

Fernando Seiti Yamamoto

**ALGORITMO EXATO PARA A ÁRVORE DE STEINER
EUCLIDIANO NO PLANO**

**Curitiba,
Março, 2013**

Fernando Seiti Yamamoto

**ALGORITMO EXATO PARA A ÁRVORE DE STEINER
EUCLIDIANO NO PLANO**

Trabalho apresentado para obtenção do Grau de
Bacharel em Ciência da Computação pela Universi-
dade Federal do Paraná

Orientador: André Luiz Pires Guedes

**Curitiba,
Março, 2013**

Sumário

Lista de Figuras	p. iii
Lista de Tabelas	p. v
Resumo	p. vi
1 Introdução	p. 7
2 Definição	p. 9
2.1 Método de Torricelli	p. 9
2.2 Método de Simpson	p. 10
2.3 Heinen	p. 11
2.4 Razão de Steiner	p. 11
2.5 Árvore de steiner cheia	p. 12
3 Algoritmo Exato	p. 14
3.1 Geração de ASC	p. 14
3.1.1 Algoritmo de Melzak	p. 14
3.1.1.1 Fase de divisão	p. 15
3.1.1.2 Fase de reconstrução	p. 15
3.1.1.3 Exemplo de execução	p. 15
3.1.2 Winter	p. 18
3.2 Concatenação	p. 19
4 Conclusão	p. 21

Referências..... p. 22

Lista de Figuras

Figura 1	Exemplo de árvore de Steiner	8
Figura 2	Método de Torricelli	10
Figura 3	Método de Simpson	10
Figura 4	Solução com ponto de Steiner e solução sem ponto de Steiner	11
Figura 5	Comparação entre árvore geradora mínima e árvore de Steiner	12
Figura 6	Combinação de árvores de Steiner cheias ligadas nos vértices destacados	13
Figura 7	Topologias possíveis para quatro vértices	14
Figura 8	Exemplo do algoritmo de Melzak	15
Figura 9	Algoritmo de Melzak Passo 1	16
Figura 10	Algoritmo de Melzak Passo 2	16
Figura 11	Algoritmo de Melzak Passo 3	16
Figura 12	Algoritmo de Melzak Passo 4	17
Figura 13	Lune	18

Figura 14	ASCs utilizadas para a solução da figura 1		19
Figura 15	Execução do geosteiner3.1 para uma instância grande do problema		20

Lista de Tabelas

Tabela 1	Função $f(n)$		17
----------	---------------	-------	----

Resumo

O objetivo deste trabalho é apresentar um estudo sobre uma solução exata de uma variação do problema da árvore de Steiner. Árvores de Steiner são árvores de tamanho mínimo que abranjem um conjunto de vértices ao mesmo tempo em que vértices auxiliares podem ser incluídos na árvore para minimizar o seu tamanho.

Palavras-chave: árvores, steiner.

1 Introdução

O objetivo do problema da Árvore de Steiner é encontrar uma árvore que conecte determinados vértices chamados terminais a fim de se obter uma árvore de tamanho mínimo, ou seja, que a soma dos valores de suas arestas seja a mínima possível, ao mesmo tempo em que vértices extras conhecidos como pontos de Steiner podem ser adicionados para minimizar o tamanho da árvore.

Existem diversas variações do problema das árvores de Steiner como árvores de Steiner em grafos, árvores de Steiner reticulados, árvores de Steiner Euclidianos no $\mathbb{R}^n$ entre outros.

Tais problemas possuem aplicações práticas como a construção de rodovias, redes ferroviárias, dutos de óleo e gás, redes elétricas e circuitos elétricos são alguns exemplos onde a árvore de Steiner ajuda a minimizar o custo de material utilizado.

Porém em 1977 Garey, Graham e Johnson [4] provaram que o problema da árvore de Steiner mínima se encontrava na classe dos problemas NP, portanto diferentes abordagens foram consideradas para resolver o problema, como algoritmos heurísticos, programação dinâmica e soluções físicas como película de sabão.

Atualmente um dos melhores algoritmos, senão o melhor, que resolve esse problema é o geosteiner desenvolvido em conjunto por Warme, Winter e Zachariasen [2].

No presente trabalho é apresentado um estudo para soluções exatas do problema da árvore de Steiner euclidiano no Plano ($\mathbb{R}^2$). Uma das variações mais estudadas onde todo o plano deve ser considerado para os possíveis pontos de steiner, o que torna o problema mais complicado.

Na figura 1 temos um exemplo de uma árvore de Steiner gerado pelo geosteiner3.1 utilizando coordenadas aproximadas das capitais dos estados do Brasil como vértices terminais (pontos destacados na imagem) que são o conjunto de vértices que devem ser considerados para a solução do problema e os pontos de steiner (vértices de grau três) que são os vértices extras utilizados para minimizar o tamanho da árvore.

