Triangularização de Peso Mínimo

Francisco Elói Soares de Araújo

Resumo

O Problema da Triangularização de Peso Mínimo pode ser formulado da seguinte maneira. Dado um conjunto S de pontos no plano, determinar uma triangularização de S que minimiza a somatória do comprimento das arestas. Não existe nenhum algoritmo polinomial para resolução do problema nem existe prova de que o mesmo seja NP-difícil. O objetivo desse trabalho é mostrar algumas linhas de pesquisa e referência bibliográficas sobre o assunto.

1 Introdução

Inicialmente vamos formalizar nosso problema. Seja $S = \{p_0, p_1, \ldots, p_{n-1}\}$ um conjunto de pontos no plano. Denotaremos por x_i e y_i as coordenadas do ponto p_i. Assumiremos, para simplificar a exposição, que não existem três pontos colineares em S. Denotaremos por $p_i p_j$ o segmento de reta com extremos p_i e p_j (fecho convexo dos pontos p_i e p_j). Denotaremos por $E(S)$ o conjunto dos segmentos de reta cujos extremos são pontos de S. Chamaremos também os elementos de $E(S)$ de *arestas*. Diremos que dois segmentos de reta se *cruzam* se eles se intersectam em qualquer ponto que não seja um extremo comum. Uma *triangularização* $T(S)$ de S é um conjunto maximal de arestas que conectam pontos de S e não se cruzam duas a duas. O *peso de uma aresta* $p_i p_j$ de $T(S)$, que denotaremos por $w(p_i, p_j)$, é a distância euclidiana entre seus extremos, isto é,

$$w(p_i, p_j) = \left((x_i - x_j)^2 + (y_i - y_j)^2\right)^{1/2}.$$

O *peso W de uma triangularização* $T(S)$ é a soma dos pesos das arestas de $T(S)$, isto é,

$$W\left(T(S)\right) = \sum_{p_i p_j \in T(S)} w(p_i, p_j).$$

Uma *triangularização de peso mínimo de* S é uma triangularização de S que tem peso mínimo entre todas as outras triangularizações de S. Denotaremos por $MWT(S)$ o conjunto de todas as triangularizações de peso mínimo de S.

Problema 1: **Triangularização de Peso Mínimo.**
Instância: Um conjunto S de pontos no plano.
Questão: Encontrar um elemento de $MWT(S)$.

O problema foi primeiramente descrito por Shamos e Hoey [SH75]. Sua solução tem aplicações em aproximação numérica de dados bivariados.

Não se conhece nenhum algoritmo polinomial para a solução do problema. Entretanto, tentativas em mostrar que o problema é NP-difícil também falharam sendo, portanto, a sua complexidade um problema em aberto.

Na seção 2 teremos uma idéia de como, utilizando o algoritmo de programação dinâmica de Gilbert para triangularização de peso mínimo de polígonos em tempo $O(n^3)$, podemos efetuar a triangularização de peso mínimo de uma figura particular que chamamos de célula. Na seção 3 veremos uma teoria para desenvolver algoritmos exatos a partir de subgrafos de $mwt(S)$, onde $mwt(S)$ é o grafo que contém os vértices de S e todas as arestas que aparecem em todas as triangularizações de peso mínimo de S. Na seção 4 veremos uma maneira de obter um subgrafo de $mwt(S)$. Dois algoritmos aproximados, $MST-T$ e $G-ST-T$, que possuem fator de aproximação $\Omega(n)$, serão mostrados na seção 5. Finalmente, na seção 6, mostraremos alguns problemas de decisão associados ao problema de triangularização de peso mínimo já provados serem NP-completos.

