçado em tempo $O(\log n + k)$, onde k é o número de vértices no caminho mínimo. Um mapa de caminhos mínimos ocupa espaço linear, pois o tamanho de uma subdivisão planar é linear.

Um *particionamento em janela* de um polígono simples relativo a um ponto s é um mapa de caminhos mínimos para a métrica das dobras. As células do particionamento são tais que a distância de dobras de s para pontos de uma célula é constante. Os pontos que são visíveis a s constituem a célula J_s dos pontos com distância de dobras 1. As células com distância de dobras 2 consistem em regiões no complemento de J_s que são visíveis das janelas de J_s , onde uma *janela* de J_s é uma parte da fronteira de J_s .

2.2.5 Mapa de Nível de Altura

O v_z -*mapa* de uma superfície poliédrica S é a interseção de S com o plano horizontal $z = v_z$, onde v_z é a altura de um vértice v . O v_z -*mapa* consiste no conjunto de pontos da superfície que podem ser alcançados a partir de v por um caminho de altura constante. Os pontos compartilhados por dois ou mais ciclos ou caminhos em um v_z -*mapa* devem ser vértices sela, pois os pontos limitados por estes ciclos ou caminhos estão alternadamente acima e abaixo do vértice compartilhado. Seja v um vértice sela de S e considere o v_z -*mapa* de S , chamamos os ciclos e caminhos que contêm v de *componente principal* M_v de v . O *mapa de nível de altura* [38] é a subdivisão planar que resulta da projeção de todas as componentes principais de vértices sela e extremos locais de S sobre o plano (veja Fig. 2.7).

Visto que um v_z -*mapa* tem tamanho linear, temos que a complexidade do mapa de nível de altura pode ser quadrática. Contudo, uma estrutura de dados que representa *implicitamente* o mapa de nível de altura pode ser construída em tempo $O(n \log n)$, usando

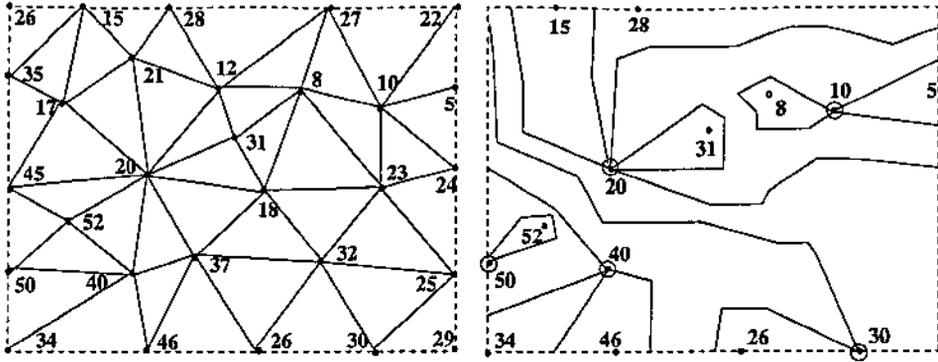


Figura 2.7: Uma superfície poliédrica e seu mapa de nível de altura [38].

espaço linear [38].

O grafo direcionado G_S cujos nós correspondem aos vértices sela (componentes principais) e extremos locais de S ; e cujo arco direcionado (v_1, v_2) existe se e somente se v_1 e v_2 estão na fronteira de uma região no mapa de nível de altura com $v_{1z} > v_{2z}$ (veja Fig. 2.8). A versão não-direcionada T_S de G_S é uma árvore cujo grau de um nó v em G_S é o número de regiões incidentes ao vértice sela v . As componentes principais são os nós internos e os extremos locais correspondem às folhas de G_S . Além disso, toda região do mapa de nível de altura corresponde a um arco de G_S e vice-versa.

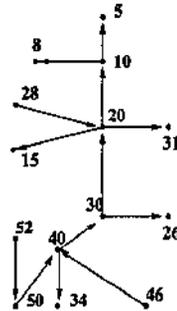


Figura 2.8: O grafo direcionado G_S [38].

O mapa de nível de altura tem as seguintes propriedades [38]:

1. Toda região é incidente a exatamente dois vértices especiais que aparecem em componentes diferentes da fronteira desta região. Um extremo local também é considerado como uma fronteira da região em que está.
2. Dois pontos s e t em uma superfície poliédrica S podem ser conectados por um caminho com uma altura constante se e somente se s e t estão na mesma região com $s_z = t_z$.

3. Todos os pontos em uma região possuem altura entre a altura da fronteira mais alta e da fronteira mais baixa. Todo caminho de altura monotonicamente crescente (decrecente) de qualquer ponto em uma região eventualmente alcança a fronteira mais alta (baixa) desta região.
4. Se dois pontos s e t em uma superfície poliédrica S estão em regiões correspondentes aos arcos a_s e a_t em G_S , respectivamente, então todo caminho entre s e t deve cruzar as componentes principais correspondentes aos nós entre a_s e a_t em G_S .
5. Se os nós em G_S entre a_s e a_t são (v_1, \dots, v_k) , então existe um caminho com diferença de altura total $|s_z - v_{1z} + \sum_{i=1}^{k-1} |v_{iz} - v_{i+1z}| + |v_{kz} - t_z|$.

2.2.6 Funil

Considere um vértice $s \in F$ e dois caminhos mínimos $p(s, v_1)$ e $p(s, v_2)$. Estes dois caminhos podem coincidir por alguma distância antes que divirjam em um vértice r para nunca mais se encontrarem. O *funil* $F_{v_1, v_2}(s)$ de s relativo aos vértices v_1 e v_2 é o conjunto das arestas nos caminhos mínimos de r , chamado *ápice do funil*, aos vértices $\tau_{v_1}(s, v_2)$ e $\tau_{v_2}(s, v_1)$ [60]: $F_{v_1, v_2}(s) = p(r, \tau_{v_1}(s, v_2)) \cup p(r, \tau_{v_2}(s, v_1))$ (veja Fig. 2.9).

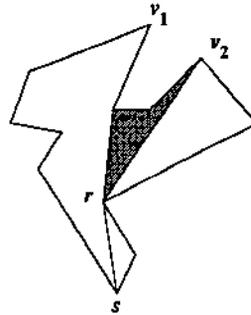


Figura 2.9: Um funil para o vértice s .

A *diferença dos funis* de dois vértices $s, t \in F$ é definida como $\Lambda F_{v_1, v_2}(s, t) = F_{v_1, v_2}(s) - F_{v_1, v_2}(t)$, ou seja, o conjunto de arestas que estão em $F_{v_1, v_2}(s)$ mas não estão em $F_{v_1, v_2}(t)$. O par de funis $(F_{v_1, v_2}(s), F_{v_1, v_2}(t))$ é *fechado* quando os caminhos $p(\tau_{v_1}(s, v_2), \tau_{v_2}(s, v_1))$ e $p(\tau_{v_1}(t, v_2), \tau_{v_2}(t, v_1))$ são disjuntos, exceto nas extremidades, e *aberto* do contrário.

2.2.7 Triangulação Geodésica

Uma *pipa* é o objeto geométrico formado pela união de três caminhos mínimos correspondendo a $p(v_1, v_2)$, $p(v_2, v_3)$ e $p(v_3, v_1)$, onde $v_1, v_2, v_3 \in F$ são três vértices do espaço livre [24, 32]. Geralmente, uma pipa consiste em um triângulo geodésico, um polígono

simples formado de três cadeias; e três *caudas*, curvas lineares por partes emanando dos três vértices onde as cadeias se interceptam (veja Fig. 2.10).

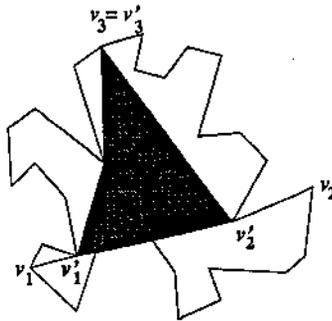


Figura 2.10: Uma pipa relativa aos vértices v_1, v_2 e v_3 .

Formalmente, a cauda (v_1, v_1') emanando do vértice v_1 é o subcaminho comum maximal dos caminhos mínimos $p(v_1, v_2)$ e $p(v_1, v_3)$. As caudas (v_2, v_2') e (v_3, v_3') são definidas de forma semelhante. A região limitada pelas três cadeias ligando v_1', v_2' e v_3' é o *triângulo geodésico* (v_1', v_2', v_3') . As cadeias são subcaminhos dos caminhos geodésicos originais. Os pontos v_1', v_2' e v_3' são chamados *ápices* do triângulo geodésico (veja Fig. 2.10).

Uma *triangulação geodésica* de F é uma decomposição de F em triângulos geodésicos cujas fronteiras não se cruzam. Uma triangulação geodésica é combinatorialmente e topologicamente como uma triangulação planar. Portanto, isto induz à *árvore de decomposição geodésica* T de grau 3, onde cada nó em T corresponde a um triângulo geodésico e ligamos dois nós se os triângulos correspondentes compartilham dois de seus vértices. Esta triangulação é obtida quando computamos uma hierarquia de caminhos mínimos entre pares de vértices.

2.3 Problemas Bidimensionais

Apresentamos alguns problemas de caminhos mínimos em duas dimensões. Além disso, resumimos o estado da arte em algoritmos eficientes que encontram soluções para estes problemas. Os problemas estão agrupados segundo o espaço que consideram.

A versão tridimensional do problema de caminhos mínimos não é discreta, pois um caminho mínimo pode interceptar um conjunto contínuo de pontos interiores às arestas de um poliedro. Este problema é *NP*-difícil [17], conhecendo-se apenas algoritmos exatos de tempo exponencial [16, 113, 120]. Dada a seqüência de arestas dos obstáculos pelas quais um caminho mínimo passa, o cálculo dos pontos de interseção entre o caminho e estas arestas requer a solução de equações algébricas de alto grau. Isto deve ser realizado por métodos numéricos aproximativos ou por cálculos algébricos simbólicos. Além disso, existem apenas algoritmos de tempo duplamente exponencial para o subproblema que

encontra a seqüência de arestas dos obstáculos pelas quais o caminho mínimo passa [16, 113, 120].

2.3.1 Interior de um Polígono Simples

Shamos [117] considerou o problema de caminhos mínimos entre dois pontos no interior de um polígono simples (PIPS) (veja Fig. 2.11). Neste caso, existe um único caminho mínimo entre dois pontos, pois o interior do polígono é simplesmente conexo.

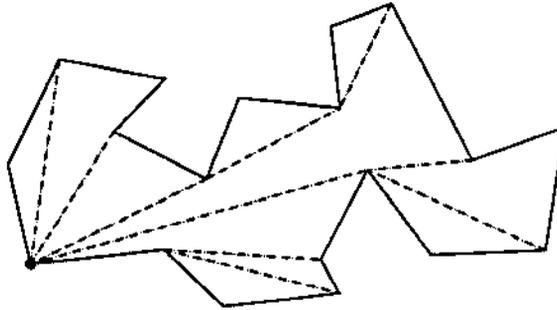


Figura 2.11: Caminhos mínimos no interior de um polígono simples.

Guibas e Hershberger [54, 58] apresentaram uma estrutura de dados para computar distâncias de caminhos mínimos entre dois pontos arbitrários no interior de um polígono simples em tempo $O(\log n)$, depois de um pré-processamento que depende tempo e espaço $\Theta(n)$, onde n denota o número de vértices do polígono. Esta estrutura de dados atribui um rótulo a cada aresta de uma triangulação e permite computar o caminho mínimo entre dois vértices do polígono em tempo $O(h^2)$, onde h é o maior rótulo de uma aresta interceptando o caminho mínimo. A estrutura de dados computa uma representação compacta do caminho mínimo da qual o caminho mínimo pode ser derivado em tempo proporcional ao número de arestas no caminho. Guibas *et al.* [55] mostraram como computar a árvore de caminhos mínimos de um vértice do polígono a todos os outros em tempo linear usando funis.

Goodrich e Tamassia [52] apresentaram uma estrutura de dados dinâmica para subdivisões planares conexas que suporta consultas de caminhos mínimos. As operações de atualização são inclusão e exclusão de vértices e arestas e são completas para subdivisões conexas. A estrutura ocupa espaço linear e depende tempo $O(\log^2 n)$ para atualizações, onde n é o tamanho atual da subdivisão.

Ghosh [51] e Suri [122] consideraram o problema de encontrar um caminho entre dois pontos com um número mínimo de dobras no interior de um polígono simples e desenvolveram algoritmos ótimos de tempo linear. Estes algoritmos computam um particionamento em janelas do polígono relativo a um ponto s e são baseados no algoritmo para árvore

de caminhos mínimos de Guibas *et al.* [55]. Hershberger e Snoeyink [60] apresentaram algoritmos para caminhos O -orientados em um polígono simples.

De Berg [37] estudou o problema da distância de dobras retilínea em um polígono simples sem buracos. McDonald e Peters [87] apresentaram um algoritmo de tempo linear para encontrar o caminho mínimo tanto em comprimento euclidiano como em distância de dobras em um polígono simples.

2.3.2 Plano com Obstáculos Poligonais

O problema de caminhos mínimos euclidiano com obstáculos poligonais (POPD) consiste em computar caminhos mínimos de um ponto dado, a *origem*, para todos os outros pontos no plano, que evitam um conjunto de obstáculos poligonais simples disjuntos com n vértices no total (veja Fig. 2.12). Este problema generaliza o de caminhos mínimos no interior de um polígono simples quando consideramos o complemento do polígono como um obstáculo.

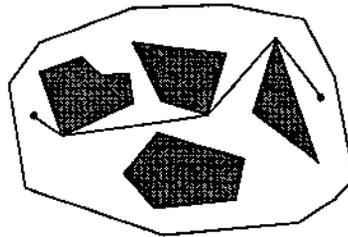


Figura 2.12: Caminhos mínimos no plano com obstáculos.

Um caminho mínimo entre dois pontos no plano evitando uma coleção de obstáculos poligonais deve ser uma linha poligonal cujos vértices são vértices dos obstáculos poligonais. Além disso, sua interseção com as fronteiras dos obstáculos são arcos convexos dos obstáculos, onde um *arco convexo* de um obstáculo é um subconjunto conexo fechado da fronteira do obstáculo tal que um segmento de reta entre dois pontos do arco intercepta o interior do obstáculo.

O algoritmo mais eficiente conhecido para resolver este problema foi dado por Hershberger e Suri [61] que apresentaram um algoritmo de tempo $O(n \log n)$ e espaço $O(n \log n)$. Este algoritmo está próximo às cotas inferiores de tempo $\Omega(n + h \log h)$ e espaço $\Omega(n)$, onde h é o número de obstáculos.

Métodos que constroem um grafo visibilidade e encontram um caminho mínimo usando o algoritmo de Dijkstra [42] resolvem o problema em tempo $O(n^2)$ [4, 5, 127]. Estes métodos assumem que se existe um caminho mínimo entre dois pontos, então existe um caminho mínimo correspondente no grafo visibilidade. Além disso, utilizam o fato que n

ordenações circulares não são independentes e podem ser realizadas em tempo total $O(n^2)$ [79].

A abordagem de varredura linear para o projeto de algoritmos em geometria computacional realiza uma simulação discreta de uma varredura linear sobre o plano, encontrando objetos e atualizando uma estrutura que representa o estado da varredura.

O caso especial onde os obstáculos são segmentos de reta paralelos (POSGRP) pode ser resolvido em tempo $\Theta(n \log n)$ pela construção do mapa de caminhos mínimos [82]. O algoritmo explora a monotonicidade de caminhos mínimos na direção perpendicular a dos segmentos e realiza uma varredura linear nesta direção. Tompa [123] estudou uma versão mais simples, onde os segmentos são substituídos por semi-retas (POSMRP).

A monotonicidade de caminhos mínimos ocorre também no caso onde todos os obstáculos poligonais possuem projeções disjuntas em uma certa direção (veja Fig. 2.13). Neste caso, caminhos mínimos são monotônicos nesta direção, perdendo a monotonicidade apenas nas extremidades do caminho para ultrapassar um obstáculo cuja projeção contém a projeção da origem ou do destino. O mapa de caminhos mínimos é construído em tempo $O(n \log n)$, mas a reta sobre a qual os obstáculos têm projeções disjuntas é encontrada em tempo $O(n^2 \log n)$ [90].

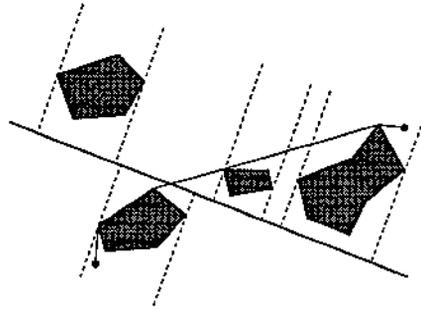


Figura 2.13: Caminhos mínimos para obstáculos com projeções disjuntas.

Quando os obstáculos poligonais simples possuem envoltórias convexas disjuntas, o problema (POECD) pode ser resolvido em tempo $O(h^2 + k \log k)$ usando um grafo visibilidade com $O(h^2)$ arestas e $k = O(\min\{n, h^2\})$ nós, onde h é o número de obstáculos. A construção do grafo visibilidade depende tempo $O\left(h^2 \log\left(\frac{n}{h}\right) + n\right)$ e espaço $O(h^2)$ [115].

Clarkson *et al.* [33], Mitchell [93] e Widmayer [128] apresentaram algoritmos de tempo $O(n \log^2 n)$ e espaço $O(n \log n)$ para determinar um caminho de comprimento L_1 mínimo de uma origem s a qualquer ponto consulta t que evita os interiores de um conjunto de obstáculos poligonais simples disjuntos determinados por um conjunto de n vértices. O caminho de s a t pode ser uma linha poligonal arbitrária, mas o comprimento de cada segmento é de acordo com a métrica L_1 . Clarkson *et al.* e Widmayer usaram uma técnica que envolve computar um grafo esparso que preserva caminhos mínimos. Mitchell

usou o método Dijkstra contínuo¹ para construir o mapa de caminhos mínimos através de consultas de arrastamento de segmento [20, 21]. O algoritmo de Mitchell pode ser generalizado para resolver o problema envolvendo distâncias com k orientações fixas em tempo $O(kn \log^2 n)$ e espaço $O(kn \log n)$.

Quando os obstáculos são polígonos disjuntos, a abordagem de grafo visibilidade resolve o problema de encontrar um caminho retilíneo mínimo de um ponto origem s a um ponto consulta t no plano em tempo $O(n^2)$ [77, 92]. O caso especial no qual todos os obstáculos são retângulos foi resolvido em tempo $\Theta(n \log n)$ [40] através de varredura planar quando se explora a monotonicidade de tais caminhos mínimos em uma direção paralela a um dos eixos (veja Fig. 2.14). Mitchell [92] apresentou um algoritmo $\Theta(n \log n)$ para o problema de caminhos mínimos retilíneos quando os obstáculos são polígonos retilíneos. Sack [116] também estudou o problema de computar caminhos mínimos retilíneos entre obstáculos retilíneos.

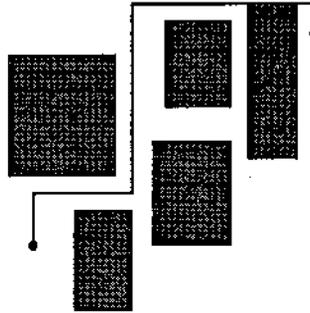


Figura 2.14: Um caminho mínimo retilíneo evitando obstáculos retangulares.

Mitchell *et al.* [98] apresentaram um algoritmo para computar distâncias de dobras não-retilíneas evitando múltiplos polígonos em tempo $O(n^2 \alpha^2(n) \log n)$ e espaço $O(n^2)$, onde $\alpha(n)$ é o inverso da *função de Ackermann*.

O problema de encontrar um caminho retilíneo com um número mínimo de dobras ou segmentos entre um ponto s e um ponto consulta t que evita uma coleção de obstáculos retilíneos disjuntos se reduz a duas buscas em largura em um grafo interseção de $O(n)$ segmentos de reta ortogonais. Um *grafo interseção* tem cada nó associado a um conjunto, enquanto suas arestas representam a existência de interseção entre dois conjuntos. Dessa forma, a estrutura de dados de Imai e Asano [67, 68] resolve este problema em tempo $\Theta(n \log n)$ e espaço $O(n \log n)$, enquanto Das e Narasimhan [36] resolvem o problema em tempo $\Theta(n \log n)$ e espaço linear.

De Berg *et al.* [39] estudaram o problema de encontrar caminhos retilíneos evitando n segmentos de reta paralelos aos eixos com uma métrica que generaliza a métrica L_1 e a métrica das dobras. Neste caso, o comprimento de um caminho é o seu comprimento na

¹O método Dijkstra contínuo é apresentado na subseção 2.4.4.

métrica L_1 mais uma constante não-negativa vezes o número de dobras no caminho. O algoritmo apresentado constrói uma estrutura em tempo $O(n^2)$ que usa espaço $O(n \log n)$ através de um grafo visibilidade retilíneo.

2.3.3 Superfície de um Poliedro Convexo

O problema de encontrar um caminho mínimo sobre a superfície S de um poliedro convexo (PSPC) foi resolvido por Mount [100] em tempo $O(n^2 \log n)$ e espaço $O(n \log n)$, onde n é o número de vértices do poliedro. Este problema foi sugerido originalmente por Dudeney em 1903 [48].

Um ponto $q \in S$ é denominado *ponto de saliência* se existem pelo menos dois caminhos mínimos em S da origem para q . Um caminho mínimo sobre a superfície de um poliedro convexo não intercepta vértices ou pontos de saliência do poliedro, e não cruza uma aresta ou face do poliedro mais que uma vez [120]. Além disso, exceto no caso onde compartilham um subcaminho comum, dois caminhos mínimos se interceptam em no máximo um ponto e esta interseção é transversal (os caminhos cruzam um ao outro). Se dois caminhos mínimos compartilham um subcaminho comum, então um caminho é um subcaminho do outro ou um subcaminho inicial de um coincide com um subcaminho final do outro.

Geralmente, os algoritmos realizam um pré-processamento que constrói um mapa de caminhos mínimos sobre a superfície do poliedro tal que qualquer ponto na mesma região do mapa tem um caminho mínimo que intercepta a mesma seqüência de arestas. Esta seqüência de arestas tem a seguinte propriedade de *desdobramento*: desdobramos as faces, f_1, f_2, \dots, f_k , que contêm as arestas interceptadas pelo caminho mínimo ($e_i = f_i \cap f_{i+1}$) até que todas estejam coplanares. Para isso, rotacionamos f_{i+1} em torno de e_i até que se torne coplanar a f_i , mas em lados diferentes de e_i . Dessa forma, o caminho mínimo se torna um segmento de reta. Os pontos de interseção deste segmento com as arestas desdobradas são determinados e transformados inversamente em pontos na superfície poliédrica.

Hwang *et al.* [66] apresentaram um algoritmo de tempo $O(n^6 \log n)$ que constrói uma subdivisão do poliedro para o problema de caminhos mínimos de todos os pares. O algoritmo pode reportar apenas os caminhos mínimos entre qualquer par de pontos nas arestas do poliedro. Um caminho mínimo entre qualquer par de pontos nas arestas do poliedro pode ser obtido em tempo $O(k + \log n)$, onde k é o número de arestas cruzadas pelo caminho mínimo.

2.3.4 Problema da Geodésica Discreta

Uma generalização do problema de caminhos mínimos evitando obstáculos no plano consiste em computar caminhos mínimos na métrica euclidiana sobre a superfície de um poliedro (não necessariamente convexo) de um ponto s para qualquer ponto t . Para isso,

considere uma superfície onde cada obstáculo se torna um prisma ortogonal infinito cuja base está no plano. Um caminho mínimo nesta superfície de um ponto no plano a outro está contido totalmente no plano e evita todos os prismas, implicando imediatamente em um caminho mínimo que evita obstáculos. Este problema é denominado *problema da geodésica discreta* (PGD), pois a superfície é modelada de forma discreta [95].

Um caminho mínimo sobre uma superfície poliédrica, possivelmente não convexa, se desdobra em um segmento de reta, não se intercepta e pode passar por qualquer face apenas uma vez. Um caminho que não se intercepta é denominado *simples*.

Chen e Han [26] apresentaram algoritmos exatos de tempo $O(n^2)$. Para uma origem fixa, o algoritmo obtém um mapa de caminhos mínimos tal que o comprimento de um caminho mínimo para um ponto consulta pode ser encontrado em tempo $O(\log n)$ e a enumeração deste caminho pode ser feita em tempo proporcional ao número de arestas que o caminho intercepta.

2.3.5 Problema da Região Valorada

O *problema da região valorada* (PRV) consiste em encontrar um caminho em uma subdivisão planar que minimiza o custo total segundo uma *métrica euclidiana valorada* (veja Fig. 2.15). A distância valorada entre dois pontos s e t na região f é o produto $\mu_f |st|$, onde $\mu_f \in [0, \infty]$ é um custo associado à região f que representa o custo unitário de se percorrer aquela região e $|st|$ é a distância euclidiana entre s e t .

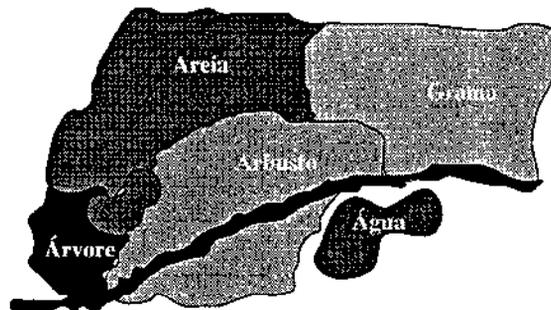


Figura 2.15: O problema da região valorada.

O problema da região valorada generaliza o problema de caminhos mínimos evitando obstáculos no plano. Neste caso, os custos associados ao espaço livre e aos obstáculos são 1 e $+\infty$, respectivamente.

Mitchell e Papadimitriou [97] usaram um algoritmo baseado na metodologia Dijkstra contínuo para encontrar um caminho $(1 + \varepsilon)$ -ótimo. A complexidade de tempo do algoritmo é $O(n^8 L)$, onde $L = \log \left(\frac{nNW}{\varepsilon} \right)$ é a complexidade binária da instância do problema,

n é o número de vértices da subdivisão, N é a maior coordenada inteira de qualquer vértice, e W é o maior custo associado a qualquer face.

Um caso especial do problema da região valorada aparece quando os custos das regiões são 0, 1 ou $+\infty$. Regiões de custo infinito são obstáculos, de custo unitário são espaços livres e de custo nulo são regiões onde se mover não custa nada. Uma aplicação deste caso está no *problema do esconderijo máximo* (PEM) onde em um espaço com obstáculos e regiões visíveis ou escondidas a um conjunto de inimigos, procuramos um caminho entre dois pontos evitando obstáculos que minimiza o comprimento em regiões expostas a um observador inimigo. Gewali *et al.* [50] e Mitchell [91] resolveram este problema em tempo $O(n^2)$ usando um grafo de visibilidade estendido.

Um caminho que evita obstáculos considerando regiões com custo nulo deve ser localmente tangente aos obstáculos formando *ângulos convexos*. Além disso, um caminho entra em uma região de custo nulo em um vértice (com ângulo maior que $\frac{\pi}{2}$ entre as arestas incidentes ao vértice) ou como um segmento normal ao interior de uma aresta (veja Fig. 2.16).

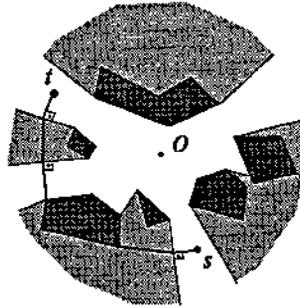


Figura 2.16: O problema do esconderijo máximo.

Quando a subdivisão valorada é retilínea e o comprimento do caminho é medido segundo a *métrica retilínea valorada*, há algoritmos eficientes baseados em grafos que preservam caminhos mínimos [25].

2.3.6 Restrições de Altura

Algumas maneiras para definir o custo de um caminho segundo restrições de altura são relativas a altura máxima do caminho, a diferença de altura total e a inclinação máxima do caminho. Enumeramos alguns problemas envolvendo caminhos mínimos que consideram restrições de altura. Sejam s e t pontos em uma superfície poliédrica:

1. Existe um caminho entre s e t que esteja abaixo de uma altura z ?
2. Existe um caminho entre s e t que esteja em uma altura constante?

3. Existe um caminho entre s e t cuja altura seja monotonicamente decrescente?
4. Qual a diferença de altura mínima total de qualquer caminho entre s e t ?

De Berg e van Kreveld [38] definiram o conceito de mapa de nível de altura que possui propriedades relacionadas a caminhos com restrições de altura. Assim, depois de construir esta estrutura em tempo $O(n \log n)$, onde n é o número de vértices da superfície poliédrica, os problemas de decisão são resolvidos em tempo de consulta $O(\log n)$. Van Kreveld [125] estudou problemas de caminhos mínimos abaixo de uma dada altura, evitando regiões com grande inclinação e minimizando a diferença total de altura.

Vários problemas de caminhos mínimos induzidos pela métrica euclideana são reduzidos ao problema que estudamos, denominado PGAD. Na figura 2.17, apresentamos uma hierarquia para estes problemas considerando a redução de um problema a outro mais geral. Dessa forma, um arco conecta um problema a outro se o primeiro pode ser reduzido ao segundo. Assim, podemos observar no diagrama abaixo que o PGAD resolve os demais problemas.

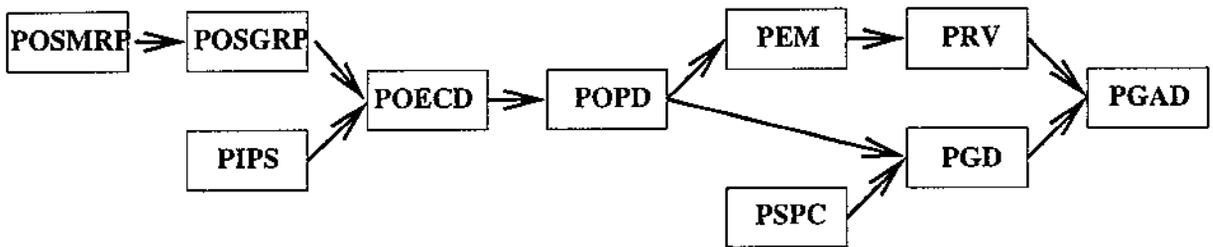


Figura 2.17: Hierarquia de problemas de caminhos mínimos bidimensionais.

O problema PGAD generaliza o problema da geodésica discreta quando existe apenas uma força constante em toda a superfície poliédrica atuando sobre o corpo. Neste caso, o trabalho total realizado para se mover um corpo consideraria apenas a força peso. Se a superfície poliédrica representa um plano, a única força que atua no corpo corresponde a uma força de atrito constante em cada face ou aresta, portanto o problema da região valorada é um caso especial do problema PGAD.

2.4 Abordagens Utilizadas

O planejamento de caminho mínimo utiliza abordagens heurísticas, aproximativas e exatas. Uma *heurística* encontra uma solução muito boa, mas não garante a otimalidade global. *Algoritmos aproximativos* computam caminhos cujo comprimento pode ser um fator constante maior que o comprimento do caminho mínimo. Um *esquema aproximativo* para um problema de otimização é um algoritmo aproximativo que toma como entrada

não apenas uma instância do problema, mas também um valor $\varepsilon > 0$ tal que para qualquer ε fixo, o esquema é um algoritmo aproximativo com erro relativo ε [35].

2.4.1 Heurísticas

Geralmente, uma heurística descarta partes do espaço onde nenhum caminho pode passar e, então, um algoritmo de caminhos mínimos exato pode ser usado no espaço resultante [9].

A heurística seguinte resolve o problema da geodésica discreta quando a superfície é representada por uma matriz de elevação. Se o espaço é representado por dados digitais, a heurística resolve o problema da região valorada.

Um caminho mínimo pode ser encontrado por um algoritmo de caminhos mínimos [42] sobre um *grafo grade* [70] onde as arestas ligam vértices de quadrados adjacentes (veja Fig. 2.18). O custo atribuído a cada aresta considera a média de custo dos dois quadrados vezes a distância euclidiana entre os vértices correspondentes. O algoritmo depende tempo $O(n \log n)$, onde n é o número de quadrados na grade. Keirse e Mitchell [71] discutem as desvantagens da digitalização e os possíveis tratamentos para isto. Quek *et al.* [111] apresentaram uma técnica que usa grafos grade de várias resoluções em um algoritmo hierárquico.

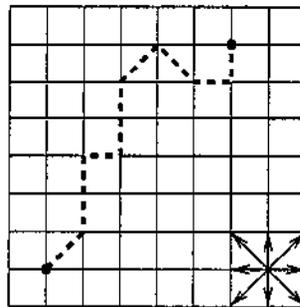


Figura 2.18: Um grafo grade.

Outra heurística para o problema da geodésica discreta encontra um caminho entre os pontos de origem e destino através da interseção da superfície com um plano vertical que contém estes pontos. A heurística melhora o caminho iterativamente aplicando o critério de otimalidade local que consiste em desdobrar o caminho em um segmento de reta. O *critério de otimalidade local* consiste na regra obedecida pelo caminho geodésico ao passar por uma aresta.

Para o problema da região valorada, a seguinte heurística usa um *grafo região* (veja Fig. 2.19). Suponha que as regiões de uma subdivisão planar são convexas. Os nós do grafo região correspondem às regiões e uma aresta liga dois nós se as regiões correspondentes

são adjacentes. O custo de cada aresta é atribuído segundo a distância valorada entre o centro das duas regiões. Um caminho mínimo é obtido neste grafo, resultando em uma seqüência de regiões interceptadas pelo caminho. Realizamos um pós-processamento, buscando tornar o caminho localmente ótimo na fronteira destas regiões.

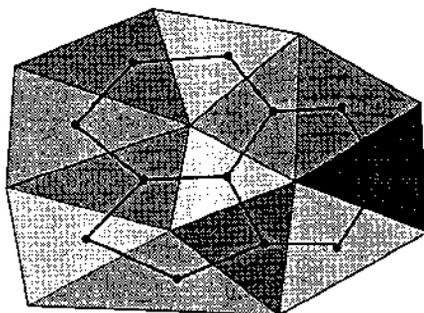


Figura 2.19: Grafo região de uma subdivisão planar.

Um método para computar um caminho em uma superfície poliédrica que evite regiões com grande inclinação consiste em ordenar o conjunto de todas as faces e arestas da superfície por inclinação. Inclua cada face ou aresta segundo esta ordem até que um caminho seja possível entre a origem e o destino sobre o subconjunto de faces incluídas. Verificar se um caminho existe pode ser feito em tempo linear utilizando busca em profundidade. Portanto, esta heurística despende tempo $O(n^2)$ [125]. Utilizando uma busca binária para a menor inclinação de uma face tal que o caminho exista sobre as faces com no máximo esta inclinação, a cota de tempo pode ser melhorada para $O(n \log n)$.

2.4.2 Algoritmos Aproximativos

A simplicidade e eficiência são as principais virtudes de algoritmos aproximativos. Geralmente, os algoritmos aproximativos transformam o espaço em grafos e computam uma aproximação do caminho mínimo no grafo ou utilizam projeções planares para representar o espaço, reduzindo o problema a algum outro problema de caminhos mínimos resolvido de forma exata.

Caminhos com diferença total de altura e comprimento euclidiano mínimos entre pontos s e t são computados utilizando um grafo G obtido de uma superfície poliédrica S estendida [125]. A superfície estendida inclui o conjunto de curvas de nível contendo todo vértice v da superfície, de forma que as novas arestas são horizontais e os novos vértices são pontos onde as novas arestas cruzam arestas de S . Além disso, adicionamos s e t como vértices para a superfície com curvas de nível.

O grafo G tem um nó para cada vértice na superfície estendida e um arco entre dois nós se os vértices correspondentes estão conectados por uma aresta. O número de arestas

e vértices é limitado por $O(n^2)$ e o grafo G pode ser construído em tempo linear no seu tamanho.

O custo de um arco é o comprimento euclidiano mais a diferença de altura multiplicada por uma constante grande o bastante para garantir que o caminho mínimo no grafo tem diferença de altura total mínima. O valor desta constante é o perímetro da projeção da superfície dividido pela menor diferença não-negativa de altura entre quaisquer dois vértices da superfície. Isto garante que cada subcaminho não-horizontal é mais custoso do que todos os subcaminhos horizontais em um caminho.

O caminho de custo mínimo encontrado no grafo valorado G usando qualquer algoritmo de caminhos mínimos tem diferença de altura mínima e consiste em uma aproximação do caminho mínimo euclidiano. O algoritmo despende tempo $O(n^2 \log n)$ e ocupa espaço $O(n^2)$.

Hershberger e Suri [62] propuseram um algoritmo linear para computar um caminho 2-ótimo entre dois pontos s e t na superfície de um poliedro convexo. O algoritmo encontra o menor caminho sobre a cunha bi-planar $W(H_s, H_t)$, onde H_s e H_t são os planos definidos pelas faces que contêm s e t ; ou sobre as cunhas tri-planares $W(H_s, H_i, H_t)$, onde H_i são planos de horizonte. Os *planos de horizonte* são planos verticais passando pelas arestas que formam a fronteira entre as faces positivas e negativas. As *faces positivas* (*negativas*) possuem produtos vetoriais positivos (negativos) entre a normal e o vetor unitário correspondendo ao eixo z positivo. Um caminho sobre a superfície de um poliedro convexo é obtido quando projetamos um caminho de uma cunha sobre este poliedro.

Caminhos mínimos quando projetados em um plano satisfazendo outra propriedade são denominados uma *xy-aproximação* para caminhos sobre uma superfície poliédrica que satisfazem esta propriedade. Uma *xy-aproximação* de um caminho mínimo entre dois pontos abaixo da altura z é encontrada em tempo $O(n \log n)$ [125]. Para isso, projetamos a fronteira das regiões acima de z sobre o plano, obtendo um conjunto de obstáculos, e computamos um caminho mínimo planar que evita estes obstáculos (veja Fig. 2.20).

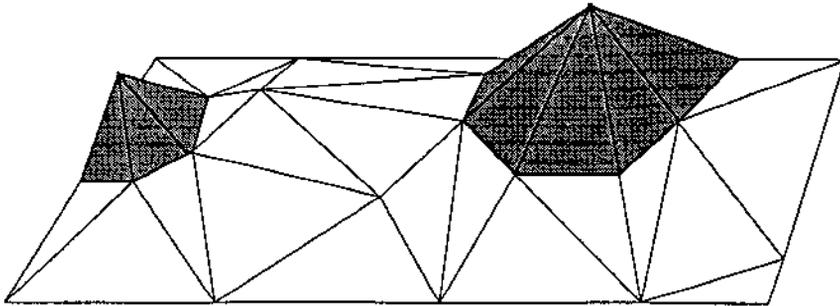


Figura 2.20: Regiões acima de z .

Uma *xy-aproximação* do caminho mínimo que evita faces com grande inclinação é

obtida reduzindo este problema a um problema planar. As faces que possuem inclinação maior que a permitida são projetadas sobre o plano, onde se tornam obstáculos. Novamente, computamos um caminho mínimo que evita obstáculos. Este caminho, projetado sobre a superfície, evita regiões de grande inclinação e tem comprimento mínimo na projeção planar.

2.4.3 Esquemas Aproximativos

Alguns problemas têm duas fontes de dificuldade distintas [97, 120]: a *componente combinatoria*, que precisa computar a seqüência de arestas interceptando um caminho mínimo, e a *componente algébrica*, que precisa computar os pontos de cruzamento dada a seqüência de arestas interceptando um caminho mínimo. A dificuldade destes subproblemas leva a considerar esquemas aproximativos.

Bajaj [8] mostrou a dificuldade do subproblema algébrico para o problema de caminhos mínimos tridimensional provando que, no pior caso, os pontos de interseção entre um caminho mínimo e as arestas dos poliedros são definidos por polinômios irredutíveis de alto grau. Canny e Reif [17] provaram que o subproblema combinatório é *NP*-Difícil.

O primeiro resultado na direção de esquemas aproximativos se deve a Papadimitriou [105] que apresentou um algoritmo que despende tempo $O(n^4 \frac{(L + \log(\frac{n}{\epsilon}))^2}{\epsilon^2})$ para computar um caminho de comprimento $(1 + \epsilon)$ -ótimo para o problema tridimensional, onde L é o número de bits de precisão no modelo computacional e n é o número total de vértices dos poliedros. Choi *et al.* [29] ajustaram a análise do algoritmo de Papadimitriou, removendo várias lacunas e inconsistências no processo. Clarkson [30] apresentou um algoritmo que computa um caminho $(1 + \epsilon)$ -ótimo em tempo $O(\frac{n^2 \text{poli} \log n}{\epsilon^4})$.