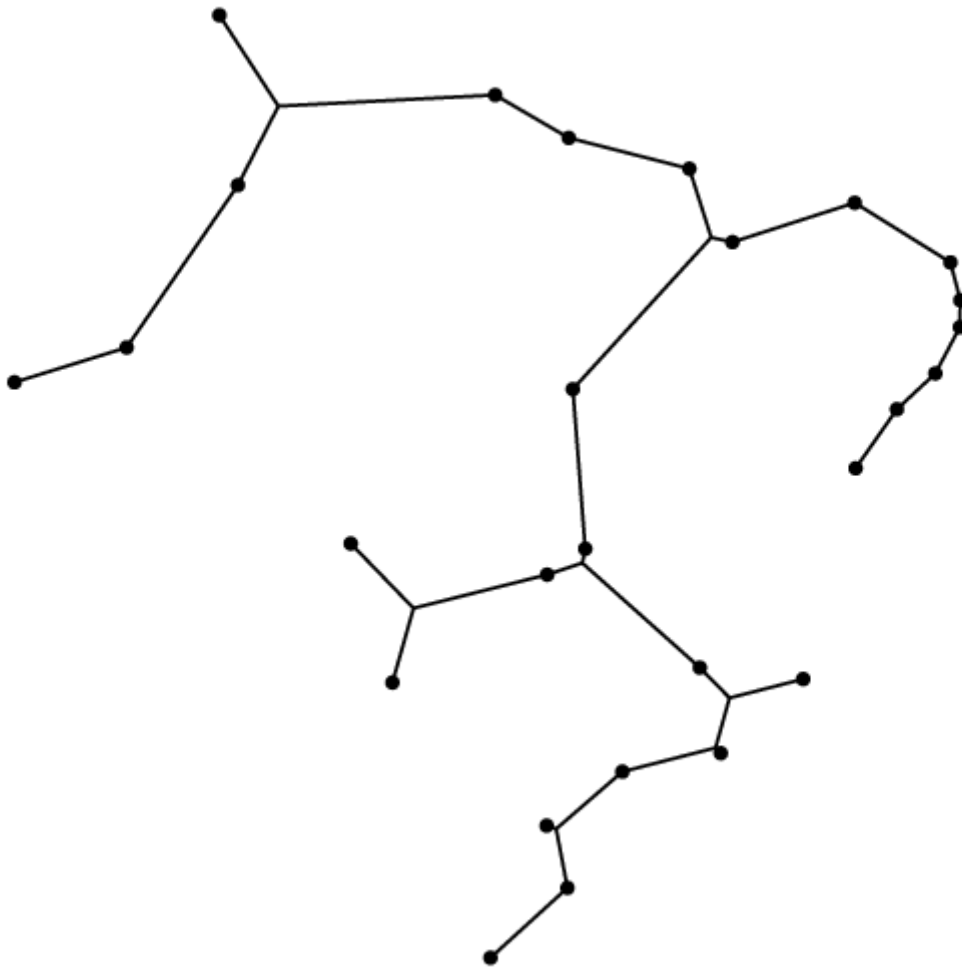


Figura 1: Exemplo de árvore de Steiner

No capítulo 2 daremos uma breve definição dos conceitos relacionados ao problema para então abordar o algoritmo no capítulo 3 onde explicaremos a solução exata do problema e concluir no capítulo 4.

2 Definição

Neste capítulo trataremos de algumas definições que ajudarão na compreensão do algoritmo utilizado para resolver a árvore de Steiner.

Seja n o número de vértices no plano euclidiano, vamos definir a solução para $n < 4$.

Para o primeiro caso onde $n = 1$ a distância para si mesmo é zero.

Agora consideremos o caso onde $n = 2$. A menor distância entre dois pontos no plano euclidiano é uma reta dada pela equação $d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$.

Para o caso onde $n = 3$, Pierre Fermat propôs o problema para encontrar um ponto que minimiza a distância entre três pontos no plano.

Para esse caso temos o método de Torricelli e o método de Simpson para resolver o problema.

2.1 Método de Torricelli

Torricelli descobriu uma solução em 1640 para o problema de Fermat.

Consideremos três pontos no plano euclidiano. A solução de Torricelli propõe que seja criado um triângulo com esses três pontos. Então é criado um triângulo equilátero para cada lado do triângulo formado anteriormente. Em seguida, cada triângulo equilátero é circunscrito com um círculo. O ponto onde os três círculos se encontram é conhecido como o ponto de Torricelli (Figura 2).

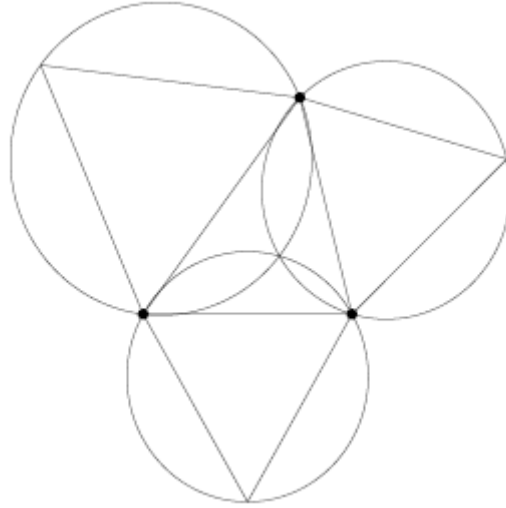


Figura 2: Método de Torricelli

2.2 Método de Simpson

Simpson propôs uma alternativa para encontrar o ponto de Torricelli onde, da mesma maneira que Torricelli, são criados os 3 triângulos equiláteros. Então, para cada triângulo equilátero formado, uma reta é desenhada para ligar o vértice de triângulo equilátero que não faz parte do triângulo original, e o vértice do triângulo original que não faz parte do triângulo equilátero. A intersecção destas três retas coincide com o ponto de Torricelli. Essas retas são conhecidas como linhas de Simpson (Figura 3).