2 Triangularização de Células

Embora não se conheça um algoritmo polinomial para triangularizar um conjunto qualquer de pontos, em 1979, P. D. Gilbert (segundo [HP94]) descreveu um algoritmo que determina uma triangularização de peso mínimo de polígonos simples em tempo $O(n^3)$. O algoritmo de Gilbet pode ser estendido para triangularizar uma figura mais geral que chamamos de *célula*. Uma célula é representada por um grafo G cujos vértices é um conjunto V de n pontos no plano e cujas arestas é um conjunto A tal que:

- existe um polígono simples P que é um subgrafo de G e todos os vértices de G ou são vértices da fronteira de P ou são pontos que estão no interior de P;
- se $p \in V$ não é vértice da fronteria de P então existe um único caminho $C = px_1x_2 \ldots x_m$ tal que $x_1x_2 \ldots x_{m-1}$ não são vértices da fronteira de P mas x_m é um vértice da fronteira de P;
- se $f_1 \in A$ e $f_2 \in A$ então f_1 e f_2 não se cruzam.

Descreveremos uma célula *cel* pelos vértices $p_0, p_1, \ldots, p_{k-1}$ encontrados em sentido anti-horário. Note que os vértices de *cel* podem aparecer na sua descrição mais de uma vez. Veja Figura 1 e observe como descrevemos uma célula. Para o propósito de descrever uma célula, toda ocorrência de um mesmo ponto na seqüência é tratada como se fossem pontos diferentes. Observe que uma célula com n vértices tem n arestas e, ao percorrermos a célula no sentido anti-horário para sua descrição, esse passeio não passa mais do que duas vezes pela mesma aresta, o que implica que a descrição da célula tem $O(n)$ elementos e, portanto, o algoritmo de Gilbert modificado para encontrar uma triangularização de peso mínimo de uma célula também gasta tempo $O(n^3)$.

Chamaremos o algoritmo de Gilbert modificado para triangularização de peso mínimo de uma célula, de algoritmo *OCT ("Optimal cell triangulation")*.

3 Subgrafo da Triangularização de Peso Mínimo

A partir do conceito de célula obtido na seção anterior e sua relação com triangularização de peso mínimo, muitos trabalhos têm sido desenvolvidos para encontrar uma célula que contenha todos os pontos de S e seja um subgrafo de um elemento de $MWT(S)$. Denotaremos por $mwt(S)$

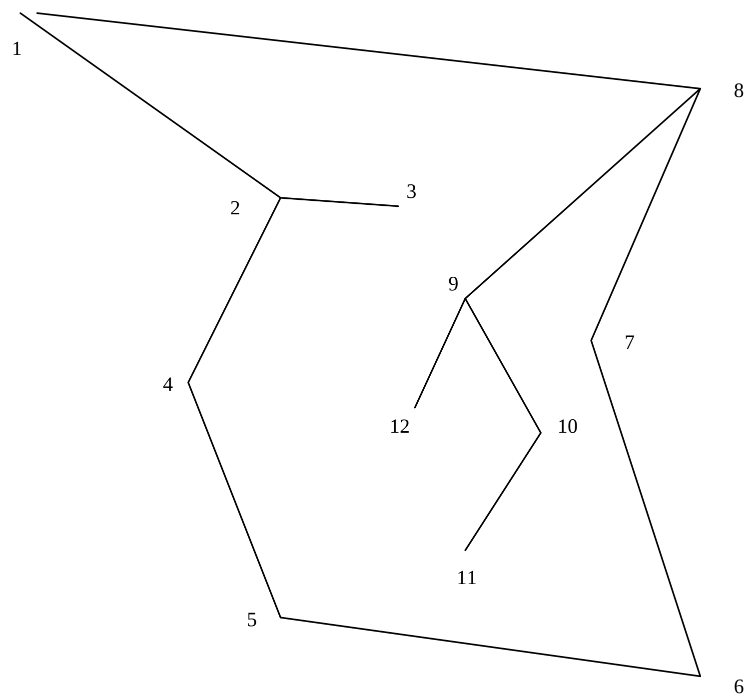

Fig. 1: Uma célula com 12 pontos descrita pela seqüencia $cel = (1,2,3,2,4,5,6,7,8,9,10,11,10,9,12,9,8)$.

o grafo tal que S é o conjunto de vértices e as arestas são aquelas que aparecem em todas as triangularizações de peso mínimo de S.

