

Um algoritmo Branch and Bound exato para o problema da árvore geradora de dilatação mínima

Patrícia Fernanda Hongo RA103715

27 de maio de 2015

1 Introdução

Um problema clássico em Teoria dos Grafos é o de encontrar redes que otimizem certas propriedades geométricas. Este trabalho trata do problema de encontrar uma árvore geradora de dilatação mínima para um grafo geométrico, problema conhecido como Minimum Dilation Spanning Tree Problem – MDSTP. O conceito de dilatação foi introduzido por Chew em 1986 [8], e teve sua definição formal para grafos arbitrários introduzida por Peleg e Ullman [12]. Atualmente, resolvedores no estado da arte são capazes de produzir soluções exatas para instâncias de até 20 vértices [3].

Uma motivação para o problema está na construção de uma malha de estradas que minimize a razão entre a distância entre duas cidades na malha e sua distância euclidiana. O problema também está presente em aplicações de várias áreas, incluindo robótica, design de circuitos integrados, entre outros [11, 1].

O trabalho está estruturado da seguinte forma: a Seção 2 apresenta uma descrição formal do problema. As Seções 3 e 4 apresentam problemas relacionados e o atual estado da arte para o MDSTP. As Seções 5 e 6 apresentam as dificuldades inerentes à busca de métodos para sua resolução exata e propostas de abordagens usando o paradigma branch and bound.

2 Descrição do problema

Um grafo $G = (V, E)$ é um par ordenado, onde V é um conjunto não vazio de elementos denominados vértices e E é um conjunto de pares de vértices. Um grafo simples é tal que há no máximo uma aresta entre dois vértices, e todas as arestas são incidentes a dois vértices distintos.

Define-se um caminho entre dois vértices u e v como sendo uma sequência $(x_1, x_2, \dots, x_n)$ de vértices distintos tal que $u = x_1$, $v = x_n$ e x_i é adjacente a x_{i+1} , para todo $i < n$. Um caminho de menor comprimento entre u e v em G é denotado por $\pi_G(u, v)$. Um grafo é dito conexo se há um caminho entre todos os seus pares de vértices. Um grafo conexo e minimal no número de arestas é denominado uma árvore.

Considere o grafo $G' = (V', E')$ tal que $V' \subseteq V$, e os extremos das arestas de E' pertencem a V' . Dizemos que G' é um *subgrafo* de G . Se $V' = V$, G' é também um subgrafo gerador de G .

Seja $P = \{p_1, p_2, \dots, p_n\}$ um conjunto de pontos no plano no espaço euclidiano. Define-se um *grafo geométrico* $G = (V, E)$ como sendo o grafo simples e completo com conjunto de vértices $V = P$. O peso de uma aresta uv é definido como sendo a distância euclidiana d_{uv} entre os vértices u e v .

Dado um grafo geométrico G e um subgrafo $G' = (V', E')$, define-se a dilatação entre dois vértice u e v como

$$\rho_{G',G}(u, v) = \frac{|\pi_{G'}(u, v)|}{|\pi_G(u, v)|}$$

A dilatação de G' é dada por

$$\rho(G', G) = \max_{u,v \in V, u \neq v} \frac{|\pi_{G'}(u, v)|}{|\pi_G(u, v)|}$$

O problema da árvore geradora de dilatação mínima consiste em encontrar uma subárvore geradora G' de G que minimize $\rho(G', G)$ [13]. A dilatação também é conhecida como *stretch factor* ou *distorção* de G [11]. As Figuras 1 e 2 mostram duas árvores geradoras e suas respectivas dilatações¹.

3 Problemas similares

O conceito de dilatação abre uma gama de problemas variados, cuja investigação pode levar a *insights* importantes para a resolução do MDSTP. A seguir, apresentamos brevemente três problemas relacionados.