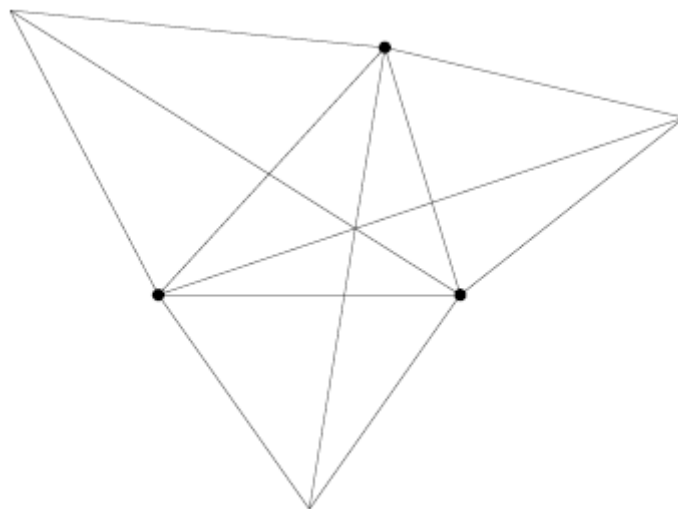


Figura 3: Método de Simpson

2.3 Heinen

Os métodos de Torricelli e de Simpson apenas funcionam se todos os ângulos internos do triângulo formado com os três pontos forem menores que 120° . Se essa condição não é satisfeita, o método de Torricelli produz um ponto fora do triângulo inicial e no método de Simpson as linhas não se encontram.

Heinen descobriu que para o caso de algum vértice do triângulo original possuir um ângulo maior que 120° , o vértice com o ângulo obtuso será o vértice que irá minimizar a distância entre os outros vértices (Figura 4).



Figura 4: Solução com ponto de Steiner e solução sem ponto de Steiner

E para os casos em que todos os ângulos são menores que 120° , o método de Torricelli e Simpson podem ser utilizados para obter a solução. E as três arestas que se encontram no ponto de Steiner formam um ângulo de 120° entre si. Sendo essa uma condição importante para os casos onde $n > 3$.

Um outro fato interessante descoberto por Heinen é que as linhas de Simpson têm o mesmo tamanho e que esse tamanho equivale ao tamanho da árvore gerada pelos três vértices ligados ao ponto de Torricelli.

2.4 Razão de Steiner

Uma árvore geradora mínima é uma versão simplificada do problema da árvore de Steiner, onde o objetivo também é encontrar uma árvore mínima, porém a inclusão de vértices auxiliares não são permitidos.

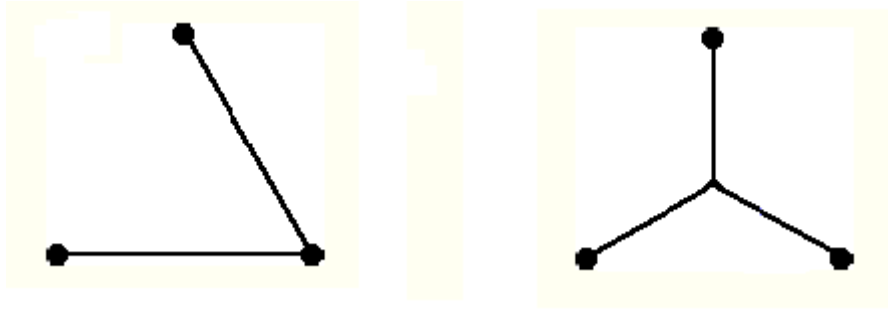


Figura 5: Comparação entre árvore geradora mínima e árvore de Steiner

A razão entre o tamanho da árvore de steiner e a árvore geradora mínima é conhecida como razão de Steiner.

Em 1992 Du e Hwang [3] provaram que o valor da razão de Steiner é de $\frac{\sqrt{3}}{2}$. Ou seja, a árvore de Steiner euclidiana é no máximo cerca de 13,4% menor que a árvore geradora mínima. Então para aplicações de custos elevados, essa possível economia é considerável.

2.5 Árvore de steiner cheia

As árvores de Steiner mínima(ASM) seguem algumas propriedades [8]:

- Pontos de Steiner incidem exatamente três arestas com ângulo de 120° entre si.
- Árvores de Steiner mínimas com n terminais tem no máximo $n - 2$ pontos de Steiner.
- ASMs são uniões de árvores Steiner cheias (Figura 6).