Hershberger e Suri [62] projetaram um algoritmo que despende tempo $O(n \log n)$ para aproximar as distâncias geodésicas entre um ponto origem fixo s e todos os vértices de um poliedro convexo. O algoritmo determina os hemisférios meridional $S(\omega)$ e setentrional $N(\omega)$ do poliedro convexo segundo um ângulo $\omega < \pi$. O *hemisfério meridional (setentrional)* é o conjunto de faces que formam um ângulo *diedral* menor ou igual (maior) que ω com o plano H_s , definido pela face que contem s . A cadeia poligonal na fronteira entre $N(\omega)$ e $S(\omega)$ é subdividida de forma que os ângulos $\eta_i = \angle v_i s v_{i+1}$ obedçam à condição $\sqrt{\frac{1 + \text{sen } \eta_i}{1 - \text{sen } \eta_i}} < 1 + \epsilon$, onde v_i e v_{i+1} são vértices consecutivos da cadeia. O algoritmo computa caminhos $2.38(1 + \epsilon)$ -ótimos de s a todos os vértices dessa cadeia através do desdobramento de cunhas bi-planares e do diagrama de Voronoi aditivo desses vértices projetados sobre o plano horizontal.

2.4.4 Metodologias

Apresentamos duas metodologias para resolução de problemas de caminhos mínimos: a árvore seqüência [26] e o método Dijkstra contínuo [95] inspirado no algoritmo de Dijkstra [42] para caminhos mínimos em grafos.

Árvore Seqüência

Chen e Han [26] introduziram uma estrutura chamada *árvore seqüência* que é construída segundo a projeção do ponto origem sobre as arestas de uma superfície poliédrica, utilizando sua imagem desdobrada. Esta projeção é um segmento fechado tal que o caminho mínimo a qualquer ponto neste segmento pode ser desdobrado em um segmento de reta. A imagem desdobrada do ponto origem consiste na sua imagem no sistema de coordenadas planar de uma face depois do desdobramento por rotação de uma seqüência de faces, de forma que todas estejam desdobradas em um plano comum.

Um nó da árvore seqüência (excluindo a raiz) é uma tripla formada por uma aresta do poliedro, pela imagem desdobrada do ponto origem relativa a aresta e pela projeção do ponto origem na aresta (veja Fig. 2.21). Dessa forma, cada nó representa uma aresta do poliedro e, quando a árvore é construída, define uma seqüência de arestas que consiste naquelas que são representadas pelos nós no caminho da raiz a este nó. Isto garante um caminho geodésico de um ponto contido na sombra deste nó ao ponto origem através desta seqüência.

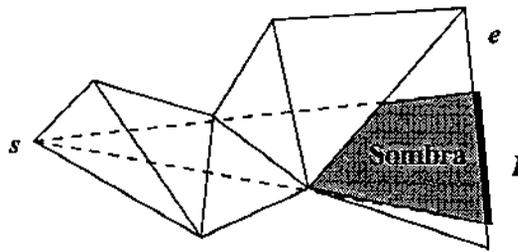


Figura 2.21: A imagem do ponto origem s , sua projeção I e sombra.

O algoritmo constrói uma subdivisão da superfície poliédrica tal que qualquer ponto na mesma região tem a mesma seqüência de arestas no caminho mínimo. Além disso, este esquema planar (*inward layout*) proporciona a computação da subdivisão da superfície uma vez para todas as faces ao invés de uma vez para cada face. Esta técnica despende tempo $O(n^2)$ e espaço $\Theta(n)$ [26].

Dijkstra Contínuo

A metodologia Dijkstra contínuo simula o efeito de propagar uma frente de onda a partir de um ponto origem s a todos os pontos do espaço livre. A *frente de onda* a uma distância δ de s é o conjunto dos pontos do espaço livre que estão à distância geodésica δ de s . A essência desta técnica é usada em [100, 120], mas Mitchell et al. [95] foram os primeiros a formalizar e generalizar a técnica [93, 96, 97].

O método trata certas entidades (vértices, arestas ou faces) como nós em um grafo e a entidade com a menor distância do ponto origem é sempre extraída para expansão. Neste caso, as arestas do poliedro representam os nós de um grafo. Assim, um sinal é propagado da origem s para todos os pontos do espaço livre. A estrutura da frente de onda muda devido a certos *eventos discretos* que ocorrem quando o sinal alcança vértices e arestas pela primeira vez. Neste momento, processamos o evento atualizando a estrutura combinatória da frente de onda. Um vértice v propaga o sinal adiante quando o recebe pela primeira vez. O ponto v é considerado como *permanentemente rotulado* com $d(v)$. O rótulo $d(v)$ é o comprimento do caminho mínimo de s a v . Continuamos propagando o sinal até que todo espaço o tenha recebido.

Metodologia Dijkstra-Contínuo(s)

1. $d(s) := 0$.
2. Permanente(s).
3. *Eventos* := Inicializa(s).
4. enquanto (*Eventos* $\neq \emptyset$) faça
 - (a) $q := \text{Desenfileira}(\textit{Eventos})$.
 - (b) Permanente(q).
 - (c) Propagar(q).

Como não existe distância única do ponto origem a uma aresta, mantemos uma descrição discreta da *função distância* que consiste nos *intervalos de otimalidade* (subsegmentos) de cada aresta. Estes intervalos formam uma cobertura que subdivide a aresta em regiões para as quais o caminho mínimo tem a mesma estrutura discreta, ou seja, passa pela mesma seqüência de vértices e arestas. Dessa forma, a seqüência define univocamente o caminho mínimo da origem ao ponto do intervalo, possibilitando a computação da distância entre estes pontos.

Capítulo 3

Caracterização de Caminhos Mínimos

Geralmente, um algoritmo resolve um problema de caminhos mínimos explorando certos fatos sobre o comportamento local dos caminhos mínimos. Este comportamento local está relacionado ao processo de obter as características de caminhos mínimos, denominado *caracterização de caminhos mínimos*, onde se determina um critério de otimalidade local.

A seção 3.1 especifica o problema de caminhos mínimos que resolvemos e define a função distância FGAD considerada neste problema. Na seção 3.2, apresentamos a caracterização de caminhos geodésicos e mínimos em uma superfície poliédrica segundo a função distância FGAD. Mostramos que caminhos geodésicos são lineares por partes. Definimos o critério de otimalidade local que especifica como um caminho intercepta uma aresta da superfície poliédrica. A seção 3.3 introduz o conceito de caminho localmente f -livre e algumas propriedades relativas à sua interseção. A seção 3.4 define intervalos de otimalidade e mostra que são conjuntos conexos que formam uma cobertura mutuamente disjunta de uma aresta.

3.1 Definição do Problema

Seja S uma superfície poliédrica, possivelmente não convexa, definida por um conjunto de faces, arestas e vértices. A complexidade dos algoritmos que resolvem problemas em S é analisada segundo o número de faces, arestas ou vértices do poliedro. Assumimos que todas as faces são triângulos, pois polígonos simples podem ser triangulados em tempo linear no número de vértices [22].

Consideramos uma *superfície poliédrica limitada*, ou seja, com um número finito de faces limitadas. A superfície S é armazenada em uma estrutura de dados (*quad-edge* [57]) que permite obter as arestas incidentes a um vértice e as faces incidentes a uma aresta.

em tempo constante.

Abordaremos o caso de movimentar um corpo pontual através do *caminho de trabalho mínimo*. Consideramos a quantidade total de trabalho que deve ser realizado para se mover o corpo através do caminho, assumindo que existe coeficiente de atrito e inclinação constantes numa mesma face. A *função distância* que considera direção de movimento provém da Mecânica Newtoniana. Esta função significa o trabalho despendido por uma *força externa* F para realizar o movimento entre pontos em uma mesma face, de forma que a aceleração resultante do corpo durante o movimento seja nula (velocidade constante). Esta força externa está contida no plano da face pois deve anular a resultante das forças consideradas que está contida neste plano. Assumimos uma velocidade inicial positiva e ignoramos a força de atrito estático. Além disso, não consideramos a componente da força externa aplicada ao corpo na mesma direção do movimento, mas o módulo desta força na sua direção que pode ser diferente da direção de movimento.

As forças que atuam no corpo, além da força externa F , são o peso P ($|P| = m.g$), a normal N ($|N| = |P| \cos \alpha_f$) e o atrito A ($|A| = \mu_f |N|$) (veja Fig. 3.1), onde m é a massa do corpo, g é a aceleração da gravidade, α_f é o ângulo agudo de inclinação da face f com o plano horizontal e μ_f é o coeficiente de atrito cinético da face f . Assumimos que μ_f é finito e positivo. Dessa forma, não consideramos obstáculos ou regiões onde se mover não custa nada.

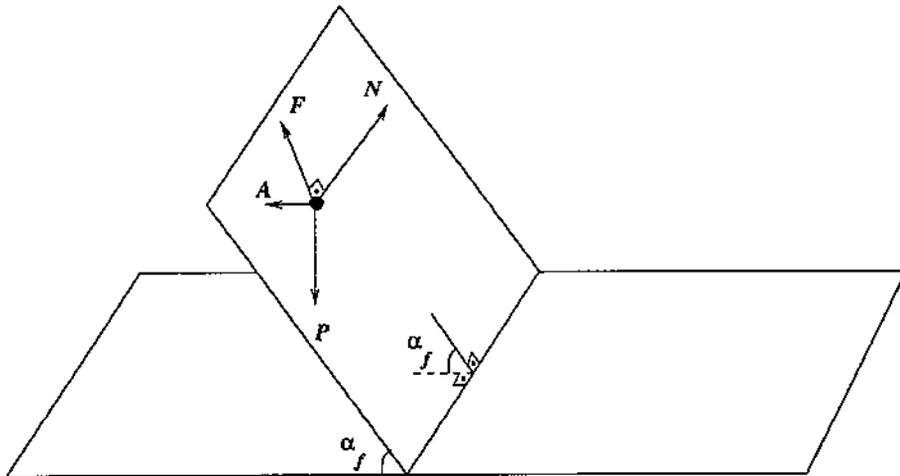


Figura 3.1: Diagrama do corpo livre.

A soma vetorial das forças peso e normal ($P \oplus N$) resulta em uma força R_{PN} contida na face e com a direção do gradiente, ou seja, perpendicular à interseção entre a face e o plano horizontal (veja Fig. 3.2). Além disso, as forças peso-normal R_{PN} e atrito A implicam em uma força resultante $R_{PNA} = R_{PN} \oplus A$ de acordo com um ângulo β (veja Fig. 3.2). Dessa forma, como a resultante de todas as forças é nula, concluímos que F

tem a mesma direção e módulo de R_{PNA} , mas com sentido oposto. Portanto, os módulos das forças R_{PN} , R_{PNA} e F são:

$$\begin{aligned} |R_{PN}| &= |P| \operatorname{sen} \alpha_f = mg \operatorname{sen} \alpha_f, \\ |R_{PNA}| &= \sqrt{|R_{PN}|^2 + 2|R_{PN}||A| \cos \beta + |A|^2}, \\ |F| &= mg \sqrt{\operatorname{sen}^2 \alpha_f + 2\mu_f \operatorname{sen} \alpha_f \cos \alpha_f \cos \beta + \mu_f^2 \cos^2 \alpha_f}. \end{aligned}$$

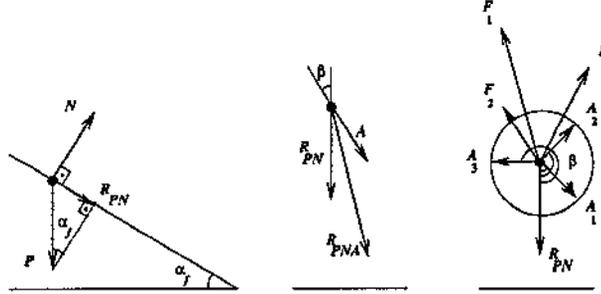


Figura 3.2: Resultante das forças que atuam no corpo.

Definimos a noção de comprimento de acordo com a função distância *geodésica atritada direcional* (FGAD). A cada face f , associamos um coeficiente de atrito cinético $\mu_f > 0$ que especifica a resistência ao movimento no interior da face f . Da mesma forma, a cada aresta e , associamos um coeficiente de atrito cinético $\mu_e > 0$. O comprimento FGAD de um segmento de reta indo de um ponto s a um ponto t na aresta e é o módulo da força externa F com $\beta \in \{0, \pi\}$ vezes a distância euclidiana entre s e t :

$$mg|\mu_e \cos \alpha_e + \operatorname{sen} \alpha_e \cos \beta||st|,$$

onde α_e é o ângulo de inclinação de e com sua projeção ortogonal no plano horizontal. O comprimento FGAD de um segmento de reta indo de um ponto s a um ponto t no interior da face f é o módulo da força F vezes a distância euclidiana entre s e t :

$$mg \sqrt{\operatorname{sen}^2 \alpha_f + 2\mu_f \operatorname{sen} \alpha_f \cos \alpha_f \cos \beta + \mu_f^2 \cos^2 \alpha_f} |st|,$$

onde β é o ângulo entre a força de atrito e a componente do peso projetada no plano da face.

Procuramos encontrar um *caminho mínimo*, ou seja, um caminho de comprimento mínimo segundo a função FGAD de um ponto *origem* s a um ponto *destino* t que esteja completamente na superfície S (veja Fig. 3.3). Assumimos que s e t são vértices da superfície S . Na realidade, o problema que resolvemos consiste em encontrar um caminho ε -ótimo de uma origem a qualquer ponto na superfície.

Formalmente, o problema de caminhos mínimos que resolvemos é especificado da seguinte forma:

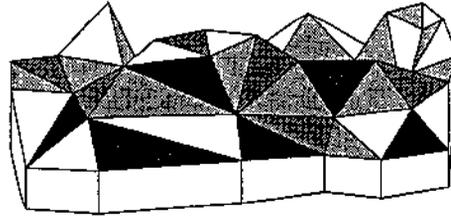


Figura 3.3: Superfície poliédrica com coeficientes de atrito cinético atribuídos a cada face.

PROBLEMA DA GEODÉSICA ATRITADA DIRECIONAL (PGAD)

INSTÂNCIA. *Um ponto origem s em uma superfície poliédrica triangulada S ; uma atribuição de coeficientes de atrito cinético a arestas e faces; e uma tolerância de erro $\varepsilon > 0$.*

QUESTÃO. *Construir uma estrutura que permita computar um caminho ε -ótimo (segundo a função distância geodésica atritada direcional (FGAD)) de s a qualquer ponto consulta t , tal que o caminho esteja na superfície S .*

3.2 Caminhos Geodésicos e Mínimos

Assumindo que cada face tem coeficiente de atrito cinético e inclinação constantes, o seguinte lema conclui que caminhos geodésicos são sempre lineares por partes. Caminhos *lineares por partes* são caminhos cuja interseção com qualquer face é a união de segmentos de reta disjuntos.

Lema 3.1 *Seja f uma face com $\mu_f > 0$. Sejam s e t pontos no interior de f . Um subcaminho de s a t inteiramente contido no interior de f é geodésico se e somente se é um segmento de reta.*

Prova. A prova consiste em verificar que o arco de uma curva parametrizada representando um subcaminho de um ponto a outro no interior de uma mesma face será localmente ótimo se e somente se for um segmento de reta (veja Fig. 3.4).

Se o subcaminho de s a t é geodésico, então deve ser localmente ótimo segundo a função distância FGAD.

Assumimos que o subcaminho de s a t é o arco de uma *curva parametrizada* diferenciável $\Phi : I \rightarrow \mathbb{R}^2$ de um intervalo aberto $I \subset \mathbb{R}$ em \mathbb{R}^2 que representa o plano da face f , onde Φ é uma função que leva $i \in I$ em um ponto $\Phi(i) = (x(i), y(i)) \in \mathbb{R}^2$. O vetor tangente $\Phi'(i)$ é o vetor $(x'(i), y'(i))$, onde $x'(i)$ indica a derivada primeira de $x(i)$ em $i \in I$.

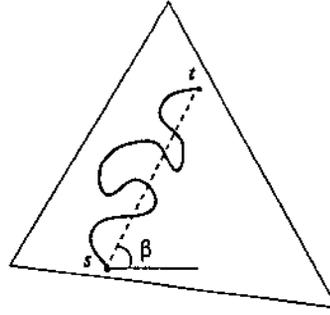


Figura 3.4: Subcaminho contido no interior de uma face.

O comprimento d_{i_0, i_1} do arco da curva parametrizada Φ segundo a função distância FGAD, de $\Phi(i_0) = s$ para $\Phi(i_1) = t$, onde $i_0, i_1 \in I$ é [43]:

$$\begin{aligned} d_{i_0, i_1} &= \int_{i_0}^{i_1} |F_i| |\Phi'(i)| di \\ &= \int_{i_0}^{i_1} |F_{x_i} V_x \oplus F_{y_i} V_y| |\Phi'(i)| di \\ &= \left| \int_{i_0}^{i_1} F_{x_i} V_x |\Phi'(i)| di \oplus \int_{i_0}^{i_1} F_{y_i} V_y |\Phi'(i)| di \right|. \end{aligned}$$

As forças $F_{x_i} V_x$ e $F_{y_i} V_y$ são as componentes horizontal e vertical da força externa F_i , onde V_x e V_y são os vetores unitários na direção dos eixos. Denotamos por $|st|$ o comprimento euclidiano do segmento de reta \overline{st} . Dessa forma, temos o seguinte:

$$\begin{aligned} \left| \int_{i_0}^{i_1} F_{x_i} V_x |\Phi'(i)| di \right| &= \left| \int_{i_0}^{i_1} (R_{PN_x} V_x \oplus A_{x_i} V_x) |\Phi'(i)| di \right| \\ &= \int_{i_0}^{i_1} (0 - |A| \cos \beta_i) |\Phi'(i)| di \\ &= -|A| \int_{i_0}^{i_1} \cos \beta_i |\Phi'(i)| di \\ &= -|A| \cos \beta |st|, \\ \left| \int_{i_0}^{i_1} F_{y_i} V_y |\Phi'(i)| di \right| &= \left| \int_{i_0}^{i_1} (R_{PN_y} V_y \oplus A_{y_i} V_y) |\Phi'(i)| di \right| \\ &= \int_{i_0}^{i_1} (-|R_{PN}| - |A| \sin \beta_i) |\Phi'(i)| di \\ &= \int_{i_0}^{i_1} -|R_{PN}| |\Phi'(i)| di - \int_{i_0}^{i_1} |A| \sin \beta_i |\Phi'(i)| di \\ &= -|R_{PN}| \int_{i_0}^{i_1} |\Phi'(i)| di - |A| \sin \beta |st|, \end{aligned}$$

onde $|\Phi'(i)| = \sqrt{(x'(i))^2 + (y'(i))^2}$ denota o módulo do vetor $\Phi'(i)$, A é a força de atrito e R_{PN} é a resultante das forças peso e normal.

O comprimento do segmento de reta \overline{st} segundo a função distância FGAD é:

$$\begin{aligned} d_{s,t} &= |F| |st| \\ &= |F_x V_x \oplus F_y V_y| |st| \\ &= |F_x V_x| |st| \oplus |F_y V_y| |st|, \end{aligned}$$

onde

$$\begin{aligned} |F_x V_x| |st| &= |(R_{PN_x} V_x \oplus A_x V_x)| |st| \\ &= -|A| \cos \beta |st|, \\ |F_y V_y| |st| &= |(R_{PN_y} V_y \oplus A_y V_y)| |st| \\ &= -|R_{PN}| |st| - |A| \sin \beta |st|. \end{aligned}$$

Dessa forma, devemos mostrar que $d_{s,t} \leq d_{i_0, i_1}$. Contudo, como

$$|F_x V_x| |st| = \left| \int_{i_0}^{i_1} F_{x_i} V_x |\Phi'(i)| di \right|,$$

necessitamos apenas demonstrar que $|F_y V_y| |st| \leq \left| \int_{i_0}^{i_1} F_{y_i} V_y |\Phi'(i)| di \right|$. Para isso, como $|st| \leq \int_{i_0}^{i_1} |\Phi'(i)| di$, temos

$$|(-|R_{PN}| |st|) V_y| \leq \left| \left(-|R_{PN}| \int_{i_0}^{i_1} |\Phi'(i)| di \right) V_y \right|.$$

Portanto,

$$|(-|R_{PN}| |st| - |A| \sin \beta |st|) V_y| \leq \left| \left(-|R_{PN}| \int_{i_0}^{i_1} |\Phi'(i)| di - |A| \sin \beta |st| \right) V_y \right|,$$

ou seja, o comprimento do segmento de reta \overline{st} é menor ou igual ao comprimento do arco de qualquer curva parametrizada segundo a função distância FGAD. Dessa forma, o subcaminho representado por este arco será geodésico se e somente se for um segmento de reta. Além disso, como o subcaminho geodésico é um segmento de reta, a suposição de que a curva Φ é diferenciável se torna desnecessária. \square

Corolário 3.1 *Caminhos geodésicos sobre superfícies poliédricas segundo a função distância FGAD são lineares por partes.*

O lugar geométrico dos pontos t de uma face f com distância δ constante de um ponto origem s , segundo a função distância FGAD, consiste na curva representada pela seguinte equação polar:

$$|st| = \frac{\delta}{mg\sqrt{(\mu_f^2 \cos^2 \alpha_f + 2\mu_f \cos \alpha_f \sin \alpha_f \cos \beta_t + \sin^2 \alpha_f)},}$$

onde β_t é o ângulo formado entre o vetor da força R_{PN} na direção do gradiente e o vetor da força de atrito A na direção do segmento \overline{st} . Esta curva assume a forma geométrica de uma oval (veja Fig. 3.5). A curva é uma circunferência quando $\alpha_f = 0$, mas tem uma forma degenerada quando $mg\sqrt{(\mu_f^2 \cos^2 \alpha_f + 2\mu_f \cos \alpha_f \sin \alpha_f \cos \beta_t + \sin^2 \alpha_f)} = 0$. Isto ocorre apenas quando $\cos \beta_t = -1$. Neste caso, temos $(\mu_f \cos \alpha_f - \sin \alpha_f)^2 = 0$, ou seja, $\mu_f = \tan \alpha_f$. Assumimos que $\mu_f \neq \tan \alpha_f$ para garantir a convexidade estrita da função distância FGAD. Assim, evitando o caso degenerado, temos neste lugar geométrico uma forte indicação da convexidade da função FGAD. Contudo, a demonstração algébrica¹ deste fato é necessária para assegurar a existência de um critério de otimalidade local.

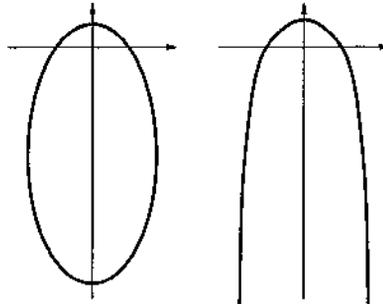


Figura 3.5: Lugar geométrico dos pontos equidistantes da origem em uma face.

Chamamos de *ângulo de incidência* θ , o ângulo agudo entre um segmento de um caminho geodésico que cruza (entra) a fronteira da face f e um vetor perpendicular à fronteira da face f (veja Fig. 3.6). O *ângulo de refração* θ' é o ângulo agudo entre um segmento de um caminho geodésico que cruza (sai) a fronteira da face f' e um vetor perpendicular à fronteira de f' .

Um caminho geodésico deve cruzar o interior de uma aresta e de acordo com um critério de otimalidade local para o problema da geodésica atritada direcional.

Lema 3.2 *Sejam f e f' duas faces que compartilham uma aresta e . Sejam s um ponto no interior de f e t um ponto no interior de f' . Seja p um caminho geodésico entre s*

¹Esta demonstração está apresentada no apêndice B.

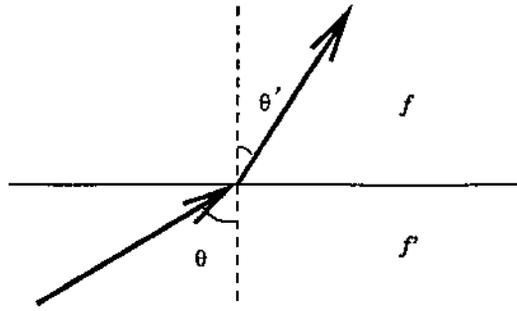


Figura 3.6: Ângulos de incidência e refração.

e t que cruza o interior de e em um único ponto x^ , então x^* pode ser univocamente determinado.*

Prova. A prova consiste na solução de um problema de minimização do comprimento de um caminho de um ponto $s(0, -y_0, z_0)$ na face f a um ponto $t(x_1, y_1, z_1)$ na face f' passando por um único ponto x^* da aresta $e = f \cap f'$ segundo a função distância FGAD (veja Fig. 3.7). As faces f e f' são determinadas pelos pontos s, t e pela aresta e cujos pontos têm ordenada nula e altura igual a $(ax + b)$, onde a, b são constantes e x é a abscissa do ponto em e .

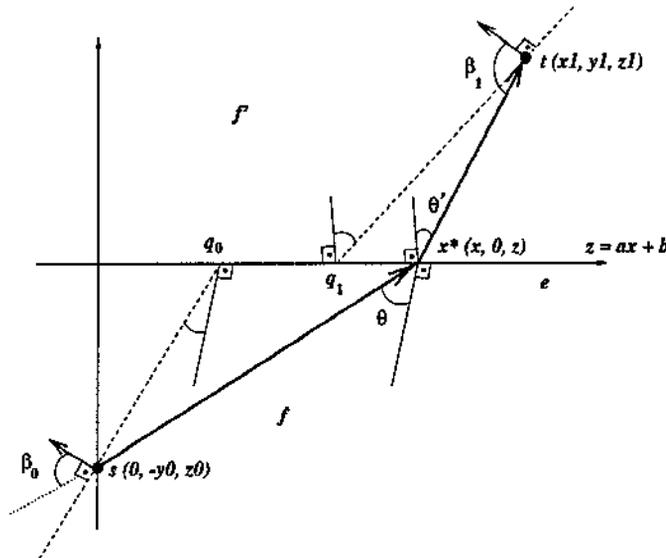


Figura 3.7: Critério de otimalidade local.

Dessa forma, devemos encontrar o ponto de mínimo x^* da seguinte função² com uma variável x real:

$$\sqrt{\mu_f^2 \cos^2 \alpha_f + 2\mu_f \cos \alpha_f \sin \alpha_f \cos \beta_{f_x} + \sin^2 \alpha_f} |sx^*| + \\ + \sqrt{\mu_{f'}^2 \cos^2 \alpha_{f'} + 2\mu_{f'} \cos \alpha_{f'} \sin \alpha_{f'} \cos \beta_{f'_x} + \sin^2 \alpha_{f'}} |x^*t|,$$

onde $\mu_f, \mu_{f'}$ são os coeficientes de atrito; $\alpha_f, \alpha_{f'}$ são os ângulos de inclinação das faces com o plano horizontal; e $\beta_{f_x}, \beta_{f'_x}$ são os ângulos entre a força de atrito A na direção do movimento e a força resultante R_{PN} na direção do gradiente para as respectivas faces. Observe que os ângulos³ β_{f_x} e $\beta_{f'_x}$ variam de acordo com x .

A convexidade da função distância FGAD entre pontos em uma mesma face garante a existência de um ponto único x_s (x_t) em e cuja distância de s (para t) é mínima. Portanto, o ponto x^* deve estar no intervalo $[x_s, x_t]$ (veja Fig. 3.8).

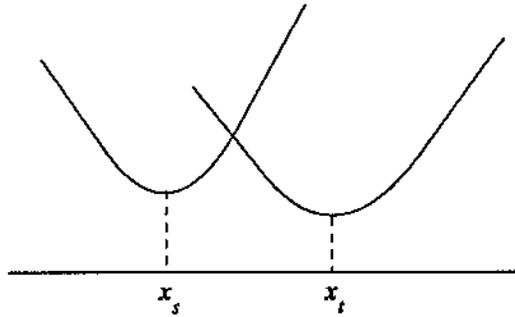


Figura 3.8: O ponto x^* deve estar no intervalo $[x_s, x_t]$.

A determinação do ponto de mínimo x_s se faz através da derivada primeira da função distância FGAD de s a x^* . De forma análoga, encontramos o ponto x_t . No apêndice C, obtemos a seguinte expressão que determina x_s em função do ângulo de incidência θ_s na aresta e , onde c_{s_0}, \dots, c_{s_4} são constantes:

$$c_{s_0} + c_{s_1} \sin \theta_s + c_{s_2} \sin^2 \theta_s + c_{s_3} \sin^3 \theta_s + c_{s_4} \sin^4 \theta_s = 0.$$

O ponto x^* é univocamente determinado pela derivada primeira da função que representa a distância FGAD de s a x^* adicionada à distância FGAD de x^* a t (veja Exp. C.1):

$$\frac{c_5 (\cos \beta_{f_{x^*}})' |sx^*| + (2c_5 \cos \beta_{f_{x^*}} + c_6) (|sx^*|)'}{\sqrt{2c_5 \cos \beta_{f_{x^*}} + c_6}} + \\ + \frac{c_7 (\cos \beta_{f'_{x^*}})' |x^*t| + (2c_7 \cos \beta_{f'_{x^*}} + c_8) (|x^*t|)'}{\sqrt{2c_7 \cos \beta_{f'_{x^*}} + c_8}} = 0,$$

²Podemos verificar a convexidade estrita desta função no apêndice B.

³Detalhes sobre os ângulos β_{f_x} e $\beta_{f'_x}$ são encontrados no apêndice A.

onde $c_5 = \mu_f \cos \alpha_f \sin \alpha_f$, $c_6 = \mu_f^2 \cos^2 \alpha_f + \sin^2 \alpha_f$, $c_7 = \mu_{f'} \cos \alpha_{f'} \sin \alpha_{f'}$ e $c_8 = \mu_{f'}^2 \cos^2 \alpha_{f'} + \sin^2 \alpha_{f'}$.

Simplificamos a equação acima, buscando uma expressão equivalente em função dos ângulos de incidência θ e refração θ' (veja Exp. C.2):

$$\frac{\sin \theta (c_5 \cos \beta_{f_x^*} + c_6) + c_9}{\sqrt{2c_5 \cos \beta_{f_x^*} + c_6}} + \frac{\sin \theta' (c_7 \cos \beta_{f'_x^*} + c_8) + c_{10}}{\sqrt{2c_7 \cos \beta_{f'_x^*} + c_8}} = 0, \quad (3.1)$$

onde $c_9 = \frac{c_5 c_4}{\sqrt{1+a^2|s q'|}}$, q' é o ponto de interseção entre a reta passando por s na mesma direção do vetor da força R_{PN} e a aresta e , c_{10} é uma constante análoga a c_9 relativa à face f' . Observe que $\cos \beta_{f_x^*} = \pm \cos \psi \cos \theta \pm \sin \psi \sin \theta$ e $\cos \beta_{f'_x^*} = \pm \cos \psi' \cos \theta' \pm \sin \psi' \sin \theta'$, sendo ψ e ψ' constantes. Portanto, a expressão algébrica acima determina univocamente x^* . \square

O ângulo crítico $\theta_{f,f'}$ para e consiste no ângulo de incidência quando o ângulo de refração $\theta' = \frac{\pi}{2}$. Neste caso, o critério de otimalidade local (veja Eq. 3.1) assume a seguinte forma:

$$\frac{\sin \theta_{f,f'} (c_5 (\pm \cos \psi \cos \theta_{f,f'} \pm \sin \psi \sin \theta_{f,f'}) + c_6) + c_9}{\sqrt{2c_5 (\pm \cos \psi \cos \theta_{f,f'} \pm \sin \psi \sin \theta_{f,f'}) + c_6}} + \frac{\pm c_7 \sin \psi' + c_8 + c_{10}}{\sqrt{2c_7 \sin \psi' + c_8}} = 0.$$

Então, analogamente a θ_s e θ_t , o ângulo crítico $\theta_{f,f'}$ é dado pela expressão

$$c_{f,f'_0} + c_{f,f'_1} \sin \theta_{f,f'} + c_{f,f'_2} \sin^2 \theta_{f,f'} + c_{f,f'_3} \sin^3 \theta_{f,f'} + c_{f,f'_4} \sin^4 \theta_{f,f'} = 0,$$

onde c_{f,f'_i} são constantes para $i = 0, \dots, 4$ (veja Eq. C.3).

Um caminho geodésico *usa criticamente* parte de uma aresta e quando encontra a aresta e no ângulo crítico $\theta_{f,e}$ em um ponto q interior a e , percorre a aresta e por alguma distância e deixa a aresta e para o interior de f' no ângulo crítico $\theta_{e,f'}$ em um ponto r interior a e (veja Fig. 3.9(a)).

Um caminho geodésico é *criticamente refletido* por e da face f quando é incidente na aresta e no ângulo crítico $\theta_{f,e}$ em um ponto q interior a e , percorre a aresta e por alguma distância e sai da aresta e de volta à face f em um ponto r interior a e , deixando a aresta no ângulo $\theta_{e,f}$ (veja Fig. 3.9(b)).

Resumindo o comportamento das geodésicas com respeito a ângulos críticos, obtemos o seguinte corolário:

Corolário 3.2 [97] *Se um caminho geodésico p passa pelo interior da aresta e , então o ângulo de incidência é menor (em valor absoluto) que o ângulo crítico $\theta_{f,f'}$. Se p percorre alguma parte $\bar{q}\bar{r}$ do interior de e , então o ângulo de incidência no ponto q é igual ao ângulo crítico $\theta_{f,f'}$, e o ângulo de refração em r deve também ser igual a $\theta_{f,f'}$.*

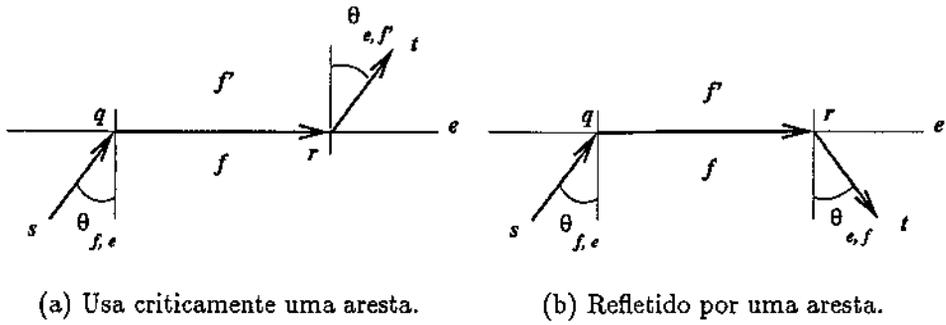


Figura 3.9: Caminhos geodésicos sobre uma aresta.

Generalizamos o Lema 3.2 para considerar ângulos críticos, ou seja, caminhos que podem usar criticamente parte de uma aresta ou que podem ser criticamente refletidos. Neste caso, temos um problema de minimização em duas variáveis.

Lema 3.3 *Um caminho geodésico cruza a aresta $e = f \cap f'$ de uma de duas maneiras: intercepta a aresta e em um único ponto de cruzamento e obedece ao critério de otimalidade local naquele ponto, ou encontra a aresta e em um ângulo crítico $\theta_{f,e}$, percorre a aresta por alguma distância e sai da aresta para a outra face (para a mesma face) em um ângulo crítico $\theta_{e,f}$ ($\theta_{e,f}$).*

Prova. A prova segue da solução de um problema de programação convexa (não-linear) [85] em duas variáveis reais x e x' , onde x e x' são as abscissas dos pontos q e r , respectivamente (veja Fig. 3.9). Desejamos minimizar a função

$$d(x, x') = \sqrt{\mu_f^2 \cos^2 \alpha_f + 2\mu_f \cos \alpha_f \sin \alpha_f \cos \beta_{f_x} + \sin^2 \alpha_f} |sx^*| + (\mu_e \cos \alpha_e - \sin \alpha_e) |x^* x'^*| + \sqrt{\mu_{f'}^2 \cos^2 \alpha_{f'} + 2\mu_{f'} \cos \alpha_{f'} \sin \alpha_{f'} \cos \beta_{f'_x} + \sin^2 \alpha_{f'}} |x'^* t|$$

sujeita a

$$g(x, x') = x - x' \leq 0.$$

As condições de Karush-Kuhn-Tucker [76] implicam as três relações:

$$\begin{aligned} \nabla d(x, x') + l \nabla g(x, x') &= 0, \\ l g(x, x') &= 0, \\ l &\geq 0, \end{aligned}$$

onde $\nabla d(x, x')$ e $\nabla g(x, x')$ são vetores gradiente e l é o multiplicador de Lagrange. Dessa forma, se $l = 0$ temos $\nabla d(x, x') = \left(\frac{\partial d(x, x')}{\partial x}, \frac{\partial d(x, x')}{\partial x'} \right) = 0$, ou melhor,

$$\frac{c_5 (\cos \beta_{f_x})' |sx|}{\sqrt{c_6 + 2c_5 \cos \beta_{f_x}}} + \sqrt{c_6 + 2c_5 \cos \beta_{f_x}} (|sx|)' + \sqrt{1 + a^2} (\mu_e \cos \alpha_e - \sin \alpha_e) = 0 \text{ e}$$

$$\frac{c_7(\cos \beta_{f'_{x'}})' |x't|}{\sqrt{c_8 + 2c_7 \cos \beta_{f'_{x'}}}} + \sqrt{c_8 + 2c_7 \cos \beta_{f'_{x'}}} (|x't|)' - \sqrt{1 + a^2}(\mu_e \cos \alpha_e - \text{sen } \alpha_e) = 0.$$

Simplificando em função dos ângulos críticos $\theta_{f,e}$ e $\theta_{e,f'}$, temos

$$\frac{\text{sen } \theta_{f,e}(c_6 + c_5 \cos \beta_{f_x}) + c_9}{\sqrt{c_6 + 2c_5 \cos \beta_{f_x}}} + (\mu_e \cos \alpha_e - \text{sen } \alpha_e) = 0 \text{ e}$$

$$\frac{\text{sen } \theta_{e,f'}(c_8 + c_7 \cos \beta_{f'_{x'}}) + c_{10}}{\sqrt{c_8 + 2c_7 \cos \beta_{f'_{x'}}}} - (\mu_e \cos \alpha_e - \text{sen } \alpha_e) = 0.$$

Por outro lado, o multiplicador de Lagrange é diferente de zero se e somente se $g(x, x') = 0$, ou seja, $x = x'$. Neste caso, o caminho cruza a aresta no único ponto de cruzamento e obedece ao seguinte critério de otimalidade local:

$$\frac{(c_6 + c_5 \cos \beta_{f_x}) \text{sen } \theta + c_9}{\sqrt{c_6 + 2c_5 \cos \beta_{f_x}}} + \frac{(c_8 + c_7 \cos \beta_{f'_{x'}}) \text{sen } \theta' + c_{10}}{\sqrt{c_8 + 2c_7 \cos \beta_{f'_{x'}}}} = 0.$$

Portanto, dependendo do multiplicador de Lagrange, um caminho geodésico intercepta a aresta em um único ponto ($l \neq 0$), ou percorre a aresta e sai em um ângulo crítico $\theta_{e,f'}$ ($l = 0$) segundo o critério de otimalidade local. Uma demonstração semelhante existe para caminhos criticamente refletidos. \square

A interseção de um caminho geodésico p com uma aresta e será um conjunto, possivelmente vazio, de pontos e segmentos. Estes pontos são chamados *pontos de cruzamento* da aresta e para o caminho p e os segmentos, *segmentos compartilhados* por e e p . Portanto, podemos caracterizar o comportamento de caminho geodésico nas arestas.

Teorema 3.1 [97] *Seja p um caminho geodésico através de uma superfície poliédrica triangulada atritada finita S .*

1. *Se p passa pelo interior da aresta e de S em um ponto de cruzamento q , então p obedece ao critério de otimalidade local em q .*
2. *Suponha que \overline{qr} é um segmento compartilhado por p e pela aresta e . Se q não é um vértice, então o ângulo de incidência em q é crítico. Se p encontra q a partir de f , então o ângulo de incidência é $\theta_{f,e}$ ou $\theta_{f,f'}$. Da mesma forma, se r não é um vértice, então o ângulo de refração em r é crítico.*

Prova. Imediato dos Lemas 3.2 e 3.3. \square

A convexidade da função distância FGAD especifica univocamente um caminho geodésico que intercepta uma seqüência de arestas:

Lema 3.4 *Se p é um caminho geodésico de um ponto s a um ponto t que intercepta a seqüência de arestas $E = (e_1, \dots, e_k)$ com $e_i \neq e_{i+1}$ (tal que não exista segmentos compartilhados), então p é o único caminho geodésico ligando s a t .*

Prova. A demonstração consiste em mostrar que a função do comprimento FGAD do caminho que vai do ponto s a um ponto t que intercepta a seqüência de arestas E é uma função estritamente convexa dos pontos de cruzamento em cada aresta (veja Fig. 3.10).