3.1 Árvore Geradora Mínima

O problema da árvore Geradora Mínima (MST em inglês) é um dos problemas clássicos em Teoria dos Grafos e consiste em encontrar uma árvore geradora que minimize a soma dos pesos das arestas. Apesar de ser um dos problemas mais básicos da área, ainda se desconhece se existe um algoritmo determinístico linear para o MST. O algoritmo mais eficiente conhecido foi encontrado por Chazelle e tem complexidade $O(m \log(\alpha))$, onde α é o inverso da função de Ackerman [6]. Para o problema da árvore geradora mínima euclidiana, entretanto, é conhecida a cota inferior $\Omega(n \log n)$, obtida a partir da redução do problema do par de pontos mais próximo, assim como um algoritmo $O(n \log n)$ para construção da árvore utilizando diagrama de Voronoi [14].

Naturalmente, surge a questão do quão boa é a dilatação de árvore geradora mínima. Infelizmente, a qualidade da aproximação provida pela árvore geradora mínima se mostra

¹Imagens obtidas em [9]

muito inferior àquela produzida por outras heurísticas [9], conforme pode-se observar nas Figuras 1 e 2.



Figura 1: Árvore geradora mínima para uma instância de 300 pontos, dilatação 18.053

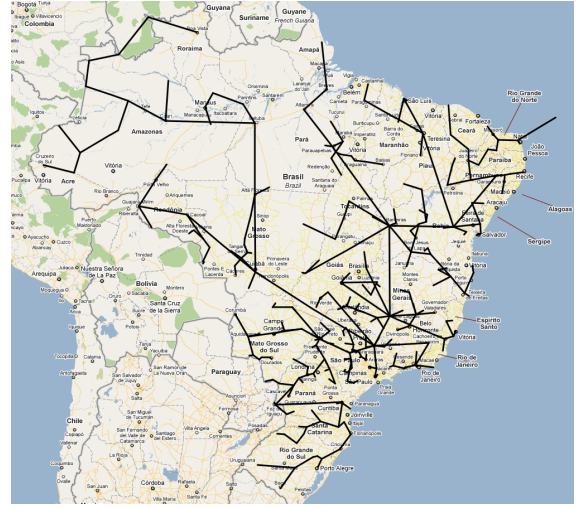


Figura 2: Solução obtida pela meta-heurística GRASP, com dilatação 10.332

3.2 t -spanners

Um conceito relacionado com a dilatação de um grafo é o de t -spanner. Um t -spanner de G é um subgrafo G' de G com dilatação $\rho(G', G) \leq t$. Grande parte dos trabalhos experimentais no problema de dilatação buscam encontrar t -spanners que minimizem o peso total das arestas, grau dos vértices, diâmetro, número de arestas, ou número de cruzamentos, dado um valor de t [15, 10, 4]. Vários desses problemas de minimização são NP-difíceis [5, 4].

3.3 Triangulação de Dilatação Mínima

O problema da Triangulação de Dilatação Mínima consiste em encontrar uma triangulação planar que minimize a dilatação, dado um conjunto P de pontos no plano. A complexidade desse problema permanece em aberto, mas acredita-se que seja NP-difícil [2]. O trabalho [3] mostra que o MSTP e o problema da triangulação de dilatação mínima compartilham muitas das propriedades geométricas que permitem reduzir o problema dado um limitante superior ρ^h para o valor da dilatação.

4 Estado da arte

O problema de encontrar um árvore geradora de dilatação mínima é NP-difícil [7]. Dessa forma, resolvê-lo exige a aplicação de métodos de otimização combinatória. O primeiro algoritmo exato para o problema foi proposto por [3] e se baseia no uso de meta-heurísticas

para obtenção soluções iniciais de alta qualidade, no uso do limitante primal produzido pela meta-heurística para realização de um pré-processamento geométrico, e na formulação do problema como um programa linear inteiro misto. Essa abordagem foi capaz de resolver à otimalidade instâncias de até 20 vértices. As subseções abaixo explicam brevemente como cada uma dessas etapas foi utilizada para compor um algoritmo exato para o MDSTP.

4.1 Meta-heurística GRASP

A produção de soluções iniciais de alta qualidade é um fator essencial na abordagem de [3], sendo utilizada para obter limitantes primais e reduzir o modelo MILP via pré-processamento geométrico. A meta-heurística escolhida pelos autores foi o *Greedy Randomized Adaptive Search Procedure* (GRASP).