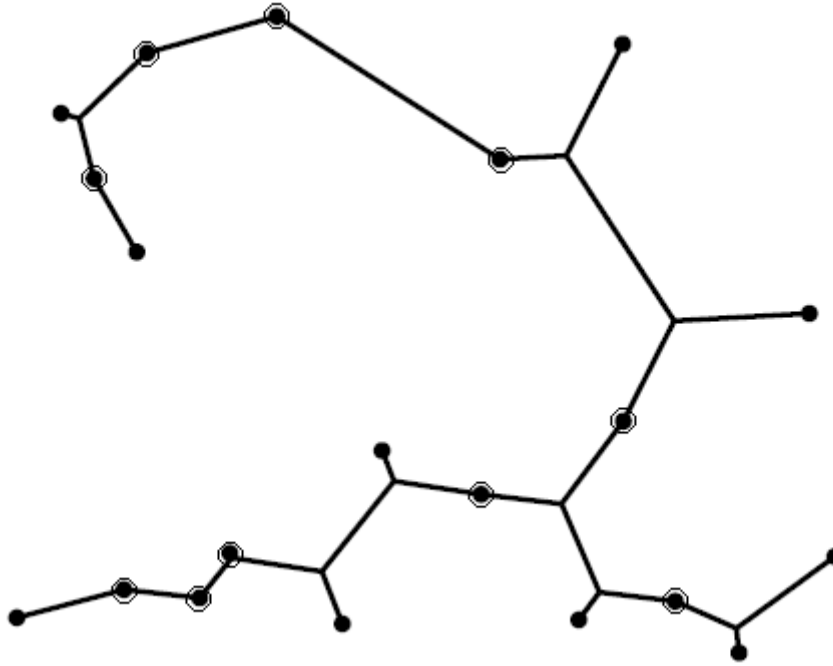


Figura 6: Combinação de árvores de Steiner cheias ligadas nos vértices destacados

Primeiramente, se um ponto de Steiner tem apenas uma aresta, então ele é uma folha e pode ser excluído e consequentemente diminuindo o tamanho da árvore. E o ponto de Steiner também não pode ter grau de tamanho dois, pois bastaria retirar esse ponto e ligar os vértices adjacentes a esse ponto por uma reta, diminuindo o tamanho da árvore. E arestas não podem incidir no mesmo ponto em um ângulo menor que 120° para satisfazer a condição angular. Portanto os pontos de Steiner são conectados por exatamente três arestas.

Árvores de Steiner cheia são árvores de Steiner que possuem exatamente dois pontos de Steiner a menos do que terminais. Ou seja, possuem o número máximo de pontos de Steiner permitidos.

Propriedades das árvores de Steiner cheias(ASC):

- Terminais possuem grau um.
- Se dois ASC compartilham um terminal z , então as duas arestas incidentes a z (uma de cada ASC) fazem pelo menos um ângulo de 120° entre si. Ou seja, um terminal não pode estar em mais do que três ASCs.

3 Algoritmo Exato

Os algoritmos atuais para solucionar o problema da árvore de Steiner Euclidiana seguem basicamente a mesma estrutura onde o problema é dividido em duas fases, a geração das árvores de Steiner cheias e a sua concatenação para encontrar a árvore de Steiner mínima(ASM).

3.1 Geração de ASC

3.1.1 Algoritmo de Melzak

A primeira solução finita para a árvore de Steiner euclidiana foi proposta em 1961 por Melzak [5] que dado um conjunto de vértices n o algoritmo consiste em gerar todas as possíveis topologias para o cada subconjunto de vértices e então encontrar a ASC de cada topologia se possível. As ASCs então são concatenadas para se encontrar a ASM.

Para encontrar os menores ASCs é necessário considerar todas as possíveis topologias cheias, ou seja, todas as possíveis maneiras de se conectar k terminais e $k - 2$ pontos de steiner onde o grau de condição seja satisfeita (terminais tem grau 1 e pontos de Steiner tem grau 3).

Por exemplo, para $k = 4$, existem três diferentes topologias cheias (Figura 7).

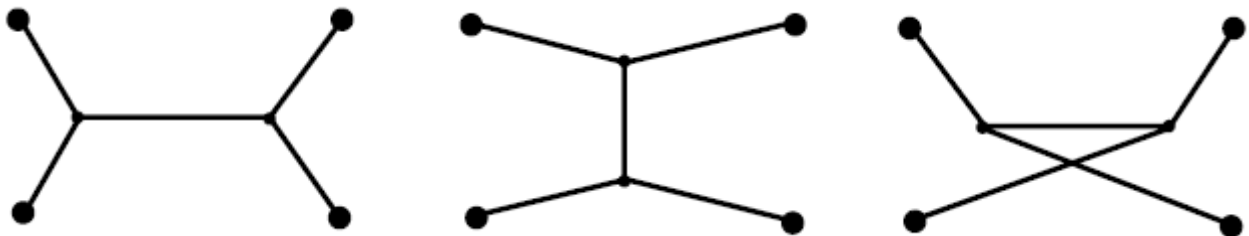


Figura 7: Topologias possíveis para quatro vértices

Para encontrar as ASC de uma topologia cheia o algoritmo de Melzak é dividido em duas fases: divisão e reconstrução.

3.1.1.1 Fase de divisão

Dado um conjunto de vértices n e sua respectiva topologia, consideramos dois terminais adjacentes a um ponto de steiner dessa topologia. Então substituímos esses dois pontos pelo seu ponto equilátero correspondente. Note que é possível duas direções para se criar o ponto equilátero. Essa etapa é executada até o conjunto ser reduzido a apenas dois vértices.