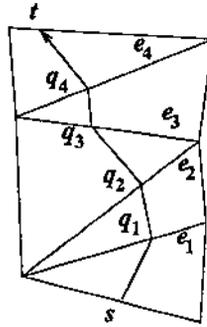


Figura 3.10: Um caminho geodésico que intercepta a seqüência de arestas E .

A função FGAD do comprimento do caminho que vai do ponto s a um ponto t que intercepta a seqüência de arestas E é dada por $d(q_1, \dots, q_k) =$

$$\begin{aligned} & \sqrt{\mu_1^2 \cos^2 \alpha_1 + 2\mu_1 \cos \alpha_1 \sin \alpha_1 \cos \beta_1 + \sin^2 \alpha_1} |sq_1| \\ & + \sum_{i=1}^{k-1} \sqrt{\mu_{i+1}^2 \cos^2 \alpha_{i+1} + 2\mu_{i+1} \cos \alpha_{i+1} \sin \alpha_{i+1} \cos \beta_{i+1} + \sin^2 \alpha_{i+1}} |q_i q_{i+1}| \\ & + \sqrt{\mu_{k+1}^2 \cos^2 \alpha_{k+1} + 2\mu_{k+1} \cos \alpha_{k+1} \sin \alpha_{k+1} \cos \beta_{k+1} + \sin^2 \alpha_{k+1}} |q_k t|, \end{aligned}$$

onde q_i é o ponto de cruzamento na aresta e_i . Desejamos mostrar que $d(q_1, \dots, q_k)$ é uma função estritamente convexa dos pontos de cruzamento em cada aresta. Pelo apêndice B, temos que as seguintes funções são estritamente convexas:

$$\begin{aligned} & \sqrt{\mu_1^2 \cos^2 \alpha_1 + 2\mu_1 \cos \alpha_1 \sin \alpha_1 \cos \beta_{1q_1} + \sin^2 \alpha_1} |sq_1| \text{ e} \\ & \sqrt{\mu_{k+1}^2 \cos^2 \alpha_{k+1} + 2\mu_{k+1} \cos \alpha_{k+1} \sin \alpha_{k+1} \cos \beta_{k+1q_k} + \sin^2 \alpha_{k+1}} |q_k t|. \end{aligned}$$

A função $\sqrt{\mu_{i+1}^2 \cos^2 \alpha_{i+1} + 2\mu_{i+1} \cos \alpha_{i+1} \sin \alpha_{i+1} \cos \beta_{i+1q_i q_{i+1}} + \sin^2 \alpha_{i+1}} |q_i q_{i+1}|$ é estritamente convexa em duas variáveis escalares que determinam os pontos q_i e q_{i+1} . Isto segue da convexidade estrita desta função quando um dos pontos q_i ou q_{i+1} é fixo. Assim, a interseção de um plano ortogonal ao plano horizontal e paralelo ao eixo das abscissas ou

ordenadas com esta função implica em uma curva estritamente convexa. Generalizando a direção do plano ortogonal, temos que a função em duas variáveis é estritamente convexa.

Pelo Teorema B.1, como a função $d(q_1, \dots, q_k)$ é um somatório de funções estritamente convexas, então $d(q_1, \dots, q_k)$ é estritamente convexa. Portanto, a função $d(q_1, \dots, q_k)$ possui um único mínimo global, e qualquer mínimo local deve ser global. Visto que p é um mínimo local, é também o único mínimo global interceptando a seqüência de arestas E . \square

Um *ponto crítico de entrada* de um caminho geodésico p na face f consiste em um ponto q (o mais próximo da origem s) interior a uma aresta $e = f \cap f'$ extremo de um segmento compartilhado quando p alcança q a partir de f . Da mesma forma, um *ponto crítico de saída* do caminho p para face f é um ponto r interior a $(f \cap f')$ (o mais distante da origem s) extremo de um segmento compartilhado quando p vai de r para a face f .

Sejam v e v' vértices consecutivos encontrados na lista de pontos descrevendo um caminho geodésico p . A caracterização de caminhos geodésicos (Teorema 3.1) implica que a estrutura do subcaminho de p entre v e v' será uma lista alternada de pontos de cruzamento e segmentos compartilhados. Um caminho geodésico p pode ser univocamente especificado por uma lista de vértices, arestas e faces cujos interiores contêm uma porção de p . Arestas e faces podem ser repetidas nesta lista.

Finalmente, obtemos a seguinte caracterização de caminhos geodésicos e mínimos em superfícies poliédricas:

Teorema 3.2 *A forma geral de um caminho geodésico ou mínimo é um caminho linear por partes que vai alternadamente por uma seqüência de vértices, seqüências de arestas (possivelmente vazias) e segmentos compartilhados, tal que o caminho obedece ao critério de otimalidade local em cada aresta de qualquer seqüência de arestas e nas extremidades de cada segmento compartilhado.*

Prova. Segue dos Lemas 3.2 e 3.3. \square

3.3 Caminhos Localmente f -Livres

Seja f uma face de uma superfície poliédrica S e e uma de suas três arestas. Um *caminho localmente f -livre* para $q \in e$ é um caminho geodésico do ponto origem s que alcança q pelo exterior da face f . Formalmente, existe um $\delta > 0$, tal que p não passa pela interseção do interior de f com a bola de raio δ centrada em q e é localmente ótimo. Um *caminho mínimo localmente f -livre* para q , denotado por $p_f(q)$, é um caminho localmente f -livre para q que tem comprimento mínimo dentre todos os caminhos localmente f -livres para q .

Um caminho mínimo para um ponto q no interior de $e = f \cap f'$ deve ser localmente f -livre ou localmente f' -livre, pois o caminho é linear por partes. Assim, o caminho mínimo para q pode ser obtido pelo menor dentre um caminho mínimo localmente f -livre $p_f(q)$ e um caminho mínimo localmente f' -livre $p_{f'}(q)$.

Algumas propriedades importantes de caminhos mínimos localmente f -livres no problema PGAD referem-se à sua interseção e cruzamento (veja Fig. 3.11).

Lema 3.5 [97] *Caminhos mínimos e caminhos mínimos localmente f -livres são simples.*

Prova. Se um caminho não é simples, então pode ser encurtado e permanece f -livre. \square

Lema 3.6 [95] *Se $p(q_1)$ e $p(q_2)$ são caminhos mínimos de s aos pontos q_1 e q_2 , então estes caminhos mínimos podem se interceptar apenas em qualquer vértice v de uma superfície poliédrica S , de forma que o subcaminho de $p(q_1)$ tem o mesmo comprimento que o subcaminho de $p(q_2)$ de s a v .*

Prova. Seja γ o ponto de interseção de $p(q_1)$ e $p(q_2)$. Os subcaminhos para γ devem ter comprimento igual, pois a substituição de um subcaminho de s a γ pelo outro encurtaria um dos dois caminhos. Além disso, se γ for interior a uma face ou aresta, então cada caminho poderia ser encurtado por um ponto interior γ_1 antes de γ no subcaminho de $p(q_1)$, a um ponto interior γ_2 , depois de γ no subcaminho de $p(q_2)$. Portanto, os caminhos poderiam ser localmente encurtados e não seriam caminhos mínimos. \square

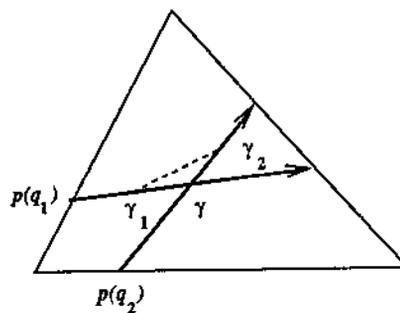


Figura 3.11: Caminhos mínimos que se interceptam podem ser encurtados.

Lema 3.7 [97] *Sejam $p_f(q_1)$ e $p_f(q_2)$ dois caminhos mínimos localmente f -livres para pontos q_1 e q_2 na aresta $e = f \cap f'$. Considere os segmentos de cada caminho como orientados na direção do caminho partindo da origem. Se um segmento de $p_f(q_1)$ e um segmento de $p_f(q_2)$ são incidentes na mesma aresta e' e ambos pertencem à mesma face f_γ , então aqueles segmentos não podem se cruzar.*

Prova. Se os segmentos se cruzam em algum ponto γ interior a f_γ , então os dois subcaminhos de s a γ devem ter comprimentos iguais. Portanto, cada caminho poderia ser encurtado por um ponto γ_1 , interior a f_γ antes de γ no subcaminho de $p_f(q_1)$, a um ponto γ_2 , interior a f_γ depois de γ no subcaminho de $p_f(q_2)$, e ser localmente ajustado, tal que seria novamente geodésico e ainda seria localmente f -livre, pois os caminhos são incidentes na mesma aresta. Se os caminhos forem incidentes em arestas diferentes de f_γ , então o caminho encurtado se tornaria não localmente f -livre uma vez que é perturbado na otimalidade local. Um argumento similar se aplica ao caso no qual o ponto de cruzamento γ está no interior de uma aresta. \square

3.4 Intervalos de Otimalidade

Um *intervalo de otimalidade* para raiz r e seqüência de arestas E relativo a um par aresta-face (e, f) consiste no subintervalo de uma aresta $e = f \cap f'$ para o qual os caminhos mínimos localmente f -livres para a aresta têm a mesma estrutura discreta (a mesma seqüência de arestas E , depois de passar pela mesma raiz r). Note que r pode ser uma extremidade de e ou um dos vértices de f' quando $E = \emptyset$.

Lema 3.8 [97] *Sejam (e, f) um par aresta-face e $p_f(q)$ um caminho mínimo localmente f -livre de s para q interior a e . Sejam r a raiz de $p_f(q)$ e E a última seqüência de arestas de $p_f(q)$. O conjunto I de pontos em e para os quais existe um caminho mínimo localmente f -livre com raiz r e última seqüência de arestas E é conexo, ou seja, um segmento contido em e .*

Prova. Sejam a e b pontos de e tais que exista caminhos mínimos localmente f -livres $p_f(a)$ e $p_f(b)$ com raiz r e última seqüência de arestas E . Seja $p_f(q)$ qualquer caminho mínimo localmente f -livre para um ponto q de e entre a e b . O primeiro segmento de reta de $p_f(q)$ a partir de q não pode nos levar à face f (pela definição de caminho localmente f -livre), então deve nos levar à face f_{k+1} (veja Fig. 3.12). O segmento não pode cruzar os caminhos $p_f(a)$ ou $p_f(b)$ (Lema 3.7), então deve cruzar a aresta e_k entre os pontos a_k e b_k . Continuando pelas arestas e_{k-1}, \dots, e_1 , o caminho $p_f(q)$ deve ficar entre os caminhos $p_f(a)$ e $p_f(b)$ até que a raiz r seja alcançada. Assim, $p_f(q)$ deve passar pelo mesmo conjunto de arestas que os caminhos $p_f(a)$ e $p_f(b)$ e ter raiz r . Portanto, existe um único caminho mínimo localmente f -livre $p_f(q)$ que também tem raiz r e última seqüência de arestas E , ou seja, $q \in I$. \square

Lema 3.9 [97] *Intervalos de otimalidade relativos a um par aresta-face (e, f) formam uma cobertura de e e possuem interiores mutuamente disjuntos.*

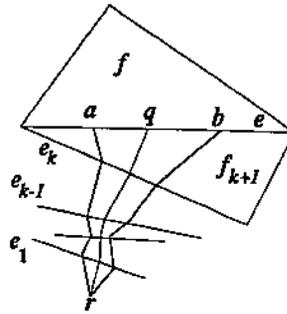


Figura 3.12: Intervalos de otimalidade são conjuntos conexos.

Prova. Todo ponto $q \in e$ deve estar em algum intervalo de otimalidade relativo a (e, f) . Se $p_f(q)$ é um caminho mínimo localmente f -livre para um ponto $q \in e$ que está nos intervalos de otimalidade para (r, E) e (r', E') relativo a (e, f) , então q deve satisfazer a equação $d(r) + d_{r,E}(q) = d(r') + d_{r',E'}(q)$, onde existe apenas um ponto q satisfazendo esta equação, ou seja, cujas distâncias da origem passando por (r, E) e por (r', E') são iguais. Se houvesse dois pontos, q e q' , então teríamos um cruzamento dos caminhos mínimos localmente f -livres para q e q' . Assim, não pode haver intervalo contido em e cujos pontos estão em dois ou mais intervalos de otimalidade. Portanto, os intervalos de otimalidade devem ter interiores disjuntos. \square

Seja r um vértice que é a raiz de uma seqüência de arestas E . Seja I um intervalo de otimalidade com raiz r e última seqüência de arestas E . O único ponto c em I que é mais próximo a r pelos caminhos que vão de r a I por E é chamado *ponto fronteira* do intervalo. O ponto fronteira existe e sua unicidade se deve à convexidade estrita da função FGAD, ao fecho e à limitação de I .

A interseção do caminho f -livre para c com a face f' consiste em um segmento de reta $\overline{c'c}$, onde o ponto c' é denominado o *ponto de acesso* do intervalo I . Os segmentos de reta $\overline{c'c}$, para todo intervalo de otimalidade com respeito a (e, f) , formam uma partição da face f' em regiões denominadas *canais de acesso*. Um canal de acesso é determinado por um par de intervalos de otimalidade adjacentes (veja Fig. 3.13).

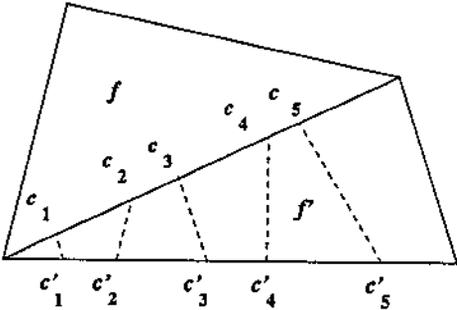


Figura 3.13: Partição da face f' em canais de acesso.

Capítulo 4

O Algoritmo Dijkstra Contínuo Direcional

Aplicamos a *metodologia Dijkstra contínuo* [93, 95, 97] para resolver o problema da geodésica atritada direcional (PGAD). Esta metodologia, baseada no algoritmo de Dijkstra para caminhos mínimos em grafos [42], simula o movimento contínuo de uma *frente de onda* que se propaga, a partir de um ponto origem s , pela superfície S (veja Fig. 4.1). A frente de onda a distância d é o conjunto de pontos tal que o comprimento do caminho mínimo de s a estes pontos é d . A frente de onda alcança vértices e arestas em pontos denominados *eventos* e o algoritmo considera apenas as mudanças que ocorrem nestes eventos. Apresentamos uma versão de um algoritmo Dijkstra contínuo para o problema PGAD no qual o número de eventos tem a mesma cota superior, $O(n^4)$, que o algoritmo para o problema da região valorada [97].

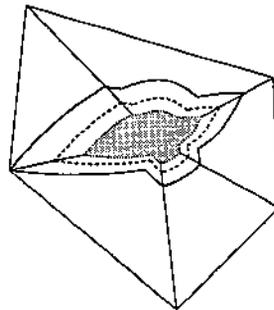


Figura 4.1: A frente de onda sobre uma superfície poliédrica.

A seção 4.1 introduz as estruturas de dados usadas pelo algoritmo Dijkstra contínuo direcional. Na seção 4.2, consideramos a especificação do algoritmo e descrevemos as

funções e procedimentos principais para propagar um intervalo de otimalidade sobre uma superfície poliédrica. A seção 4.3 descreve as funções numéricas utilizadas pelo algoritmo. Uma destas funções encontra os pontos de interseção entre um caminho geodésico e uma seqüência de arestas. A outra função encontra o ponto à igual distância do ponto origem em dois intervalos não disjuntos. A seção 4.4 apresenta a prova de corretude do algoritmo, onde demonstramos que os intervalos de otimalidade, quando o algoritmo termina, permitem computar os caminhos mínimos a partir da origem. Na seção 4.5, realizamos a análise de complexidade do algoritmo e apresentamos uma extensão que obtém caminhos mínimos para pontos no interior de uma face.

4.1 Estruturas de Dados

O algoritmo usa estruturas de dados como pares aresta-face, intervalos de otimalidade candidatos e pontos de evento. Um *par aresta-face* (e, f) tem associado uma lista ordenada de intervalos candidatos na ordem que aparecem sobre a aresta. Inicialmente, a lista de intervalos de (e, f) está vazia, mas forma uma cobertura da aresta e na conclusão do algoritmo.

Um *intervalo candidato* I é um segmento contido em uma aresta com a seguinte informação associada: sua extensão, $[a, b]$; sua raiz, r ; seus *primeiros pontos de cruzamento*, (u_a, u_b) , dos caminhos geodésicos de r para a e b que cruzam a primeira aresta da última seqüência de arestas; seus *últimos pontos de cruzamento*, (w_a, w_b) , dos caminhos geodésicos para a e b que cruzam a última aresta da última seqüência de arestas; seu par aresta-face, (e, f) ; sua distância, $d(r)$; seu ponto fronteira, c ; seu ponto de acesso, c' ; e seu predecessor \bar{I} , o intervalo candidato que propagou I .

Um ponto de evento q tem associado: uma distância $d(q)$, que é o comprimento do menor caminho para o ponto q computado até então; um intervalo candidato I , que contém o ponto de evento; o tipo do evento $\Gamma(q)$, que indica se o ponto de evento é um ponto fronteira; e a topologia $\Upsilon(q)$, que informa se o ponto de evento é interior a uma aresta ou um vértice da superfície. Inicialmente, o único ponto de evento é a origem s . À medida que o algoritmo procede, os pontos de evento são encontrados e cada um será *rotulado permanentemente*, significando que o valor $d(q)$ representa o comprimento de um caminho mínimo de s a q . O algoritmo mantém uma *fila de prioridades*, *Eventos*, com pontos de evento de algum intervalo candidato.

4.2 Algoritmo

Inicialmente, o algoritmo rotula permanentemente a origem s com uma distância $d(s) = 0$ e inicializa a lista de eventos e as listas de intervalos candidatos dos pares aresta-face,

considerando as arestas das faces que contêm s . O algoritmo procede selecionando o evento da fila de prioridades com a menor distância, propagando a frente de onda pelo intervalo correspondente para as arestas das faces que contêm o intervalo, determinando os intervalos de pontos que são otimamente alcançados pelos caminhos passando por I naquelas arestas e atualizando a fila de eventos e a lista de intervalos candidatos dos pares aresta-face envolvidos.

Algoritmo Dijkstra-Contínuo(s)

1. $d(s) := 0$.
2. Permanente(s).
3. para (cada aresta $e_{i,f} \subset f | s \in f$)
 - (a) $I_{i,f} := \text{Difundir}(s, e_{i,f})$.
 - (b) Atualizar-Intervalo($I_{i,f}, s$).
4. enquanto ($\text{Eventos} \neq \emptyset$) faça
 - (a) $q := \text{Desenfileira}(\text{Eventos})$.
 - (b) Permanente(q).
 - (c) se ($\Upsilon(q) = \text{Vértice}$) então
 - i. para (cada aresta $e_{i,f} \subset f | q \in f$)
 - A. $I_{i,f} := \text{Difundir}(q, e_{i,f})$.
 - B. Atualizar-Intervalo($I_{i,f}, q$).
 - (d) se ($\Gamma(q) = \text{Fronteira}$) então
 - i. para (cada aresta $e_i \subset f, f'$)
 - A. se ($e \neq e_i \subset f$) então
 - $I_i := \text{Projetar}(I, e_i)$.
 - se ($I_i \neq \emptyset$) então Atualizar-Intervalo(I_i, q).
 - B. se ($pc_a(I)$) então
 - $I_a := \text{Usar-Criticamente}(I, e_i, a)$.
 - se ($I_a \neq \emptyset$) então Atualizar-Intervalo(I_a, q).
 - C. se ($pc_b(I)$) então
 - $I_b := \text{Usar-Criticamente}(I, e_i, b)$.
 - se ($I_b \neq \emptyset$) então Atualizar-Intervalo(I_b, q).

Nas rotinas usadas pelo algoritmo Dijkstra contínuo, consideramos as seguintes notações. Denotamos por $e_{i,f}$ uma aresta de uma face f que contém um ponto evento q . Denotamos por e_i uma aresta de uma face que contém a aresta e do intervalo de otimalidade candidato associado ao evento q . $I_{i,f}$ denota um intervalo na aresta $e_{i,f}$, enquanto I_i é um intervalo na aresta e_i . I_a e I_b são intervalos que representam segmentos compartilhados. $pc_a(I)$ e $pc_b(I)$ verificam se uma extremidade de um intervalo candidato é um ponto crítico. Um intervalo I está associado a aresta e , enquanto um intervalo I_i

tem extremidades $[a_i, b_i]$, primeiros (últimos) pontos de cruzamento $[u_{a_i}, u_{b_i}]$ ($[w_{a_i}, w_{b_i}]$) e raiz r_i .

A função $Difundir(q, e_{i,f})$ encontra o subconjunto da aresta $e_{i,f}$ que é alcançado pelo vértice q segundo o critério de otimalidade local. A função retorna um intervalo candidato em $e_{i,f}$ cujos pontos são acessíveis por q . Para executar esta função, devemos verificar se o caminho mínimo para o vértice q é f -livre.

Função $Difundir(q, e_{i,f})$

1. $I_{i,f} := (e_{i,f}, q)$.
2. $\theta := \text{Ângulo-Crítico}(f, e_{i,f})$.
3. se $(q \in e_{i,f})$ então
 - (a) se $(\exists \theta)$ então $pc_a(I_{i,f}) := \text{Verdadeiro}$.
4. senão
 - (a) $I_{i,f} := \text{Retocar-Intervalo}(I_{i,f}, \theta)$.
 - (b) $I_{i,f} := \text{Fronteira}(I_{i,f})$.
5. retorne $(I_{i,f})$.

O passo 1 inicializa o intervalo $I_{i,f}$ com a extensão da aresta $e_{i,f}$, enquanto a raiz, os primeiros pontos de cruzamento, os últimos pontos de cruzamento e o predecessor de $I_{i,f}$ são o vértice q . O passo 2 obtém o ângulo crítico θ na aresta $e_{i,f}$ pela face f . No passo 3, se a aresta $e_{i,f}$ for incidente a q e se existe θ , então a aresta $e_{i,f}$ é usada criticamente e a função assinala a existência de um ponto crítico. O passo 4.a retoca o intervalo segundo o ângulo crítico θ , pois a aresta $e_{i,f}$ não incide em q . No passo 4.b, os pontos fronteira e de acesso são calculados pela função $\text{Fronteira}(I_{i,f})$. O passo 5 retorna o intervalo candidato $I_{i,f}$ (veja Fig. 4.2).

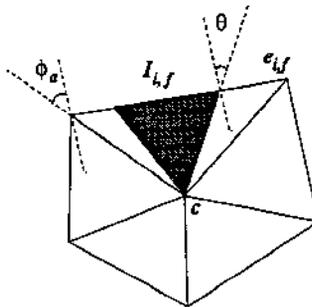


Figura 4.2: A função $Difundir(c, e_{i,f})$.

A função *Retocar-Intervalo*(I, θ) realiza o ajuste nas extremidades do intervalo I para que o ângulo de incidência nas extremidades não seja maior em valor absoluto que o ângulo crítico θ .

Função *Retocar-Intervalo*(I, θ)

1. $[\phi_a, \phi_b] := \text{Incidência}(I)$.
2. se $([\phi_a, \phi_b] \cap [-\theta, \theta] = \emptyset)$ então $I := \emptyset$.
3. se $([\phi_a, \phi_b] \not\subset [-\theta, \theta])$ então
 - (a) se $(-\theta \in [\phi_a, \phi_b])$ então
 - i. $a := \text{Encontrar-Ângulo}(-\theta, I)$.
 - ii. $pc_a(I) := \text{Verdadeiro}$.
 - (b) se $(\theta \in [\phi_a, \phi_b])$ então
 - i. $b := \text{Encontrar-ângulo}(\theta, I)$.
 - ii. $pc_b(I) := \text{Verdadeiro}$.
4. **retorne**(I).

O passo 1 calcula os ângulos de incidência, ϕ_a e ϕ_b , às extremidades do intervalo I . No passo 2, a função verifica se o intervalo I contém pontos que são alcançados com ângulos de incidência no intervalo $[-\theta, \theta]$. No passo 3, os ângulos de incidência não são menores em valor absoluto ao ângulo crítico. No passo 3.a; se $-\theta \in [\phi_a, \phi_b]$, então a função retoca o intervalo I e assinala sua extremidade a como um ponto crítico. No passo 3.b; se $\theta \in [\phi_a, \phi_b]$, então a função retoca o intervalo I e assinala sua extremidade b como um ponto crítico. O passo 4 retorna o intervalo candidato I .

A função *Encontrar-Ângulo*(θ, I) determina o ponto a em I tal que o caminho geodésico da raiz de I pela última seqüência de arestas alcançará o ponto a com ângulo de incidência θ . A função retorna o ponto de alcance a , o primeiro ponto de cruzamento u correspondente, e o último ponto de cruzamento w correspondente. Se não existe um ponto a , a função retorna \emptyset .

Função *Encontrar-Ângulo*(θ, I)

1. $\phi := \text{Refração}(\theta, I)$.
2. se $(\exists \phi)$ então
 - (a) $a := \text{Caminho-Geodésico}(\phi, I, e)$.
 - (b) se $(a \in I)$ então **retorne**(a).
3. **retorne**(\emptyset).

No passo 1, chamamos a função $Refração(\theta, I)$ que calcula o ângulo de incidência ϕ na primeira aresta da seqüência de arestas E de I tal que um caminho geodésico que inicia na raiz r de E no ângulo ϕ passe por E e incida na aresta e que contém I com ângulo de incidência θ . Iniciando com o ângulo θ na aresta e , determinamos os ângulos de incidência em cada aresta na última seqüência de arestas E . Para isso, utilizamos os predecessores dos intervalos e a especificação unívoca do ângulo de incidência através do critério de otimalidade local. Quando o ângulo de incidência em alguma aresta é maior em valor absoluto ao ângulo crítico, não existe ângulo ϕ . No passo 2.a, traçamos o caminho geodésico de r com ângulo ϕ pela última seqüência de arestas. Se o caminho geodésico traçado deixa a seqüência, então não existe um ponto a . Caso contrário, o caminho geodésico intercepta a aresta e em algum ponto a com ângulo de incidência θ . Se $a \in I$, então retornamos a no passo 3.b. Caso contrário, quando não existe um ponto a , retornamos \emptyset no passo 3.

A função $Projetar(I, e_i)$ encontra o subconjunto de e_i que é alcançado por caminhos geodésicos através de I , obedecendo ao critério de otimalidade local. A função retorna um intervalo candidato em e_i , cujos pontos são acessíveis por I . Se nenhuma parte de e_i é acessível, retorna \emptyset .

Função $Projetar(I, e_i)$

1. $[\phi_a, \phi_b] := \text{Incidência}(I)$.
2. $(a_i, b_i) := \text{Critério-Otimalidade}(e, e_i, [\phi_a, \phi_b])$.
3. se $(a_i \in e_i)$ então
 - (a) $u_{a_i} := u_a$.
 - (b) $w_{a_i} := a$.
4. senão $I_i := \emptyset$.
5. se $(b_i \in e_i)$ então
 - (a) $u_{b_i} := u_b$.
 - (b) $w_{b_i} := b$.
6. senão
 - (a) $b_i := v$.
 - (b) $(u_{b_i}, w_{b_i}) := \text{Encontrar-Ponto}(I, v)$.
7. $\theta := \text{Ângulo-Crítico}(f, e_i)$
8. $I_i := \text{Retocar-Intervalo}(I_i, \theta)$.
9. $r_i := r$.
10. $\bar{I}_i := I$.
11. $I_i := \text{Fronteira}(I_i)$.

12. `retorne(I_i)`.

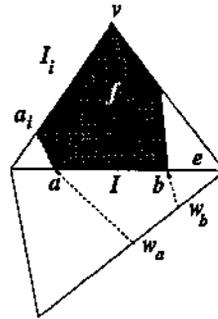


Figura 4.3: A função $Projetar(I, e_i)$.

O passo 1 calcula os ângulos de incidência, ϕ_a e ϕ_b , às extremidades do intervalo I , determinados pelos pontos w_a e w_b (veja Fig. 4.3). O passo 2 obtém os pontos a_i e b_i que resultam do critério de otimalidade local nos pontos extremos de I na aresta e_i pela face f . Os passos 3 a 6 atribuem as extremidades, primeiros e últimos pontos de cruzamento do intervalo I_i . O passo 6.b chama $Encontrar-Ponto(I, v)$ que retorna o primeiro e último pontos de cruzamento no caminho geodésico da raiz de I para o vértice v passando pela última seqüência de arestas de I . $Encontrar-Ponto(I, v)$ é um procedimento de busca binária que calcula um pequeno intervalo contendo o primeiro ponto de cruzamento no caminho geodésico de r para v . No passo 8, ajustamos o intervalo I segundo o ângulo crítico θ . Os passos 9 a 11 atribuem a raiz, intervalo predecessor, pontos fronteira e de acesso ao intervalo I_i . O passo 12 retorna o intervalo I_i .

A função $Usar-Criticamente(I, e_i, a)$ estabelece o ponto $a \in I$ como um ponto crítico para a aresta e . Do ponto a , caminhos podem seguir sobre a aresta e por alguma distância e sair com ângulo crítico para uma das faces adjacentes a e . Portanto, deve-se instanciar intervalos candidatos nas arestas e_i opostas a e .

Função $Usar-Criticamente(I, e_i, a)$

1. se $(e_i = e)$ então
 - (a) $d(v^*) := \min\{d(v^*), d(a) + d_{a,e}(a, v^*)\}$.
 - (b) $Incluir-Evento(Eventos, v^*)$.
 - (c) `retorne(\emptyset)`.
2. $\theta := \hat{Angulo-Critico}(f, e)$.
3. se $(e \neq e_i \subset f \wedge \exists \theta)$ então
 - (a) $a_i := Intersecao(a, \theta, e, e_i)$.

- (b) se $(v^* \in e_i)$ então
 - i. se $(a_i \in e_i)$ então $I_i := [a_i, v^*]$.
 - ii. senão $I_i := e_i$.
 - (c) senão
 - i. se $(a_i \in e_i)$ então $I_i := [a_i, v]$.
 - ii. senão retorne(\emptyset).
4. $\theta' := \text{Ângulo-Crítico}(f', e)$.
 5. se $(e \neq e_i \subset f' \wedge \exists \theta')$ então
 - (a) $a'_i := \text{Interseção}(a, \theta', e, e_i)$.
 - (b) se $(v^* \in e_i)$ então
 - i. se $(a'_i \in e_i)$ então $I_i := [a'_i, v^*]$.
 - ii. senão $I_i := e_i$.
 - (c) senão
 - i. se $(a'_i \in e_i)$ então $I_i := [a'_i, v']$.
 - ii. senão retorne(\emptyset).
 6. $r_i := a$.
 7. $\bar{I}_i := a$.
 8. $I_i := \text{Fronteira}(I_i)$.
 9. retorne(I_i).

No passo 1, se a aresta e_i é a aresta que contém o intervalo I , então a função inclui o vértice v^* na fila de prioridades *Eventos* e retorna \emptyset (veja Fig. 4.4). O passo 2 computa o ângulo crítico na aresta e pela face f . O passo 3 obtém o ponto a_i que resulta da incidência do caminho geodésico pela face f na aresta e_i e ajusta o intervalo I_i para estar contido em e_i segundo os vértice v^* ou v . O passo 4 encontra o ângulo crítico na aresta e pela face f' . O passo 5 obtém o ponto a'_i que resulta da incidência do caminho geodésico criticamente refletido pela face f' na aresta e_i e ajusta o intervalo I_i para estar contido em e_i segundo os vértices v^* ou v' . Os passos 6 a 8 atribuem a raiz, intervalo predecessor, pontos fronteira e de acesso ao intervalo I_i . Finalmente, o passo 9 retorna o intervalo I_i .

O procedimento *Atualizar-Intervalo*(I, q) atualiza os intervalos existentes no par aresta-face (e, f) do novo intervalo candidato I . O procedimento verifica dominância entre intervalos e realiza os ajustes necessários. Dizemos que um intervalo I *domina* outro I_1 em um ponto de interseção γ quando existe um caminho geodésico passando por (r, E) com comprimento menor do que um caminho geodésico passando por (r_1, E_1) para γ . O procedimento cria uma instância do intervalo ajustado I e inclui os eventos correspondentes na fila de prioridades.

Procedimento *Atualizar-Intervalo*(I, q)

- iii. se $(d(r_1) + \lambda_1 < d(r) + \lambda)$ então retorne(\emptyset).
 - iv. senão $a := b_1 := \text{Ponto-Empate}(I_1, I)$.
- (c) se $(I \supset I_1)$ então
- i. $\lambda := \text{Encontrar-Ponto}(\bar{I}, a_1)$.
 - ii. $\lambda_1 := \text{Encontrar-Ponto}(\bar{I}_1, a_1)$.
 - iii. se $(d(r) + \lambda < d(r_1) + \lambda_1)$ então $I_1 := \text{Excluir-Intervalo}(I_1)$
 - iv. senão $a := b_1 := \text{Ponto-Empate}(I_1, I)$.
- (d) se $(a_1 \in I \wedge b_1 \notin I)$ então
- i. $\lambda := \text{Encontrar-Ponto}(\bar{I}, b)$.
 - ii. $\lambda_1 := \text{Encontrar-Ponto}(\bar{I}_1, b)$.
 - iii. se $(d(r) + \lambda < d(r_1) + \lambda_1)$ então $I_1 := \text{Excluir-Intervalo}(I_1)$
 - iv. senão
 - A. $\lambda := \text{Encontrar-Ponto}(\bar{I}, a_1)$.
 - B. $\lambda_1 := \text{Encontrar-Ponto}(\bar{I}_1, a_1)$.
 - C. se $(d(r_1) + \lambda_1 < d(r) + \lambda)$ então retorne(\emptyset)
 - D. senão $a := b_1 := \text{Ponto-Empate}(I_1, I)$.
- (e) se $(a_1 > b)$ então
- i. $\gamma := \overline{w_b b} \cap \overline{w_{a_1} a_1}$.
 - ii. $\lambda := \text{Encontrar-Ponto}(\bar{I}, \gamma)$.
 - iii. $\lambda_1 := \text{Encontrar-Ponto}(\bar{I}_1, \gamma)$.
 - iv. se $(d(r) + \lambda < d(r_1) + \lambda_1)$ então $I_1 := \text{Excluir-Intervalo}(I_1)$
 - v. senão retorne(\emptyset).

2. retorne(I).

A função $\text{Ponto-Empate}(I_1, I)$ determina o ponto de empate que será a nova fronteira entre I e seu vizinho I_1 . O procedimento $\text{Excluir-Intervalo}(I_1)$ exclui I_1 da lista de intervalos do par aresta-face correspondente e exclui quaisquer entradas de pontos de evento em I_1 da fila de prioridades.

4.3 Funções Numéricas

O algoritmo Dijkstra contínuo usa duas funções numéricas para resolver subproblemas numéricos do PGAD. A função $\text{Encontrar-Ponto}(I, a)$ determina os pontos de interseção entre um caminho geodésico e uma seqüência de arestas. A função $\text{Ponto-Empate}(I, I')$ encontra o *ponto de empate* q que satisfaz a equação $d(r) + d_{r,E}(q) = d(r') + d_{r',E'}(q)$ para os intervalos candidatos I e I' .

A função $\text{Encontrar-Ponto}(I, a)$ determina um caminho da raiz r de I para a que intercepta a última seqüência de arestas de I . O problema de encontrar o caminho geodésico de r a a é intratável. Portanto, a função retorna um caminho cujo comprimento é no máximo $(1 + \varepsilon)$ vezes o comprimento do caminho mínimo, onde $\varepsilon \in (0, 1)$ é a *tolerância de erro*.

A função realiza uma busca binária para determinar um ponto $u \in [u_a, u_b]$ tal que o caminho geodésico de r por u passe pelo ponto a . Quando u está contido em um intervalo com tamanho menor que um certo δ , a função avança para a aresta seguinte procurando encontrar um caminho geodésico aproximado para a . $\delta = \frac{wh_e}{\delta k W}$, onde w (W) é o módulo mínimo (máximo) que a força externa pode assumir, h é o menor comprimento de um segmento perpendicular a uma aresta passando pelo vértice oposto e k é o tamanho da seqüência de arestas associada a I . A função retorna o comprimento do caminho de r para a , o primeiro e o último pontos de cruzamento do caminho para a .

Na especificação das funções numéricas usamos a seguinte notação. Denotamos por j um índice para uma aresta da última seqüência de arestas associada a um intervalo candidato I . A aresta que contém o intervalo I é denotada por e . Denotamos por $[a_i, b_i]$ as extremidades de um intervalo na aresta e_i e por q_i o ponto de interseção do caminho geodésico nesta aresta.

Função Encontrar-Ponto(I, a)

1. $j := 1$.
2. $a_1 := u_a$.
3. $b_1 := u_b$.
4. enquanto ($j \neq k$) faça
 - (a) enquanto ($|a_j b_j| \geq \delta$) faça
 - i. $u = \frac{a_j + b_j}{2}$.
 - ii. $a' := \text{Caminho-Geodésico}(u, I, e)$.
 - iii. se ($a' \in e$) então
 - A. se ($a \leq a'$) então $b_j := u$
 - B. senão $a_j := u$.
 - iv. senão
 - A. se ($a' \geq e_i$) então $b_j := u$
 - B. senão $a_j := u$.
 - (b) $q_j = \frac{a_j + b_j}{2}$.
 - (c) $a_{j+1} := \text{Caminho-Geodésico}(a_j, I, e_{j+1})$.
 - (d) $b_{j+1} := \text{Caminho-Geodésico}(b_j, I, e_{j+1})$.
 - (e) $j := j + 1$.
5. $\lambda := d(r) + d_{r,E}(r, a)$.
6. retorne (λ, q_1, q_k).

O passo 4.a verifica o tamanho do intervalo que contém u . O passo 4.a.ii traça o caminho geodésico originado na raiz r de I passando por u . O caminho geodésico termina quando encontra a aresta e (passo 4.a.iii) ou quando deixa a última seqüência de arestas

de I (passo 4.a.iv). O passo 4.a.iii.A verifica se a está à esquerda de a' , enquanto o passo 4.a.iv.A verifica se a' está à direita da aresta e_i que não foi alcançada pelo caminho geodésico. No passo 4.b, a função avança pela seqüência de arestas. O passo 4.c (4.d) faz a_{j+1} (b_{j+1}) ser o ponto na aresta seguinte, e_{j+1} , que intercepta o caminho geodésico que passa por e_j quando o caminho geodésico é traçado a partir de r passando por a_j (b_j). O passo 5 calcula o comprimento do caminho de r para a passando por q_1, q_2, \dots, q_k .

A função *Ponto-Empate*(I, I') procura o ponto de empate para ajustar os intervalos candidatos I e I' que se sobrepõem. Se não existe um ponto cuja diferença entre as distâncias de s passando pelas raízes de I e I' é menor que ϵ , então a função retorna uma extremidade do intervalo que domina o outro.

Função Ponto-Empate(I, I')

1. $j, j' := 1$.
2. $[a_1, b_1] := [u_a, u_b]$.
3. $[a'_1, b'_1] := [u'_a, u'_b]$.
4. **enquanto** ($j < k \wedge j' < k'$) **faça**
 - (a) **se** ($|a_j b_j| < \delta$) **então**
 - i. $q_j = \frac{a_j + b_j}{2}$.
 - ii. $[a_{j+1}, b_{j+1}] := [\text{Caminho-Geodésico}(a_j, I, e_{j+1}), \text{Caminho-Geodésico}(b_j, I, e_{j+1})]$.
 - iii. $j := j + 1$.
 - (b) **senão se** ($|a'_{j'} b'_{j'}| < \delta$) **então**
 - i. $q'_{j'} = \frac{a'_{j'} + b'_{j'}}{2}$.
 - ii. $[a'_{j'+1}, b'_{j'+1}] := [\text{Caminho-Geodésico}(a'_{j'}, I', e_{j'+1}), \text{Caminho-Geodésico}(b'_{j'}, I', e_{j'+1})]$.
 - iii. $j' := j' + 1$.
 - (c) **senão**
 - i. $u = \frac{a_j + b_j}{2}$.
 - ii. $u' = \frac{a'_{j'} + b'_{j'}}{2}$.
 - iii. $w := \text{Caminho-Geodésico}(u, I, e)$.
 - iv. $w' := \text{Caminho-Geodésico}(u', I', e)$.
 - v. **se** ($w, w' \in e$) **então**
 - A. **se** ($w' \geq b$) **então** $b'_j := u'$ **senão** $a'_{j'} := u'$.
 - B. **se** ($w \geq b'$) **então** $b_j := u$ **senão** $a_j := u$.
 - C. **se** ($w' \geq w$) **então**
 - $I'_L := [a'_{k'}, w']$.
 - $I_L := [a_k, w]$.
 - $a'_{j'} := \text{Ponto-Empate}(I_L, I'_L)$.
 - $I_R := [w, b_k]$.
 - $I'_R := [w', b'_{k'}]$.
 - $b_j := \text{Ponto-Empate}(I_R, I'_R)$.
 - D. **senão**

- $\gamma := p_f(w) \cap p_f(w')$.
- se $(d(r') + d(r', \gamma) < d(r) + d(r, \gamma))$ então
 - $I_R := [a_k, w]$.
 - $I'_R := [w', b'_{k'}]$.
 - $a_j := \text{Ponto-Empate}(I_R, I'_R)$.
- senão
 - $I'_L := [w', a'_{k'}]$.
 - $I_L := [a_k, w]$.
 - $b'_{j'} := \text{Ponto-Empate}(I_L, I'_L)$.

vi. senão

- A. se $(w \geq e_i)$ então $b_j := u$ senão $a_j := u$.
- B. se $(w' \geq e'_i)$ então $b'_{j'} := u'$ senão $a'_{j'} := u'$.

