PROBLEMAS DE DESLOCAMENTO NO PLANO EM GEOMETRIA COMPUTACIONAL

Aluno: Natan Costa Lima
Orientador: Professor Carlos Eduardo Ferreira
Conselho Nacional de Desenvolvimento científico e tecnológico
Palavras Chave: Geometria - Computacional - Plano - Deslocamento

Departamento de Ciência da Computação
Instituto de Matemática e Estatística
Universidade de São Paulo

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

1. Introdução

Problemas de deslocamento no plano em geometria computacional têm diversas aplicações, principalmente na área de robótica. Muitos dos problemas encontrados foram extensivamente estudados, como o problema de construir o grafo de visibilidade, o problema do menor caminho euclidiano e alguns problemas relacionados a galerias de arte.

Um problema clássico que surge é o problema em que o robô precisa atravessar uma galeria sem colidir com obstáculos ou então fazer o mesmo pelo menor caminho possível em relação a distância euclidiana. Há diversas soluções interessantes para estes problemas e uma delas é utilizando a construção de um grafo de visibilidade, uma abordagem útil pois além de possibilitar a resolução do problema ainda nos fornece uma estrutura com informações sobre a visibilidade dos pontos envolvidos na forma de grafo.

Outro problema interessante é o do vigia, Watchman route problem. Neste problema, temos uma galeria e queremos traçar um caminho curto para que o vigia consiga enxergar toda galeria enquanto anda por esse caminho.

2. Grafo de visibilidade

Grafos de visibilidade são usados para computarmos caminhos de distância euclidiana mínima dentro de um ambiente com obstáculos. Para poder apresentar um método de construção, vamos precisar do seguinte lema:

Lema 1. (Menor caminho)
Seja P um conjunto de polígonos (que por conveniência chamaremos de obstáculos), qualquer caminho mínimo entre dois pontos s e t, através de P, é um caminho polígonal onde seus vértices são vértices dos polígonos em P, s e t.

Com este lema, podemos definir melhor o que é um grafo de visibilidade que denotaremos por $GVis(P)$.
Definição 1 (Grafo de visibilidade)
$GVis(P)$, o grafo de visibilidade de P é o grafo onde seus vértices são os vértices contidos em $P \bigcup \{s,t\}$, sendo s e t os pontos para os quais queremos a menor distância e os arcos em $GVis(P)$ são entre vértices onde há caminho em linha reta entre eles, e que não colida com obstáculos, com custo igual a distância entre esses vértices..

Assim se construirmos o grafo de visibilidade, basta rodarmos o algoritmo de Dijkstra para conseguirmos um caminho de menor distância entre s e t.

2.1 Algoritmos para construção do grafo

Vamos considerar n como $|P|+2$ para incluirmos s e t entre nossos vértices do grafo. Um algoritmo com complexidade $O(n^3)$ é imediato, basta testarmos para todos os n^2 pares de vértices (i,j), se o segmento de reta $\overline{ij}$ cruza com algum segmento pertencente aos polígonos em P. Em caso negativo, o par (i,j) se enxerga e incluímos a aresta ij no grafo.

Para melhorar a complexidade, se ao invés de olharmos para os vértices em ordem arbitrária considerarmos uma ordem especial, podemos desenvolver um algoritmo mais eficiente que o algoritmo ingênuo. Usaremos também uma técnica chamada linha de varredura para processar os segmentos dos polígonos.

Basicamente o que vamos fazer é fixar um vértice $p \in P \bigcup \{s,t\}$ e ordenar o restante dos vértices por ângulo ao redor de p, esta será a ordem de nossos eventos, cada vértice pode inserir ou remover segmentos de nossa linha de varredura dependendo de qual lado no sentido horário está o restante do segmento e fazendo algumas consultas nesta estrutura poderemos dizer se um determinado ponto é visível para p ou não adicionando assim as arestas de $GVis(P)$. Vamos a uma definição formal do algoritmo.