A fase de construção do método utiliza m algoritmo de Prim randomizado. Vértices são adicionados à árvore incompleta se sua distância até a árvore se encontra no intervalo $[(1-\alpha)d_m, d_m]$, onde $\alpha \in [0, 1]$ e d_m é o custo de uma aresta que seria escolhida pelo algoritmo Prim. A cada iteração da fase de construção, é escolhido um novo valor de α no intervalo $\{0.0, 0.1, \dots, 1.0\}$.

A fase de busca local utiliza perturbações que podem ser computadas eficientemente, as quais foram nomeadas *Triangle* e *Path*. Sejam u, v, r vértices distintos de uma árvore T . Suponha que $\pi_G(u, a) \cdot (a, z) \cdot \pi_G(z, r)$ e $\pi_G(u, a) \cdot (a, z) \cdot \pi_G(z, r)$ sejam, respectivamente, os caminhos de u a r e de v a r em T . O método *Triangle* substituirá a aresta (a, z) pela aresta (a, b) , recalculando a dilatação dos $O(n^2)$ pares de vértices afetados. Como a vizinhança de *Triangle* é de apenas duas soluções, o método *Path* o repetirá em triplas de vértices consecutivos de um caminho de maior dilatação, obtendo uma vizinhança de tamanho $2k$, onde k é o comprimento do caminho.

Finalmente, é aplicada a técnica *path relinking* à um conjunto de soluções elite. Dadas duas árvores geradoras, S e S' , o método adiciona iterativamente arestas de S' a S , formando ciclos. A remoção de uma aresta do ciclo gera uma nova árvore, um passo mais próxima de S' .

4.2 Modelo MILP

A modelagem do MDSTP foi baseada no modelo *multicommodity flow network*, no qual há um fluxo unitário entre cada par distinto de vértices. Para cada aresta ij do grafo original, são criados os arcos (i, j) e (j, i) . Define-se a variável x_{ij}^{ab} como sendo valor do fluxo entre os vértices a e b que passa pelo arco (i, j) . A variável assume valor 1 se o arco faz parte do caminho entre a e b na solução ótima. A formulação utilizada pode ser vista a seguir:

$$\min \rho \tag{1}$$

$$\sum_{i \in P \setminus \{a\}} x_{ia}^{ab} - \sum_{i \in P \setminus \{a\}} x_{ai}^{ab} = -1, \quad \forall a, b \in P, a < b \tag{2}$$

$$\sum_{i \in P \setminus \{j\}} x_{ij}^{ab} - \sum_{i \in P \setminus \{j\}} x_{ji}^{ab} = 0, \quad \forall j \in P \setminus \{a, b\}, \forall a, b \in P, a < b, \quad (3)$$

$$\sum_{i \in P \setminus \{b\}} x_{ib}^{ab} - \sum_{i \in P \setminus \{b\}} x_{bi}^{ab} = 1, \quad \forall a, b \in P, a < b, \quad (4)$$

$$\sum_{(i,j) \in A} \frac{d_{ij}}{d_{ab}} x_{ij}^{ab} - \rho \leq 0, \quad \forall a, b \in P, a < b, \quad (5)$$

$$x_{ij}^{ab} + x_{ji}^{ab} - x_{ij}^{ij} \leq 0, \quad \forall \{i, j\} \in E, \forall a, b \in P, a < b, \quad (6)$$

$$\sum_{\{i,j\} \in E} x_{ij}^{ij} = |P| - 1, \quad (7)$$

$$x \in \mathbb{B}^{|A| \cdot |E|}. \quad (8)$$

Note que a formulação possui $O(n^4)$ restrições e variáveis. Como o tamanho do modelo é polinomial no tamanho da entrada, diz-se que ele é compacto. Entretanto, seu tamanho ainda é massivo e representa grande parte da dificuldade de resolução do MILP, motivo pelo qual são utilizadas rotinas de pré-processamento para redução do número de variáveis e restrições.