5. $q_k := \frac{a_k + b_k}{2}$.

6. $q'_{k'} := \frac{a'_{k'} + b'_{k'}}{2}$.

7. $\lambda := d(r) + d(r, q_k)$.

8. $\lambda' := d(r') + d(r', q'_{k'})$.

9. se $(\lambda' \notin ((1 - \varepsilon)\lambda, (1 + \varepsilon)\lambda))$ então

(a) $\lambda := \text{Encontrar-Ponto}(\vec{I}, b)$.

(b) $\lambda' := \text{Encontrar-Ponto}(\vec{I}', b)$.

(c) se $(d(r) + \lambda < d(r') + \lambda')$ então retorne (b) senão retorne (a') .10. senão retorne $\left(\frac{q_k + q'_{k'}}{2}\right)$.

O passo 4.a verifica o tamanho do intervalo $\overline{a_j b_j}$ e avança sobre a seqüência de arestas, fazendo a_{j+1} (b_{j+1}) ser o ponto na aresta e_{j+1} que é interceptado pelo caminho geodésico originado de q_{j-1} passando por a_j (b_j). O passo 4.b é análogo ao passo 4.a considerando I' . O passo 4.c.iii traça o caminho geodésico originado de r passando por u . O passo 4.c.iv traça o caminho geodésico originado de r' passando por u' . Novamente, os caminhos geodésicos terminam quando encontram a aresta e ou quando deixam a seqüência de arestas correspondente. O passo 4.c.v verifica se os caminhos geodésicos traçados encontram a aresta e , então analisa os seguintes casos: $w' \notin [a, b]$ (passo 4.c.v.A), $w \notin [a', b']$ (passo 4.c.v.B), w' está à direita de w (passo 4.c.v.C) e w está à esquerda de w' (passo 4.c.v.D). Todos os casos levam a ajustes nos intervalos $\overline{a_j b_j}$ e $\overline{a'_{j'} b'_{j'}}$. No último caso, o caminho geodésico para w deve cruzar o caminho geodésico para w' e o ajuste depende da distância ao ponto de interseção γ . O passo 4.c.vi.A verifica a situação onde o caminho geodésico para w deixa a seqüência de arestas. O passo 4.c.vi.B faz o mesmo considerando w' . O passo 6 calcula o comprimento do caminho de r para q_k , enquanto o passo 7 calcula o comprimento do caminho de r' para $q'_{k'}$.

4.4 Prova de Corretude

Apresentamos a prova de que, na conclusão do algoritmo Dijkstra contínuo direcional, as listas de intervalos de otimalidade dos pares aresta-face permitem computar corretamente os caminhos mínimos do ponto origem aos demais pontos da superfície.

Teorema 4.1 *As listas de intervalos de otimalidade dos pares aresta-face permitem computar corretamente os caminhos mínimos localmente f -livres do ponto origem aos pontos desses intervalos dentre os caminhos localmente f -livres que passam pelas arestas por intervalos cujo ponto fronteira já foi rotulado permanentemente.*

Prova. A prova procede por indução no número de eventos rotulados permanentemente. No caso base, quando o ponto origem s é rotulado permanentemente (passo 2), os intervalos candidatos são os intervalos nas arestas das faces incidentes ao ponto origem. A função $Difundir(s, e_{i,f})$ pode criar um intervalo em cada aresta, obedecendo ao critério de otimalidade local (passo 3.a). Os intervalos são incluídos nas listas de intervalos correspondentes aos respectivos pares aresta-face pelo procedimento $Atualizar-Intervalo(I_{i,f}, s)$ (passo 3.b), tendo em vista que as listas de intervalos estão vazias inicialmente. Dessa forma, estes intervalos permitem computar corretamente os caminhos mínimos passando apenas pelo intervalo degenerado que consiste no ponto origem já rotulado permanentemente e isto estabelece a base da indução.

A hipótese de indução consiste em assumir a validade do teorema quando k eventos foram rotulados permanentemente. Um intervalo candidato terá seu ponto fronteira rotulado permanentemente, assegurando que alguma parte dele se tornará parte de um intervalo final de otimalidade com os mesmos atributos que o intervalo candidato.

O passo de indução requer a demonstração de que, ao rotularmos permanentemente o $k + 1$ -ésimo evento, a introdução de um intervalo I criado não invalida o teorema. Portanto, o melhor caminho localmente f -livre para pontos de I é por sua raiz r e última seqüência de arestas correspondente, passando por intervalos predecessores cujos pontos fronteira já foram rotulados permanentemente.

O critério de otimalidade local garante que todos os intervalos candidatos possíveis propagados através do ponto evento são criados corretamente pela função $Difundir(q, e_{i,f})$ (passo 4.c), $Projetar(I, e_i)$ (passo 4.d.i.A) ou $Usar-Criticamente(I, e_i, a)$ (passos 4.d.i.B e 4.d.i.C). Estes intervalos são incluídos pelo procedimento $Atualizar-Intervalo(I, q)$, cuja corretude será mostrada no lema seguinte. \square

Vamos demonstrar que os intervalos candidatos são incluídos corretamente pelo procedimento $Atualizar-Intervalo(I, q)$, considerando os intervalos já existentes. Um intervalo candidato será excluído em $Atualizar-Intervalo(I, q)$, quando garantimos que o intervalo de otimalidade correspondente ao candidato será vazio.

Lema 4.1 *Se incluímos o intervalo I entre os intervalos I_1 e I_2 na aresta e ou excluimos o intervalo I_1 no procedimento Atualizar-Intervalo(I, q), então a lista de intervalos resultantes da aresta e subdivide corretamente os pontos de e segundo os melhores caminhos localmente f -livres computados até então.*

Prova. Visto que o ponto fronteira $\bar{c} \in \bar{I}$ foi localizado no canal de acesso (I_1, I_2) , sabemos que \bar{c} está à direita do segmento $\overline{c'_1 c_1}$, onde c'_1 e c_1 são os pontos de acesso e fronteira do intervalo I_1 . Portanto, \bar{c} está à direita do caminho geodésico a partir de r_1 para b_1 , pois $\bar{c} \notin \bar{I}_1$. Visto que \bar{I} é conexo, então todos os pontos de \bar{I} estão à direita do caminho geodésico a partir de r_1 para qualquer ponto de I_1 .

Analisando as disposições entre os intervalos I e I_1 , identificamos os seguintes casos distintos: (*) $I \cap I_1 = \emptyset$ e I está à direita de I_1 ; (a) $a_1 \notin I$ e $b_1 \in I$; (b) $I \subset I_1$; (c) $I \supset I_1$; (d) $a_1 \in I$ e $b_1 \notin I$; e (e) $I \cap I_1 = \emptyset$ e I está à esquerda de I_1 (veja Fig. 4.5).

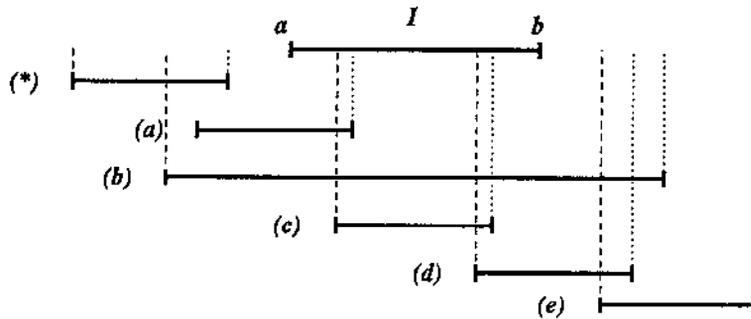


Figura 4.5: Casos distintos entre os intervalos I e I_1 .

(*) Assuma que $I \cap I_1 = \emptyset$ e I está à direita de I_1 .

Seja w a interseção do caminho geodésico originado em r para a extremidade a do intervalo I com a aresta e' . Visto que $a \in I$, então $w \in \bar{I}$, pelo método usado para construir I de seu predecessor, \bar{I} . O ponto w está estritamente à direita do segmento $\overline{w_1 b_1}$, o último segmento no caminho geodésico a partir de r_1 para b_1 . Além disso, a está estritamente à direita do segmento $\overline{w_1 b_1}$, então os segmentos $\overline{w a}$ e $\overline{w_1 b_1}$ não se interceptam (veja Fig. 4.6).

Neste caso, os caminhos mínimos localmente f -livres da origem aos pontos de I e I_1 não se cruzam. Dessa forma, não existe ponto em I que seja melhor alcançado por r_1 e vice-versa.

(a) Assuma que $a_1 \notin I$ e $b_1 \in I$.

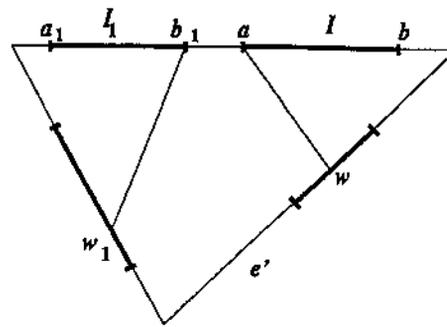


Figura 4.6: Caso (*).

Agora, a está estritamente à esquerda do segmento $\overline{w_1 b_1}$, então os segmentos $\overline{w_1 a}$ e $\overline{w_1 b_1}$ devem interceptar-se em algum ponto γ que é interior a ambos os segmentos (veja Fig. 4.7). Além disso, para todo ponto $q' \in [a, b_1]$, existe um ponto $q'_1 \in (q', b_1]$ tal que o caminho geodésico originado em r para q' cruza o caminho geodésico originado em r_1 para q'_1 . Analogamente, para todo ponto $q'_1 \in (a, b_1]$, existe um ponto $q' \in [a, q'_1)$ tal que o caminho geodésico originado em r_1 para q'_1 cruza o caminho geodésico originado em r para q' .

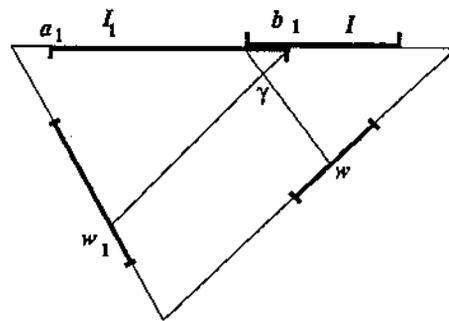


Figura 4.7: Caso (a).

Neste caso, os caminhos mínimos localmente f -livres da origem aos pontos de I e I_1 se cruzam no intervalo $[a, b_1]$. Assim, existe um único ponto q^* que é alcançado igualmente por I e I_1 , ou o intervalo $[a, b_1]$ é dominado exclusivamente por I ($q^* = a$) ou I_1 ($q^* = b_1$).

Seja $q' \in I = [q^*, b]$, onde $[q^*, b]$ é a extensão de I depois de qualquer ajuste. Se q' não está em $[a_1, b_1]$, então é melhor alcançar q' pela raiz r do que pela raiz r_1 . Se q' é interior a $[a_1, b_1]$, então $q^* \in [a_1, b_1]$. Contudo, o caminho para q^* pela raiz

r não é maior que o caminho para q^* pela raiz r_1 e é mínimo até então. Se fosse melhor alcançar q' pela raiz r_1 do que pela raiz r , então o caminho geodésico de r_1 para q' interceptaria o caminho geodésico de r para q^* . Dessa forma, contradizendo o Lema 3.7, então temos $I_1 = [a_1, q^*]$.

(b) Assuma que $I \subset I_1$.

Agora, para todo ponto $q'_1 \in (a, b_1]$, existe um ponto $q' \in [a, q'_1] \cap [a, b]$ tal que o caminho geodésico originado em r para q' cruza o caminho geodésico originado em r_1 para q'_1 (veja Fig. 4.8).

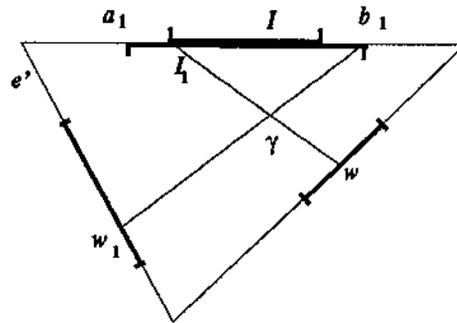


Figura 4.8: Caso (b).

Novamente, os caminhos mínimos localmente f -livres da origem aos pontos de I e I_1 se cruzam no intervalo $[a, b_1]$. Assim, existe um ponto q^* que é alcançado igualmente por I e I_1 , ou o intervalo $[a, b]$ é dominado exclusivamente por I ($q^* = a$) ou I_1 ($q^* = b_1$). Quando $q^* \in I$, $[b, b_1]$ é melhor alcançado por r .

iii. Assuma que $d(r_1) + d_{r_1, E_1}(b) < d(r) + d_{r, E}(b)$.

O melhor caminho localmente f -livre computado até então para o ponto b é pela raiz r_1 e última seqüência de arestas E_1 . Seja w_1 a interseção do caminho geodésico originado em r_1 para b com aresta e' . Visto que $b \in I_1$, então $w_1 \in \bar{I}_1$, pelo método usado para construir I_1 de seu predecessor, \bar{I}_1 . Vamos supor que existe um ponto q' no interior do intervalo candidato I para (r, E) segundo (e, f) .

Note que q' é interior a $[a, b]$, senão q' seria inacessível a partir de r ou poderia ser alcançado de alguma outra raiz de (e, f) por um caminho localmente f -livre menor. O ponto w_1 está estritamente à esquerda do segmento $\overline{wq'}$, o último segmento no caminho geodésico a partir de r através de E para q' . Contudo, b está estritamente à direita do segmento $\overline{wq'}$, então os segmentos $\overline{w_1b}$ e $\overline{wq'}$ devem interceptar-se em algum ponto γ que é interior a ambos segmentos (veja

Fig. 4.8). Visto que γ está no caminho $p_f(q')$, o subcaminho de $p_f(q')$ a partir de r para γ não é menor que o subcaminho de $p_f(b)$ a partir de r_1 para γ . Isso significa que podemos ter uma melhora estrita ao caminho para q' indo da raiz r_1 para q' (pois $d_{r_1, E_1}(w_1) + d(w_1, q') < d_{r, E}(\gamma) + d(\gamma, q')$). Assim, nenhum ponto q' interior ao intervalo candidato para (r, E) segundo (e, f) pode existir e justificamos a não inclusão do intervalo candidato I .

iv. Assuma que $d(r) + d_{r, E}(b) \leq d(r_1) + d_{r_1, E_1}(b)$.

Seja $I = [q^*, b]$ depois de qualquer ajuste, então, analogamente ao caso (a), temos $I_1 = [a_1, q^*]$.

(c) Assuma que $I \supset I_1$.

A prova para este caso é análoga ao caso (b).

(d) Assuma que $a_1 \in I$ e $b_1 \notin I$.

Novamente, a está estritamente à esquerda do segmento $\overline{w_1 b_1}$, então os segmentos \overline{wa} e $\overline{w_1 b_1}$ devem interceptar-se em algum ponto γ que é interior a ambos segmentos (veja Fig. 4.9). Além disso, para todo ponto $q'_1 \in [a_1, b_1]$, existe um ponto $q' \in [a, q'_1] \cap [a, b]$ tal que o caminho geodésico originado em r para q' cruza o caminho geodésico originado em r_1 para q'_1 . Analogamente, para todo ponto $q' \in [a, b]$, existe um ponto $q'_1 \in [q', b_1] \cap [a_1, b_1]$ tal que o caminho geodésico originado em r_1 para q'_1 cruza o caminho geodésico originado em r para q' .

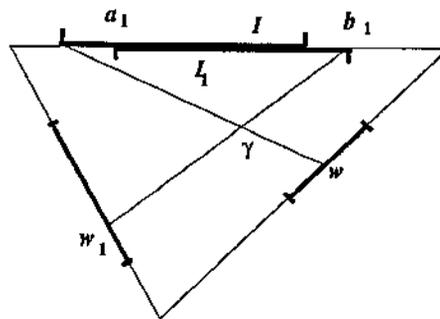


Figura 4.9: Caso (d).

Neste caso, os caminhos mínimos localmente f -livres da origem aos pontos de I e I_1 se cruzam no intervalo $[a, b_1]$. Existe um único ponto q^* alcançado igualmente por I e I_1 ou $[a_1, b]$ é melhor alcançado por I ou I_1 . Quando $[a_1, b]$ é melhor alcançado por I , temos que $d(r) + d_{r, E}(a_1) < d(r_1) + d_{r_1, E_1}(a_1)$ e $[b, b_1]$ é melhor alcançado por r . Assim, excluimos I_1 . Quando $[a_1, b]$ é dominado por I_1 , temos que

$d(r_1) + d_{r_1, E_1}(b) < d(r) + d_{r, E}(b)$ e $[a, a_1]$ é melhor alcançado por r_1 . Assim, não incluímos I . Se existe um ponto q^* , temos que $[a, a_1]$ é melhor alcançado por r_1 e $[b, b_1]$ é melhor alcançado por r , resultando os intervalos $I = [q^*, b]$ e $I_1 = [a_1, q^*]$.

iii. Assuma que $d(r) + d_{r, E}(a_1) < d(r_1) + d_{r_1, E_1}(a_1)$.

Simétrico ao caso b.iii, o melhor caminho localmente f -livre computado até então para o ponto a_1 é pela raiz r e última seqüência de arestas E . Portanto, nenhum ponto interior ao intervalo candidato para (r_1, E_1) segundo (e, f) pode existir e justificamos a exclusão do intervalo candidato I_1 .

iv.C. Assuma que $d(r_1) + d_{r_1, E_1}(b) < d(r) + d_{r, E}(b)$.

Novamente, como no caso b.iii, o melhor caminho localmente f -livre computado até então para o ponto b é pela raiz r_1 e última seqüência de arestas E_1 . Portanto, nenhum ponto interior ao intervalo candidato para (r, E) segundo (e, f) pode existir e justificamos a não inclusão do intervalo candidato I .

iv.D. Assuma que $d(r_1) + d_{r_1, E_1}(a_1) \leq d(r) + d_{r, E}(a_1)$ e $d(r) + d_{r, E}(b) \leq d(r_1) + d_{r_1, E_1}(b)$.

Análogo ao caso (a), seja $q' \in I = [q^*, b]$, então $I_1 = [a_1, q^*]$ depois de qualquer ajuste.

(e) Assuma que $I \cap I_1 = \emptyset$ e I está à esquerda de I_1 .

Agora, seja w a interseção do caminho geodésico originado em r para b com a aresta e' . Portanto, b está estritamente à esquerda do segmento $\overline{w_1 a_1}$, então os segmentos \overline{wb} e $\overline{w_1 a_1}$ devem interceptar-se em algum ponto γ que é interior a ambos segmentos (veja Fig. 4.10). Além disso, para todo ponto $q'_1 \in [a_1, b_1]$, existe um ponto $q' \in [a, b]$ tal que o caminho geodésico originado em r para q' cruza o caminho geodésico originado em r_1 para q'_1 . Analogamente, para todo ponto $q' \in [a, b]$, existe um ponto $q'_1 \in [a_1, b_1]$ tal que o caminho geodésico originado em r_1 para q'_1 cruza o caminho geodésico originado em r para q' .

Neste caso, os caminhos mínimos localmente f -livres da origem aos pontos de I e I_1 se cruzam nos intervalos $[a, b]$ e $[a_1, b_1]$. Assim, o caminho $p_f(b)$ a partir de r intercepta o caminho $p_f(a_1)$ a partir de r_1 em γ . Se $d(r) + d_{r, E}(\gamma) < d(r_1) + d_{r_1, E_1}(\gamma)$, então excluimos o intervalo I_1 , senão não incluímos I .

iv. Assuma que $d(r) + d_{r, E}(\gamma) < d(r_1) + d_{r_1, E_1}(\gamma)$.

Análogo ao caso b.iii., o melhor caminho localmente f -livre computado até então para o ponto γ é pela raiz r e última seqüência de arestas E . Portanto, nenhum ponto interior ao intervalo candidato para (r_1, E_1) segundo (e, f) pode existir e justificamos a exclusão do intervalo candidato I_1 .

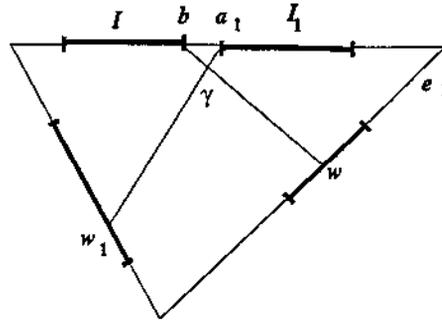


Figura 4.10: Caso (e).

v. Assuma que $d(r_1) + d_{r_1, E_1}(\gamma) \leq d(r) + d_{r, E}(\gamma)$.

Agora, o melhor caminho localmente f -livre computado até então para o ponto γ é pela raiz r_1 e última seqüência de arestas E_1 . Assim, simetricamente ao caso anterior, nenhum ponto interior ao intervalo candidato para (r, E) segundo (e, f) pode existir e justificamos a não inclusão do intervalo candidato I .

□

No algoritmo Dijkstra contínuo, os eventos ocorrem à medida que a frente de onda encontra novas arestas. Na execução do algoritmo, se um evento ocorre a uma certa distância de s , então todos os pontos mais próximos a s do que este evento já encontraram a frente de onda. Além disso, demonstramos que um ponto evento é rotulado permanentemente com o comprimento do caminho mínimo da origem.

Lema 4.2 [95] *Se o algoritmo acabou de atribuir um rótulo permanente δ para algum ponto, então para qualquer ponto q em uma aresta e tal que $d(q) \leq \delta$, existe um intervalo I para (r, E) na lista de intervalos correspondente à (e, f) que contém q tal que um caminho mínimo para q tenha raiz r , última seqüência de arestas E e comprimento $d(r) + d_{r, E}(q)$.*

Prova. Seja $q \in e$ um ponto tal que $d(q) \leq \delta$ (veja Fig. 4.11). Suponha que q não está em um intervalo candidato que corretamente identifique um caminho mínimo para q . Seja $p(q)$ um caminho mínimo para q . Agora, $p(q)$ intercepta pelo menos um intervalo candidato, então seja $I_j = [a_j, b_j]$ o último intervalo candidato das listas de intervalos interceptado por $p(q)$, e suponha que I_j é um intervalo relativo a (e_j, f_j) . Se $p_f(q)$ interceptar I_j em uma de suas extremidades, pegue qualquer intervalo. No caso onde $p_f(q) \cap I_j$ é um vértice v , pegue I_j tal que $p_f(q) \cap f_j \neq v$. A suposição sobre q diz que $q \notin I_j$. Seja $\overline{q'q'} = f_j \cap p(q)$, então o ponto q' está em uma aresta $e_{j+1} = f_j \cap f_{j+1}$. Seja $p_{f_{j+1}}(q')$ o subcaminho de $p(q)$ a partir de s a q' . Claramente, $p_{f_{j+1}}(q')$ é um caminho mínimo

localmente f_{j+1} -livre para q' . O intervalo I_j deve ter sido propagado, pois a distância ao ponto fronteira de I_j é $\leq d(q'') < d(q) \leq \delta$, e o algoritmo sempre rotula permanentemente o ponto evento mais próximo da origem. Mas então, pelo Teorema 4.1, teria sido criado um intervalo candidato na aresta e_{j+1} que contém o ponto q' . Isso contradiz o fato de que I_j é o último intervalo candidato interceptado por $p(q)$. \square

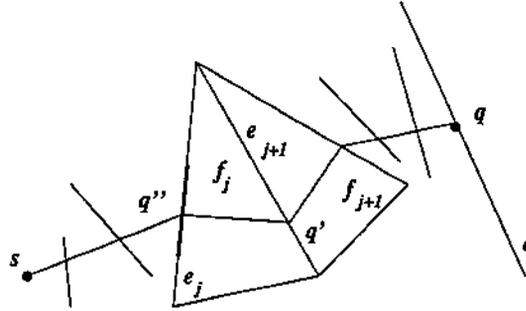


Figura 4.11: Prova do lema.

Na conclusão do algoritmo, a frente de onda encontrou todos os pontos de uma superfície poliédrica S :

Lema 4.3 [97] *Na conclusão do algoritmo, a lista de intervalos de (e, f) forma uma cobertura da aresta e .*

Prova. Suponha que exista algum ponto $q \in e = f \cap f'$ que não está coberto por um intervalo candidato relativo a (e, f) . Seja $p_f(q)$ um caminho mínimo localmente f -livre para q . Seja $I_j = [a_j, b_j]$ o último intervalo candidato interceptado por $p_f(q)$ e suponha que I_j é um intervalo relativo a (e_j, f_j) . Visto que q não está coberto, $e_j \neq e$. Seja $\overline{q''q'} = f_j \cap p_f(q)$, então o ponto q' está em uma aresta $e_{j+1} = f_j \cap f_{j+1}$. Seja $p_{f_{j+1}}(q')$ o subcaminho de $p_f(q)$ a partir de s a q' . Claramente, $p_{f_{j+1}}(q')$ é um caminho mínimo localmente f_{j+1} -livre para q' . O intervalo I_j deve ter sido propagado, senão seu ponto fronteira estaria ainda na fila de prioridades e o algoritmo não teria concluído. Mas então, pelo Teorema 4.1, teria sido criado um intervalo candidato na aresta e_{j+1} que incluía o ponto q' . Isso contradiz o fato de que I_j é o último intervalo candidato interceptado por $p_f(q)$. \square

A corretude do algoritmo assume que as funções numéricas são realizadas corretamente. Omitimos a demonstração deste fato para o problema PGAD devido a semelhança com resultados encontrados na literatura (Lema 8.1 de [97]). Neste caso, a prova de

corretude para o algoritmo Dijkstra contínuo direcional continua válida considerando intervalos de ε -otimalidade, ou seja, intervalos associados a caminhos localmente f -livres aproximados cujo comprimento é no máximo $(1 + \varepsilon)$ o caminho mínimo localmente f -livre.

4.5 Análise de Complexidade

A complexidade do algoritmo Dijkstra contínuo direcional depende do número de eventos ou intervalos de otimalidade candidatos. Vamos apresentar uma cota superior polinomial $O(n^4)$ no pior caso para o número de eventos [97].

Um subcaminho p de um caminho mínimo localmente f -livre pode cruzar uma aresta em no máximo um número linear de vezes, onde p não possui vértices ou pontos críticos (Lema 7.1 de [97]). A demonstração deste fato consiste em obter uma atribuição de triplas de pontos cruzamento a vértices de uma superfície poliédrica tal que nenhum vértice seja associado a mais que um número constante de triplas.

O número de intervalos de otimalidade sobre uma aresta e que correspondem à primeira incidência de um caminho mínimo localmente f -livre em e , ou seja, intervalos onde e ocorre apenas uma vez na última seqüência de arestas, pode ser no máximo $O(n^2)$ (Lemas 7.2 e 7.3 de [97]). A prova deste fato associa univocamente um par de intervalos de otimalidade adjacentes com mesma raiz a um vértice ou ponto crítico. Além disso, como existe um número linear de vértices e um número linear de pontos críticos por aresta (Lema 7.4 de [97]), o número total de raízes não pode ser maior que $O(n^2)$.

Portanto, pode existir no máximo $O(n^3)$ intervalos de otimalidade por par aresta-face (e, f) , pois todos $O(n^2)$ caminhos mínimos localmente f -livres que passam por e podem interceptar e em no máximo um número linear de vezes. Isto implica que o número total de intervalos de otimalidade para todos $O(n)$ pares aresta-face é $O(n^4)$. Assim, existe no máximo $O(n^4)$ pontos de evento.

Os resultados acima são válidos também para o PGAD, pois baseiam-se apenas no Lema 3.7 e em algumas propriedades métricas que também valem para a função distância FGAD. O número de pontos de evento também limita o espaço necessário para as estruturas de dados em $O(n^4)$.

O algoritmo Dijkstra contínuo direcional realiza $O(n^4)$ passos, envolvendo um número constante de chamadas a operações sobre as listas de intervalos e a funções numéricas. Assim, a complexidade do algoritmo será $O(n^4 K(n))$, onde $K(n)$ é a complexidade da função *Encontrar-Ponto*(I, q). Mitchell e Papadimitriou [97] apresentam uma cota $O(n^4 L)$ para a complexidade das funções numéricas, onde L é o número de bits necessários para especificar a instância do problema ($L = \log N + \log(W/w) + \log n + \log(1/\varepsilon)$), N é o valor inteiro da máxima coordenada, w (W) é o módulo mínimo (máximo) que a força externa pode assumir e ε é a tolerância de erro). Portanto, a complexidade total do algoritmo

será $O(n^8L)$.

O algoritmo que obtemos computa o caminho mínimo de s a qualquer ponto t em qualquer aresta $e = f \cap f'$ de uma superfície poliédrica. Podemos estender o algoritmo para encontrar o caminho mínimo de s a qualquer ponto da superfície poliédrica. Para isso, usamos uma abordagem exaustiva que compara todos os intervalos de otimalidade relativos a (e_i, f) , uma vez que tenhamos localizado t na face f com arestas denotadas por e_i através de técnicas de localização de pontos em tempo $O(\log n)$ [72, 108]. Deve haver um caminho mínimo para t que passa por um destes intervalos. Assim, $O(n^3)$ chamadas a *Encontrar-Ponto*(I, t) depende tempo $O(n^4L)$ e resulta o comprimento de um caminho ϵ -ótimo para t . Dessa forma, um caminho ϵ -ótimo pode ser produzido em tempo $O(n^7L)$ quando traçamos o caminho pelos predecessores dos intervalos de otimalidade.

Teorema 4.2 *Um caminho ϵ -ótimo de s para qualquer ponto t sobre uma superfície poliédrica S pode ser encontrado em tempo de pré-processamento $O(n^8L)$, tempo de consulta $O(n^7L)$ e espaço $O(n^4)$.*

Capítulo 5

Conceitos Básicos de Proximidade

Enquanto outros problemas de geometria computacional envolvem propriedades de incidência ou convexidade dos objetos geométricos, os *problemas de proximidade* (*closest-point* [109]) envolvem propriedades métricas e de esparsidade.

Seja Q um conjunto de m sítios em um certo espaço, os problemas de proximidade procuram enumerar pares de sítios em Q satisfazendo certas condições: pares separados por uma distância menor ou igual a um certo δ , k pares de sítios mais próximos em Q e mk pares representando os k vizinhos mais próximos de cada sítio em Q .

Abordamos problemas que consideram a proximidade de pontos em superfície poliédrica. Utilizamos uma estrutura computacional que contém toda a informação de proximidade relevante, o diagrama de Voronoi [126].

A *abordagem de lugar geométrico* consiste em um paradigma para o projeto de algoritmos geométricos onde se procura definir lugares geométricos e organizá-los em uma estrutura de dados. Os *lugares geométricos de proximidade* são um particionamento do espaço em regiões, cada região sendo o lugar geométrico dos pontos mais próximos a um sítio de um conjunto Q do que a qualquer outro sítio de Q segundo uma certa métrica. O diagrama de Voronoi é uma estrutura de dados que organiza os lugares geométricos de proximidade.

A seção 5.1 apresenta o problema do par mais próximo e suas generalizações. Na seção 5.2, introduzimos o problema de todos vizinhos mais próximos. A seção 5.3 considera o problema da árvore geradora mínima. Na seção 5.4, discutimos problemas de proximidade induzidos por problemas de caminhos mínimos. A seção 5.5 introduz o diagrama de Voronoi euclideano. Na seção 5.6, consideramos generalizações deste diagrama segundo a métrica, espaço e objeto geométrico que representa um sítio. A seção 5.7 descreve o diagrama de Voronoi de ordem k e a seção 5.8 apresenta o diagrama de Voronoi segundo uma função distância de caminhos mínimos.

5.1 Par Mais Próximo

O *problema do par mais próximo* para um conjunto finito Q de m sítios (objetos geométricos) consiste em encontrar um par de sítios q_i e q_j cuja distância entre q_i e q_j é mínima. A distância entre dois objetos é definida como a menor distância segundo uma certa métrica entre um ponto em um objeto e um ponto no outro. Este problema aparece quando se deseja detectar e evitar colisões entre objetos móveis ou estacionários. O problema do par mais próximo pode ser considerado como uma generalização ao *problema de detecção de interseção*.

Quando os sítios são pontos no plano considerando a distância euclideana, este problema pode ser resolvido em tempo $\Theta(m \log m)$ [13, 63, 118]. Bentley e Shamos [12, 13] resolvem o problema com um algoritmo divisão-e-conquista que usa uma abordagem multi-dimensional. O algoritmo divide o conjunto de pontos através da reta mediana M e encontra recursivamente um par mais próximo nos dois subconjuntos resultantes. Seja Q' o conjunto de sítios q_k a uma distância menor do que δ da mediana M , onde δ é o comprimento mínimo dos segmentos ligando os pares mais próximos nos subconjuntos separados por M . O algoritmo realiza uma varredura em tempo linear no conjunto Q' e encontra um par mais próximo de Q .

Chang e Lee [19] analisaram a complexidade de caso médio do algoritmo de Bentley e Shamos para o plano. Chang e Lee assumiram que o conjunto de pontos está uniformemente distribuído por um processo estocástico em uma região limitada. Isto garante que o número esperado de pontos na faixa definida pela mediana ($d(q_k, M) \leq \delta$) é $O(\sqrt{m})$. Dessa forma, a complexidade de caso médio é linear.

O *problema de escalamento* [10, 46, 119] determina o fator de escala máximo δ tal que um conjunto de objetos geométricos escalados por este fator, segundo pontos centrais nos respectivos núcleos, sejam disjuntos dois a dois (veja Fig. 5.1). Quando os objetos não são disjuntos inicialmente, temos $\delta < 1$. Este problema aparece em computação gráfica, robótica e cartografia computacional, quando precisamos redimensionar janelas, espaços de trabalho para robôs e rótulos geográficos, respectivamente.

O problema do par mais próximo para pontos é um caso particular do problema de escalamento quando todos os objetos considerados são discos unitários a serem escalados segundo seus centros. Além disso, o problema de escalamento consiste em uma especialização do problema do par mais próximo geral considerando a métrica euclideana.

Formann [46] resolve o problema de escalamento para objetos convexos (bolas de alguma métrica L_p) no plano em tempo $\Theta(m \log m)$. Utilizando o paradigma de *busca paramétrica* [88] refinado por Cole [34], Formann resolve este problema para objetos estrelados e x -monotônicos em tempo $O(m \log^2 m)$.

Bartling e Hinrichs [10] resolvem o problema do par mais próximo para objetos con-

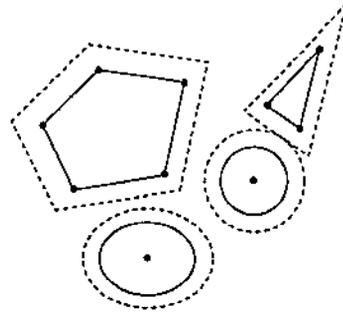


Figura 5.1: O problema de escalamento.

vexos e compactos no plano. Quando os objetos são m segmentos, discos circulares ou polígonos convexos com número de arestas constante, a solução do problema considerando a métrica euclidiana depende tempo $\Theta(m \log m)$.

Seja Q_1, Q_2, \dots, Q_k uma partição exaustiva ($Q = Q_1 \cup Q_2 \cup \dots \cup Q_k$) e mutuamente exclusiva ($Q_i \cap Q_j = \emptyset$) do conjunto de m sítios Q . O problema de encontrar a menor distância entre dois objetos em subconjuntos distintos é uma generalização do problema do par mais próximo denominada *versão colorida* [119].

Formalmente, o problema do par mais próximo que resolvemos na seção 6.3 é especificado da seguinte forma:

PROBLEMA DO PAR MAIS PRÓXIMO EM SUPERFÍCIE POLIÉDRICA

INSTÂNCIA. *Um conjunto de pontos Q em uma superfície poliédrica triangulada S ; uma atribuição de coeficientes de atrito cinético a arestas e faces; e uma tolerância de erro $\epsilon > 0$.*

QUESTÃO. *Encontrar um par de pontos do conjunto Q que são mais próximos um ao outro de acordo com a função distância de caminhos mínimos induzida pela função distância FGAD.*

5.2 Todos Vizinhos Mais Próximos

O *problema de todos vizinhos mais próximos* para um conjunto Q finito de m sítios procura um vizinho mais próximo para cada sítio no conjunto Q (veja Fig. 5.2). Para um conjunto de pontos no plano euclidiano, existem algoritmos com complexidade $\Theta(m \log m)$ [64, 124] que encontram a solução diretamente sem construir o diagrama de Voronoi correspondente.

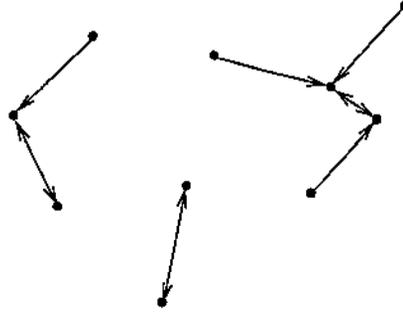


Figura 5.2: O problema de todos vizinhos mais próximos.

Se a seqüência de pontos forma um polígono convexo no plano, podemos encontrar todos os pares mais próximos em tempo linear [81]. Contudo, quando a seqüência de pontos define um polígono monotônico, estrelado ou simples, resolvemos o problema em tempo $\Theta(m \log m)$ [2].

Vaidya [124] apresentou um algoritmo que resolve o problema d -dimensional em qualquer métrica L_p . Hinrichs *et al.* [64] usaram o paradigma de varredura, onde uma reta vertical R varre o plano atualizando a seção de cruzamentos em cada ponto de transição (veja Fig. 5.3). O algoritmo mantém como invariante a interseção da reta de varredura R com o diagrama de Voronoi $V(Q_R)$ dos pontos à esquerda de R . Os pontos $q_i \in Q_R$ cuja região de Voronoi intercepta R são os pontos ativos, enquanto as arestas de $V(Q_R)$ que interceptam a reta R são os bissetores ativos. Dessa forma, os pontos de transição consistem nos pontos em Q e nas interseções dos bissetores ativos (vértices de Voronoi). Quando um ponto em Q é alcançado por R , um novo intervalo é incluído na invariante $V(Q_R) \cap R$ provocando, possivelmente, o desaparecimento de algum intervalo existente. Se o ponto de transição é a interseção de dois bissetores, ocorre uma mudança topológica em $V(Q_R) \cap R$, pois R alcançou um vértice de $V(Q_R)$ e um intervalo é excluído.

Uma generalização do problema de todos vizinhos mais próximos consiste no problema que considera uma coleção de conjuntos mutuamente exclusivos de sítios, Q_1, Q_2, \dots, Q_k , com um total de m sítios, e para cada sítio q_i encontra um sítio q_j mais próximo a q_i e em um conjunto distinto ao conjunto que contém q_i .

Aggarwal *et al.* [2] resolveram o problema generalizado em tempo $\Theta(m \log m)$ quando os sítios são pontos no plano. O algoritmo usa diagramas de Voronoi de subconjuntos da união Q de todos os conjuntos Q_i e algoritmos de localização de pontos. Observe que localizar cada ponto $q_j \in Q_i$ no diagrama de Voronoi de $Q - Q_i$ resolve o problema em tempo $O(km \log m)$ devido à construção dos k diagramas de Voronoi $V(Q - Q_i)$. O algoritmo constrói o diagrama de Voronoi $V(Q)$. Para cada subconjunto Q_i , encontra o conjunto Q'_i de pontos $q_j \in Q - Q_i$ tal que a região de Voronoi de q_j tenha uma aresta de Voronoi em $V(Q)$ contida no bissetor entre q_j e algum ponto em Q_i , constrói o diagrama

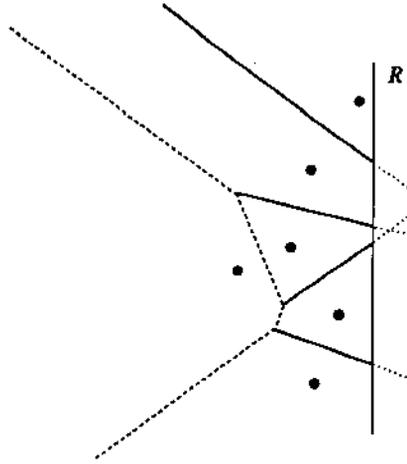


Figura 5.3: Varredura planar por uma reta vertical sobre o diagrama de Voronoi.

de Voronoi $V(Q'_i)$ e localiza os pontos de Q_i em $V(Q'_i)$.