Denotaremos por $conv(S)$ o fecho convexo definido por S. Denotaremos por $CH(S)$ as arestas extremas de $conv(S)$. Claramente $CH(S)$ são arestas de $mwt(S)$.

Diversas propriedades foram observadas para que se determinasse um algoritmo que devolvesse um subgrafo de $mwt(S)$ (o ideal seria se soubessemos determinar o próprio grafo $mwt(S)$). É claro que, se estamos interessados em algoritmos exatos e rápidos é interessante que o número de componentes do subgrafo de $mwt(S)$ obtido pelo algoritmo seja o menor possível (observe que se o subgrafo de $mwt(S)$ possuir uma só componente, poderemos acrescentar a ele as arestas de $CH(S)$, remover arestas de circuitos (enquanto houver circuitos) que não sejam arestas extremas de $conv(S)$ e obter então uma célula que, como vimos, pode, em tempo $O(n^3)$, ser triangularizada otimamente usando o algoritmo de Gilbert).

Suponha que temos um subgrafo de $mwt(S)$ que contém as arestas extremas de $conv(S)$ e contenha k componentes conexas. Existem, no máximo, $\binom{n^2}{k-1}$ maneiras de escolher $k-1$ arestas para conectar essas componentes. Entre algumas dessas escolhas definimos células diferentes que podem ser, cada uma, computadas em $O(n^3)$ (observe que em algumos maneiras de escolher e acrescentar arestas, não produzimos um grafo conexo e em algumas outras, temos arestas se cruzando). Portanto computaríamos $MWT(S)$ em tempo $O(n^{O(k)})$ o que implica se algum algoritmo gerar um subgrafo de $mwt(S)$ com um número constante de componentes, o mesmo poderá ser usado como subrotina de um algoritmo polinomial para Triangularização de Peso Mínimo (obviamente, devido o exposto até agora, tal algoritmo não existe).

Na próxima seção descreveremos um algoritmo desenvolvido por Dickerson e Montague [DM96] que, em centenas de testes com conjuntos acima de 220 pontos obtidos uniforme e aleatoriamente, os autores obtiveram como saída, em todas as instâncias, um subgrafo de $mwt(S)$ conexo. Além disso, os mesmos fizeram testes empíricos comparando o algoritmo desenvolvido com o subgrafo de $mwt(S)$ obtido por Keil [Kei94] chamado "β-skeleton" e observaram que o primeiro continha substancialmente mais arestas que o segundo.

4 Triangularizações Localmente Ótimas

Dado um conjunto S de pontos e um polígono P cujos vértices são elementos de S (podem haver elementos de S que não são vértices de P), dizemos que P é *branco em relação a S* se não existir nenhum ponto de S que seja interior a P. Diremos, simplesmente, que P é branco se S estiver claro no contexto.

Em uma triangularização $T(S)$ uma *aresta f é localmente ótima* (em relação a $T(S)$) se uma das seguintes situações ocorre.

• A aresta f é aresta de $conv(S)$.

• A aresta f é adjacente a dois triângulos brancos que juntos formam um quadrilátero não-convexo.

• A aresta f é adjacente a dois triângulos brancos que formam um quadrilátero convexo e $w(f) \leq w(f')$, onde f' é a outra diagonal do quadrilátero.

Diremos que uma *triangularização é localmente ótima* se cada uma de suas arestas é localmente ótima.

Claramente, uma triangularização de $MWT(S)$ deve ser localmente ótima. Infelizmente, S pode permitir diferentes triangularizações que são localmente ótimas mas não são globalmente ótimas. Denotaremos por $LOT(S)$ o conjunto de todas as triangularização localmente ótimas. O grafo $lot(S)$ é o grafo contendo todas as arestas que aparecem em todas as triangularizações localmente ótimas.