$VISIBILITY - GRAPH(P,s,t)$
1. $V \leftarrow P \bigcup \{s,t\}$
2. **para cada** $p \in V$ **faça**
3. $\quad O \leftarrow ORDENA(p, V/p)$
4. $\quad T \leftarrow INICIALIZA(p)$
5. $\quad$ **para cada** $q \in O$ **faça**
6. $\quad\quad$ **se** $VISIVEL(T,p,q)$**faça**
7. $\quad\quad\quad$ Adicione a aresta pq em $GVis(P)$
8. $\quad\quad ADICIONA(T,p,q)$
9. $\quad\quad REMOVA(T,p,q)$

Abaixo uma descrição das rotinas.
$ORDENA(p,C)$ simplesmente ordena os vértices do conjunto C pelo ângulo a partir do eixo x usando como pivo o vértice p.
$INICIALIZA(p)$ inicializa a estrutura da linha de varredura com os segmentos que cruzam a semi-reta paralela ao eixo x e que têm como início p.
$VISIVEL(T,p,q)$ decide se q é visível a partir de p vendo se o segmento $\overline{pq}$ cruza com o segmento mais perto de p na linha de varredura.
$ADICIONA(T,p,q)$ e $INSERE(T,p,q)$ adicionam e inserem segmentos na linha de varredura dependendo se os segmentos que têm como ponta q têm outra ponta no sentido horário ou anti-horário em relação a q pela ordenação usada.

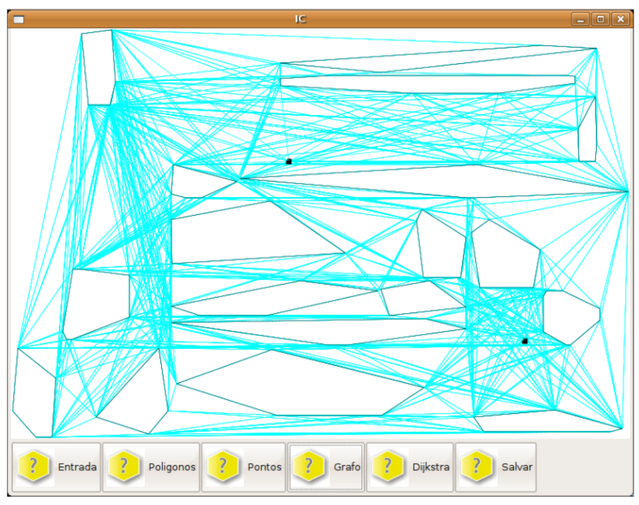

Figura 1: *Grafo de visibilidade.(Fonte: Programa desenvolvido durante a iniciação)*

3. O problema da rota do vigia

Iremos apresentar aqui outro problema de visibilidade, conhecido como problema da rota do vigia (*Watchman Route*). O problema consiste em acharmos uma rota interna a um polígono tal que todos os pontos do polígono são visíveis de algum ponto da rota. Uma possível aplicação seria pensar como a rota de um vigia noturno que deseja cuidar de uma galeria, e gostaríamos que além de vigiar toda a galeria, ainda o fizesse em pouco tempo, para isto temos o objetivo de minimizar o tamanho da rota.

Vamos mostrar aqui que este problema é NP-difícil para polígonos arbitrários.

3.1 Definindo o problema

Problema do vigia
Entrada: Polígono P com buracos, inteiro k
Questão: Existe uma rota de tamanho menor ou igual a k, tal que esta não intersecte o exterior de P ou o interior de algum dos buracos de P e todos os pontos do polígono são visíveis de algum ponto da rota?

O problema do vigia é NP-Difícil através de uma redução ao problema do caixeiro viajante geométrico retilinear (**geometric traveling salesman**).
Eis a definição:
Caixeiro viajante geométrico retilinear
Entrada: Conjunto S com n pontos no plano, inteiro b
Questão: Existe um caminho de comprimento menor ou igual a b que visita todos os pontos de S através de um caminho retilinear?