A correteza deste algoritmo se baseia no seguinte fato: se $q_j \notin Q_i$ é um vizinho mais próximo de um ponto em Q_i , então a região de Voronoi de q_j tem uma aresta de Voronoi em $V(Q)$ contida no bissetor entre q_j e algum ponto em Q_i . Além disso, os diagramas de Voronoi $V(Q - Q_i)$ e $V(Q)$ são iguais na região complementar ao fecho da união das regiões de Voronoi em $V(Q)$ associadas a pontos em Q_i .

A análise de complexidade mostra que os conjuntos Q'_i são encontrados em tempo total linear, enquanto $V(Q)$ e os outros diagramas são construídos em tempo total $O(m \log m)$. Portanto, o algoritmo encontra um vizinho mais próximo em um conjunto distinto para cada ponto em tempo $\Theta(m \log m)$.

Formalmente, o problema de todos vizinhos mais próximos que resolvemos é especificado da seguinte forma:

PROBLEMA DE TODOS VIZINHOS MAIS PRÓXIMOS EM SUPERFÍCIE POLIÉDRICA

INSTÂNCIA. Um conjunto de pontos Q em uma superfície poliédrica triangulada S ; uma atribuição de coeficientes de atrito cinético a arestas e faces; e uma tolerância de erro $\epsilon > 0$.

QUESTÃO. Encontrar o ponto mais próximo a cada ponto do conjunto Q de acordo com a função distância de caminhos mínimos induzida pela função distância FGAD.

5.3 Árvore Geradora Mínima

O problema da árvore geradora mínima consiste em encontrar uma árvore de comprimento mínimo que conecta todos os m sítios de um conjunto Q (veja Fig. 5.4). Este problema, geralmente, tem aplicações em vários tipos de redes.

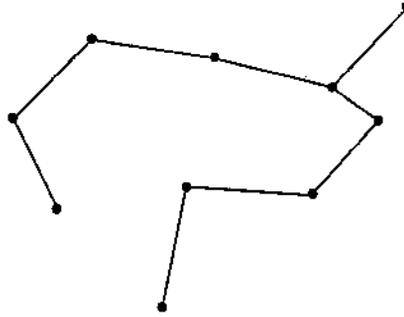


Figura 5.4: Uma árvore geradora mínima.

O problema da árvore geradora mínima é encontrado em Teoria dos Grafos, onde se procura uma subárvore com a soma mínima dos pesos das arestas de um grafo valorado G que contém todos os nós. Dijkstra [42], Kruskal [75] e Prim [110] resolveram o problema em tempo polinomial. Cheriton e Tarjan [27] apresentaram um algoritmo $O(E)$ para grafos densos ($E > m^{1+\epsilon}$), onde E é o número de arestas no grafo, m o número de nós e ϵ é alguma constante positiva.

O algoritmo geral de Cheriton e Tarjan seleciona a aresta com menor peso como a primeira aresta da árvore. O algoritmo realiza um passo geral até conectar todos os nós. Neste passo, escolhe uma árvore T na floresta definida pelas arestas já selecionadas, seleciona a aresta e com menor peso incidente a um nó em T e a um nó em outra árvore T' , exclui todas as arestas com peso menor que e incidentes a T e inclui a aresta e na árvore geradora mínima. Para grafos planares, Cheriton e Tarjan resolveram o problema em tempo linear no pior caso.

Seja Q um conjunto de sítios particionado em k subconjuntos Q_i mutuamente exclusivos e exaustivos. Uma generalização do problema da árvore geradora mínima é encontrar a árvore de comprimento mínimo que contém exatamente um sítio de cada subconjunto Q_i (veja Fig. 5.5). Se cada subconjunto Q_i é unitário, temos o problema da árvore geradora mínima original. Este problema aparece quando se deseja determinar centros de serviço regionais que devem ser interconectados.

Myung *et al.* [103] demonstraram que esta generalização para grafos é um problema NP -difícil. Para isso, reduzem o problema da cobertura de nós, que é NP -completo, ao problema generalizado [49]. Além disso, encontrar uma solução ϵ -ótima para o problema da árvore geradora mínima generalizado também é NP -difícil. Portanto, para qualquer

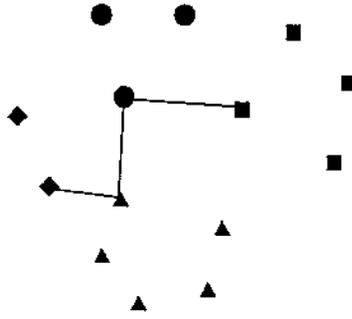


Figura 5.5: O problema da árvore geradora mínima generalizado.

heurística de tempo polinomial, não existe constante real ε tal que a solução encontrada pela heurística tenha comprimento menor ou igual a ε vezes o comprimento da solução ótima. A demonstração assume que $P \neq NP$ e chega a uma contradição, pois se uma heurística encontra uma solução ε -ótima, então resolve o problema da cobertura de nós e temos $P = NP$.

Outra generalização para o problema da árvore geradora mínima consiste em encontrar uma árvore de comprimento mínimo contendo pelo menos k sítios. Este problema surge quando se precisa minimizar o custo de instalação de um serviço para k cidades em uma região com m cidades. Ravi *et al.* [112] mostraram que este problema para pontos no plano é NP -difícil e apresentaram um algoritmo exato polinomial para o caso onde os pontos estão na fronteira de uma região convexa.

Geralmente, algoritmos aproximativos são propostos para resolver esta generalização. Estes algoritmos usam técnicas de cota inferior para comparar o custo de suas soluções com a solução ótima. Além disso, o valor da cota inferior deve ser próximo ao valor da solução ótima para que o algoritmo aproximativo possa garantir um fator de aproximação próximo a 1. Blum *et al.* [14] e Mitchell [94] apresentaram algoritmos que possuem fatores de aproximação $O(1)$ e $2\sqrt{2}$, respectivamente.

Formalmente, o problema da árvore geradora mínima que resolvemos na seção 6.3 é especificado da seguinte forma:

PROBLEMA DA ÁRVORE GERADORA MÍNIMA EM SUPERFÍCIE POLIÉDRICA

INSTÂNCIA. Um conjunto de pontos Q em uma superfície poliédrica triangulada S ; uma atribuição de coeficientes de atrito cinético a arestas e faces; e uma tolerância de erro $\varepsilon > 0$.

QUESTÃO. Encontrar uma árvore que conecta todos os pontos do conjunto Q de comprimento total mínimo de acordo com a função distância de caminhos mínimos induzida

pela função distância FGAD.

5.4 Proximidade de Caminhos Mínimos

Um problema de caminhos mínimos adicionado a um conjunto Q finito de sítios leva a vários problemas de proximidade.

Guha e Suzuki [53] resolveram problemas de proximidade em um plano com n retângulos isotéticos (lados paralelos aos eixos) como obstáculos considerando a métrica de caminhos mínimos retilíneos. Os algoritmos encontram um par mais próximo em um conjunto de m sítios Q e um vizinho mais próximo para cada sítio de Q em tempo $O((n+m)\log(n+m))$. A árvore geradora mínima é obtida em tempo $O((n+m)\log(n+m)\log n)$ pela redução ao diagrama de Voronoi (veja Fig. 5.6).

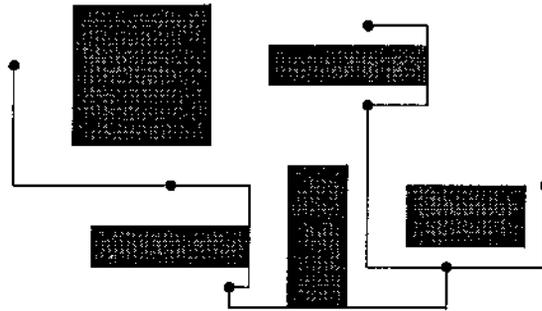


Figura 5.6: Uma árvore geradora mínima retilínea com obstáculos retangulares isotéticos.

Os algoritmos, baseados em varredura planar, consideram um tipo de caminho mínimo geometricamente especial denominado z -caminho, onde z é alguma das direções $\pm x$ e $\pm y$. Um z -caminho consiste em um caminho monotônico na direção z que percorre a direção ortogonal a z apenas quando alcança um obstáculo retangular ou para evitar perder a z -monotonicidade.

Larson e Li [77] estudaram o problema que procura todas as distâncias retilíneas mínimas entre um conjunto de m pontos no plano com obstáculos poligonais. O algoritmo obtido depende tempo $O(m(m^2+n^2))$, onde n é o número total de vértices dos obstáculos. Hwang [65] obteve um algoritmo $O(m \log m)$ para a árvore geradora mínima retilínea.

Wu *et al.* [130] apresentaram um algoritmo para a árvore geradora mínima de caminhos mínimos retilíneos com n obstáculos retilíneos, ou seja, polígonos com arestas apenas horizontais e verticais. Wu *et al.* utilizam um grafo, semelhante a uma grade, que define um subespaço contendo um caminho mínimo para qualquer par de sítios evitando os obstáculos retilineares. A árvore geradora mínima para pontos é construída em tempo $O(m \log m + n^2 \log n)$ por um método híbrido que encontra caminhos mínimos enquanto constrói a árvore.

5.5 Diagrama de Voronoi Euclidiano

Seja Q um conjunto finito de m pontos distintos no plano euclidiano. A *região de Voronoi* associada a q_i relativa a Q é o conjunto de todos os pontos que estão mais próximos a q_i do que a qualquer outro elemento de Q (veja Fig. 5.7). A união de todas as fronteiras das regiões de Voronoi é chamada *diagrama de Voronoi* de Q . Este diagrama apareceu quando o matemático russo, G. Voronoi, investigou a geometria dos números [126].

A fronteira comum de duas regiões de Voronoi é uma *aresta de Voronoi* e representa os pontos que possuem dois vizinhos mais próximos em Q . Duas arestas encontram-se em um *vértice de Voronoi*, este ponto possui mais que dois vizinhos mais próximos no conjunto Q . As regiões são exaustivas e mutuamente exclusivas exceto pelas fronteiras.

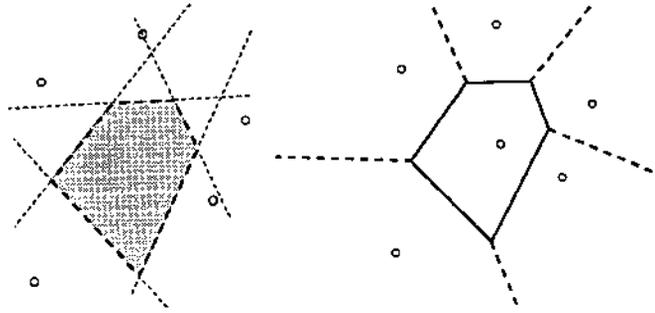


Figura 5.7: Uma região e um diagrama de Voronoi euclidiano.

Apenas alguns bissetores contribuem realmente para o contorno de uma região de Voronoi e o número total de arestas do diagrama de Voronoi é apenas de ordem linear. Pelo menos três algoritmos, assintoticamente ótimos no pior caso, foram propostos para construção do diagrama de Voronoi euclidiano de m pontos no plano em tempo $\Theta(m \log m)$ e espaço $\Theta(m)$.

Shamos e Hoey utilizaram a *abordagem divisão-e-conquista* [118], onde o problema é quebrado (divisão) em dois subproblemas de igual tamanho que são resolvidos recursivamente e, então, unimos as subsoluções (conquista).

Brown reduziu o problema de construir o diagrama de Voronoi de m pontos no d -espaço euclidiano para a construção da envoltória convexa de m pontos no $d + 1$ -espaço [15].

Fortune usou a técnica de varredura linear [47], onde uma reta horizontal percorre o plano enquanto o conjunto de arestas de Voronoi ativas interceptadas é armazenada ordenadamente conforme a ordem das interseções com a reta de varredura. Esta informação é atualizada quando a reta alcança pontos de evento que alteram o conjunto de arestas.

5.6 Diagramas de Voronoi Generalizados

Apresentamos algumas generalizações do diagrama de Voronoi de acordo com a distância, o espaço e o objeto geométrico que representa um sítio. A definição de cada diagrama será feita através da região de Voronoi correspondente.

O conceito de *diagrama de Voronoi abstrato* [73, 74] é baseado em sistemas de curvas simples (bissetores) que bissectam o plano em regiões ilimitadas (semi-panos). As regiões de Voronoi, definidas pela interseção de todos os semi-planos contendo um dado ponto, formam uma partição do plano em conjuntos caminho-conexos com interiores não vazios. O diagrama de Voronoi abstrato consiste na união das fronteiras destas regiões.

5.6.1 Distância

O diagrama de Voronoi que construímos utiliza uma função distância de caminhos mínimos. Neste caso, a distância entre dois pontos consiste no comprimento do caminho mínimo segundo a função distância FGAD entre estes pontos.

Diagrama de Voronoi Retilíneo

No *diagrama de Voronoi retilíneo*, a distância de um sítio $q_i \in Q$ a um ponto r segundo a métrica retilínea é $d_R(q_i, r) = |q_{ix} - r_x| + |q_{iy} - r_y|$ (veja Fig. 5.8). A região de Voronoi consiste no conjunto de pontos aos quais a distância retilínea de um ponto em Q é menor ou igual à distância retilínea de qualquer outro ponto em Q [65]. Uma aplicação deste diagrama está no projeto de circuitos integrados.

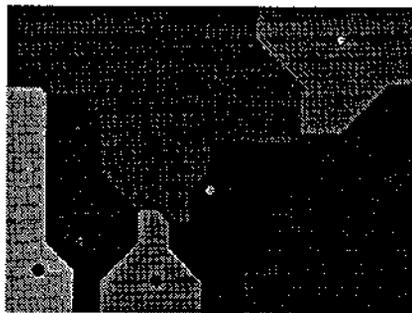


Figura 5.8: Diagrama de Voronoi retilíneo.

Diagrama de Voronoi Fluvial

No *diagrama de Voronoi fluvial*, a *distância fluvial* $d_F(q_i, r)$ denota o tempo mínimo necessário para um fluido emanar de um sítio q_i em Q para alcançar r (veja Fig. 5.9).

A região de Voronoi consiste do conjunto de pontos dos quais a distância fluvial de q_i é menor ou igual à distância fluvial de qualquer outro sítio em Q [121]. Este diagrama pode ser aplicado para considerar o efeito do vento na difusão de poluição.

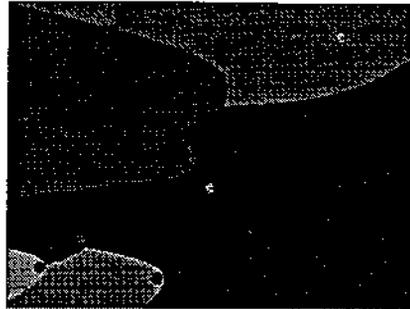


Figura 5.9: Diagrama de Voronoi fluvial.

Diagrama de Voronoi Valorado

No *diagrama de Voronoi valorado*, a região de Voronoi é o conjunto de pontos r aos quais a *distância valorada*, $d_V(q_i, r) = A_{q_i} + M_{q_i} d(q_i, r)$, do sítio q_i em Q é menor ou igual a distância valorada de qualquer outro sítio em Q , onde $d(q_i, r)$ é a distância euclideana, A_{q_i} é uma constante aditiva e M_{q_i} é uma constante multiplicativa [6]. Quando A e M são os mesmos para todos os sítios em Q , temos o diagrama de Voronoi euclideano. Quando M é o mesmo para todos os pontos em Q ($M_{q_i} = M_{q_j} = 1$) e A não é necessariamente o mesmo, temos o *diagrama de Voronoi aditivo* (veja Fig. 5.10(a)). Quando A é o mesmo para todos os pontos em Q ($A_{q_i} = A_{q_j} = 0$) e M não é necessariamente o mesmo, temos o *diagrama de Voronoi multiplicativo* (veja Fig. 5.10(b)). No caso genérico, temos o *diagrama de Voronoi composto* (veja Fig. 5.10(c)).



(a) Aditivo.

(b) Multiplicativo.

(c) Composto.

Figura 5.10: Diagramas de Voronoi valorados.

Este diagrama pode ser aplicado ao problema de encontrar uma área de mercado para um conjunto de empresas (q_i) cujos preços dos produtos dependem do custo de produção A_{q_i} e do custo unitário de transporte M_{q_i} .

5.6.2 Espaço

O espaço abordado pelo diagrama de Voronoi de caminhos mínimos que estudamos é uma superfície poliédrica S , possivelmente não convexa tal que uma reta ortogonal a um certo plano intercepta S em apenas um ponto, com coeficientes de atrito cinético atribuídos a suas arestas e faces.

Diagrama de Voronoi Tridimensional

O *diagrama de Voronoi tridimensional*, embora tenha um número linear de regiões, pode ter um número quadrático de arestas e vértices. A complexidade do diagrama de Voronoi para dimensões maiores (no espaço euclidiano) tem crescimento exponencial [7].

Diagrama de Voronoi Esférico

O *diagrama de Voronoi esférico* considera a esfera de raio unitário, onde as coordenadas de um ponto r satisfazem $r_x^2 + r_y^2 + r_z^2 = 1$. A *distância esférica*, $d_E(q_i, r)$, de um sítio $q_i \in Q$ a um ponto r é o comprimento do menor arco do grande círculo passando por q_i e r : $d_E(q_i, r) = \arccos(q_{ix}r_x + q_{iy}r_y + q_{iz}r_z) \leq \pi$. A região de Voronoi consiste no conjunto de pontos sobre a esfera para os quais a distância esférica de q_i é menor ou igual à distância esférica de qualquer outro sítio em Q [89].

Este diagrama está relacionado à superfície do globo, onde se deseja encontrar uma pista para um pouso de emergência ou a origem de um certo poluente disperso na atmosfera por um conjunto de fábricas. Outras generalizações semelhantes ao diagrama de Voronoi esférico, segundo o espaço considerado, são diagramas em um cilindro, cone ou toro [41, 73].

5.6.3 Objeto Geométrico

O objeto geométrico que representa um sítio, considerado pelo diagrama de Voronoi de caminhos mínimos que investigamos, é o ponto. Além disso, assumimos que o conjunto Q de sítios está contido no conjunto de vértices do poliedro.

Diagrama de Voronoi de Segmentos

No *diagrama de Voronoi de segmentos*, a distância entre um segmento de reta $q_i \in U$ e um ponto r , $d_U(q_i, r)$, é a distância euclideana para r do ponto mais próximo em q_i : $d_U(q_i, r) = \min_{s \in q_i} d(s, r)$ (veja Fig. 5.11). A região de Voronoi consiste no conjunto de pontos dos quais a distância de segmento q_i é menor ou igual à distância de qualquer outro segmento em U [44, 80].



Figura 5.11: Diagrama de Voronoi de segmentos.

Este diagrama tem aplicações em localização de facilidades e agrupamento. Além disso, o território marítimo de um país consiste na interseção da região de Voronoi associada ao país com o conjunto de pontos a uma distância menor que 200 milhas de algum ponto no país.

Diagrama de Voronoi de Áreas

No *diagrama de Voronoi de áreas*, a distância entre uma área $q_i \in A$ e um ponto r , $d_A(q_i, r)$, é a distância euclideana para r do ponto mais próximo em q_i : $d_A(q_i, r) = \min_{s \in q_i} d(s, r)$. A região de Voronoi consiste no conjunto de pontos dos quais a distância da área q_i é menor ou igual à distância de qualquer outra área em A [44, 80].

As técnicas usadas na construção de diagramas de Voronoi de segmentos podem ser estendidas para o caso que considera polígonos disjuntos. Neste caso, o conjunto de polígonos pode ser tratado como um conjunto de segmentos de reta não disjuntos.

Diagrama de Voronoi Circular

Yap [131] constrói o diagrama para arcos circulares em tempo $O(m \log m)$, mas o algoritmo requer que os arcos circulares se interceptem apenas nas extremidades. Sharir [119] constrói o diagrama de Voronoi para um conjunto de discos circulares.

5.7 Diagrama de Voronoi de Ordem k

O diagrama de Voronoi $V_k(Q)$ de ordem k para um conjunto finito Q de m sítios foi introduzido por Shamos e Hoey [118] para tratar o problema de k pontos mais próximos. Lee [78] apresentou o primeiro algoritmo para construir este diagrama em tempo $O(k^2 m \log m)$ e espaço $O(k^2 m)$. Além disso, mostrou que o número de regiões, arestas e vértices deste diagrama no plano é $O(k(m - k))$.

5.7.1 Diagrama da Seqüência

A região de Voronoi para uma seqüência ordenada Q' de k sítios (q_1, q_2, \dots, q_k) consiste no conjunto de pontos dos quais a distância do ponto q_i é menor ou igual à distância do ponto q_j (se e somente se $i < j$) que é menor ou igual à distância de qualquer outro ponto em $Q - Q'$ (veja Fig. 5.12). A construção de $V_{k+1}(Q)$ é o particionamento de cada região de $V_k(Q)$ em regiões de $V_{k+1}(Q)$. Isto equivale a obter a interseção de cada região de Voronoi de $V_k(Q)$ com o diagrama de Voronoi $V_1(Q - Q')$. Este diagrama unifica os problemas de ponto-mais-próximo e ponto-mais-distante, pois o lugar geométrico dos pontos cujos k vizinhos mais próximos estão no conjunto Q' é também o lugar geométrico dos pontos cujos $m - k$ vizinhos mais distantes estão no conjunto $Q - Q'$. Assim, o diagrama de ponto-mais-próximo de ordem k é o diagrama de ponto-mais-distante de ordem $(m - k)$.

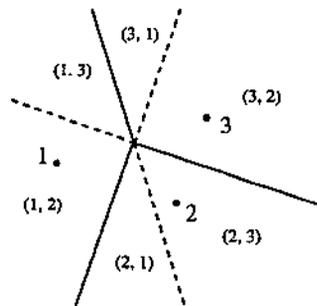


Figura 5.12: Diagrama de Voronoi de ordem $k = 2$.

5.7.2 Diagrama do Subconjunto

A região de Voronoi consiste no conjunto de pontos dos quais a distância para um subconjunto Q' de k pontos de Q é menor ou igual à distância para qualquer outro ponto em $Q - Q'$, ou seja, o conjunto de pontos que são mais próximos a qualquer ponto em Q' do que a qualquer ponto de Q que não está em Q' .

Chazelle e Edelsbrunner [23] desenvolveram duas versões de um algoritmo melhor para grandes valores de k . O primeiro depende tempo $O(m^2 \log m + k(m - k) \log^2 m)$ e espaço

$O(k(m - k))$, enquanto o segundo depende tempo $O(m^2 + k(m - k)\log^2 m)$ e espaço $O(m^2)$. Utilizando amostragem aleatória de pontos, Clarkson [31] determina o diagrama em tempo $O(km^{1+\epsilon})$, onde ϵ é um termo constante.

5.8 Diagrama de Voronoi de Caminhos Mínimos

Restrições de movimento impedem a utilização de um caminho direto forçando a tomada do caminho mínimo que satisfaz estas restrições. O caminho mínimo entre dois pontos q_i e q_j em um conjunto finito Q é definido como o caminho de comprimento mínimo dentre todos os caminhos contínuos possíveis ligando q_i a q_j que satisfazem as restrições de movimento.

A região de Voronoi $R(q_i)$ de caminhos mínimos associada a q_i consiste no conjunto de pontos r dos quais o comprimento do caminho mínimo de q_i para r é menor ou igual ao comprimento do caminho mínimo originado de qualquer outro sítio em Q para r . Dessa forma, a *distância geodésica* entre dois pontos usa uma métrica ou função distância induzida por um problema de caminhos mínimos.

Quando a métrica ou função distância utilizada na definição do comprimento de um caminho é convexa, as regiões de Voronoi particionam a superfície poliédrica no *diagrama de Voronoi de caminhos mínimos* $V(Q)$.

Aronov [3] apresentou um algoritmo de divisão-e-conquista com *fase de extensão* que depende tempo $O((n + m)\log(n + m)\log n)$ para computar o diagrama de Voronoi de m sítios em um polígono simples de n lados (veja Fig. 5.13).

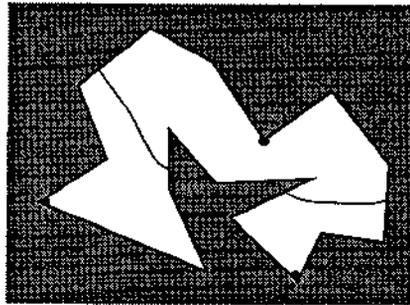


Figura 5.13: Diagrama de Voronoi em um polígono simples.

Guha e Suzuki [53] construíram o diagrama de Voronoi de caminhos mínimos retilíneo com obstáculos retangulares isotéticos em tempo $O((n + m)\log(n + m)\log n)$ por um algoritmo de divisão-e-conquista com fase de extensão. Mitchell [93] computou o diagrama de Voronoi usando a metodologia Dijkstra contínuo em tempo $O((n + m)\log^2(n + m))$ considerando polígonos simples como obstáculos. Mount [101] usou uma abordagem semelhante para o diagrama de Voronoi em superfície poliédrica, generalizando o problema

da geodésica discreta para o caso de múltiplas origens. O diagrama de Voronoi é construído em tempo $O((n + m)^2 \log(n + m))$ e espaço $O((n + m)^2)$, onde n é a complexidade do poliedro e m é o número de sítios.

Capítulo 6

O Diagrama de Voronoi de Caminhos Mínimos

A construção do diagrama de Voronoi de caminhos mínimos de um conjunto Q de m sítios requer a descrição de uma subdivisão planar. Esta descrição consiste nas coordenadas dos vértices de Voronoi e do conjunto de arestas de Voronoi armazenado em uma estrutura de dados como a *quad-edge* [57].

A versão Voronoi do problema de caminhos mínimos PGAD procura encontrar a origem $q_i \in Q$ mais próxima a um ponto consulta t , onde as distâncias são consideradas de acordo com o comprimento do caminho mínimo de q_i para t segundo a função distância FGAD.

A seção 6.1 especifica o diagrama de Voronoi de caminhos mínimos que estudamos e o algoritmo que propomos para sua construção. Na seção 6.2, apresentamos um procedimento que constrói uma subdivisão da superfície poliédrica que implica na descrição geométrica e topológica do diagrama de Voronoi que abordamos. Na seção 6.3, realizamos a redução dos problemas de proximidade que consideramos ao problema de construir o diagrama de Voronoi de caminhos mínimos VGAD.

6.1 Diagrama de Voronoi GAD

Formalmente, o diagrama de Voronoi VGAD que construímos é especificado da seguinte forma:

PROBLEMA DO DIAGRAMA DE VORONOI-GAD

INSTÂNCIA. *Um conjunto de pontos Q em uma superfície poliédrica triangulada S ; uma atribuição de coeficientes de atrito cinético a arestas e faces; e uma tolerância de erro*

$\varepsilon > 0$.

QUESTÃO. Construir o diagrama de Voronoi de caminhos mínimos que permita identificar o sítio $q_i \in Q$ tal que um caminho ε -ótimo p^* sobre a superfície S (segundo a função distância geodésica atritada direcional) de q_i a um dado ponto consulta t é o menor dentre todos os caminhos ε -ótimos originados de outro sítio para t e permita também determinar o caminho p^* .

O algoritmo para o diagrama de Voronoi VGAD consiste em uma generalização do algoritmo de caminhos mínimos para o caso de múltiplas origens onde se deseja encontrar um caminho mínimo de um conjunto de origens Q para qualquer ponto consulta sobre a superfície. Modificamos a inicialização do algoritmo principal para incluir os intervalos de otimalidade associados a cada ponto de origem. Assumimos que cada ponto origem é um vértice da superfície poliédrica.

Algoritmo Voronoi-Contínuo(Q)

1. para ($1 \leq j \leq m$)

(a) $d(q_j) := 0$.

(b) Permanente(q_j).

(c) para (cada aresta $e_{i,f} \subset f | q_j \in f$)

i. $I_{i,f} := \text{Difundir}(q_j, e_{i,f})$.

ii. Atualizar-Intervalo($I_{i,f}, q_j$).

2. enquanto ($\text{Eventos} \neq \emptyset$) faça

(a) $q := \text{Desenfileira}(\text{Eventos})$.

(b) Permanente(q).

(c) se ($\Upsilon(q) = \text{Vértice}$) então

i. para (cada aresta $e_{i,f} \subset f | q \in f$)

A. $I_{i,f} := \text{Difundir}(q, e_{i,f})$.

B. Atualizar-Intervalo($I_{i,f}, q$).

(d) se ($\Gamma(q) = \text{Fronteira}$) então

i. para (cada aresta $e_i \subset f, f'$)

A. se ($e \neq e_i \subset f$) então

• $I_i := \text{Projetar}(I, e_i)$.

• se ($I_i \neq \emptyset$) então Atualizar-Intervalo(I_i, q).

B. se ($pc_a(I)$) então

• $I_a := \text{Usar-Criticamente}(I, e_i, a)$.

• se ($I_a \neq \emptyset$) então Atualizar-Intervalo(I_a, q).

C. se ($pc_b(I)$) então

• $I_b := \text{Usar-Criticamente}(I, e_i, b)$.

• se ($I_b \neq \emptyset$) então Atualizar-Intervalo(I_b, q).

Teorema 6.1 *As listas de intervalos de otimalidade dos pares aresta-face permitem computar corretamente os caminhos mínimos localmente f -livres dos sítios aos pontos desses intervalos dentre os caminhos localmente f -livres que passam pelas arestas por intervalos cujo ponto fronteira já foi rotulado permanentemente.*

Prova. A prova procede por indução no número de eventos rotulados permanentemente. No caso base, quando os sítios q_j são rotulados permanentemente (passo 1), os intervalos candidatos estão nas arestas das faces incidentes aos sítios. A função $Difundir(q_j, e_{i,f})$ cria apenas um intervalo obedecendo ao critério de otimalidade local em cada aresta (passo 1.c.i), pois se existir dois sítios em uma mesma face, podemos subdividi-la em tempo constante, garantindo que cada face contenha apenas um sítio. Os intervalos são incluídos nas listas de intervalos correspondentes aos respectivos pares aresta-face pelo procedimento *Atualizar-Intervalo*($I_{i,f}, q_j$) (passo 1.c.ii). Dessa forma, estes intervalos permitem computar corretamente os caminhos mínimos passando apenas pelos intervalos degenerados que são os sítios e isto estabelece a base da indução.

A hipótese e o passo de indução são semelhantes ao caso de origem única (Seção 4.4). Todos os intervalos possíveis são criados e incluídos corretamente considerando os intervalos existentes nas listas de intervalos. Isto é garantido pelo critério de otimalidade local e por um resultado semelhante ao Lema 4.1. Neste caso, o Lema 3.7 deve ser estendido para considerar caminhos mínimos localmente f -livres originados de sítios distintos, q_1 e q_2 . Neste caso, se os caminhos se cruzam em algum ponto γ interior a uma face, então um dos subcaminhos de q_1 a γ ou de q_2 a γ pode ter comprimento menor. Portanto, este caminho poderia ser encurtado e ser localmente ajustado tal que seria novamente geodésico e ainda seria localmente f -livre, pois os caminhos são incidentes na mesma aresta. \square

Podemos garantir, assumindo a validade de um resultado análogo ao Lema 4.1 para múltiplas origens, que na conclusão do algoritmo *Voronoi-Contínuo*(Q), as listas de intervalos formam uma cobertura das arestas de uma superfície poliédrica S tal que os intervalos permitem computar corretamente caminhos mínimos para pontos sobre as arestas de S originados nos sítios $q_j \in Q$.

A complexidade do algoritmo *Voronoi-Contínuo*(Q) tem uma análise semelhante à do algoritmo *Dijkstra-Contínuo*(s). Dessa forma, podemos obter os intervalos de otimalidade que permitem computar um caminho ϵ -ótimo de um sítio q_j para qualquer ponto t sobre uma superfície poliédrica em tempo $O((n + m)^8 L)$, onde n é o número de vértices da superfície e m é o número de sítios. Contudo, como assumimos que os sítios são vértices da superfície ($m = O(n)$), temos que o algoritmo *Voronoi-Contínuo*(Q) despende tempo $O(n^8 L)$.

6.2 A Subdivisão de Voronoi

A descrição e a topologia do diagrama de Voronoi de caminhos mínimos são encontradas quando construímos uma subdivisão de cada face f da superfície poliédrica em células associadas aos intervalos de otimalidade relativos a (e_i, f) , tal que um ponto t está em uma célula C_j se e somente se existe um caminho mínimo sobre a superfície poliédrica de uma origem q_k a t pelo intervalo I_j associado.

Esta subdivisão, denominada *subdivisão de Voronoi*, é um superconjunto do diagrama de Voronoi, pois os vértices e arestas de Voronoi estão contidos nos bissetores geodésicos que constituem a fronteira das células da subdivisão de Voronoi em cada face.

O procedimento *Subdivisão-Voronoi*(f) constrói a subdivisão de f em células C_j que indicam os caminhos mínimos para pontos em f . O procedimento propaga os intervalos de otimalidade relativos a (e_i, f) para o interior da face f , considerando os pontos de interseção entre os arcos bissetores na fronteira das células.

Se um ponto t está na fronteira das células C_i e C_j , então $d(r_i) + d_{r_i, E_i}(t) = d(r_j) + d_{r_j, E_j}(t)$. Seja $h_{i,j}$ o *bissetor geodésico* contendo a fronteira entre as células C_i e C_j , definido por esta equação. Consideramos $h_{i,j}$ sendo orientado na direção em que a distância de r_i e r_j para t em f cresce. No PGAD, os bissetores geodésicos que limitam as células serão curvas polinomiais de grau muito alto. Neste caso, usaremos a rotina numérica *Ponto-Bissetor*(I_{i-1}, I_i, I_{i+1}) para determinar o ponto de interseção entre dois bissetores geodésicos. Se as células C_i e C_j compartilham uma fronteira, então denotamos por γ_k , o ponto dessa fronteira que é o mais próximo aos pontos r_i e r_j e está associado ao intervalo I_k , como um ponto evento para a subdivisão (veja Fig. 6.1).

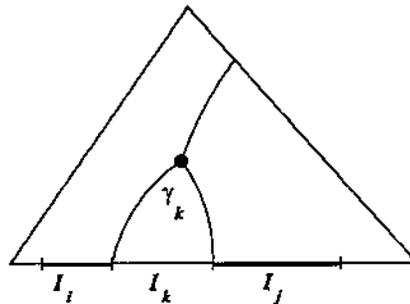


Figura 6.1: Um ponto evento para a subdivisão.

Procedimento *Subdivisão-Voronoi*(f)

1. $Intervalos := (I_0, I_1, \dots, I_k, I_{k+1})$.
2. para $(1 \leq i \leq k)$
 - (a) $\gamma_i := \text{Ponto-Bissetor}(I_{i-1}, I_i, I_{i+1})$.

- (b) se $(\gamma_i = \emptyset)$ então $\delta_i := +\infty$
- (c) senão
 - i. $\lambda := \text{Encontrar-Ponto}(\gamma_i, I_i)$.
 - ii. $\delta_i := d(r_i) + \lambda$.
- (d) Incluir(*Subeventos*, δ_i).

3. enquanto(*Subeventos* $\neq \emptyset$) faça

- (a) $\delta_j := \text{Desenfileira}(\textit{Subeventos})$.
- (b) Excluir(*Subeventos*, δ_j).
- (c) Excluir(*Intervalos*, I_j).
- (d) $\gamma_{j-1} := \text{Ponto-Bissetor}(I_{j-2}, I_{j-1}, I_{j+1})$.
- (e) se $(\gamma_{j-1} = \emptyset)$ então $\delta_{j-1} := +\infty$
- (f) senão
 - i. $\lambda := \text{Encontrar-Ponto}(\gamma_{j-1}, I_{j-1})$.
 - ii. $\delta_{j-1} := d(r_{j-1}) + \lambda$.
- (g) $\gamma_{j+1} := \text{Ponto-Bissetor}(I_{j-1}, I_{j+1}, I_{j+2})$.
- (h) se $(\gamma_{j+1} = \emptyset)$ então $\delta_{j+1} := +\infty$
- (i) senão
 - i. $\lambda := \text{Encontrar-Ponto}(\gamma_{j+1}, I_{j+1})$.
 - ii. $\delta_{j+1} := d(r_{j+1}) + \lambda$.
- (j) se $(q_j \neq q_{j-1})$ então *Aresta*($q_j, q_{j-1}, \gamma_j, v_{j-1,j}, h_{j-1,j}$).
- (k) se $(q_j \neq q_{j+1})$ então *Aresta*($q_j, q_{j+1}, \gamma_j, v_{j,j+1}, h_{j,j+1}$).
- (l) se $(q_j \neq q_{j-1}, q_{j+1})$ então $v_{j-1,j+1} := \text{Vértice}(\gamma_j, q_j, q_{j-1}, q_{j+1}, h_{j-1,j}, h_{j,j+1}, h_{j-1,j+1})$.

O passo 1 inicializa a lista *Intervalos* como uma lista circular contendo os intervalos de otimalidade relativos a (e_i, f) . O passo 2 computa δ_i como a distância ao ponto de interseção γ_i entre $h_{i-1,i}$ e $h_{i,i+1}$ caso estes bissetores se interceptem em f , do contrário $\delta_i = +\infty$. δ_i é incluído na fila de prioridades *Subeventos*. Os pontos γ_i representam uma das extremidades do próximo arco de bissetor a estar contido em uma aresta de Voronoi. O passo 3.a determina o próximo evento como o ponto com o menor valor δ_j em *Subeventos*. Os passos 3.b e 3.c excluem δ_j de *Subeventos* e I_j de *Intervalos*. Os passos 3.d a 3.f atualizam δ_{j-1} como a distância ao ponto de interseção γ_{j-1} entre $h_{j-2,j-1}$ e $h_{j-1,j+1}$ caso estes bissetores se interceptem em f , do contrário $\delta_{j-1} = +\infty$. Os passos 3.g a 3.i atualizam δ_{j+1} como a distância ao ponto de interseção γ_{j+1} entre $h_{j-1,j+1}$ e $h_{j+1,j+2}$ caso estes bissetores se interceptem em f , do contrário $\delta_{j+1} = +\infty$. Os bissetores $h_{j-1,j}$ e $h_{j,j+1}$ completam a fronteira da célula C_j , enquanto o bissetor $h_{j-1,j+1}$ continua a construção da fronteira das células C_{j-1} e C_{j+1} . O passo 3.j inclui o arco do bissetor $h_{j-1,j}$ com extremidades $v_{j-1,j}$ e γ_j na aresta de Voronoi entre os sítios q_j e q_{j-1} , onde q_i indica o sítio correspondente aos caminhos mínimos para os pontos do intervalo I_i . Da

mesma forma, o passo 3.k inclui o arco do bissetor $h_{j,j+1}$ com extremidades $v_{j,j+1}$ e γ_j na aresta de Voronoi entre os sítios q_j e q_{j+1} . O passo 3.l atualiza a subdivisão, estabelecendo o ponto evento $v_{j-1,j+1}$ como um vértice de Voronoi dos sítios q_{j-1} , q_j e q_{j+1} com arcos $h_{j-1,j}$, $h_{j,j+1}$, $h_{j-1,j+1}$ e mesmas coordenadas de γ_j .

A função numérica *Ponto-Bissetor*(I_{i-1}, I_i, I_{i+1}) encontra o ponto de interseção aproximado entre os bissetores relativos a I_{i-1}, I_i e I_i, I_{i+1} . A função realiza $O(L = \log \frac{\sqrt{3N}}{\delta})$ passos, executando a função *Ponto-Empate*(I, I'). Assim, a complexidade da função numérica *Ponto-Bissetor*(I_{i-1}, I_i, I_{i+1}) é $O(n^4 L^2)$.

Função *Ponto-Bissetor*(I_{i-1}, I_i, I_{i+1})

1. $[h_0, h_1] := \text{Perpendicular}(v, e)$.
2. $[a', b'] := [a_i, b_i]$.
3. enquanto $(h_1 - h_0 > \delta)$ faça
 - (a) $e' := \frac{h_1 + h_0}{2}$.
 - (b) $I'_{i-1} := \text{Projetar}(I_{i-1}, e')$.
 - (c) $I'_i := \text{Projetar}(I_i, e')$.
 - (d) $I'_{i+1} := \text{Projetar}(I_{i+1}, e')$.
 - (e) $a' := \text{Ponto-Empate}(I'_{i-1}, I'_i)$.
 - (f) $b' := \text{Ponto-Empate}(I'_i, I'_{i+1})$.
 - (g) se $(a' < b')$ então $h_0 := e'$ senão $h_1 := e'$.
4. retorne $(\frac{a'+b'}{2})$.