Lema 1: [DM96] Para qualquer conjunto de pontos S no plano, $lot(S)$ é um subgrafo de $mwt(S)$. Além disso, se a aresta f não está em nenhum subgrafo de $LOT(S)$ então não está em nenhum subgrafo de $MWT(S)$.

O lema acima particiona as arestas em três conjuntos distintos: aquelas que estão em *todas* as triangularizações localmente ótimas; aquelas que estão em *nenhuma* das triangularizações localmente ótimas; aquelas que estão em *algumas* (mas não todas) triangularizações localmente ótimas. As duas propriedades a seguir são fáceis de serem verificadas e permitem-nos construir um algoritmo para encontrar um subgrafo de $lot(S)$.

Lema 2: [DM96] Seja $f \in E(S)$. Seja $T_L(f)$ o conjunto de triângulos brancos (sobre todas as triangularizações) que são limitados por f de um lado, digamos o esquerdo, e $T_R(f)$ o conjunto de triângulos brancos que são limitados por f do lado direito. Se não existe $t_1 \in T_L(f)$ e $t_2 \in T_R(f)$ tal que f seja localmente ótimo então f não pertence a qualquer subgrafo de $LOT(S)$.

Lema 3: [DM96] A aresta f pertence a todos os subgrafos de $LOT(S)$ se e somente se qualquer aresta que intersecta f não pertence a qualquer subgrafo de $LOT(S)$.

Algoritmos LT
Entrada: conjunto S de pontos no plano.
Saída: um subgrafo de $mwt(S)$.

1. $candTris$:= Lista de todos os triângulos brancos de S;
2. $candEdges$:= Lista de todas as arestas de $E(S)$;
3. $edgesInMWT$:= Lista vazia;
4. Remova arestas extremas de $conv(S)$ de $candEdges$ e insira na lista $edgesInMWT$;
5. Repita
   Para cada $f \in candEdges$ teste se f é localmente ótima para toda combinação de triângulos brancos vizinhos a f de $candTris$. Se para algum quadrilátero, f não for localmente ótima então f não está em $lot(S)$ e deve ser removido da lista $candEdges$. Além disso, remova todos os triângulos da lista $candTris$ que contém f.
   até que não haja mais arestas de $candEdges$ para serem removidas;
6. Para cada $f \in candEdges$, se f não intersecta nenhuma outra aresta em $candEdges$, adicione f à lista $edgesInMWT$.

Teorema 4: [DM96]
Quando o algoritmo LT termina, as aresta em edgesInMWT formam um subgrafo de $lot(S)$ e, portanto, um subgrafo de $mwt(S)$.
A complexidade de tempo do algoritmo descrito é $O(n^6)$ no pior caso.

5 Algoritmos Aproximados

Segundo Heath e Pemmaraju [HP94], muitos algoritmos aproximados para a solução do problema de triangularização de peso mínimo têm sido propostos. Nenhum deles, entretanto, tem um fator de aproximação constante. As aproximações das triangularizações gulosa e de Delaunay têm sido estudadas. A triangularização gulosa pode ser construída em tempo $O(n^2)$ por um algoritmo guloso que escolhe a menor aresta que não cruza com qualquer outra aresta previamente escolhida. A triangularização de Delaunay pode ser construída em tempo $O(n \log n)$. Lloyd [Llo77] mostrou simples contra-exemplos de que ambos os algoritmos não produzem uma triangularização de peso mínimo. Denotaremos por $GT(S)$ e $DT(S)$ o conjunto das triangularizações gulosa e de Delaunay respectivamente.

Levcopoulus [Lev87] mostrou que o fator de aproximação de $GT(S)$ é $\Omega(n^{1/2})$. Kirkpatrick [Kir80] mostrou que o fator de aproximação de $DT(S)$ é $\Omega(n)$. Portanto, $GT(S)$ e $DT(S)$ podem ter aproximações arbitrariamente ruins.