3.1.1.2 Fase de reconstrução

Inicialmente os dois vértices restantes do conjunto n após a fase de divisão são conectados por uma reta. Então o ponto equilátero é substituído pelos seus respectivos pontos iniciais excluídos na fase de divisão.

No ponto em que a reta criada anteriormente interceptar o arco dos dos vértices originais do ponto equilátero é inserido o ponto de Steiner, caso contrário essa instância do problema é descartada. Se essa condição for satisfeita os dois vértices originais ao ponto equilátero e o vértice que fazia parte da reta com o ponto equilátero são ligados ao ponto de Steiner. Essa etapa é executada recursivamente para todos os pontos equiláteros criados na fase de divisão até que se obtenha a árvore com todos os vértices do conjunto n ou até que não seja possível localizar o ponto de Steiner.

3.1.1.3 Exemplo de execução

Para entendermos melhor o algoritmo, vamos considerar um problema com quatro terminais v_1, v_2, v_3 e v_4 e sua respectiva topologia [6] (Figura 8).

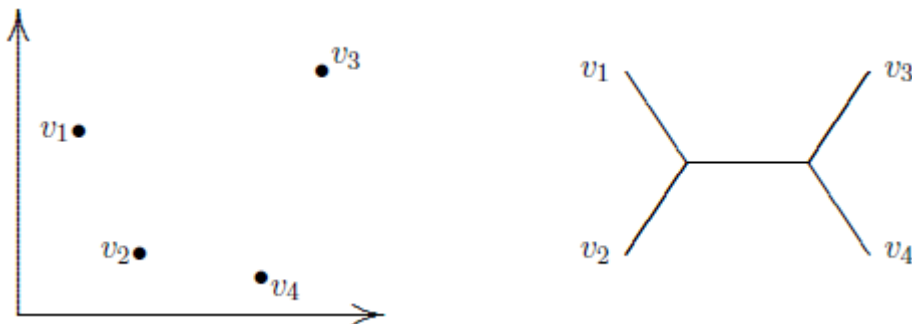


Figura 8: Exemplo do algoritmo de Melzak

No primeiro passo vamos considerar os dois vértices v_1 e v_2 adjacentes a um mesmo ponto de steiner. E substituímos esses pontos pelo ponto equilátero v_5 . Na figura x podemos ver as duas possibilidades de gerar o ponto equilátero e sua topologia resultante.

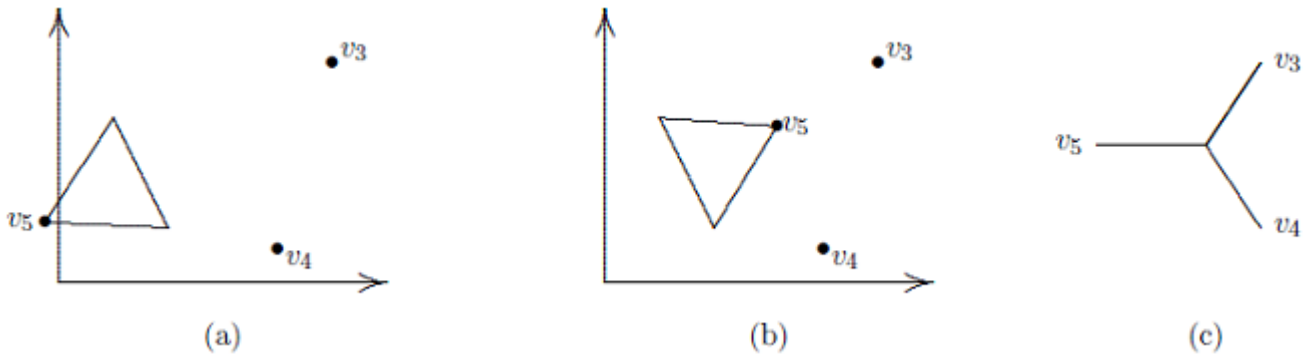


Figura 9: Algoritmo de Melzak Passo 1

Em seguida repetimos o procedimento para cada um dos dois subproblemas gerados. Vamos substituir os dois vértices v_5 e v_4 adjacentes ao mesmo ponto de Steiner, pelas duas possibilidades de ponto equilátero v_6 para cada um dos subproblemas.