O passo 1 inicializa o intervalo $[h_0, h_1]$ do segmento perpendicular a aresta e na face f passando pelo vértice v oposto a e . Uma reta paralela a e que contém o ponto de interseção dos bissetores no interior de f deve interceptar este intervalo. O passo 2 atribui a extensão do intervalo I_i ao segmento $\overline{a'b'}$. O passo 3 verifica o tamanho do intervalo $[h_0, h_1]$. No passo 3.a, a reta e' paralela a aresta e passa pelo ponto médio do intervalo $[h_0, h_1]$. Os passos 3.b a 3.d projetam os intervalos sobre a reta e' , enquanto os passos 3.e e 3.f encontram os pontos de empate a' e b' dos intervalos projetados em e' . No passo 3.g, se a' está à esquerda de b' , então h_0 aumenta, senão h_1 diminui. O passo 4 retorna o ponto de interseção aproximado.

O procedimento *Subdivisão-Voronoi*(f) encontra os vértices da subdivisão em ordem crescente da distância a partir da origem (veja Fig. 6.2). Cada vértice corresponde ao fechamento de uma célula C_j quando excluimos o intervalo I_j da lista *Intervalos* e consideramos o bissetor geodésico entre as células correspondentes aos intervalos vizinhos a I_j na lista *Intervalos*.

Lema 6.1 [95] *O procedimento* *Subdivisão-Voronoi*(f) *constrói corretamente as células* C_j *correspondentes aos intervalos de otimalidade de* (e_i, f) .

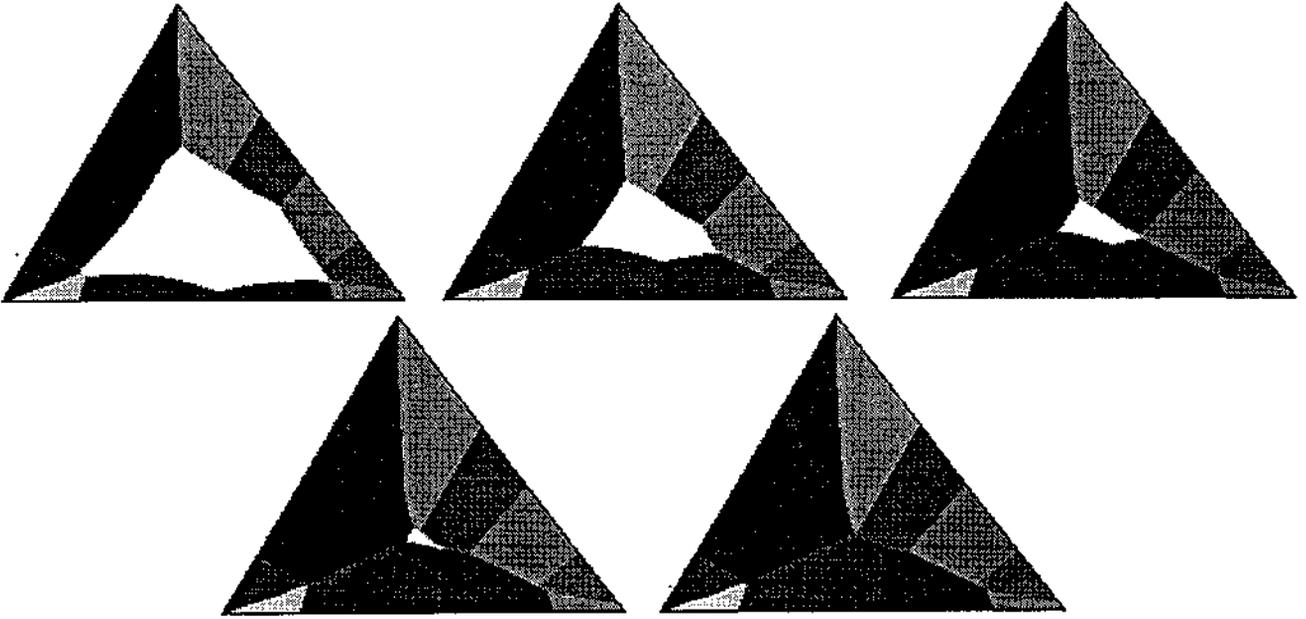


Figura 6.2: Células em uma subdivisão de uma face.

Prova. A prova procede por indução no número de vértices encontrados para a subdivisão. Suponha que δ_k é a distância de um ponto origem q_j ao ponto evento γ_i quando k vértices foram encontrados para a subdivisão. A subdivisão induzida pelo conjunto inicial de bissetores está correta a uma distância δ_1 do sítio mais próximo à face. Suponha, indutivamente, que a subdivisão está correta a uma distância δ_k e considere o efeito do evento γ_i . O evento γ_i corresponde ao fechamento de uma célula C_i e às células C_{i-1} e C_{i+1} se tornarem adjacentes depois da distância δ_k . Se t é um ponto a distância δ de qualquer origem, onde $\delta_k \leq \delta < \delta_{k+1}$, então t é localizado apropriadamente segundo a subdivisão criada pelo procedimento para pontos entre o k -ésimo e o $k+1$ -ésimo evento e não pode ser alcançado melhor pelo intervalo I_i , pois $t \notin C_i$. Portanto, o bissetor entre C_{i-1} e C_{i+1} localiza corretamente t entre os intervalos correspondentes. A localização de t relativa a todos os outros intervalos deve estar correta, pois todos os outros bissetores são disjuntos para distâncias que estão entre δ_k e δ_{k+1} e, pela hipótese de indução, assumimos que a subdivisão estava correta até então. \square

Lema 6.2 [95] *O procedimento Subdivisão-Voronoi(f) depende tempo $O(n^7 L^2)$ e usa espaço $O(n^3)$.*

Prova. Visto que, em cada iteração do procedimento, excluimos um dos intervalos da lista *Intervalos*, pode haver no máximo $O(n^3)$ execuções do passo 3, ou seja, o número de intervalos de otimalidade relativos a (e_i, f) . Cada execução envolve um número constante

de operações na lista *Subeventos*, cálculos de interseção entre bissetores e cálculos de distância a pontos. Dessa forma, a complexidade de tempo é $O(n^3(\log n + K(n) + G(n)))$, onde $K(n)$ é a complexidade da função *Encontrar-Ponto*(I, q) e $G(n) = O(n^4 L^2)$ é a complexidade da função *Ponto-Bissetor*(I_{i-1}, I_i, I_{i+1}). Assim, a complexidade de tempo do procedimento *Subdivisão-Voronoi*(f) é $O(n^7 L^2)$. \square

Executamos o procedimento *Subdivisão-Voronoi*(f) para cada face f , tomando um tempo $O(n^8 L^2)$, pois existe um total de $O(n^4)$ intervalos de otimalidade para todas as faces. Para associar um ponto t na superfície poliédrica a uma região de Voronoi, localizamos t em alguma face f [72, 108]. Então, localizamos t em alguma célula de f . A localização de pontos em uma célula envolve testar em que lado de uma curva polinomial o ponto está através da função numérica *Encontrar-Ponto*(I, q). Assim, a complexidade de uma localização de ponto é $O(n^4 L \log n)$. O caminho mínimo do sítio q_i associado a t é obtido em tempo $O(k)$ através dos k intervalos predecessores para o intervalo associado à célula obtida na localização de t até o sítio correspondente.

6.3 Redução de Problemas

Os problemas de proximidade em superfície poliédrica que abordamos são encontrar um par mais próximo em um conjunto finito Q de sítios, encontrar um vizinho mais próximo para cada sítio em Q e computar a árvore geradora mínima de Q . Apresentamos a redução destes problemas de proximidade ao diagrama de Voronoi de caminhos mínimos de Q . Portanto, a construção do diagrama de Voronoi de caminhos mínimos permite a resolução destes problemas com a mesma cota de tempo.

O *bissetor geodésico* de dois sítios q_i e q_j é o conjunto de pontos na superfície poliédrica que têm igual distância de caminho mínimo de q_i e de q_j . Um *vizinho mais próximo geodésico* de um sítio q_i é um sítio q_j tal que o comprimento do caminho mínimo de q_i a um ponto do bissetor de q_i e q_j seja mínimo dentre todos os outros sítios em Q . Um ponto do bissetor de q_i e q_j cuja distância é mínima denomina-se *ponto médio geodésico* de q_i e q_j .

Lema 6.3 *Seja $q_i, q_j \in Q$ tais que q_j é o vizinho mais próximo de q_i , então as regiões de Voronoi de q_i e q_j definem uma aresta do diagrama de Voronoi de caminhos mínimos VGAD.*

Prova. Seja c um ponto na superfície poliédrica cujas distâncias de q_i e q_j são iguais e mínimas. Suponha que c não está em uma aresta de Voronoi de q_i . Dessa forma, o caminho de q_i a c intercepta uma aresta de Voronoi entre q_i e q_k em u . Assim, $d(q_i, u) < d(q_i, c)$. Como u não deve ser o ponto médio geodésico o que está a uma distância mínima de q_i e

q_k , então $d(q_i, o) + d(q_k, o) \leq 2d(q_i, u) < 2d(q_i, c) = d(q_i, c) + d(q_j, c)$. Portanto, teríamos q_k mais próximo a q_i do que q_j , contradizendo a suposição de que q_j é o vizinho mais próximo de q_i . \square

Teorema 6.2 *O problema de Todos Vizinhos Mais Próximos em Superfície Poliédrica é transformável em tempo linear ao diagrama de Voronoi VGAD.*

Prova. Um vizinho mais próximo de um sítio q_i é obtido verificando cada aresta de Voronoi da região $R(i)$ (Lema 6.3). Toda aresta pertence a duas regiões de Voronoi, então cada aresta será examinada apenas duas vezes. Portanto, todos os vizinhos mais próximos são encontrados em tempo linear utilizando o diagrama de Voronoi VGAD. \square

O diagrama de Voronoi VGAD também pode ser usado para resolver o problema do Par Mais Próximo em Superfície Poliédrica. Este problema é transformável ao problema de Todos Vizinhos Mais Próximos em Superfície Poliédrica, pois um dos pares obtidos pelo problema de Todos Vizinhos Mais Próximos em Superfície Poliédrica é um par mais próximo que pode ser determinado em tempo linear. Portanto, temos:

Teorema 6.3 *O problema de Par Mais Próximo em Superfície Poliédrica é transformável em tempo linear ao problema de Todos Vizinhos Mais Próximos em Superfície Poliédrica.*

Geralmente, os algoritmos que resolvem o problema da árvore geradora mínima em grafos são baseados na existência de uma árvore geradora mínima que contém a menor aresta com um nó em Q_1 e o outro em Q_2 , onde $\{Q_1, Q_2\}$ é uma partição do conjunto de nós [110].

Dessa forma, um algoritmo deve examinar o comprimento da aresta de cada nó em Q_1 para cada nó em Q_2 . Contudo, a seguinte propriedade do diagrama de Voronoi de caminhos mínimos afirma a necessidade de examinar apenas as arestas do grafo planar dual, onde o conjunto de nós (duais às regiões de Voronoi) é o conjunto de sítios e existe uma aresta entre dois nós se existe uma aresta de Voronoi entre as regiões de Voronoi correspondentes. O comprimento de uma aresta ligando dois nós (relativos a $R(q_i)$ e $R(q_j)$) é duas vezes o comprimento do caminho mínimo do sítio q_i a um ponto médio geodésico de q_i e q_j .

Lema 6.4 *Para qualquer partição $\{Q_1, Q_2\}$ do conjunto de sítios Q , a menor aresta (par mais próximo) entre nós q_i de Q_1 e q_j de Q_2 pertence ao conjunto das arestas adjacentes a q_i no grafo dual.*

Prova. Assumimos que o comprimento δ do caminho mínimo de q_i a um ponto médio geodésico o de q_i e q_j é o menor entre nós de Q_1 e Q_2 , onde $q_i \in Q_1$ e $q_j \in Q_2$. Além disso, temos que q_j pertence ao conjunto de sítios adjacentes a q_i no grafo dual de Q . Caso contrário, o bissetor de q_i e q_j não contém uma aresta de Voronoi da fronteira da região de Voronoi $R(q_i)$ (Lema 6.3). Dessa forma, o caminho mínimo de q_i ao ponto médio o intercepta $R(q_i)$ em um ponto u . A aresta de Voronoi contendo u está contida em um bissetor entre q_i e q_k , onde $q_k \in Q$. Pela desigualdade do triângulo, temos $d(q_k, o) < d(q_k, u) + d(u, o)$. Como $d(q_k, u) = d(q_i, u)$, temos $d(q_k, o) < d(q_i, u) + d(u, o) = d(q_i, o)$. Portanto, o comprimento do caminho mínimo de q_k para o ponto médio o é menor que δ . Assim, quando $q_k \in Q_1$, temos $d(q_j, o') < d(q_j, o)$; e quando $q_k \in Q_2$, temos $d(q_i, o'') < d(q_i, o)$; onde o' e o'' são pontos médios dos bissetores entre q_j, q_k e q_i, q_k , respectivamente. Dessa forma, temos q_k como vizinho mais próximo de q_i ou q_j , contradizendo a suposição de que o comprimento do caminho mínimo de q_i a um ponto médio o de q_i e q_j é o menor entre nós de Q_1 e Q_2 . \square

Portanto, o problema de computar uma árvore geradora mínima de Q sobre uma superfície poliédrica pode ser reduzido em tempo linear ao diagrama de Voronoi de caminhos mínimos, pois é equivalente a computar uma árvore geradora mínima no grafo dual ao diagrama de Voronoi VGAD.

Teorema 6.4 *O problema da Árvore Geradora Mínima em Superfície Poliédrica é transformável em tempo linear ao diagrama de Voronoi de caminhos mínimos VGAD.*

Pelos Teoremas 6.2, 6.3 e 6.4, temos que os problemas de proximidade abordados são resolvidos em tempo $O(n^8 L^2)$ e espaço $O(n^4)$ pela construção do diagrama de Voronoi de caminhos mínimos VGAD.

Capítulo 7

Implementação

Uma razão para implementar um algoritmo geométrico inclui realizar uma especificação mais detalhada do algoritmo, identificando e tratando pormenores omitidos no seu projeto.

A implementação de algoritmos geométricos é uma atividade que pode envolver a interação com o usuário e a apresentação dos objetos geométricos produzidos. Dessa forma, o desenvolvimento de programas visualizadores e bibliotecas para interface tem auxiliado o desenvolvimento de atividades práticas relativas a algoritmos geométricos.

Implementamos o algoritmo Dijkstra contínuo para o problema da geodésica discreta [95] como um módulo externo ao visualizador *Geomview*. O módulo externo, escrito em C, permite a escolha de pontos origem sobre uma superfície poliédrica e anima a execução do algoritmo usando um painel de controle da biblioteca *FORMS* para interface com o usuário.

O módulo externo tem cerca de 2.000 linhas de código com 35 funções e procedimentos. O algoritmo Dijkstra contínuo representa 45% do código escrito. As rotinas geométricas básicas e estruturas de dados são aproximadamente 34% do código. A apresentação dos objetos geométricos, envolvendo a saída de arquivos formatados para o *Geomview* e a interface utilizando a biblioteca *FORMS* corresponde a apenas 21% das linhas escritas.

Na seção 7.1, apresentamos o programa *Geomview* para visualização de objetos geométricos. A seção 7.2 introduz a biblioteca *FORMS* para construção de painéis de interação. A seção 7.3 ilustra o módulo externo implementado. A estrutura de dados *quad-edge* que representa a superfície poliédrica é introduzida na seção 7.4. Na seção 7.5, discutimos a fila de prioridades e a lista de intervalos. A seção 7.6 descreve rotinas geométricas básicas usadas no algoritmo que resolve o problema da geodésica discreta. Na seção 7.7, indicamos alguns melhoramentos que realizamos neste algoritmo.

7.1 Geomview

O *Geomview* [102] é um programa interativo para visualização e manipulação de objetos geométricos tridimensionais no espaço euclidiano, hiperbólico ou esférico. O *Geomview* permite visualizar objetos estáticos ou funcionar como uma ferramenta que apresenta objetos geométricos produzidos dinamicamente por outros programas.

O *Geomview* foi desenvolvido por membros do *Geometry Center* na Universidade de Minnesota e executa em estações *Silicon Graphics (SGI)*, dentre vários outros sistemas. Particularmente, utilizamos a versão 1.6.1 do *Geomview* em uma *SGI Indigo2* para o sistema operacional *IRIS*. O *Geomview* está disponível gratuitamente pela Internet em 'ftp://ftp.geom.umn.edu'.

O *Geomview* mostra objetos cuja geometria é descrita em arquivos com sintaxe definida segundo a biblioteca de objetos gráficos *OOGL* sobre a qual o *Geomview* foi construído. Além disso, permite o controle interativo sobre o ponto de vista, a velocidade de movimento e a aparência de superfícies, dentre outros. O *Geomview* suporta tipos de dados como poliedros, quadriláteros, malhas retangulares, superfícies Bezier e muitos outros. Inicialmente, o *Geomview* apresenta o painel de controle principal, o painel de ferramentas e uma câmara (veja Fig. 7.1). O *painel principal* tem um menu por onde se pode acessar os demais painéis de controle, uma lista de objetos (*targets*) e uma lista de módulos externos instalados no sistema (*modules*). O *painel de ferramentas* permite selecionar o tipo de movimento e a *câmara* mostra os objetos geométricos carregados.

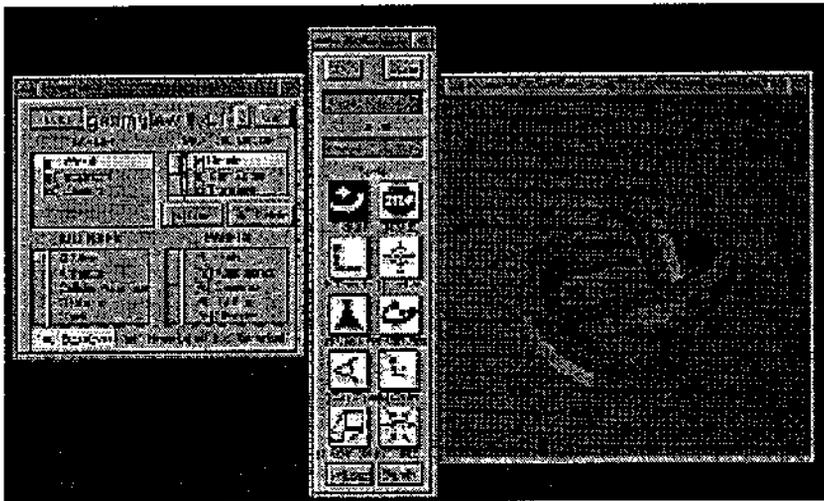


Figura 7.1: A apresentação inicial do *Geomview*.

O *Geomview* apresenta objetos originados de outro programa que interage com o *Geomview* e executa simultaneamente. À medida que este programa altera a descrição geométrica de um objeto, a imagem mostrada pelo *Geomview* reflete as mudanças. Pro-

gramas que geram objetos e usam o *Geomview* para apresentação são chamados *módulos externos*. Geralmente, um módulo externo consiste na implementação de algum algoritmo que cria um objeto geométrico que muda a forma enquanto o algoritmo executa e os aspectos de apresentação são deixados para o *Geomview*.

Um módulo externo se comunica com o *Geomview* através da linguagem de comandos do *Geomview* (gcl). Esta linguagem, apesar de ter a sintaxe de lisp, expressa apenas comandos a serem executados em uma dada ordem. Dessa forma, um módulo externo pode controlar qualquer aspecto do *Geomview*.

Quando o *Geomview* chama um módulo externo, cria pipes conectados à entrada e saída padrão do módulo. Pipes são como arquivos usados para comunicação entre programas. O *Geomview* interpreta o que o módulo escreve na saída padrão como um comando gcl. Dessa forma, o módulo pode informar ao *Geomview* qualquer mudança de um objeto geométrico em tempo de execução. Se o módulo externo pede alguma informação ao *Geomview*, o *Geomview* escreve a informação na entrada padrão do módulo.

7.2 FORMS

Um módulo se comunica com o usuário por um painel de controle ou respondendo a certos eventos. Um painel para interface com usuário pode utilizar a biblioteca *FORMS* de domínio público em 'ftp://ftp.laue.phys.uwm.edu/pub/XForms' ou qualquer outra biblioteca para GUI (*Graphical User Interface*).

A biblioteca *FORMS* é uma biblioteca de rotinas em C, escrita por Mark Overmars, que permite construir painéis de interação. A biblioteca *FORMS* usa os serviços proporcionados por Xlib e pode ser compilada por um compilador ANSI C ou C++ em máquinas com o sistema de janelas X instalado.

A utilização da biblioteca *FORMS* envolve definir um conjunto de painéis, especificando os objetos que compõem cada painel: botões, textos, campos de entrada, dentre outros (veja Fig. 7.2). Depois que um painel é definido, podemos mostrá-lo na tela. A rotina *fl-do-forms()* cuida da interação entre o usuário e o painel. Esta rotina resgata eventos de uma fila de eventos, retornando qualquer mudança no estado do painel devido a alguma ação do usuário. A rotina envia o evento obtido pelo objeto apropriado e notifica à aplicação qual ação deve ser tomada. O controle é retornado ao programa que realiza a ação correspondente e passa o controle para *fl-do-forms()* novamente.

7.3 Módulo Externo

O módulo externo implementado responde a eventos (clique do mouse) selecionando um vértice da superfície poliédrica como ponto origem e tem um painel de controle *FORMS*

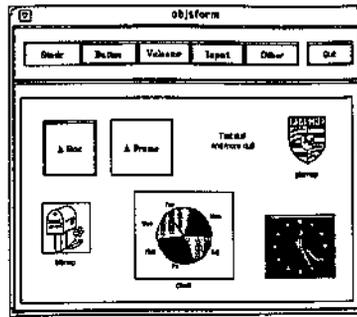


Figura 7.2: Um painel de interação *FORMS*.

para interface com usuário (veja Fig. 7.3). O painel de controle contém um botão luminoso para identificar se a execução do algoritmo será instantânea ou através de uma animação passo a passo. Além disso, o painel possui quatro botões para controlar a animação: tocar, avançar, pausa e parar. O botão tocar inicia a execução instantânea ou animada. O botão avançar realiza um passo do laço principal do algoritmo que permanece em pausa. O botão pausa interrompe a execução do algoritmo, e o botão parar termina a execução. Na animação, o módulo apresenta os objetos geométricos a cada passo, enquanto a execução instantânea apresenta os objetos apenas quando o algoritmo conclui.

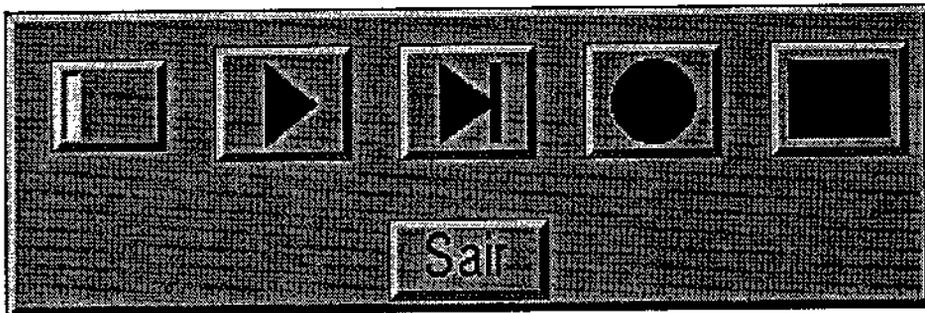


Figura 7.3: O painel de controle do módulo externo.

Os objetos geométricos são apresentados sobre uma superfície poliédrica carregada no *Geomview* (veja Fig. 7.4). Identificamos os pontos origem e o evento rotulado permanentemente em cada passo através de esferas centradas nestes pontos. Além disso, o módulo mostra a árvore de caminhos mínimos para os eventos rotulados permanentemente e a árvore dos eventos atuais ainda não rotulados. Assim, temos dois objetos geométricos que consistem em dois conjuntos de linhas poligonais abertas, onde cada linha poligonal representa um caminho mínimo (computado até então) de uma origem para um evento (ainda não) rotulado permanentemente.

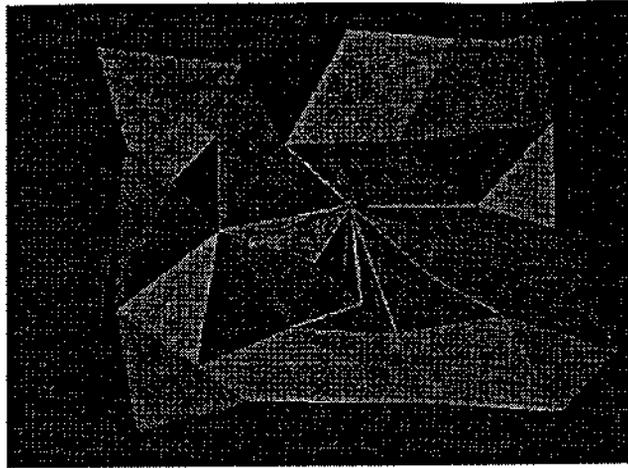


Figura 7.4: Os objetos geométricos apresentados no Geomview pelo módulo externo.

7.4 Quad-edge

A implementação do algoritmo Dijkstra contínuo representa a superfície poliédrica através da estrutura de dados *quad-edge*. Para isso, usamos a biblioteca *libquad* escrita por J. Stolfi e baseada na implementação original de J. Roth.

A *quad-edge* [57] é uma estrutura de dados que permite a representação de grafos não-direcionados imersos em uma superfície, onde um operador topológico (*Splice*) e uma primitiva (*MakeEdge*) para criação de aresta são suficientes para construção e modificação de qualquer diagrama.

A *quad-edge* representa, simultaneamente e uniformemente, os diagramas primal e dual; permite trocar de uma representação para outra em tempo constante; e manipular seus parâmetros geométricos e topológicos independentemente.

Uma *álgebra de aresta* é um objeto combinatório finito que captura as propriedades topológicas de uma subdivisão S . As arestas da álgebra são particionadas em grupos que consistem em quatro versões orientadas e direcionadas de uma aresta não-direcionada de S e quatro versões das arestas duais. Uma subdivisão tem duas orientações possíveis relativas ao sentido de rotação e duas direções definindo uma ordem entre os pontos de uma linha. O grupo contendo uma aresta e é a *órbita* de e sob a subálgebra gerada pelas funções *Rot* e *Flip*. A aresta e *Rot* é a versão rotacionada (dual) de e , direcionada da direita para a esquerda, e orientada tal que se mover no sentido anti-horário na face à direita de e corresponde a se mover no sentido anti-horário segundo a origem de e . A aresta e *Flip* é a versão da aresta e com orientação oposta e mesma direção. A estrutura de dados *quad-edge* implementa a álgebra de aresta. Selecionamos uma aresta e_0 canônica em cada grupo e qualquer aresta e deste grupo é uma versão e_0 *Rot* ^{r} *Flip* ^{f} , onde $r \in \{0, 1, 2, 3\}$ e $f \in \{0, 1\}$.

Representamos o grupo de arestas contendo e na estrutura de dados por um *vetor de aresta* $e[0 \dots 3]$, onde $e[r]$ corresponde à aresta $e_0 \text{ Rot}^r$. Uma *referência de aresta* é uma tripla (e, r, f) que representa uma aresta $e = e_0 \text{ Rot}^r \text{ Flip}^f$. Cada elemento $e[r]$ de um vetor de aresta contém dois campos: *Data* e *Next*. O campo *Data* guarda informações geométricas e não topológicas, enquanto o campo *Next* contém uma referência à aresta $e_0 \text{ Rot}^r \text{ Onext}$, onde $e \text{ Onext}$ é a aresta seguinte a e com a mesma origem no sentido anti-horário e $e \text{ Lnext}$ é a aresta seguinte com a mesma face à esquerda no sentido anti-horário.

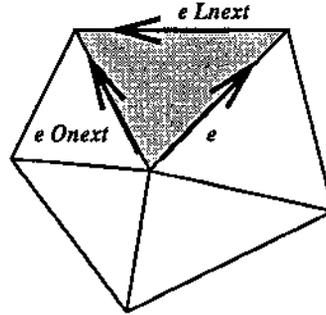


Figura 7.5: As primitivas *Onext* e *Lnext* da *quad-edge*.

A seqüência de arestas em torno de um vértice $e \text{ Org}$ é obtida pela aplicação da primitiva *Onext*, enquanto obtemos a seqüência de arestas ao redor da face à esquerda de uma aresta e aplicando a primitiva *Lnext*.

7.5 Fila de Prioridades e Lista de Intervalos

Uma *fila de prioridades* é uma estrutura de dados para um conjunto cujos elementos estão associados a um valor denominado *chave* [35]. Uma fila de prioridades suporta as operações de incluir um elemento no conjunto, obter e excluir o elemento com a menor chave.

Uma aplicação para filas de prioridades está na simulação de eventos discretos. Os elementos na fila são eventos que devem ser simulados na ordem de um tempo de ocorrência (chave) associado a cada evento. Dessa forma, a simulação de um evento pode causar outros eventos a serem simulados. O algoritmo Dijkstra contínuo usa a operação *Desenfileira(Eventos)* em cada passo do laço principal para escolher o próximo evento e simular uma frente de onda se propagando. Quando novos eventos são produzidos, eles são incluídos na fila de prioridades pelo operador *Incluir-Evento(Eventos, q)*.

Um *heap* implementa uma fila de prioridades necessitando tempo $O(\log n)$ para qualquer operação sobre um conjunto de n elementos. Contudo, utilizamos a representação mais simples de uma fila de prioridades que é uma lista desordenada. Dessa forma, as operações *Desenfileira(Eventos)* e *Excluir-Evento(Eventos, q, I)* dependem tempo linear,

e a operação *Incluir-Evento*(*Eventos*, *q*, *I*) depende tempo constante, onde *q* é um evento que representa um ponto sobre a superfície poliédrica em um intervalo candidato *I*.

A operação *Retirar-Intervalo*(*Eventos*, *I*) exclui todos os eventos correspondentes ao intervalo candidato *I* da fila de prioridades em tempo linear. Esta operação é realizada sempre que excluimos um intervalo *I* de alguma lista de intervalos.

As listas de intervalos foram implementadas como listas ordenadas duplamente ligadas. O procedimento *Incluir-Intervalo*(*LI*, *I*) depende tempo linear para encontrar a posição de *I* na lista de intervalos ordenada *LI* e tempo constante para incluir o intervalo na lista. O procedimento *Excluir-Intervalo*(*I*) utiliza os apontadores para intervalos predecessor e sucessor para excluir o intervalo *I* da lista de intervalos correspondente em tempo constante.

7.6 Rotinas Básicas

Geralmente, o projeto de algoritmos geométricos resulta em uma especificação que usa conceitos geométricos usuais representados por estruturas de dados e rotinas complexas. Assim, a implementação destes algoritmos requer um esforço adicional para a tradução da semântica intrínseca à geometria.

A rotina *Orientação*(*s*, *q*, *r*) retorna a orientação dos três pontos (*s*, *q*, *r*), onde a orientação da tripla (*s*, *q*, *r*) é o sentido em que o segmento \overline{st} roda em torno de *s* enquanto o ponto *t* vai de *q* para *r* sobre o segmento \overline{qr} . Assumimos que *q* está à direita de *r* quando o sentido de orientação é anti-horário e dizemos que a tripla (*s*, *q*, *r*) tem orientação positiva. De forma semelhante, se a tripla tem orientação negativa, o ponto *q* está à esquerda de *r*. A função obtém o sinal do determinante das coordenadas dos três pontos que é precisamente a orientação da tripla (*s*, *q*, *r*). Assim, a orientação de (*s*, *q*, *r*) é dada por

$$\begin{vmatrix} 1 & s_x & s_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{vmatrix}.$$

A função *Interseção*(*q*₀, *r*₀, *q*₁, *r*₁) retorna o ponto de interseção entre as retas *R*₀ e *R*₁ determinadas pelos pares de pontos (*q*₀, *r*₀) e (*q*₁, *r*₁), respectivamente. A reta *R*_{*i*}, para *i* = 0, 1, tem seus coeficientes definidos pela expressão $y = a_i x + b_i$, ou melhor,

$$y = \left(\frac{r_{iy} - q_{iy}}{r_{ix} - q_{ix}} \right) x + \left(\frac{r_{ix} q_{iy} - q_{ix} r_{iy}}{r_{ix} - q_{ix}} \right).$$

Esta equação não está bem definida quando *R*_{*i*} é perpendicular ao eixo das abscissas. Neste caso, a reta *R*_{*i*} é representada pela equação $x = r_{ix} = q_{ix}$. As coordenadas (*x*, *y*) do ponto de interseção devem satisfazer o sistema das equações lineares correspondentes

às duas retas R_0 e R_1 . A solução deste sistema linear é possível e determinada quando as retas são disjuntas e transversais. Nos casos onde as retas são coincidentes ou paralelas, a função retorna um ponto qualquer ou \emptyset , respectivamente.

Seja e uma aresta especificada pelos pontos a e b . Sejam f e f' as faces especificadas pela aresta e e pelos vértices v e v' , respectivamente, não incidentes em e . A função $Imagem(a, b, v, v')$ retorna um ponto coincidente a v' quando rotacionamos f' em torno da aresta e tal que f e f' se tornem coplanares. O ponto obtido é a *imagem* do ponto v' sobre o plano da face f . Inicialmente, a função translada os pontos de forma que o ponto a se torne coincidente com a origem dos eixos. Rotacionamos os pontos em torno do eixo das alturas tal que b_y seja igual a zero e em torno do eixo das ordenadas tal que b_z seja igual a zero. Neste momento, temos a aresta e contida no eixo das abscissas. Rotacionamos os pontos em torno do eixo das abscissas tal que v_z seja nulo. Finalmente, rotacionamos v' em torno do eixo das abscissas tal que os pontos a, b, v e v' estejam no plano horizontal $z = 0$ e $v_y v'_y < 0$ para que f e f' não estejam sobrepostas. A função realiza as transformações inversas sobre o ponto v' e retorna o ponto resultante.

A função $Ponto-Empate(I, I')$ retorna o ponto q na interseção entre os intervalos candidatos I e I' tal que $d(r) + d_{r,E}(r, q) = d(r') + d_{r',E}(r', q)$ (veja Fig. 7.6). Inicialmente, a função realiza transformações sobre os pontos dos intervalos e suas raízes para que estejam contidos no plano horizontal $z = 0$ e a aresta que contém os intervalos esteja contida no eixo das abscissas. Assim, a ordenada do ponto q deve ser igual a zero e a abscissa x deve satisfazer a equação da hipérbole cujos focos são as raízes dos intervalos. Dessa forma, deduzimos a equação quadrática $(4c_1^2 - c_2^2)x^2 - (8c_1^2r_x + 2c_2c_3)x + 4c_1^2(r_x^2 + r_y^2) - c_3^2 = 0$, onde $c_1 = |d(r) - d(r')|$, $c_2 = 2(r_x - r'_x)$ e $c_3 = r_x'^2 + r_y'^2 - r_x^2 - r_y^2 - c_1^2$. Quando esta equação não tem solução real, a hipérbole não intercepta os intervalos de otimalidade I, I' e um dos intervalos domina o outro. A equação tem solução em no máximo dois pontos. Assim, a função retorna o único ponto, quando existe, que está contido na interseção dos intervalos e está associado ao braço da hipérbole onde $c_1 = d(r) - d(r')$.

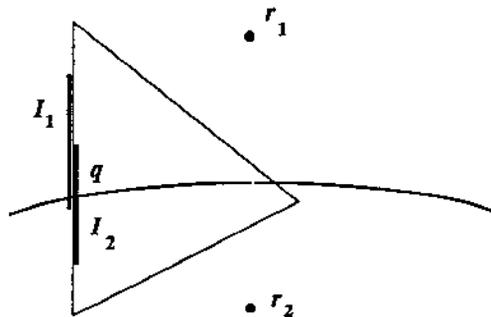


Figura 7.6: A função $Ponto-Empate(I, I')$.

A função $Canal-Acesso(I)$ retorna um par (I_1, I_2) de intervalos de otimalidade ad-

jacentes ($b_1 = a_2$) tal que o ponto fronteira \bar{c} do intervalo \bar{I} predecessor de I está na região limitada pelos segmentos $\overline{c'_1 c_1}$ e $\overline{c'_2 c_2}$ (veja Fig. 7.7). I_1 (I_2) é vazio quando \bar{c} está na região limitada pela aresta e_1 (e_2) e pelo segmento $\overline{c'_2 c_2}$ ($\overline{c'_1 c_1}$). A função realiza uma busca linear sobre a lista de intervalos do par aresta-face associado ao intervalo I . A função pára quando o ponto de acesso c'_i correspondente a um intervalo I_i está à esquerda de \bar{c} segundo um ponto q no interior do intervalo I e retorna o par (I_{i-1}, I_i) . Quando q , \bar{c} e c'_i são colineares, o ponto q é substituído pelo vértice v oposto a aresta e na face f . Neste caso, a busca continua se c'_i está à esquerda de \bar{c} segundo v . Além disso, os pontos \bar{c} e c'_i , quando coincidentes, são substituídos por pontos no interior de \bar{I} e \bar{I}_i , respectivamente.

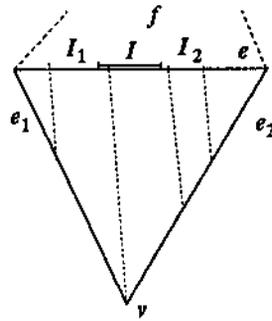


Figura 7.7: A função $\text{Canal-Acesso}(I)$.

A função $\text{Projetar}(I, e)$ retorna o intervalo de otimalidade I' dos pontos q sobre a aresta e cujo segmento \overline{qr} intercepta o intervalo I , onde r é a imagem da raiz do intervalo I sobre o plano da face associada a I (veja Fig. 7.8). A função rotaciona o ponto r em torno do intervalo I tal que r seja coplanar a I e a aresta e . A função encontra as interseções q_a e q_b da reta suporte da aresta e com as semi-retas \overrightarrow{ra} e \overrightarrow{rb} , respectivamente. Os casos possíveis a serem considerados são: $q_a, q_b \in e$, $q_a \in e \wedge q_b \notin e$ e $q_a \notin e \wedge q_b \in e$. No primeiro caso, as extremidades de I' são q_a e q_b . No segundo caso, as extremidades de I' são q_a e v ou I' é vazio, onde v é o vértice oposto ao intervalo I na aresta e . O terceiro caso é simétrico ao segundo. Os casos degenerados onde as semi-retas \overrightarrow{ra} e \overrightarrow{rb} são paralelas à reta suporte de e são tratados assumindo como interseção uma das extremidades de e .

7.7 Melhoramentos no Algoritmo

Durante a implementação, identificamos algumas omissões no algoritmo Dijkstra contínuo para o problema da geodésica discreta (PGD) [95]. Melhoramos o algoritmo e preenchemos algumas lacunas deixadas na prova de corretude apresentada por Mitchell *et al.* [95]. Além disso, verificamos que a lista de intervalos $ILIST$ sugerida por Mitchell *et al.*, bem como os ponteiros para cada intervalo candidato para o qual um vértice é a raiz, são supérfluos.

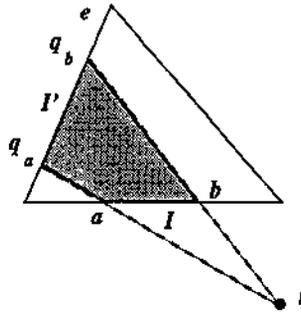


Figura 7.8: A função $Projeta(I, e)$.

No PGD, a função correspondente a $Atualizar-Intervalo(I, q)$ realiza chamadas a função $Ponto-Empate(I, I_1)$ para obter o ponto q' em $I \cap I_1$ tal que $d(r) + |\bar{r}q'| = d(r_1) + |\bar{r}_1q'|$, onde \bar{r} é a imagem da raiz r sobre o plano da face correspondente a I . Os intervalos podem ser ajustados assumindo q' como extremidade. Neste caso, o algoritmo atualiza o ponto fronteira c_1 do intervalo I_1 , mas não determina que o evento correspondente deve ser excluído da fila de prioridades quando $c_1 \notin I_1$ depois do ajuste.

A função $Atualizar-Intervalo(I, q)$, para o PGD, assume que um caminho mínimo computado até então não pode interceptar uma fronteira $\bar{c}_i c_i$ de um canal de acesso. Dessa forma, o algoritmo ignora a possibilidade do intervalo I ser dominado por algum intervalo existente. Um contra-exemplo onde o intervalo I não é incluído ocorre quando propagamos um evento que representa um vértice e uma aresta oposta a este vértice já foi melhor alcançada por um intervalo de otimalidade existente.