Plaisted e Hong [PH87] desenvolveram um algoritmo de complexidade $O(n^5)$ e mostraram que o peso da triangularização que seu algoritmo produz tem um fator de aproximação $O(\log n)$. Smith [Smi89] melhorou o tempo de execução desse algoritmo para $O(n^2 \log n)$. O limitante inferior da performance da heurística de Plaisted e Hong é o melhor conhecido.

Descreveremos agora dois algoritmos de aproximação para triangularização de peso mínimo.

Algoritmo $MST-T$ (Algoritmo de triangularização da Árvore Espalhada Mínima)
Entrada: conjunto S de pontos no plano.
Saída: uma triangularização $MST-T$.

1. Compute $DT(S)$;
2. Compute Árvore Espalhada Mínima $MST(S)$ do grafo $(S, DT(S))$ usando a seguinte atribuição de custos para aresta de $DT(S)$;
$c\left((p_i, p_j)\right) = \begin{cases} 0 & \text{se } (p_i, p_j) \in CH(S), \\ w(p_i, p_j) & \text{caso contrário.} \end{cases}$
3. Use OCT para triangularização de peso mínimo da célula definida por $CH(S) \cup MST(S)$.

Observe que passos 1 e 2 do algoritmo $MST-T$ consomem tempo $O(n \log n)$ e passo 3 gasta tempo $O(n^3)$ sendo, portanto, a complexidade de tempo do algoritmo $O(n^3)$.

Denotaremos por $R(T(S))$ o fator de aproximação da triangularização $T(S)$, isto é,

$$R(T(S)) = \frac{W(T(S))}{W(MWT(S))}.$$

Lingas [Lin87] mostrou que

$$R(MST-T(S)) = O\left(\frac{n * jump(S)}{W(MWT(S))} + \log n\right),$$

onde $jump(S) = \max\{w(p_i, p_j) | (p_i, p_j) \in MST(S) \setminus CH(S)\}$.

Lingas também mostrou um exemplo no qual $n * jump(S)/W(MWT(S)) = \Omega(n)$. Segundo Heath e Pemmaraju [HP94], a literatura mostra alguns exemplos onde $MWT-T(S)$ não produz uma triangularização de peso mínimo.

O segundo algoritmo difere do primeiro apenas na escolha da árvore espalhada.

Algoritmo $G-ST-T$ (Algoritmo de triangularização da Árvore Espalhada Mínima)
Entrada: conjunto S de pontos no plano.
Saída: uma triangularização $G-ST-T$.

1. Compute $GT(S)$;
2. Compute Árvore Espalhada Mínima $GMST(S)$ do grafo $(S, GT(S))$ usando a seguinte atribuição de custos para aresta de $GT(S)$;
$c\left((p_i, p_j)\right) = \begin{cases} 0 & \text{se } (p_i, p_j) \in CH(S), \\ w(p_i, p_j) & \text{caso contrário.} \end{cases}$
3. Use OCT para triangularização de peso mínimo da célula definida por $CH(S) \cup GMST(S)$.

Teorema 5: [HP94]
A complexidade de tempo do algoritmo $G-ST-T$ é $O(n^3)$.

Para qualquer conjunto de pontos S de tamanho n

$$R(G-ST-T(S)) = O\left(\frac{n * greedy\_jump(S)}{W(MWT(S))} + \log n\right),$$

onde $greedy\_jump(S) = \max\{w(p_i,p_j)|(p_i,p_j) \in GMST(S) \setminus CH(S)\}$.

Heath e Pemmaraju [HP94] mostraram resultdos experimentais que indicavam que a escolha de $G-ST-T(S)$ produz triangularização de peso menor do que $MST-T(S)$ e quase sempre mínima (ótima).

6 Problemas Associados à Triangularização de Peso Mínimo

Enquanto permanece em aberto a conjectura de ser o problema da triangularização de peso mínimo NP-difícil, para alguns outros problemas de decisão associados a esse já sabemos serem eles NP-completos.