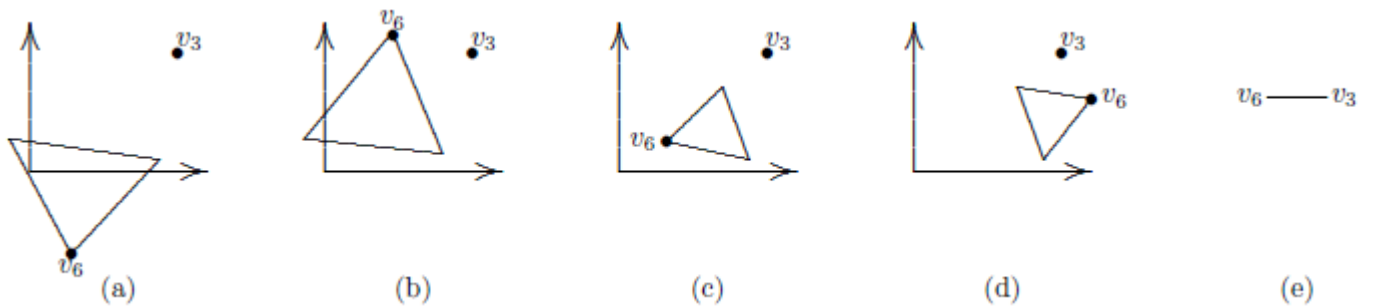


Figura 10: Algoritmo de Melzak Passo 2

Para a próxima etapa utilizaremos apenas o primeiro subproblemas da figura 10(a) e iniciaremos a fase de reconstrução ligando os dois vértices restantes com uma reta conforme a figura 11.

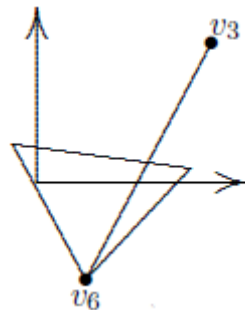


Figura 11: Algoritmo de Melzak Passo 3

Inserimos os dois últimos vértices removidos v_4 e v_5 , que foram substituídos pelo vértice v_6 na árvore para então verificar se a reta de v_6 a v_3 intersecta o arco v_4v_5 . No nosso caso

essa intersecção existe e é nosso ponto de Steiner, caso o contrário outro subproblema deveria ser verificado. Ligamos esse ponto de Steiner aos vértices v_4 e v_5 e removemos a reta de v_6 ao ponto de Steiner, resultando na figura 12(b).

Repetimos o procedimento para os dois primeiros vértices removidos v_1 e v_2 para obter a árvore de Steiner (Figura 12(d)) da topologia.

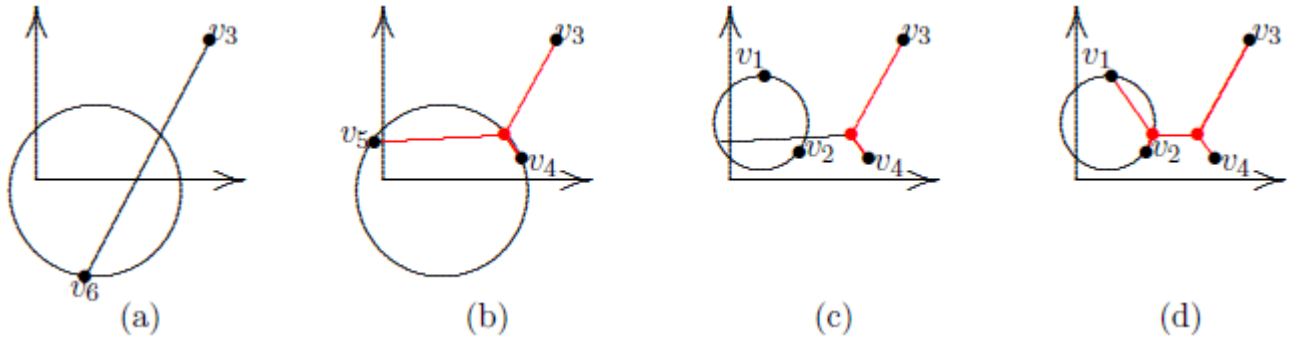


Figura 12: Algoritmo de Melzak Passo 4

A abordagem de Melzak resolve instâncias pequenas do problema devido a quantidade de topologias existentes dada pela equação:

$$f(k) = \frac{(2k-4)!}{2^{k-2}(k-2)!}$$

Tabela 1: Função $f(n)$

n	f(n)
3	1
4	3
5	15
6	105
7	945
8	10.395
9	135.135
10	2.027.025
11	34.459.425
12	654.729.075

Como mostrado na tabela 1 a execução do algoritmo se torna inviável para instâncias grandes do problema mesmo com otimizações existentes como escolher o único ponto equilátero

dentre as duas alternativas que pode fazer parte da solução para minimizar a quantidade de operações.

3.1.2 Winter

Winter [7] propôs uma abordagem diferente ao método de Melzak. Ele busca gerar as ASCs em vários subconjuntos. O objetivo é realizar testes nas subtopologias a fim de verificar se essas podem fazer parte de alguma topologia maior.

Um dos possíveis testes é o *lune pruning*. Dado dois vértices, ligamos esses vértices por uma reta e criamos uma circunferência onde o raio equivale a distância entre os dois vértices e o centro de cada circunferência é dada por cada um dos vértices (Figura 13). A área de intersecção das circunferências é chamada de *lune*. Para que a reta entre os dois vértices faça parte de uma ASM, é necessário que não exista nenhum outro vértice na *lune*.