A função $Propagar(I)$ cria um intervalo f_i -livre em cada aresta e_i oposta ao ponto fronteira c de I , quando c é um vértice. Neste momento, o algoritmo deveria verificar se o caminho mínimo para o vértice c é f_i -livre, senão poderá ser criado um intervalo de otimalidade sobre a aresta e_i cujos caminhos mínimos induzidos não são f_i -livres, contradizendo a definição de intervalo de otimalidade no par aresta-face (e_i, f_i) .

O laço principal do algoritmo PGD propaga intervalos associados apenas a eventos rotulados permanentemente como pontos fronteira. Contudo, afirmamos que qualquer evento associado a um vértice deve ser propagado independente de ser um ponto fronteira ou não. Se o algoritmo não propagar eventos correspondentes a vértices que não são pontos fronteira, o Lema 4.3 não valerá para o PGD (Lema 6.6 de [95]), ou seja, pode existir um ponto que não está contido em um intervalo de otimalidade quando o algoritmo PGD conclui. Dessa forma, um contra-exemplo para o Lema 4.3 pode ser facilmente obtido no plano (veja Fig. 7.9). Neste contra-exemplo, o evento c_1 é rotulado permanentemente e propaga o intervalo $[v_1, v_2]$ sobre a aresta $\bar{v}_1 \bar{v}_3$. O intervalo criado tem c_2 como ponto fronteira e não cobre inteiramente a aresta $\bar{v}_1 \bar{v}_3$. Além disso, o evento c_2 é rotulado permanentemente sem criar nenhum intervalo candidato. Assim, o algoritmo termina sem

os pontos dos segmentos $\overline{b_2v_3}$ e $\overline{v_2v_3}$ estarem contidos em algum intervalo de otimalidade.

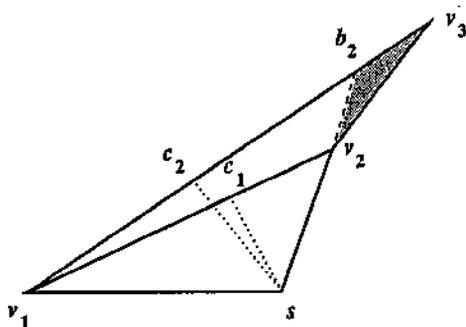


Figura 7.9: Um contra-exemplo para o Lema 6.6 [95] no PGD.

Capítulo 8

Conclusões

Buscando resultados teóricos e práticos, estudamos uma área da geometria computacional ainda pouco explorada. Analisamos superfícies poliédricas sob o enfoque da obtenção de algoritmos geométricos.

Propomos algoritmos para determinação de estruturas geométricas sobre uma superfície poliédrica. Assim, utilizando uma função distância de caminhos mínimos, abordamos problemas de caminhos mínimos e proximidade considerando superfícies de inclinação variável.

Desenvolvemos um módulo para visualização de soluções de um problema geométrico em superfície poliédrica. Implementamos o algoritmo que resolve o problema da geodésica discreta e sua extensão para o diagrama de Voronoi através do *Geomview*, alcançando objetivos práticos relativos a problemas de caminhos mínimos e proximidade. Além disso, representamos e manipulamos uma superfície poliédrica através da estrutura de dados *quad-edge*.

8.1 Resultados Obtidos

O estudo de objetos geométricos e de problemas da geometria computacional em superfície poliédrica fez necessária a definição de uma métrica para o cálculo de distâncias entre pontos neste tipo de superfície. Uma forma de se definir tais métricas que permite grande liberdade para generalizações é

$$d(q_1, q_n) = \min \sum_{i=1}^{n-1} \Delta'(q_i, q_{i+1}),$$

onde q_1, \dots, q_n é uma seqüência de pontos que descreve um caminho mínimo entre q_1 e q_n , e Δ' é a distância entre pontos em uma mesma face. Para considerar a direção de movimento, definimos a função distância FGAD baseada no trabalho realizado por um

corpo para se movimentar. Esta função distância implicou em uma versão direcional do problema de caminhos mínimos.

Obtivemos uma série de resultados relevantes relacionados a este problema. Estes resultados demonstraram a caracterização de caminho geodésico, localmente f -livre e mínimo segundo as restrições e especificações do problema, ou seja, sobre uma superfície poliédrica segundo a função distância FGAD. Assim, demonstramos que caminhos geodésicos são lineares por partes e identificamos o critério de otimalidade local correspondente a esta situação.

Neste momento, utilizando a teoria de funções convexas e programação convexa, demonstramos propriedades de convexidade estrita da função distância FGAD. A convexidade desta função garantiu a unicidade do mínimo de funções reais relacionadas ao critério de otimalidade. A convexidade da função FGAD também é um fato fundamental para a unicidade do caminho geodésico que intercepta uma seqüência de arestas como enunciado no Lema 3.4.

Verificamos a existência de ângulos críticos como no problema da região valorada [97] e deduzimos expressões para representá-los. Além disso, constatamos que um caminho geodésico p no PGAD pode ter vários segmentos críticos entre dois vértices consecutivos encontrados na lista de pontos descrevendo p .

Os lemas que apresentam propriedades de caminho localmente f -livre e intervalo de otimalidade generalizam estes conceitos para o problema PGAD embora as demonstrações sejam muito semelhantes às encontradas na literatura [95, 97].

As contribuições originais incluíram ainda o desenvolvimento de um algoritmo, baseado na metodologia Dijkstra contínuo [95, 97, 93], para resolução do problema de caminhos mínimos PGAD que generaliza o problema da região valorada e o problema da geodésica discreta. Nosso algoritmo resultou de uma extensão aos algoritmos existentes na literatura para considerar o problema PGAD que leva em conta um coeficiente de atrito cinético e um ângulo de inclinação com o plano horizontal em cada face de uma superfície poliédrica.

O algoritmo que obtivemos para resolver o problema PGAD é muito semelhante aos demais algoritmos baseados na metodologia Dijkstra contínuo. Contudo, a função $Projetar(I, e_i)$ e o procedimento $Atualizar-Intervalo(I, q)$ fazem chamada a procedimentos relacionados à caracterização dos caminhos geodésicos e à função distância FGAD. Neste nível, o algoritmo torna-se diferente dos demais assumindo o caráter direcional.

A resolução do PGAD proporcionou uma métrica para caminhos mínimos em superfície poliédrica. Esta métrica permitiu o estudo de problemas de proximidade neste espaço. Assim, resolvemos os seguintes problemas:

- Problema do Par Mais Próximo em Superfície Poliédrica,
- Problema de Todos Vizinhos Mais Próximos em Superfície Poliédrica,

- Problema da Árvore Geradora Mínima em Superfície Poliédrica.

Estendemos o problema de caminhos mínimos PGAD para o caso de múltiplas origens. Dessa forma, obtivemos um algoritmo para construir um diagrama de Voronoi com uma função distância de caminhos mínimos e provamos a corretude deste algoritmo. A especificação deste algoritmo envolveu modificar a inicialização do algoritmo de origem única para incluir os intervalos de otimalidade associados a todos os pontos de origem (sítios).

Além disso, apresentamos um procedimento para obter uma descrição do diagrama homeomorfa a uma subdivisão planar. Este procedimento constrói uma subdivisão em cada face da superfície poliédrica, identificando os vértices e arestas do diagrama de Voronoi de caminhos mínimos VGAD.

Utilizamos este diagrama de Voronoi para resolver os problemas de proximidade com a mesma cota de tempo. Para isso, demonstramos que estes problemas são redutíveis à construção do diagrama de Voronoi de caminhos mínimos.

Implementamos um algoritmo Dijkstra contínuo em superfície poliédrica que resolve o problema da geodésica discreta [95]. Desenvolvemos um módulo externo ao *Geomview* para visualização da árvore de caminhos mínimos sobre uma superfície poliédrica. Utilizamos a estrutura de dados *quad-edge* para representar uma superfície poliédrica [57]. A implementação do algoritmo para o diagrama de Voronoi foi uma adaptação ao algoritmo de caminhos mínimos para considerar múltiplas origens.

Alguns detalhes importantes no algoritmo proposto por Mitchell *et al.* que resolve o PGD foram omitidos. Identificamos e tratamos, dentre outros detalhes, a possibilidade de um intervalo sendo incluído ser dominado por intervalos existentes e a não propagação de vértices que não são pontos fronteira.

8.2 Extensões e Trabalhos Futuros

Algumas extensões e trabalhos futuros são sugeridos nesta seção. As sugestões consistem em generalizações do problema que estudamos e outras variações.

Podemos investigar uma métrica ou função distância que considere direção de movimento e seja algebricamente mais simples. Isto pode implicar na obtenção de um critério de otimalidade local mais intuitivo, possibilitando uma interpretação como a propriedade de desdobramento e a Lei de Snell para os problemas da geodésica discreta e da região valorada, respectivamente. Além disso, podemos propor uma função não convexa. A obtenção de um algoritmo exato para este caso requer o estudo de múltiplos caminhos geodésicos entre dois pontos em uma seqüência de arestas.

Algumas aplicações precisam tratar regiões com coeficiente de atrito nulo e infinito (obstáculos). Dessa forma, podemos estender o algoritmo Dijkstra contínuo direcional

para considerar este caso. Quando temos regiões com coeficiente de atrito nulo, podemos abordar um diagrama de Voronoi de caminhos mínimos com sítios representados por polígonos simples. Neste caso, o sítio consiste em regiões com coeficiente de atrito nulo e a distância entre dois sítios é o menor comprimento de um caminho mínimo entre pontos em cada um dos dois sítios.

O diagrama de Voronoi de caminhos mínimos de ordem k pode ser obtido quando propagamos um intervalo candidato mesmo quando este não implica em caminhos mínimos para seus pontos. Neste caso, o intervalo I propaga até que existam k intervalos de otimalidade correspondentes a k sítios distintos que induzam k caminhos de comprimento menor do que o comprimento do caminho mínimo a partir da origem correspondente ao intervalo I para todos os pontos do intervalo I .

Uma generalização possível para o problema da geodésica atritada direcional é o problema tridimensional, onde as regiões são poliedros que particionam o espaço euclidiano tridimensional. Consideramos uma força (atração gravitacional) constante com direção fixa em todo espaço. Além disso, temos um coeficiente de resistência implicando uma força de atrito na direção do movimento e uma força (normal) constante com direção fixa no interior de cada região poliédrica. O problema tridimensional consiste em encontrar um caminho de trabalho mínimo para se mover um corpo de uma posição a outra do espaço.

A implementação dos algoritmos que resolvem o problema da região valorada sobre uma superfície poliédrica e o problema da geodésica atritada direcional são trabalhos futuros baseados no módulo externo implementado. A implementação de um algoritmo para o PRV requer, dentre outras coisas, o tratamento de segmentos críticos e o uso de funções numéricas aproximativas, enquanto um algoritmo para o PGAD deve ainda tratar um critério de otimalidade algebricamente mais complicado.

Apêndice A

O Ângulo β_x

Neste apêndice, vamos encontrar uma expressão em função de x para o cosseno do ângulo β_x . Primeiramente, identificamos os casos distintos para a disposição da face f segundo os valores de a e b (veja Fig. A.1). A aresta e pode ter o coeficiente angular, a , positivo, nulo ou negativo. Analogamente, o coeficiente linear b pode ser maior, igual ou menor que a altura z_0 do ponto $s(0, -y_0, z_0)$.

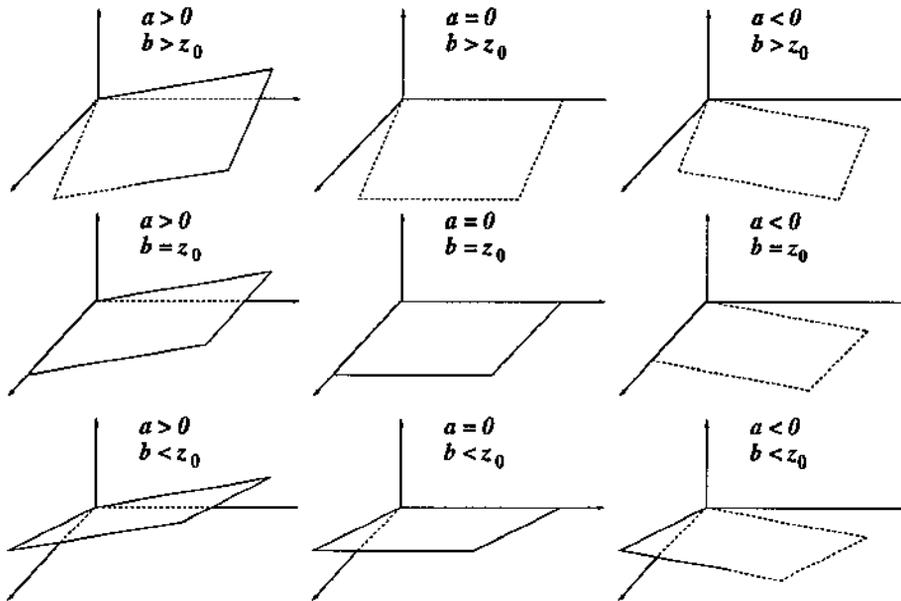


Figura A.1: Casos distintos para a disposição da face f .

Seja q o ponto sobre a aresta e com a mesma altura que s (veja Fig. A.2). Dessa forma, o ponto q tem abscissa $\frac{z_0 - b}{a}$, ou seja, q tem coordenadas $(\frac{z_0 - b}{a}, 0, z_0)$. O vetor da força resultante R_{PN} tem direção perpendicular ao segmento \overline{sq} . Seja q' o ponto de

interseção entre a reta passando por s na mesma direção do vetor da força R_{PN} e a aresta e . Dessa forma, temos o triângulo retângulo $\Delta qsq'$ com ângulo reto em s . Pelo teorema de Pitágoras, temos $|qq'|^2 = |sq|^2 + |sq'|^2$. Portanto, a abscissa do ponto q' é $q'_x = \frac{ay_0^2}{b-z_0}$. Seja r o ponto sobre a aresta e tal que o segmento \overline{sr} é perpendicular à aresta e . Agora, pelo triângulo Δors , onde o é a origem dos eixos coordenados, temos que a abscissa de r é $r_x = \frac{a(z_0-b)}{1+a^2}$.

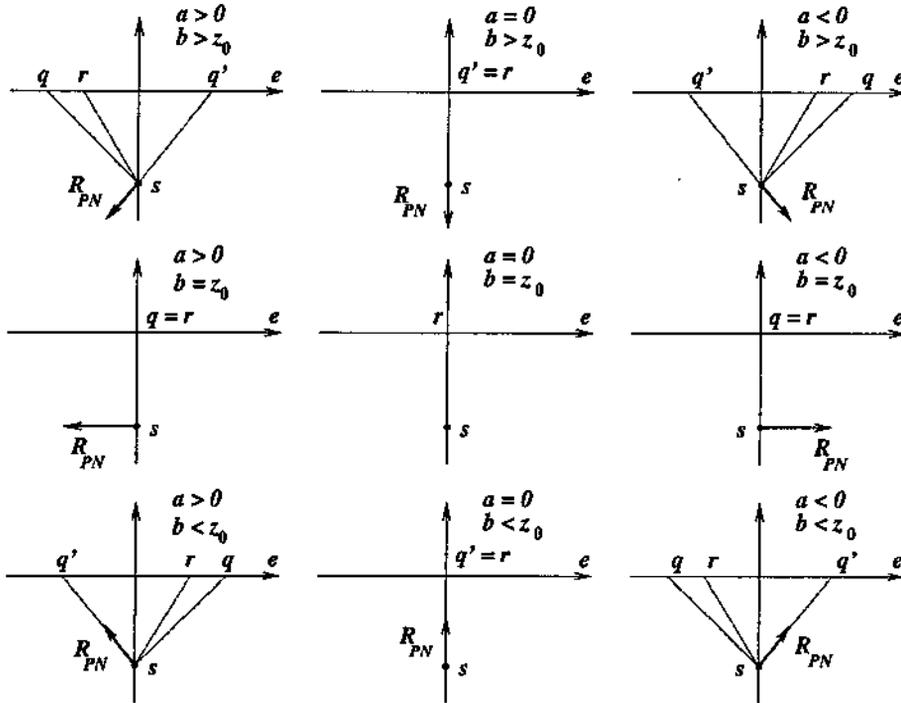


Figura A.2: Casos para a disposição da face f segundo os pontos q , q' e r .

Seja x^* um ponto sobre a aresta e . Seja ψ o ângulo $\angle rsq'$ com lados \overline{sr} e $\overline{sq'}$. Seja θ_x o ângulo de incidência do segmento $\overline{sx^*}$ em e . Observe que o ângulo θ_x varia em função de x , enquanto o ângulo ψ é constante. A tabela A.1 apresenta expressões para o ângulo β_x em função de ψ e θ_x .

Dessa forma, utilizando transformações trigonométricas obtemos expressões de $\cos \beta_x$ em função de θ_x e ψ . A tabela A.2 apresenta estas expressões e permite concluir que $\cos \beta_x$ assume a forma $c_1 \cos \theta_x + c_2 \sin \theta_x$, onde c_1 e c_2 são constantes.

Os valores de $\cos \psi$ e $\sin \psi$ são obtidos pelas leis dos senos e cossenos no triângulo $\Delta rsq'$. Por outro lado, vamos encontrar expressões para $\cos \theta_x$ e $\sin \theta_x$ em função de x . Observe que $\sin \left(\frac{\pi}{2} - \theta_x \right) = \cos \theta_x = \frac{|sr|}{|sx^*|}$ (veja Fig. A.3). Portanto, denotando por c_3 a distância constante $|sr|$ e por $d(x) = \sqrt{x^2 + y_0^2 + (z_0 - z)^2}$ a distância euclidiana entre s

	$a > 0$		$a = 0$	$a < 0$	
$b > z_0$	$x < r_x$	$\theta_x + \psi$	θ_x	$x < q'_x$	$\theta_x - \psi$
	$r_x < x < q'_x$	$-\theta_x + \psi$		$q'_x < x < r_x$	$-\theta_x + \psi$
	$x > q'_x$	$\theta_x - \psi$		$x > r_x$	$\theta_x + \psi$
$b = z_0$	$x < r_x$	$\theta_x + \frac{\pi}{2}$	β_x	$x < r_x$	$-\theta_x + \frac{\pi}{2}$
	$x > r_x$	$-\theta_x + \frac{\pi}{2}$		$x > r_x$	$\theta_x + \frac{\pi}{2}$
$b < z_0$	$x < q'_x$	$-\theta_x + \psi + \pi$	$-\theta_x + \pi$	$x < r_x$	$-\theta_x - \psi + \pi$
	$q'_x < x < r_x$	$\theta_x - \psi + \pi$		$r_x < x < q'_x$	$\theta_x - \psi + \pi$
	$x > r_x$	$-\theta_x - \psi + \pi$		$x > q'_x$	$-\theta_x + \psi + \pi$

Tabela A.1: Tabela de expressões de β_x em função de θ_x e ψ .

	$a > 0$		$a = 0$	$a < 0$	
$b > z_0$	$x < r_x$	$\cos(\theta_x + \psi)$	$\cos(\theta_x)$	$x < q'_x$	$\cos(\theta_x - \psi)$
	$r_x < x < q'_x$	$\cos(-\theta_x + \psi)$		$q'_x < x < r_x$	$\cos(-\theta_x + \psi)$
	$x > q'_x$	$\cos(\theta_x - \psi)$		$x > r_x$	$\cos(\theta_x + \psi)$
$b = z_0$	$x < r_x$	$-\text{sen } \theta_x$	β_x	$x < r_x$	$\text{sen } \theta_x$
	$x > r_x$	$\text{sen } \theta_x$		$x > r_x$	$-\text{sen } \theta_x$
$b < z_0$	$x < q'_x$	$-\cos(-\theta_x + \psi)$	$-\cos \theta_x$	$x < r_x$	$-\cos(\theta_x + \psi)$
	$q'_x < x < r_x$	$-\cos(\theta_x - \psi)$		$r_x < x < q'_x$	$-\cos(\theta_x - \psi)$
	$x > r_x$	$-\cos(\theta_x + \psi)$		$x > q'_x$	$-\cos(-\theta_x + \psi)$

Tabela A.2: Tabela de expressões de $\cos(\beta_x)$ em função de θ_x e ψ .

e x^* , temos $\cos \theta_x = \frac{c_3}{d(x)}$.

Semelhantemente, observe que $\cos\left(\frac{\pi}{2} - \theta_x\right) = \frac{rx^*}{|sx^*|}$. Assim,

$$\text{sen } \theta_x = \frac{\sqrt{\left(x - \frac{a(z_0 - b)}{1 + a^2}\right)^2 + \left(ax - \frac{a^2(z_0 - b)}{1 + a^2}\right)^2}}{d(x)},$$

colocando $(1 + a^2)$ em evidência e desradicalizando, temos

$$\text{sen } \theta_x = \frac{\sqrt{(1 + a^2)} \left(x - \frac{a(z_0 - b)}{1 + a^2}\right)}{d(x)}.$$

Agora, multiplicando o numerador e o denominador por $\sqrt{(1 + a^2)}$ e expandindo o numerador, temos

$$\text{sen } \theta_x = \frac{(1 + a^2)x + a(b - z_0)}{\sqrt{(1 + a^2)}d(x)} = \frac{d'(x)}{\sqrt{(1 + a^2)}}. \quad (\text{A.1})$$

Seja $b(x)$ uma função real que representa $\cos \beta_x$. Assumimos o caso onde $a > 0$, $b > z_0$, $x < r_x$. Dessa forma, $\cos \beta_x = \cos(\theta_x + \psi) = \cos \psi \cos \theta_x - \text{sen } \psi \text{sen } \theta_x$. Neste

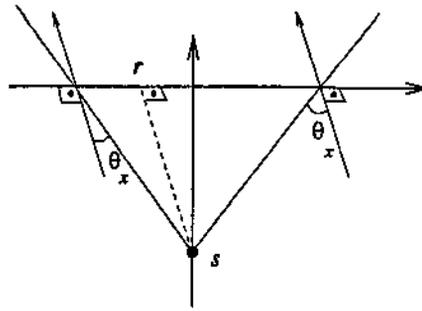


Figura A.3: Cosseno e seno do ângulo de incidência θ_x .

caso, $c_1 = \cos \psi$ e $c_2 = -\sin \psi$. Portanto,

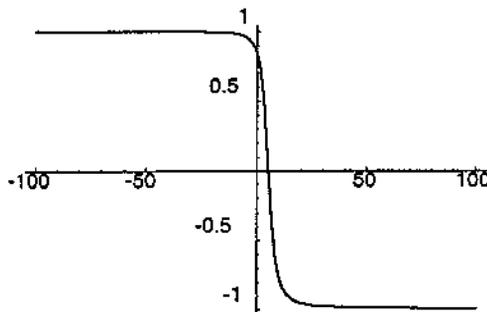
$$b(x) = \frac{c_1 c_3}{d(x)} + \frac{c_2 d'(x)}{\sqrt{1+a^2}}.$$

Contudo, pela lei dos cossenos no triângulo $\Delta q'sx^*$, podemos obter a seguinte função (veja Fig. A.4):

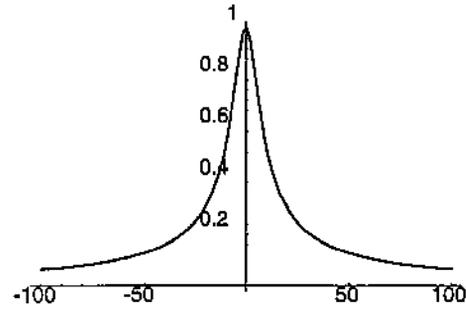
$$b(x) = \frac{d^2(x) + |sq'|^2 - |x^*q'|^2}{2d(x)|sq'|}.$$

Assim, derivando-se $b(x)$, encontramos

$$\begin{aligned} b'(x) &= \frac{(1+a^2)q'_x + a(b-z_0)}{|sq'|d(x)} - \frac{(d^2(x) + |sq'|^2 - |x^*q'|^2)[(1+a^2)x + a(b-z_0)]}{2|sq'|d^3(x)}, \\ &= \frac{(1+a^2)q'_x + a(b-z_0)}{|sq'|d(x)} - \frac{b(x)[(1+a^2)x + a(b-z_0)]}{d^2(x)}. \end{aligned}$$



(a) O $\cos \beta_x$.



(b) A derivada primeira de $\cos \beta_x$.

Figura A.4: Gráficos em função de x .

Seja $k(x) = (1 + a^2)x + a(b - z_0)$, temos $d'(x) = \frac{k(x)}{d(x)}$ e $d''(x) = \frac{1+a^2}{d(x)} - \frac{k^2(x)}{d^3(x)}$. Podemos, simplificar a expressão para $b'(x)$ da seguinte forma:

$$\begin{aligned} b'(x) &= \frac{(1+a^2)(q'_x - x) + (1+a^2)x + a(b - z_0)}{|sq'|d(x)} - \frac{b(x)k(x)}{d^2(x)}, \\ &= \frac{(1+a^2)(q'_x - x) + k(x)}{|sq'|d(x)} - \frac{b(x)d'(x)}{d(x)}. \end{aligned}$$

De forma semelhante, encontramos a seguinte expressão:

$$b''(x) = \frac{2b(x)d''(x)}{d^2(x)} - \frac{2d'(x)[(1+a^2)(q'_x - x) + d(x)d'(x)]}{|sq'|d^2(x)} - \frac{b(x)d''(x)}{d(x)} + \frac{d'^2(x) + d(x)d'''(x) - (1+a^2)}{|sq'|d(x)}.$$

Assim, chegamos à expressão algébrica do cosseno do ângulo entre as forças de atrito e resultante peso-normal. Além disso, encontramos e simplificamos as derivadas primeira e segunda desta expressão.

Apêndice B

Funções Convexas

Neste apêndice, apresentamos os conceitos básicos de funções convexas e demonstramos a convexidade da função distância FGAD.

Uma função $f : I \rightarrow \mathbf{R}$ é chamada *convexa* se a inequação

$$f\left(\frac{x+y}{2}\right) \leq \frac{f(x)+f(y)}{2} \quad (\text{B.1})$$

é satisfeita para todo $x, y \in I$, onde I é um intervalo da reta real [114]. Esta função é chamada *estritamente convexa* quando temos a desigualdade estrita. Geometricamente, a condição B.1 significa que o ponto médio do segmento entre dois pontos no gráfico da função f está sobre ou acima do ponto correspondente no gráfico (veja Fig. B.1). Na verdade, todo o segmento está acima do gráfico da função, ou seja, a inequação

$$f[\lambda x + (1 - \lambda)y] \leq \lambda f(x) + (1 - \lambda)f(y) \quad (\text{B.2})$$

é satisfeita para todo λ no intervalo aberto $(0, 1)$. Esta inequação é chamada *inequação de Jensen*. Se $-f$ é convexa, então dizemos que f é *côncava*.

Uma função convexa é contínua exceto nas extremidades do seu domínio, suas derivadas direita e esquerda existem e são não-decrescentes. Uma função convexa e finita em um intervalo fechado satisfaz a *condição de Lipschitz*, sendo limitada superior e inferiormente.

A classe de funções convexas pode ser caracterizada de várias formas. Para uma função diferenciável, a convexidade (convexidade estrita) implica uma derivada crescente (estritamente crescente). Portanto, quando f'' existe em (a, b) , então f é convexa se e somente se $f''(x) \geq 0$. Se $f''(x) > 0$ em (a, b) , então f é estritamente convexa neste intervalo.

As funções convexas são reconhecidas mais facilmente observando que são formadas por outras funções convexas e considerando algumas operações sobre as quais a classe de funções convexas permanece fechada [114]. Dessa forma, os teoremas seguintes garantem

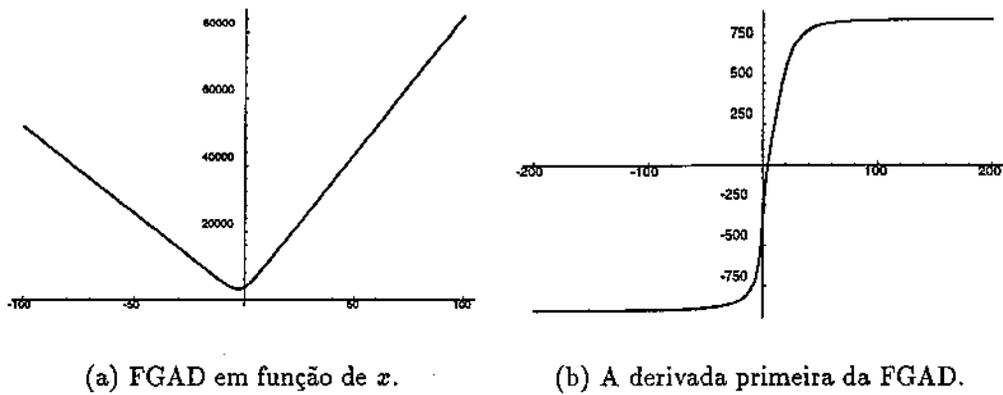


Figura B.1: A função distância FGAD.

que a soma de duas funções convexas é uma função convexa, dentre outras propriedades de funções convexas.

Teorema B.1 [114] *Se $f : I \rightarrow \mathbb{R}$ e $g : I \rightarrow \mathbb{R}$ são convexas e $\gamma \geq 0$, então $f(x) + g(x)$ e $\gamma f(x)$ são convexas em I .*

Teorema B.2 [114] *Seja $f : I \rightarrow \mathbb{R}$ e $g : J \rightarrow \mathbb{R}$ onde $\Im(f) \subset J$. Se f e g são convexas e g é crescente, então a função composta $g \circ f$ é convexa em I .*

Teorema B.3 [114] *Se $f : I \rightarrow \mathbb{R}$ e $g : I \rightarrow \mathbb{R}$ são não-negativas, decrescentes (crescentes) e convexas, então $h(x) = f(x)g(x)$ também possui estas três propriedades.*

A convexidade estrita de uma função é uma propriedade que implica em um único mínimo local e, portanto, global desta função. Assim, demonstramos a convexidade estrita da função distância FGAD para garantir a existência de um critério de otimalidade local, quando um caminho geodésico intercepta uma aresta.

Lema B.1 *A função distância FGAD do ponto $s(0, -y_0, z_0)$ no interior de uma face f ao ponto $x^*(x, 0, z)$ sobre uma aresta e da face f é estritamente convexa.*

Prova. A função distância FGAD pode ser escrita como

$$mg\sqrt{(\mu_f^2 \cos^2 \alpha_f + 2\mu_f \cos \alpha_f \sin \alpha_f b(x) + \sin^2 \alpha_f)(d^2(x))},$$

onde $b(x) = \cos \beta_x$ ¹ e $d(x)$ é a distância euclídeana entre s e x^* .

¹Detalhes sobre $\cos \beta_x$ são encontrados no apêndice A.

Dessa forma, seja $f(x) = (\mu_f^2 \cos^2 \alpha_f + 2\mu_f \cos \alpha_f \sin \alpha_f b(x) + \sin^2 \alpha_f) d^2(x)$ e $g(x) = \sqrt{x}$, então a função distância FGAD é estritamente convexa se e somente se f é estritamente convexa, pois g é estritamente convexa e crescente (Teorema B.2).

Assim, devemos mostrar que $f''(x) > 0$, ou seja,

$$2(1+a^2)(\mu_f^2 \cos^2 \alpha_f + 2\mu_f \cos \alpha_f \sin \alpha_f b(x) + \sin^2 \alpha_f) + 8\mu_f \cos \alpha_f \sin \alpha_f k(x)b'(x) + 2\mu_f \cos \alpha_f \sin \alpha_f d^2(x)b''(x) > 0,$$

onde

$$\begin{aligned} k(x) &= (1+a^2)x + a(b-z_0), \\ b'(x) &= \frac{(1+a^2)(q'_x-x)+k(x)}{|sq'|d(x)} - \frac{b(x)d'(x)}{d^2(x)}, \\ b''(x) &= \frac{2b(x)d^2(x)}{d^4(x)} - \frac{2d'(x)[(1+a^2)(q'_x-x)+k(x)]}{|sq'|d^3(x)} - \frac{b(x)d''(x)}{d^3(x)} + \frac{d^2(x)+d(x)d''(x)-(1+a^2)}{|sq'|d^3(x)}, \\ d'(x) &= \frac{k(x)}{d(x)}, \\ d''(x) &= \frac{1+a^2}{d(x)} - \frac{k^2(x)}{d^3(x)}. \end{aligned}$$

Portanto, expandindo $b'(x)$ e $b''(x)$, substituindo $d'(x)$ e $d''(x)$, e como

$$d^2(x) + d(x)d''(x) - (1+a^2) = 0,$$

a inequação $f''(x) > 0$ fica da seguinte forma:

$$\begin{aligned} &2(1+a^2)(\mu_f^2 \cos^2 \alpha_f + 2\mu_f \cos \alpha_f \sin \alpha_f b(x) + \sin^2 \alpha_f) + \\ &+ 8\mu_f \cos \alpha_f \sin \alpha_f k(x) \left(\frac{(1+a^2)(q'_x-x)+k(x)}{|sq'|d(x)} - \frac{b(x)k(x)}{d^2(x)} \right) + \\ &+ 2\mu_f \cos \alpha_f \sin \alpha_f d^2(x) \left(\frac{2b(x)k^2(x)}{d^4(x)} - \frac{2k(x)[(1+a^2)(q'_x-x)+k(x)]}{|sq'|d^3(x)} - \frac{b(x)[d^2(x)(1+a^2)-k^2(x)]}{d^4(x)} \right) > 0. \end{aligned}$$

Multiplicando $k(x)$ pela expressão de $b'(x)$, $d^2(x)$ pela expressão de $b''(x)$, e agrupando os termos semelhantes de $k(x)b'(x)$ e $d^2(x)b''(x)$, temos

$$\begin{aligned} &2(1+a^2)(\mu_f^2 \cos^2 \alpha_f + 2\mu_f \cos \alpha_f \sin \alpha_f b(x) + \sin^2 \alpha_f) + \\ &+ 4\mu_f \cos \alpha_f \sin \alpha_f \left(\frac{k(x)[(1+a^2)(q'_x-x)+k(x)]}{|sq'|d(x)} - \frac{b(x)k^2(x)}{d^2(x)} \right) - \\ &- 2\mu_f \cos \alpha_f \sin \alpha_f \left(\frac{b(x)[d^2(x)(1+a^2)-k^2(x)]}{d^2(x)} \right) > 0. \end{aligned}$$

Novamente, multiplicando os fatores $(4\mu_f \cos \alpha_f \sin \alpha_f)$ e $(-2\mu_f \cos \alpha_f \sin \alpha_f)$, e simplificando os termos semelhantes, obtemos

$$\begin{aligned} &2(1+a^2)(\mu_f^2 \cos^2 \alpha_f + 2\mu_f \cos \alpha_f \sin \alpha_f b(x) + \sin^2 \alpha_f) + \\ &+ 4\mu_f \cos \alpha_f \sin \alpha_f \left(\frac{k(x)[(1+a^2)(q'_x-x)+k(x)]}{|sq'|d(x)} \right) - \\ &- 2\mu_f \cos \alpha_f \sin \alpha_f \left(\frac{b(x)k^2(x)}{d^2(x)} \right) - 2\mu_f \cos \alpha_f \sin \alpha_f b(x)(1+a^2) > 0. \end{aligned}$$

Colocando $[-2\mu_f \cos \alpha_f \sin \alpha_f b(x)]$ em evidência, encontramos

$$\begin{aligned} &2(1+a^2)(\mu_f^2 \cos^2 \alpha_f + 2\mu_f \cos \alpha_f \sin \alpha_f b(x) + \sin^2 \alpha_f) + \\ &+ 4\mu_f \cos \alpha_f \sin \alpha_f \left(\frac{k(x)[(1+a^2)(q'_x-x)+k(x)]}{|sq'|d(x)} \right) - \\ &- 2\mu_f \cos \alpha_f \sin \alpha_f b(x) \left(\frac{k^2(x)}{d^2(x)} + (1+a^2) \right) > 0. \end{aligned}$$

Seja $c_4 = (1 + a^2)q'_x + a(b - z_0)$. Portanto, como $(1 + a^2)q'_x + a(b - z_0) = (1 + a^2)(q'_x - x) + k(x)$, temos

$$2(1 + a^2)(\mu_f^2 \cos^2 \alpha_f + 2\mu_f \cos \alpha_f \sin \alpha_f b(x) + \sin^2 \alpha_f) + 4\mu_f \cos \alpha_f \sin \alpha_f \left(\frac{c_4}{|sq'|} \frac{k(x)}{d(x)} \right) - 2\mu_f \cos \alpha_f \sin \alpha_f b(x) \left(\frac{k^2(x)}{d^2(x)} + (1 + a^2) \right) > 0.$$

Dividindo a expressão por 2 e colocando $(\mu_f \cos \alpha_f \sin \alpha_f)$ em evidência, obtemos

$$(1 + a^2)(\mu_f^2 \cos^2 \alpha_f + 2\mu_f \cos \alpha_f \sin \alpha_f b(x) + \sin^2 \alpha_f) + \mu_f \cos \alpha_f \sin \alpha_f \left[2 \frac{c_4}{|sq'|} \frac{k(x)}{d(x)} - b(x) \frac{k^2(x)}{d^2(x)} - b(x)(1 + a^2) \right] > 0.$$

Pela igualdade $d'(x) = \frac{k(x)}{d(x)}$, encontramos

$$(1 + a^2)(\mu_f^2 \cos^2 \alpha_f + 2\mu_f \cos \alpha_f \sin \alpha_f b(x) + \sin^2 \alpha_f) + \mu_f \cos \alpha_f \sin \alpha_f \left(2 \frac{c_4}{|sq'|} d'(x) - b(x) d'^2(x) - b(x)(1 + a^2) \right) > 0.$$

Observe que $\frac{c_4}{|sq'|} = \frac{(1+a^2)q'_x + a(b-z_0)}{\sqrt{q_x'^2 + y_0^2 + (aq'_x + b - z_0)^2}}$. Substituindo $q'_x = \frac{ay_0}{b-z_0}$ e simplificando, temos

$$\frac{c_4}{|sq'|} = \sqrt{\frac{a^2 y_0^2}{a^2 y_0^2 + (b - z_0)^2}} + a^2 \leq \sqrt{1 + a^2}.$$

Além disso, $d'(x)$ é uma função limitada inferior e superiormente, onde

$$\lim_{x \rightarrow \pm\infty} d'(x) = \pm\sqrt{1 + a^2}.$$

Dessa forma, como $f(x) \geq 0$ e $\mu_f \cos \alpha_f \sin \alpha_f \geq 0$, a função $f''(x)$ assume seu valor mínimo quando $\frac{c_4}{|sq'|} = \sqrt{1 + a^2}$ e $d'(x) = -\sqrt{1 + a^2}$. Assim, a inequação $f''(x) > 0$ fica da seguinte forma:

$$(1 + a^2)(\mu_f^2 \cos^2 \alpha_f + 2\mu_f \cos \alpha_f \sin \alpha_f b(x) + \sin^2 \alpha_f) + \mu_f \cos \alpha_f \sin \alpha_f (-2(1 + a^2) - 2(1 + a^2)b(x)) > 0.$$

Expandindo $f(x)$ e simplificando ou eliminando os termos semelhantes, temos

$$(1 + a^2)(\mu_f^2 \cos^2 \alpha_f + \sin^2 \alpha_f) - 2(1 + a^2)\mu_f \cos \alpha_f \sin \alpha_f > 0.$$

Colocando $(1 + a^2)$ em evidência, obtemos

$$(1 + a^2)(\mu_f^2 \cos^2 \alpha_f - 2\mu_f \cos \alpha_f \sin \alpha_f + \sin^2 \alpha_f) > 0.$$

Assim, chegamos ao quadrado perfeito $(\mu_f \cos \alpha_f - \sin \alpha_f)^2$ que é, claramente, não-negativo. Além disso, $\mu_f \cos \alpha_f - \sin \alpha_f = 0$ se e somente se $\mu_f = \tan \alpha_f$. Contudo, isso é impossível por suposição². Dessa forma, $(1 + a^2)(\mu_f \cos \alpha_f - \sin \alpha_f)^2 > 0$, ou seja, $f''(x) > 0$. Podemos concluir, então, que a função $f(x)$ é estritamente convexa. Consequentemente, a função distância FGAD também é estritamente convexa. \square

²Esta suposição é feita na seção 3.2.

Apêndice C

Critério de Otimalidade Local

Neste apêndice, encontramos as expressões relacionadas ao critério de otimalidade local. Portanto, simplificamos a derivada primeira da função distância FGAD e obtemos o ponto x_s na aresta e (mais próximo da origem s) através de uma equação em função de $\text{sen } \theta_s$.