3.2 Dificuldade do problema

Esta variante do problema do caixeiro viajante geométrico retilinear continua NP-difícil [3]. Dizemos que o problema é retilinear se os caminhos só podem ser feitos com linhas horizontais ou verticais.

A trasformação envolve a construção de uma galeria com $O(n)$ corredores retilineares envoltos em um retângulo como na figura 3. As paredes dos corredores definem $O(n^2)$ buracos convexos no interior do polígono. Por fim inserimos no lugar dos pontos, estruturas como as mostradas abaixo. Claramente a construção pode ser feita em tempo polinomial. Temos assim o seguinte resultado.

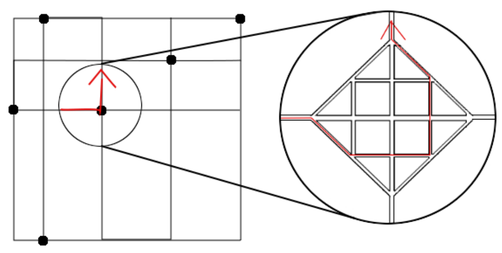

Figura 2: *Método de construção para representar o problema do caixeiro como problema do vigia*

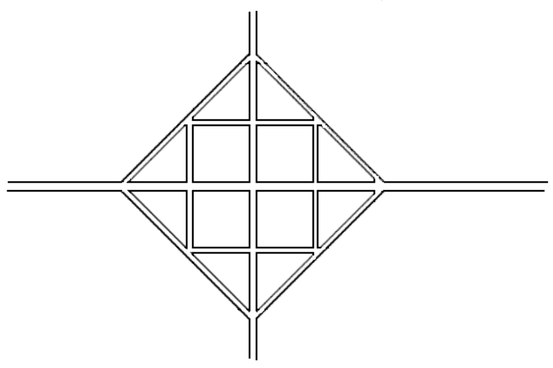

Figura 3: *Estrutura inserida para moldar o problema do caixeiro no problema do vigia.*

Teorema O problema do vigia é NP-difícil até mesmo com polígonos convexos e buracos convexos.

Para a prova devemos notar que haverá uma solução do tamanho B para o problema do caixeiro, se e somente se existir uma rota do vigia de comprimento $B+(2\sqrt{2}nx+4nx)-(2b+4c)x$, onde x é o tamanho do lado dos quadrados dentro da estrutura, b é o número de pontos em S que ficam nas arestas mais externas do retângulo R e c é o número de pontos de S que são pontas do retângulo externo R.

4. Conclusão

Existem alguns algoritmos mais eficientes para a construção do grafo de visibilidade, incluindo um algoritmo ótimo com complexidade $O(n \log n + k)$, onde k é o número de arcos no grafo [4].

Para o problema do vigia existe alguns algoritmos eficientes quando colocamos restrições aos polígonos, como no caso em que o polígono é retilinear e não tem buracos, ou então quando consideramos polígonos simples.

Geometria computacional é uma área bem interessante, apesar de apresentar certa dificuldade em tratar casos especiais como pontos colineares e erros de precisão com ponto flutuante na hora de implementar os algoritmos, é uma área antiga mas ainda em constante evolução.

Referências

[1] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computacional Geometry: Algorithms and applications (second edition)*, Springer, New York, 2005.

[2] Wei-pang Chin, *Optimum watchman routes.*, Information processing letters **28 issue 1** (1988), 39-44.

[3] M. R. Garey, R. L. Graham, and D. S. Johnson, *Some NP-complete geometric problems.*, Proceedings of the eight annual ACM symposium on Theory of computing (1976), 10-22.

[4] S. K. Ghosh and D. M. Mount, *An output-sensitive algorithm for computing visibility graphs.*, SIAM J. Comput **20** (1991), 888–910.