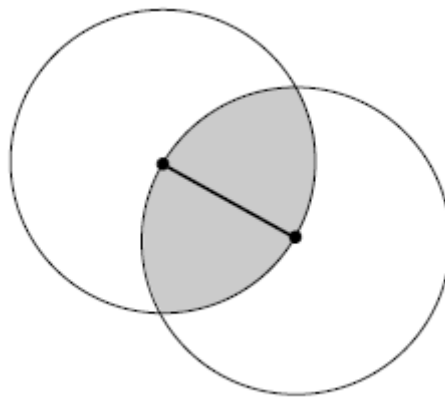


Figura 13: Lune

Com isso é possível verificar se dois vértices que possuem um ponto de Steiner adjacente podem aparecer em uma ASM. Se o ponto de Steiner não pode ser inserido fora da área, então todas as topologias que possuem essa subtopologias podem ser descartadas.

Outros testes mais complexos podem ser aplicados nessa fase para se reduzir a quantidade de ASCs gerados e consequentemente diminuindo o tempo de processamento da fase de concatenação. Permitindo resolver instâncias maiores do problema.

Na figura 14 temos todas as ASCs geradas pelo geosteiner3.1 para a instância do problema das capitais dos estados do Brasil. Essas são ASCs que restaram depois de serem processadas por algoritmos de otimizações. E então concatenadas para se obter a solução.