C.1 A Derivada Primeira

A derivada primeira da função distância FGAD de s a x^* com relação ao parâmetro x tem a seguinte forma:

$$\frac{\mu_f \cos \alpha_f \text{sen } \alpha_f d(x) b'(x)}{\sqrt{\mu_f^2 \cos^2 \alpha_f + 2\mu_f \cos \alpha_f \text{sen } \alpha_f b(x) + \text{sen}^2 \alpha_f}} + \sqrt{\mu_f^2 \cos^2 \alpha_f + 2\mu_f \cos \alpha_f \text{sen } \alpha_f b(x) + \text{sen}^2 \alpha_f} d'(x),$$

onde μ_f é o coeficiente de atrito cinético na face f , α_f é o ângulo de inclinação da face f com o plano horizontal, $d(x)$ é a distância euclidiana de s a x^* e $b(x) = \cos \beta_x$.

Seja $c_5 = \mu_f \cos \alpha_f \text{sen } \alpha_f$ e $c_6 = \mu_f^2 \cos^2 \alpha_f + \text{sen}^2 \alpha_f$, então a expressão acima fica como segue:

$$\frac{c_5 d(x) b'(x)}{\sqrt{2c_5 b(x) + c_6}} + \sqrt{2c_5 b(x) + c_6} d'(x).$$

Simplificando em um único denominador, temos

$$\frac{c_5 d(x) b'(x) + (2c_5 b(x) + c_6) d'(x)}{\sqrt{2c_5 b(x) + c_6}}. \quad (\text{C.1})$$

Expandindo $b'(x)$, cancelando $d(x)$ e simplificando com $c_4 = (1 + a^2)q'_x + a(b - z_0)$, obtemos

$$\frac{\left(\frac{c_4}{|sq'|} - b(x) d'(x)\right) + (2c_5 b(x) + c_6) d'(x)}{\sqrt{2c_5 b(x) + c_6}}.$$

Multiplicando c_5 e simplificando os termos semelhantes, encontramos

$$\frac{\frac{c_5 c_4}{|s q'|} + c_5 b(x) d'(x) + c_6 d'(x)}{\sqrt{2c_5 b(x) + c_6}}$$

Colocando $d'(x)$ em evidência, temos

$$\frac{\frac{c_5 c_4}{|s q'|} + (c_5 b(x) + c_6) d'(x)}{\sqrt{2c_5 b(x) + c_6}}$$

Agora, como $d'(x) = \sqrt{1 + a^2} \sen \theta_x$, onde θ_x é o ângulo de incidência do segmento $\overline{s x^*}$ na aresta e , obtemos

$$\frac{\frac{c_5 c_4}{|s q'|} + \sqrt{1 + a^2} (c_5 b(x) + c_6) \sen \theta_x}{\sqrt{2c_5 b(x) + c_6}}$$

Seja $c_9 = \frac{c_5 c_4}{\sqrt{1 + a^2} |s q'|}$, então a expressão acima fica da seguinte forma:

$$\frac{\sqrt{1 + a^2} (c_9 + (c_5 b(x) + c_6) \sen \theta_x)}{\sqrt{2c_5 b(x) + c_6}} \quad (C.2)$$

C.2 O ponto x_s

O ponto de mínimo x_s da função distância FGAD de s a x^* é determinado pela derivada primeira desta função. Portanto, como $b(x) = \pm \cos \psi \cos \theta_s \pm \sen \psi \sen \theta_s$, a raiz da função C.2 em $\sen \theta_s$ é dada pela seguinte equação:

$$(c_9 + (\pm c_5 \cos \psi \cos \theta_s \pm c_5 \sen \psi \sen \theta_s + c_6) \sen \theta_s) = 0.$$

Isolando $\cos \theta_s$ para encontrarmos uma expressão em função de $\sen \theta_s$, temos

$$\cos \theta_s = \frac{-c_9 \pm c_5 \sen \psi \sen^2 \theta_s - c_6 \sen \theta_s}{\pm c_5 \cos \psi \sen \theta_s} = \sqrt{1 - \sen^2 \theta_s}.$$

Agora, desradicalizando, eliminando o denominador e simplificando os termos semelhantes, encontramos

$$c_9^2 + 2c_6 c_9 \sen \theta_s + (c_6^2 \pm c_5 c_9 \sen \psi - c_5^2 \cos^2 \psi) \sen^2 \theta_s \pm 2c_5 c_6 \sen \psi \sen^3 \theta_s + c_5^2 \sen^4 \theta_s = 0.$$

Portanto, sejam

$$\begin{aligned} c_{s0} &= c_9^2, \\ c_{s1} &= 2c_6 c_9, \\ c_{s2} &= c_6^2 \pm c_5 c_9 \sen \psi - c_5^2 \cos^2 \psi, \\ c_{s3} &= \pm 2c_5 c_6 \sen \psi, \\ c_{s4} &= c_5^2, \end{aligned}$$

temos

$$c_{s_0} + c_{s_1} \operatorname{sen} \theta_s + c_{s_2} \operatorname{sen}^2 \theta_s + c_{s_3} \operatorname{sen}^3 \theta_s + c_{s_4} \operatorname{sen}^4 \theta_s = 0. \quad (\text{C.3})$$

Dessa forma, a equação C.3 permite identificar o ponto mais próximo da origem s na aresta e . Para isso, encontramos as raízes de um polinômio de quarto grau em $\operatorname{sen} \theta_s$. Finalmente, resolvemos uma equação quadrática em x_s que segue da igualdade entre as raízes reais do polinômio e a expressão A.1. Analogamente, obtemos o ponto x_t na aresta e mais próximo ao ponto destino $t(x_1, y_1, z_1)$

Apêndice D

Tabelas de Símbolos

D.1 Letras Romanas Minúsculas

<i>a</i>	Arco de um grafo direcionado, coeficiente angular, extremidade de um intervalo de otimalidade
<i>b</i>	Coeficiente linear, extremidade de um intervalo de otimalidade
<i>c</i>	Ponto de acesso e fronteira de um intervalo de otimalidade, extremidade de um arco de um bissetor geodésico
<i>d</i>	Métrica sobre um conjunto de pontos, dimensão, comprimento de um caminho mínimo
<i>e</i>	Aresta de uma superfície poliédrica
<i>f</i>	Face de uma superfície poliédrica, região de uma subdivisão planar
<i>g</i>	Mapeamento linear, aceleração da gravidade
<i>h</i>	Número de obstáculos, menor comprimento de um segmento perpendicular a uma aresta passando pelo vértice oposto, bissetor geodésico, extremidade de um intervalo sobre a altura
<i>i</i>	Índice inteiro
<i>j</i>	Índice inteiro
<i>k</i>	Número de vértices em um caminho mínimo, tamanho de uma seqüência de arestas, ordem de generalização de um problema de proximidade
<i>l</i>	Multiplicador de Lagrange
<i>m</i>	Massa de um corpo, cardinalidade de um conjunto de sítios
<i>n</i>	Cardinalidade de uma instância de um problema
<i>o</i>	Obstáculo, ponto qualquer, ponto médio geodésico
<i>p</i>	Caminho, caminho mínimo, ou geodésico
<i>q</i>	Ponto ou sítio qualquer

<i>r</i>	Raiz de uma célula ou intervalo de otimalidade
<i>s</i>	Ponto origem de um caminho
<i>t</i>	Ponto destino de um caminho
<i>u</i>	Primeiro ponto de cruzamento, ponto qualquer
<i>v</i>	Vértice de um espaço livre, ponto de evento
<i>x</i>	Eixo cartesiano das abscissas
<i>y</i>	Eixo cartesiano das ordenadas
<i>w</i>	Último ponto de cruzamento, módulo mínimo que a força externa pode assumir
<i>z</i>	Eixo cartesiano das alturas

D.2 Letras Romanas Maiúsculas

<i>A</i>	Conjunto convexo, força de atrito, constante aditiva, conjunto de áreas
<i>B</i>	Bissetor geodésico
<i>C</i>	Corpo, célula de uma mapa de caminho mínimo, célula de uma subdivisão
<i>E</i>	Seqüência de arestas, número de arestas de um grafo, distância esférica
<i>F</i>	Espaço, espaço livre, funil, força externa, distância fluvial
<i>G</i>	Grafo direcionado ou valorado, complexidade de uma rotina numérica
<i>H</i>	Plano definido por uma face
<i>I</i>	Intervalo de otimalidade
<i>J</i>	Célula de um particionamento em janela
<i>K</i>	Complexidade de uma rotina
<i>L</i>	Complexidade binária de um problema de caminho mínimo
<i>M</i>	Reta mediana, constante multiplicativa
<i>N</i>	Maior coordenada inteira de qualquer vértice, força normal
<i>O</i>	Conjunto de vetores de comprimento unitário, conjunto de obstáculos
<i>P</i>	Força peso
<i>Q</i>	Conjunto, subconjunto e seqüência ordenada de pontos ou sítios
<i>R</i>	Distância retilínea, força resultante, reta de varredura, região Voronoi
<i>S</i>	Subdivisão planar conexa, superfície poliédrica
<i>T</i>	Árvore de decomposição geodésica, árvore

U	Conjunto de segmentos
V	Vetor, diagrama de Voronoi, distância valorada
W	Maior custo associado a qualquer face, cunha biplanar, módulo máximo que a força externa pode assumir

D.3 Letras Gregas Minúsculas

α	Ângulo de inclinação com o plano horizontal
β	Ângulo entre a força de atrito e a resultante peso-normal
γ	Ponto de interseção entre caminhos ou entre bissetores geodésicos
δ	Real positivo
ε	Fator de erro relativo
η	Ângulo diedral
θ	Ângulo de incidência, refração ou crítico
λ	Quantidade escalar
μ	Custo unitário associado a uma região, coeficiente de atrito cinético
τ	Ancestral comum mais baixo
ϕ	Ângulo de incidência que determina um traçado de caminho geodésico
ψ	Ângulo constante $\angle rsq'$
ω	Ângulo que especifica a orientação de um corpo

D.4 Letras Gregas Maiúsculas

Γ	Tipo de um evento
Δ	Métrica de caminho mínimo
Λ	Diferença de funis
Υ	Topologia de um evento
Φ	Função contínua do intervalo real $[0, 1]$ em pontos de um espaço

D.5 Notações Gerais

$d_F(s, t)$	Comprimento de um caminho mínimo no espaço F do ponto s ao ponto t
$ O $	Cardinalidade do conjunto O
$C(V)$	Configuração do corpo C especificada pelo vetor V
$C(V, \omega)$	Configuração do corpo C especificada pelo vetor V e pelo ângulo ω
$\mathcal{EC}(C)$	Espaço de configuração de um corpo C

$\mathcal{E}C_P(C, O)$	Espaço proibido relativo a um corpo C e um conjunto de obstáculos O
$\mathcal{E}C_L(C, O)$	Espaço livre relativo a um corpo C e um conjunto de obstáculos O
$d_R(q, r)$	Métrica retilínea
L_p	Métrica de Minkowski
\ominus	Diferença vetorial
\oplus	Soma vetorial
$\ \cdot\ _A$	Função distância de Minkowski
T_s	Árvore de caminho mínimo relativa a um vértice s
$\tau_s(v_1, v_2)$	Ancestral comum mais baixo de v_1 e v_2 em T_s
M_s	Mapa de caminho mínimo relativo a um ponto s
C_r	Célula associada a uma raiz r
$B(r, r')$	Bissetor entre as raízes r e r'
M_v	Componente principal de um vértice v
$F_{v_1, v_2}(s)$	Funil de s relativo aos vértices v_1 e v_2
$\alpha(n)$	Inverso da função de Ackermann
$S(\omega)$	Hemisfério meridional
$N(\omega)$	Hemisfério setentrional
$p_f(q)$	Caminho localmente f -livre da origem para um ponto q
\bar{I}	Intervalo de otimalidade predecessor

Bibliografia

- [1] J. Adegest, M. Overmars, e J. Snoeyink. Minimum-link C -oriented paths: Single-source queries. *Internat. Journal Comput. Geom. Appl.*, 4(1):39–51, 1994.
- [2] A. Aggarwal, H. Edelsbrunner, P. Raghavan, e P. Tiwari. Optimal time bounds for some proximity problems in the plane. *Information Processing Letters*, 42(1):55–60, 1992.
- [3] B. Aronov. On the geodesic Voronoi diagram of point sites in a simple polygon. *Algorithmica*, 4:109–140, 1989.
- [4] Ta. Asano, Te. Asano, L. J. Guibas, J. Hershberger, e H. Imai. Visibility-polygon search and Euclidean shortest paths. Em *Proc. 26th Annual IEEE Sympos. Found. Comput. Sci.*, pág. 155–164, 1985.
- [5] Te. Asano. An efficient algorithm for finding the visibility polygon for a polygonal region with holes. *Trans. IECE Japan*, E68(9):560–562, 1985.
- [6] F. Aurenhammer e H. Edelsbrunner. An optimal algorithm for constructing the weighted Voronoi diagram in the plane. *Pattern Recogn.*, 17:251–257, 1984.
- [7] D. Avis e B. K. Bhattacharya. Algorithms for computing d -dimensional Voronoi diagrams and their duals. Em F. P. Preparata, editor, *Computational Geometry*, volume 1 of *Advances in Computing Research*, pág. 159–180. JAI Press, London, England, 1983.
- [8] C. Bajaj. The algebraic complexity of shortest paths in polyhedral spaces. Em *Proc. 23rd Allerton Conf. Commun. Control Comput.*, pág. 510–517, 1985.
- [9] R. Barrera e J. Vázquez-Gómez. A shortest path algorithm for hierarchical terrain models. Em *Proc. Auto-Carto 9*, pág. 156–263, 1989.
- [10] F. Bartling e K. Hinrichs. A plane-sweep algorithm for finding a closest pair among convex planar objects. Em *Proc. 9th Sympos. Theoret. Aspects Comput. Sci.*, volume 577 de *Lecture Notes in Computer Science*, pág. 221–232. Springer-Verlag, 1992.

- [11] B. G. Baumgart. A polyhedron representation for computer vision. Em *Proc. AFIPS Natl. Comput. Conf.*, volume 44, pág. 589–596, 1975.
- [12] J. L. Bentley. Multidimensional divide-and-conquer. *Commun. ACM*, 23(4):214–229, 1980.
- [13] J. L. Bentley e M. I. Shamos. Divide-and-conquer in multidimensional space. Em *Proc. 8th Annual ACM Sympos. Theory Comput.*, pág. 220–230, 1976.
- [14] A. Blum, P. Chalasani, e S. Vempala. A constant-factor approximation for the k -MST problem in the plane. Em *Proc. 27th Annual ACM Sympos. Theory Comput. (STOC 95)*, pág. 294–302, 1995.
- [15] K. Q. Brown. Voronoi diagrams from convex hulls. *Information Processing Letters*, 9:223–228, 1979.
- [16] J. Canny. A new algebraic method for robot motion planning and real geometry. Em *Proc. 28th Annual IEEE Sympos. Found. Comput. Sci.*, pág. 39–48, 1987.
- [17] J. Canny e J. H. Reif. New lower bound techniques for robot motion planning problems. Em *Proc. 28th Annual IEEE Sympos. Found. Comput. Sci.*, pág. 49–60, 1987.
- [18] J. Cassels. *An Introduction to the Geometry of Numbers*. Springer-Verlag, 1959.
- [19] R.-C. Chang e R. C. T. Lee. The average-case performance analysis of a closest-pair problem. *Internat. Journal Comput. Math.*, 16(2):125–130, 1984.
- [20] B. Chazelle. An algorithm for segment-dragging and its implementation. *Algorithmica*, 3:205–221, 1988.
- [21] B. Chazelle. A functional approach to data structures and its use in multidimensional searching. *SIAM Journal on Computing*, 17:427–462, 1988.
- [22] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6:485–524, 1991.
- [23] B. Chazelle e H. Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. Em *Proc. 29th Annual IEEE Sympos. Found. Comput. Sci.*, pág. 590–600, 1988.
- [24] B. Chazelle, H. Edelsbrunner, M. Grigni, L. Guibas, J. Hershberger, M. Sharir, e J. Snoeyink. Ray shooting in polygons using geodesic triangulations. *Algorithmica*, 12:54–68, 1994.

- [25] D. Z. Chen, K. S. K., e H.-Y. T. Tu. Shortest path queries among weighted obstacles in the rectilinear plane. Em *Proc. 11th Annual ACM Sympos. Comput. Geom.*, pág. 370–379, 1995.
- [26] J. Chen e Y. Han. Shortest paths on a polyhedron. Em *Proc. 6th Annual ACM Sympos. Comput. Geom.*, pág. 360–369, 1990.
- [27] D. Cheriton e R. E. Tarjan. Finding minimum spanning trees. *SIAM Journal on Computing*, 5:724–742, 1976.
- [28] L. P. Chew. Planning the shortest path for a disc in $O(n^2 \log n)$ time. Em *Proc. 1st Annual ACM Sympos. Comput. Geom.*, pág. 214–220, 1985.
- [29] J. Choi, J. Sellen, e C. K. Yap. Approximate Euclidean shortest path in 3-space. Em *Proc. 10th Annual ACM Sympos. Comput. Geom.*, pág. 41–48, 1994.
- [30] K. L. Clarkson. Approximation algorithms for shortest path motion planning. Em *Proc. 19th Annual ACM Sympos. Theory Comput.*, pág. 56–65, 1987.
- [31] K. L. Clarkson. New applications of random sampling in computational geometry. *Discrete Comput. Geom.*, 2:195–222, 1987.
- [32] K. L. Clarkson. Approximation algorithms for planar traveling salesman tours and minimum-length triangulations. Em *Proc. 2nd ACM-SIAM Sympos. Discrete Algorithms (SODA '91)*, pág. 17–23, 1991.
- [33] K. L. Clarkson, S. Kapoor, e P. M. Vaidya. Rectilinear shortest paths through polygonal obstacles in $O(n(\log n)^2)$ time. Em *Proc. 3rd Annual ACM Sympos. Comput. Geom.*, pág. 251–257, 1987.
- [34] R. Cole. Slowing down sorting networks to obtain faster sorting algorithms. *Journal ACM*, 34:200–208, 1987.
- [35] T. H. Cormen, C. E. Leiserson, e R. L. Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge, Mass., 1990.
- [36] G. Das e G. Narasimhan. Geometric searching and link distances. Em *Proc. 2nd Workshop Algorithms Data Struct.*, volume 519 of *Lecture Notes in Computer Science*, pág. 261–272. Springer-Verlag, 1991.
- [37] M. de Berg. On rectilinear link distance. *Comput. Geom. Theory Appl.*, 1(1):13–34, July 1991.

- [38] M. de Berg e M. van Kreveld. Trekking in the alps without freezing or getting tired. Em *1st Annual European Sympos. on Algorithms (ESA '93)*, volume 726 de *Lecture Notes in Computer Science*, pág. 121–132. Springer-Verlag, 1993.
- [39] M. de Berg, M. van Kreveld, B. J. Nilsson, e M. H. Overmars. Finding shortest paths in the presence of orthogonal obstacles using a combined L_1 and link metric. Em *Proc. 2nd Scand. Workshop Algorithm Theory*, volume 447 of *Lecture Notes in Computer Science*, pág. 213–224. Springer-Verlag, 1990.
- [40] P. J. de Rezende, D. T. Lee, e Y. F. Wu. Rectilinear shortest paths in the presence of rectangular barriers. *Discrete Comput. Geom.*, 4:41–53, 1989.
- [41] F. Dehne e R. Klein. An optimal algorithm for computing the Voronoi diagram on a cone. Report SCS-TR-122, School of Comput. Sci., Carleton Univ., 1987.
- [42] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [43] M. P. do Carmo. *Elementos de Geometria Diferencial*. Ao Livro Tecnico S.A., Rio de Janeiro, 1971.
- [44] R. L. Drysdale, III e D. T. Lee. Generalized Voronoi diagrams in the plane. Em *Proc. 16th Allerton Conf. Commun. Control Comput.*, pág. 833–842, 1978.
- [45] H. Edelsbrunner, L. J. Guibas, e J. Stolfi. Optimal point location in a monotone subdivision. *SIAM Journal on Computing*, 15:317–340, 1986.
- [46] M. Formann. Weighted closest pairs. Em *Proc. 10th Sympos. Theoret. Aspects Comput. Sci.*, volume 665 de *Lecture Notes in Computer Science*, pág. 270–281. Springer-Verlag, 1993.
- [47] S. J. Fortune. A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2:153–174, 1987.
- [48] M. Gardner. *The 2nd Scientific American Book of Mathematical Puzzles and Diversions*. Simon e Schuster, New York, 1961.
- [49] M. R. Garey e D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, NY, 1979.
- [50] L. Gewali, A. Meng, J. S. B. Mitchell, e S. Ntafos. Path planning in $0/1/\infty$ weighted regions with applications. *ORSA Journal Comput.*, 2(3):253–272, Summer 1990.

- [51] S. K. Ghosh. Computing visibility polygon from a convex set and related problems. *Journal Algorithms*, 12:75–95, 1991.
- [52] M. T. Goodrich e R. Tamassia. Dynamic ray shooting and shortest paths via balanced geodesic triangulations. Em *Proc. 9th Annual ACM Sympos. Comput. Geom.*, pág. 318–327, 1993.
- [53] S. Guha e I. Suzuki. Proximity problems for points on a rectilinear plane with rectangular obstacles. *Algorithmica*, 17:281–307, 1997.
- [54] L. J. Guibas e J. Hershberger. Optimal shortest path queries in a simple polygon. *Journal Comput. Syst. Sci.*, 39:126–152, 1989.
- [55] L. J. Guibas, J. Hershberger, D. Leven, M. Sharir, e R. E. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2:209–233, 1987.
- [56] L. J. Guibas, J. E. Hershberger, J. S. B. Mitchell, e J. S. Snoeyink. Approximating polygons and subdivisions with minimum link paths. *Internat. Journal Comput. Geom. Appl.*, 3(4):383–415, December 1993.
- [57] L. J. Guibas e J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Trans. Graph.*, 4:74–123, 1985.
- [58] J. Hershberger. A new data structure for shortest path queries in a simple polygon. *Information Processing Letters*, 38:231–235, 1991.
- [59] J. Hershberger e L. J. Guibas. An $O(n^2)$ shortest path algorithm for a non-rotating convex body. *Journal Algorithms*, 9:18–46, 1988.
- [60] J. Hershberger e J. Snoeyink. Computing minimum length paths of a given homotopy class. Em *Proc. 2nd Workshop Algorithms Data Struct.*, volume 519 of *Lecture Notes in Computer Science*, pág. 331–342. Springer-Verlag, 1991.
- [61] J. Hershberger e S. Suri. Efficient computation of Euclidean shortest paths in the plane. Em *Proc. 34th Annual IEEE Sympos. Found. Comput. Sci. (FOCS 93)*, pág. 508–517, 1993.
- [62] J. Hershberger e S. Suri. Practical methods for approximating shortest paths on a convex polytope in \mathbb{R}^3 . Em *Proc. 6th ACM-SIAM Sympos. Discrete Algorithms (SODA '95)*, pág. 447–456, 1995.
- [63] K. Hinrichs, J. Nievergelt, e P. Schorn. Plane-sweep solves the closest pair problem elegantly. *Information Processing Letters*, 26:255–261, 1988.

- [64] K. Hinrichs, J. Nievergelt, e P. Schorn. A sweep algorithm for the all-nearest-neighbours problem. Em *Computational Geometry and its Applications*, volume 333 of *Lecture Notes in Computer Science*, pág. 43–54. Springer-Verlag, 1988.
- [65] F. K. Hwang. An $O(n \log n)$ algorithm for rectilinear minimal spanning tree. *Journal ACM*, 26:177–182, 1979.
- [66] Y.-H. Hwang, R.-C. Chang, e H.-Y. Tu. Finding all shortest path edge sequences on a convex polyhedron. Em *Proc. 1st Workshop Algorithms Data Struct.*, volume 382 of *Lecture Notes in Computer Science*, pág. 251–266. Springer-Verlag, 1989.
- [67] H. Imai e Ta. Asano. Efficient algorithms for geometric graph search problems. *SIAM Journal on Computing*, 15:478–494, 1986.
- [68] H. Imai e Ta. Asano. Dynamic orthogonal segment intersection search. *Journal Algorithms*, 8:1–18, 1987.
- [69] H. Imai e M. Iri. Polygonal approximations of a curve-formulations and algorithms. Em G. T. Toussaint, editor, *Computational Morphology*, pág. 71–86. North-Holland, Amsterdam, Netherlands, 1988.
- [70] S. T. Jones. Solving problems involving variable terrain. Part 1: A general algorithm. *Byte*, 5(2):58–68, February 1980.
- [71] D. M. Keirse e J. S. B. Mitchell. Planning strategic paths through variable terrain data. Em *Proc. SPIE Applications of Artificial Intelligence*, volume 485, pág. 172–179, 1984.
- [72] D. G. Kirkpatrick. Optimal search in planar subdivisions. *SIAM Journal on Computing*, 12:28–35, 1983.
- [73] R. Klein. Abstract Voronoi diagrams and their applications. Em *Computational Geometry and its Applications*, volume 333 of *Lecture Notes in Computer Science*, pág. 148–157. Springer-Verlag, 1988.
- [74] R. Klein. *Concrete and Abstract Voronoi Diagrams*, volume 400 de *Lecture Notes in Computer Science*. Springer-Verlag, 1989.
- [75] J. B. Kruskal, Jr. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Amer. Math. Soc.*, 7:48–50, 1956.
- [76] H. W. Kuhn e A. W. Tucker. *Linear inequalities and related systems*. Princeton University Press, Princeton, 1956.

- [77] R. C. Larson e V. O. K. Li. Finding minimum rectilinear distance paths in the presence of barriers. *Networks*, 11:285–304, 1981.
- [78] D. T. Lee. On k -nearest neighbor Voronoi diagrams in the plane. *IEEE Trans. Comput.*, C-31:478–487, 1982.
- [79] D. T. Lee e Y. T. Ching. The power of geometric duality revisited. *Information Processing Letters*, 21:117–122, 1985.
- [80] D. T. Lee e R. L. Drysdale, III. Generalization of Voronoi diagrams in the plane. *SIAM Journal on Computing*, 10:73–87, 1981.
- [81] D. T. Lee e F. P. Preparata. The all nearest-neighbor problem for convex polygons. *Information Processing Letters*, 7:189–192, 1978.
- [82] D. T. Lee e F. P. Preparata. Euclidean shortest paths in the presence of rectilinear barriers. *Networks*, 14:393–410, 1984.
- [83] D. T. Lee e C. K. Wong. Voronoi diagrams in L_1 (L_∞) metrics with 2-dimensional storage applications. *SIAM Journal on Computing*, 9:200–211, 1980.
- [84] T. Lozano-Pérez e M. A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM*, 22:560–570, 1979.
- [85] D. G. Lvenberger. *Linear and Nonlinear Programming*. Addison-Wesley Publishing Company, 1937.
- [86] M. Mäntylä. *An Introduction to Solid Modeling*. Computer Science Press, Rockville, MD, 1988.
- [87] K. M. McDonald e J. G. Peters. Smallest paths in simple rectilinear polygons. *IEEETCAD: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11, 1992.
- [88] N. Megiddo. Linear-time algorithms for linear programming in \mathbb{R}^3 and related problems. *SIAM Journal on Computing*, 12:759–776, 1983.
- [89] R. E. Miles. Random points, sets and tessellations on the surface of a sphere. *Sankhyā: The Indian Journal of Statistics: Series A*, 33:145–174, 1971.
- [90] J. S. B. Mitchell. *Planning shortest paths*. Ph.D. thesis, Stanford Univ., Stanford, CA, 1986.

- [91] J. S. B. Mitchell. Shortest paths among obstacles, zero-cost regions, and "roads". Technical Report 764, School Oper. Res. Indust. Engrg., Cornell Univ., Ithaca, NY, 1987.
- [92] J. S. B. Mitchell. An optimal algorithm for shortest rectilinear paths among obstacles. Em *Abstracts 1st Canad. Conf. Comput. Geom.*, page 22, 1989.
- [93] J. S. B. Mitchell. L_1 shortest paths among polygonal obstacles in the plane. *Algorithmica*, 8:55-88, 1992.
- [94] J. S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple new method for the geometric k -MST problem. Em *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms (SODA '96)*, pág. 402-408, 1996.
- [95] J. S. B. Mitchell, D. M. Mount, e C. H. Papadimitriou. The discrete geodesic problem. *SIAM Journal on Computing*, 16:647-668, 1987.
- [96] J. S. B. Mitchell e C. H. Papadimitriou. Planning shortest paths. Em *SIAM Conf. Geometric Modelling & Robotics*, pág. 15-19, Albany, NY, 1985.
- [97] J. S. B. Mitchell e C. H. Papadimitriou. The weighted region problem: finding shortest paths through a weighted planar subdivision. *Journal ACM*, 38:18-73, 1991.
- [98] J. S. B. Mitchell, G. Rote, e G. Woeginger. Minimum-link paths among obstacles in the plane. *Algorithmica*, 8:431-459, 1992.
- [99] J. S. B. Mitchell e S. Suri. Separation and approximation of polyhedral objects. volume 5, pág. 95-114, 1995.
- [100] D. M. Mount. On finding shortest paths on convex polyhedra. Technical Report 1495, Department of Computer Science, University of Maryland, 1985.
- [101] D. M. Mount. Voronoi diagrams on the surface of a polyhedron. Technical Report 1496, Department of Computer Science, University of Maryland, 1985.
- [102] T. Munzner, S. Levy, e M. Philips. Geomview: A system for geometric visualization. Em *Proc. 11th Annual ACM Sympos. Comput. Geom.*, pág. C12-C13, 1995.
- [103] Y-S. Myung, C.-H. Lee, e D-W Tcha. On the generalized minimum spanning tree problem. *Networks*, 26:231-241, 1995.
- [104] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, 1994.

- [105] C. H. Papadimitriou. An algorithm for shortest-path motion in three dimensions. *Information Processing Letters*, 20:259–263, 1985.
- [106] C. H. Papadimitriou e E. B. Silverberg. Optimal piecewise linear motion of an object among obstacles. *Algorithmica*, 2:523–539, 1987.
- [107] T. K. Peucker, R. J. Fowler, J. J. Little, e D. M. Mark. The triangulated irregular network. Em *Proc. DTM Symp. Am. Soc. of Photogrammetry – Am. Congress on Survey and Mapping*, pág. 24–31, 1978.
- [108] F. P. Preparata. A new approach to planar point location. *SIAM Journal on Computing*, 10:473–482, 1981.
- [109] F. P. Preparata e M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, NY, 1985.
- [110] R. C. Prim. Shortest connection networks and some generalizations. *Bell Syst. Tech. Journal*, 36:1389–1401, 1957.
- [111] F. K. H. Quek, R. F. Franklin, e F. Pont. A decision system for autonomous robot navigation over rough terrain. Em *Proc. SPIE Applications of Artificial Intelligence*, 1985.
- [112] R. Ravi, R. Sundaram, M. V. Marathe, D. J. Rosenkrantz, e S. S. Ravi. Spanning trees short and small. Em *Proc. 5th ACM-SIAM Sympos. Discrete Algorithms (SODA '94)*, pág. 546–555, 1994.
- [113] J. H. Reif e J. A. Storer. 3-dimensional shortest paths in the presence of polyhedral obstacles. Em *Proc. 13th Internat. Sympos. Math. Found. Comput. Sci.*, volume 3:24 de *Lecture Notes in Computer Science*, pág. 85–92. Springer-Verlag, 1988.
- [114] A. W. Roberts e D. E. Varberg. *Convex Functions*. Academic Press, Inc., 1973.
- [115] H. Rohnert. Shortest paths in the plane with convex polygonal obstacles. *Information Processing Letters*, 23:71–76, 1986.
- [116] J.-R. Sack. *Rectilinear computational geometry*. Ph.D. thesis, School Comput. Sci., Carleton Univ., Ottawa, ON, 1984. Report SCS-TR-54.
- [117] M. I. Shamos. *Computational Geometry*. Ph.D. thesis, Dept. Comput. Sci., Yale Univ., New Haven, CT, 1978.
- [118] M. I. Shamos e D. Hoey. Closest-point problems. Em *Proc. 16th Annual IEEE Sympos. Found. Comput. Sci.*, pág. 151–162, 1975.

- [119] M. Sharir. Intersection and closest-pair problems for a set of planar discs. *SIAM Journal on Computing*, 14:448–468, 1985.
- [120] M. Sharir e A. Schorr. On shortest paths in polyhedral spaces. *SIAM Journal on Computing*, 15:193–215, 1986.
- [121] K. Sugihara. Voronoi diagrams in a river. *Internat. Journal Comput. Geom. Appl.*, 2(1):29–48, 1992.
- [122] S. Suri. A linear time algorithm for minimum link paths inside a simple polygon. *Computer Vision, Graphics and Image Processing*, 35:99–110, 1986.
- [123] M. Tompa. An optimal solution to a wire-routing problem. *Journal Comput. Syst. Sci.*, 23:127–150, 1981.
- [124] P. Vaidya. An optimal algorithm for the all-nearest-neighbors problem. Em *Proc. 27th IEEE Sympos. Foundations of Computer Science*, pág. 117–122, 1986.
- [125] M. van Kreveld. On quality paths on polyhedral terrains. Em *IGIS'94: Geographic Information Systems*, volume 884 de *Lecture Notes in Computer Science*, pág. 113–122. Springer-Verlag, 1994.
- [126] G. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques, deuxième memoire, recherches sur les paralleloèdres primitifs. *Journal fur die Reine und Angewandte Methematik*, 134:198–287, 1908.
- [127] E. Welzl. Constructing the visibility graph for n line segments in $O(n^2)$ time. *Information Processing Letters*, 20:167–171, 1985.
- [128] P. Widmayer. Network design issues in VLSI. Manuscript, Institut für Informatik, University Freiburg, 1989.
- [129] P. Widmayer, Y. F. Wu, e C. K. Wong. On some distance problems in fixed orientations. *SIAM Journal on Computing*, 16(4):728–746, 1987.
- [130] Y. F. Wu, P. Widmayer, M. D. F. Schlag, e C. K. Wong. Rectilinear shortest paths and minimum spanning trees in the presence of rectilinear obstacles. *IEEE Trans. Comput.*, 36:321–331, 1987.
- [131] C. K. Yap. An $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments. *Discrete Comput. Geom.*, 2:365–393, 1987.

Índice

- xy*-aproximação, 28
- abordagem
 - de lugar geométrico, 73
 - divisão-e-conquista, 81
 - lugar geométrico, 86
 - varredura linear, 20
- ancestral comum mais baixo, 13
- ângulo
 - β , 114
 - convexo, 24
 - crítico, 41
 - de incidência, 38, 40, 41
 - de refração, 38, 41
 - diedral, 29
- aproximativo
 - algoritmo, 25
 - esquema, 25
- arestas, 9
 - inclinação, 10
 - seqüência de, 3
 - raiz, 3
 - simples, 3
- árvore, 15
 - de caminhos mínimos, 13
 - de decomposição geodésica, 17
 - geradora mínima, 78, 80, 97
 - seqüência, 30
- atrato, 33
- bissetor geodésico, 13, 92, 96
- bola
 - aberta, 4
 - fechada, 4
- busca paramétrica, 74
- câmara, 100
- caminho, 3
 - $(1 + \epsilon)$ -ótimo, 4
 - O -orientado, 4
 - z -caminho, 80
 - equivalente, 4
 - geodésico, 4
 - criticamente refletido, 41
 - usa criticamente, 41
 - linear por partes, 35
 - localmente f -livre, 45
 - mínimo, 4, 34
 - caracterização, 32
 - de trabalho, 33
 - localmente f -livre, 45
 - perturbação, 4
 - retilíneo, 4
 - simples, 23
- canal de acesso, 48
- caracterização de caminhos mínimos, 32
- cauda, 17
- componente principal, 14
- condição de Lipschitz, 119
- configuração do corpo, 7
- critério de otimalidade local, 26
- curva
 - de nível, 9
 - parametrizada, 35

- simplificação, 11
- desdobramento, 22
- diagrama de Voronoi, 81
 - abstrato, 82
 - aditivo, 83
 - áreas, 85
 - aresta de Voronoi, 81
 - composto, 83
 - de caminhos mínimos, 87
 - de ordem k , 86
 - esférico, 84
 - fluvial, 82
 - multiplicativo, 83
 - região de Voronoi, 81
 - retilíneo, 82
 - segmentos, 85
 - tridimensional, 84
 - valorado, 83
 - vértice de Voronoi, 81
- envoltória convexa, 3, 81
- ε , 4, 59
- espaço
 - de configuração, 7
 - de trabalho, 7
 - livre, 8
 - métrico, 10
 - proibido, 8
- eventos discretos, 31
- faces, 9
 - aresta-adjacentes, 3
 - inclinação, 10
 - negativas, 28
 - positivas, 28
 - seqüência de, 3
- fator de erro relativo, 4, 59
- FGAD, 34
- fila de prioridades, 51, 104
- chave, 104
- força
 - soma vetorial, 33
- frente de onda, 31
- função
 - côncava, 119
 - convexa, 119
 - de Ackermann, 21
 - distância, 10, 33
 - convexa de Minkowski, 11
 - geodésica atritada direcional, 34
 - estritamente convexa, 119
- funil, 16
 - ápice, 16
 - diferença, 16
 - par aberto, 16
 - par fechado, 16
- geodésico
 - árvore de decomposição, 17
 - bissetor, 13, 92, 96
 - caminho, 4
 - métrica, 4, 87
 - ponto médio, 96
 - triangulação, 17
 - triângulo, 17
 - vizinho mais próximo, 96
- geometria computacional, 1
- Geomview, 100
- grade retilínea, 12
- grafo
 - denso, 78
 - grade, 26
 - interseção, 21
 - região, 26
 - visibilidade, 12
- hemisfério
 - meridional, 29

- setentrional, 29
- heurística, 25
- inequação de Jensen, 119
- intervalo de otimalidade, 31, 47
 - candidato, 51
 - dominância, 57
- isometria, 10
- lugar geométrico de proximidade, 73
- mapa
 - v_z -mapa, 14
 - de caminhos mínimos, 13
 - célula, 13
 - de nível de altura, 14, 25
- matriz de elevação, 9
- metodologia Dijkstra contínuo, 50
- métrica, 10
 - das dobras, 10
 - de caminho mínimo, 2
 - de orientações fixas, 11
 - esférica, 84
 - fluvial, 82
 - geodésica, 4, 87
 - Minkowski, 10
 - retilínea, 10
 - valorada, 83
 - euclideana, 23
 - retilínea, 24
- modelo de elevação digital, 9
- módulo externo, 101
- obstáculo, 6
 - arco convexo, 19
- painel
 - de ferramentas, 100
 - principal, 100
- par aresta-face, 51
- particionamento em janela, 14
- PEM, 24
- PGAD, 34
- PGD, 23
- pipa, 16
- pipe, 101
- PIPS, 18
- planejamento de caminho mínimo, 1
- planejamento de movimentos, 1
- plano de horizonte, 28
- POECD, 20
- polígono
 - estrelado, 3
 - monotônico, 3
 - retilíneo, 4
 - simples, 3
 - triangulação, 3
- ponto
 - crítico
 - de entrada, 45
 - de saída, 45
 - de acesso, 48
 - de cruzamento, 43
 - primeiro, 51
 - último, 51
 - de empate, 59
 - de evento, 50
 - rotulado permanentemente, 51
 - de referência, 7
 - de saliência, 22
 - fronteira, 48
 - imagem, 106
 - médio geodésico, 96
- POPD, 19
- POSGRP, 20
- posicionamento livre, 8
- POSMRP, 20
- problema

- árvore geradora mínima, 78
- cobertura de nós, 78
- escalamento, 74
- esconderijo máximo, 24
- geodésica discreta, 23
- par mais próximo, 74
- proximidade, 73
- região valorada, 23
- todos vizinhos mais próximos, 75
- PRV, 23
- PSPC, 22

- quad-edge, 103
 - álgebra de arestas, 103
 - órbita, 103
 - referência de aresta, 104
 - vetor de aresta, 104

- rede irregular triangulada, 9
- retângulo isotético, 80

- segmento compartilhado, 43
- subdivisão de Voronoi, 92
- subdivisão planar, 3
 - região, 3
 - externa, 3
- superfície poliédrica, 9
 - limitada, 32

- trabalho, 2
- triangulação, 3
 - geodésica, 17
- triângulo geodésico, 17
 - ápices, 17

- versão
 - caso único, 12
 - origem única, 12
 - todos os pares, 12
- vértice, 9
 - especial, 10
 - extremo local, 9
 - fronteira, 9
 - interior, 9
 - máximo local, 9
 - mínimo local, 9
 - permanentemente rotulado, 31
 - sela, 9
- VGAD, 89
- vizinho mais próximo
 - geodésico, 96