Apresentaremos alguns deles já que os mesmos poderão ser bons candidatos a redução à versão do problema de decisão da triangularização de peso mínimo caso o último realmente seja NP-completo.

Primeiramente mostraremos a versão de decisão do problema formulado por Heath e Pemmajaru [HP94].

Problema 2: **Triangularização de Peso Mínimo** - (versão problema de decisão).
Instância: Um conjunto S de pontos no plano; um número racional positivo k.
Questão: Existe uma triangularização $T(S)$ tal que $W(T(S)) \leq k$?

Lloyd [Llo77] provou ser NP-completo o seguinte problema de decisão.

Problema 3: **Existência de Triangularização.**
Instância: Um conjunto S de pontos no plano; um conjunto de segmentos de reta $E' \subseteq E(S)$.
Questão: Existe uma triangularização $T(S) \subseteq E'$?

O problema da triangularização de peso mínimo pode ser visto de várias maneiras. Considerando o grafo $G = (S, E(S))$ (grafo completo) e sendo

$$C = \{(f_i, f_j) \in E(S) \text{x} E(S) | f_i \text{ e } f_j \text{ se cruzam.}\},$$

definimos o *grafo de cruzamento* de G como sendo o grafo $G^c = (E(S), C)$. Uma visão particular de nosso problema original é descrito a seguir e provado por Heath e Pemmajaru [HP94] ser NP-completo.

Problema 4: **Conjunto Independente Maximal Mínimo Limitado.**
Instância: Um grafo $G^c = (E, C)$, onde cada vértice f de E tem peso $w(f)$; um número racional positivo k.
Questão: Existe um conjunto independente maximal $I \subseteq E$ tal que $w(f) \leq k$?

Um resultado mais significativo de NP-completude é o problema de decisão geométrico, mostrado abaixo, também provado ser NP-completo por Heath e Pemmajaru [HP94].

Problema 5: **Triangularização de Peso Mínimo Genérica.**
Instância: Um conjunto S de pontos no plano; um conjunto de segmentos de reta $E' \subseteq E(S)$; um número racional positivo k.
Questão: Existe uma triangularização em E' cujo peso não é maior que k?

Referências

[DM96] M. T. Dickerson and M. H. Montague. A (usually) connected subgraph of the minimum weight triangulation. Manuscrito, 1996.

[HP94] L. S. Heath and S. V. Pemmaraju. New results for the minimum weight triangulation problem. *Algorithmica*, 12:533–552, 1994.

[Kei94] J. M. Keil. Computing a subgraph of the minimum weight triangulation. *Computational Geometry: Theory and Applications*, 4:13–26, 1994.

[Kir80] D. G. Kirkpatrick. A note on delaunay and optimal triangulations. *Information Processing Letters*, 10:127–128, 1980.

[Lev87] C. Levcopoulos. An $\mathcal{O}(n^{1/2})$ lower bound for the non-optimality of the greedy triangulations. *Information Processing Letters*, 25:247–251, 1987.

[Lin87] A. Lingas. A new heuristic for the minimum weight triangulations. *SIAM Journal Algebraic and Discrite Methods*, 8:646–658, 1987.

[Llo77] E. L. Lloyd. On triangulations of a set of points in the plane. In *Proceedings of the Eighteenth IEEE Symposium on Foundations of Computer Science*, pages 228–240, 1977.

[PH87] D. A. Plaisted and J. Hong. A heuristic triangulation algorithm. *Journal of Algorithms*, 8:405–437, 1987.

[SH75] M. I. Shamos and D. Hoey. Closest point problems. In *Proceedings of the Sixteenth IEEE Symposium on Foundations of Computer Science*, pages 151–162, 1975.

[Smi89] W. D. Smith. Studies in computational geometry motivated by mesh generation. dissertação, 1989.