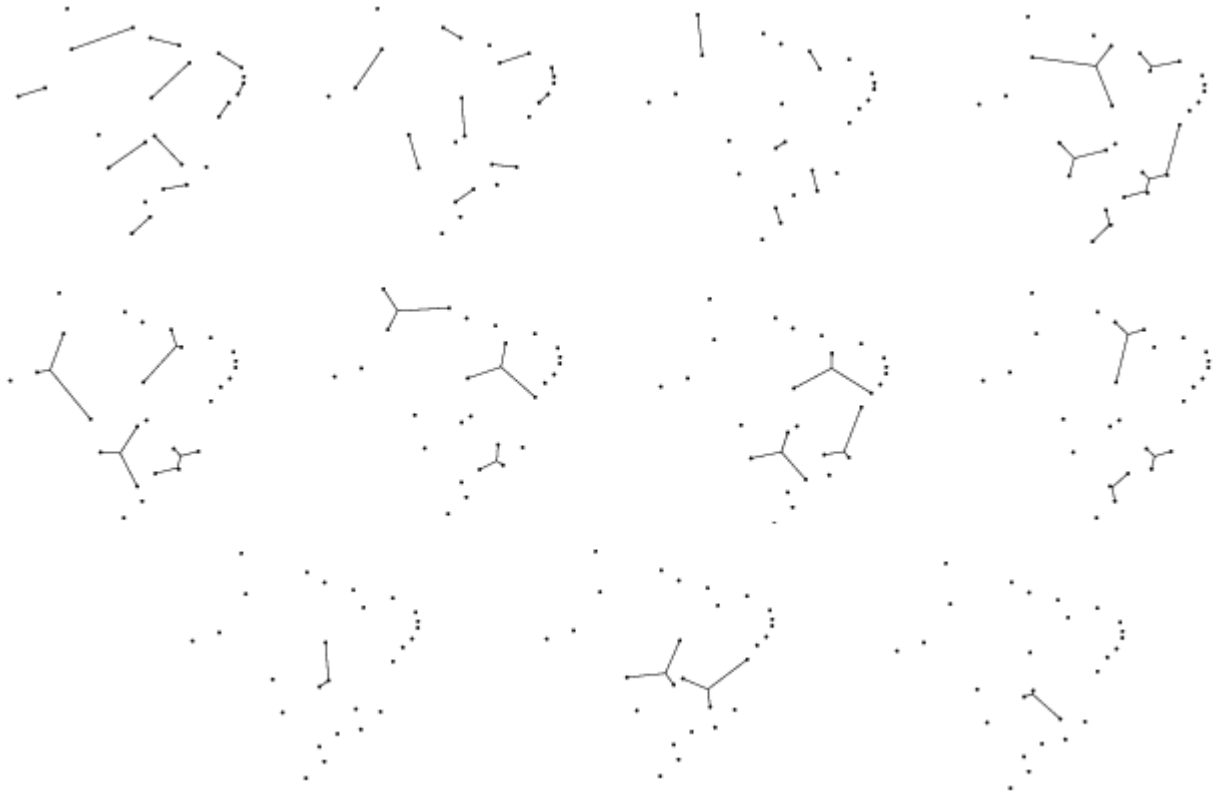


Figura 14: ASCs utilizadas para a solução da figura 1

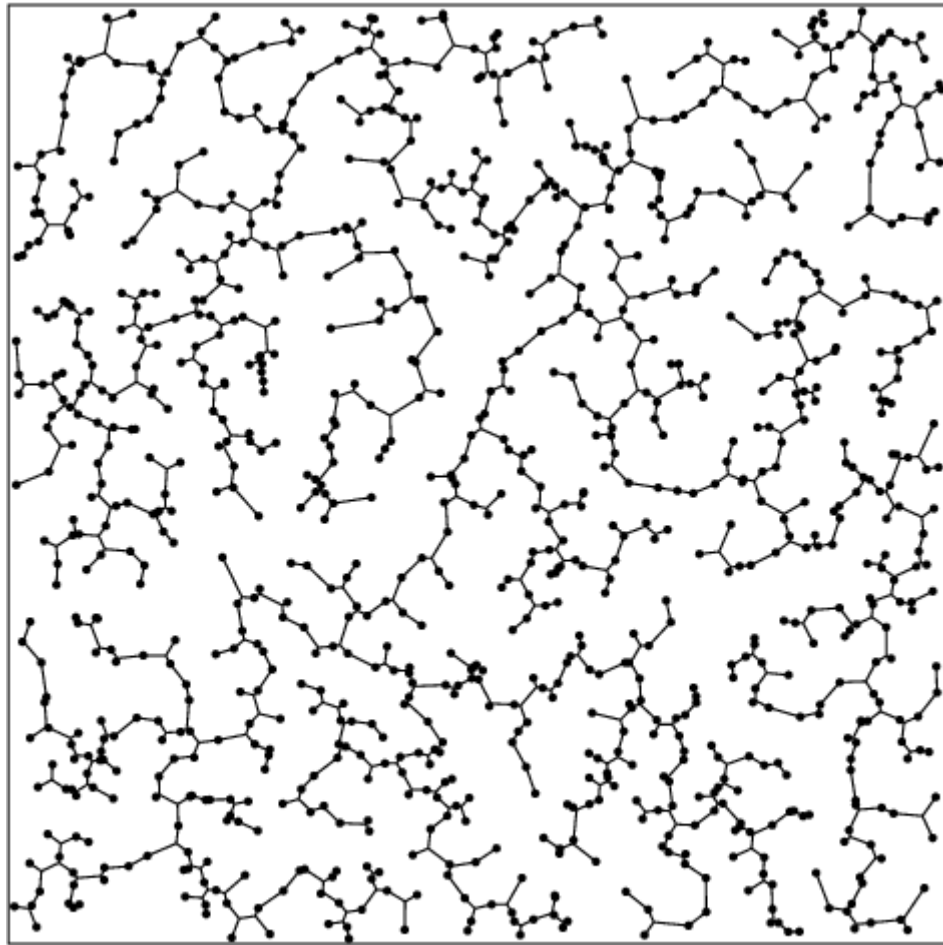
3.2 Concatenação

Um dos primeiros algoritmos utilizados para se concatenar as ASCs e encontrar a ASM foi o algoritmo de backtracking. As ASCs são combinadas recursivamente até que se encontre uma árvore que tenha todos os vértices ou que não seja possível continuar (um ciclo é formado por exemplo). Nesse caso o backtracking ocorre e outra ASC é considerado.

Otimizações iniciais são executados para verificar se cada ASC pode fazer parte de uma ASM. É verificado se cada ASC possui pelo menos um vértice em comum com qualquer uma das outras ASCs existentes. Ou seja, se a união da ASC analisada com todas as outras ASCs é nula. Então essa ASC é descartada antes de se iniciar a fase de concatenação.

Noções de matriz de compatibilidade introduzidas por Cockayne e Hewgill [1] também são utilizadas durante a concatenação. Por exemplo, duas ASCs só podem fazer parte da mesma árvore de Steiner se possuem no máximo um vértice em comum, para não criar ciclo. E o ângulo do vértice que conecta duas ASCs não pode ser maior que 120° . Além de outros testes que podem ser realizados.

A figura 15 mostra uma árvore de Steiner com mil pontos gerado pelo geosteiner3.1.



Steiner Minimal Tree: 1000 points, length = 200379.7882628364, 196.21 seconds

Figura 15: Execução do geosteiner3.1 para uma instância grande do problema

4 Conclusão

O problema da árvore de Steiner Euclidiana no plano teve muita atenção nos últimos anos e com a colaboração de diversos autores foi possível encontrar um algoritmo que pudesse resolver grandes instâncias do problema.

Diversas abordagens foram propostos para resolver o problema e soluções mais atuais do problema não foram abordados em detalhes neste trabalho e são referência para estudos futuros. Bem como outras variações do problema da árvore de Steiner como a árvore de Steiner euclidiana no espaço tridimensional que vem sendo abordado mais recentemente e inclusões de obstáculos para a construção da árvore.

Referências

- [1] COCKAYNE, E. J., AND HEWGILL, D. E. Exact computation of steiner minimal trees in the plane. *Information Processing Letters* (1986).
- [2] D. M. WARME, P. W., AND ZACHARIASEN, M. Exact algorithms for plane steiner tree problems: A computational study. Tech. rep., University of Copenhagen, Denmark, April 1998.
- [3] F. K. HWANG, D. S. R., AND WINTER, P. *The Steiner Tree Problem*. Elsevier Science Publishers B. V., Netherlands, 1992.
- [4] M. R. GAREY, R. L. G., AND JOHNSON, D. S. The complexity of computing steiner minimal trees. *SIAM J. APPL. MATH* 32, 4 (June 1977).
- [5] MELZAK, Z. A. On the problem of steiner. *Canad. Math. Bull.* 4(2) (1961), 143–148.
- [6] SOOTHILL, G. The euclidean steiner problem, April 2010.
- [7] WINTER, P. An algorithm for the steiner problem in the euclidean plane. *Networks* 15, 3 (1985), 323–345.
- [8] WINTER, P., AND ZACHARIASEN, M. Large euclidean steiner minimum trees in an hour, October 1996.