4.3 Pré-processamento geométrico

As rotinas de pré-processamento utilizam a dilatação ρ^h encontrada pela meta-heurística GRASP para fixar o valor variáveis x_{ij}^{ab} como sendo 1 ou 0. Dado que o valor de uma variável é fixada, essa informação pode ser propagada causando a fixação de outras variáveis do modelo. Uma restrição pode ser removida da formulação quando todas as suas variáveis são fixadas. Abaixo, são descritas as principais rotinas de redução.

Considere uma árvore geradora T , e considere o vértice u tal que $\pi_T(a, b) = \pi_T(a, u) \cup \pi_T(u, b)$, para algum par de vértices a, b . Pela desigualdade triangular:

$$\frac{d_{au} + d_{ub}}{d_{ab}} \leq \rho_T(a, b) \leq \rho^h$$

ou seja, o vértice u deve estar no interior da elipse de focos a e b e eixo maior de comprimento $\rho^h \cdot d_{ab}$. Esse teste permite a realização do que os autores chamam de bloqueio de vértices por elipse, conforme ilustrado na Figura 3².

²Imagem obtida em [9]

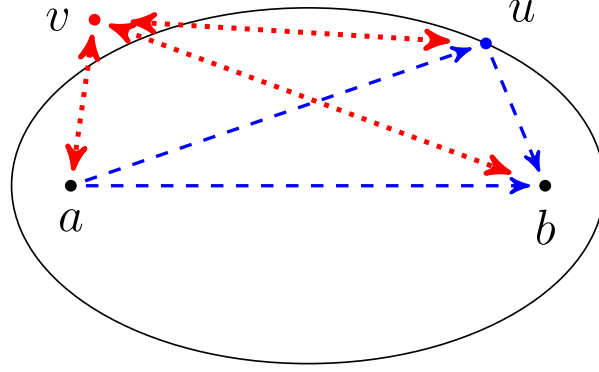


Figura 3: Bloqueio de vértice por elipse: o vértice v não pode fazer parte do caminho entre a e b .

Uma extensão dessa ideia consiste em eliminar arcos (u, v) no caminho $\pi_T(a, b)$ tais que $\frac{d_{au} + d_{uv} + d_{vb}}{d_{ab}} \geq \rho^h$. Finalmente, quando uma elipse de focos a, b não possui outros vértices em seu interior, podemos identificar a aresta ab como pertencente à solução.

5 Dificuldades do problema

Conforme visto na seção anterior, o MDSTP é NP-difícil e requer o uso de técnicas de otimização combinatória. Algoritmos no estado da arte modelam o problema como um programa linear inteiro e utilizam *solvers* comerciais para resolvê-lo, além de fazer uso de heurísticas para obtenção de soluções iniciais.

Esse problema apresenta ainda outra dificuldade: sua natureza minimax. O MDSTP busca minimizar a dilatação máxima entre todos os pares de vértices. Essa propriedade dificulta a realização de escolhas locais, visto que a adição de um vértice/aresta a uma solução parcial pode afetar a dilatação entre $O(n^2)$ pares de vértices. O efeito global que decorre de alterações locais da solução torna o cômputo da dilatação mais caro em relação a outros problemas de otimização que não são minimax.

Além disso, uma vez encontrado o par de vértices de dilatação máxima, há potencialmente inúmeras árvores cuja dilatação seja decorrente desse par de vértices, particularidade que dificulta a poda de ramos da árvore de enumeração.

6 Abordagens propostas

Atualmente, o algoritmo no estado da arte é capaz de resolver instâncias de até 20 vértices à otimalidade. O avanço para a resolução de instâncias maiores requer ou o refinamento do método ou a adoção de novas abordagens. Apresentamos a seguir uma proposta de uma nova abordagem do problema via Branch & Bound combinatório.

As componentes de um algoritmo de Branch & Bound são: a definição de um espaço de busca, a operação de ramificação, o teste de viabilidade, o cálculo dos limitantes e a escolha da ordem de busca. As subseções seguintes descrevem uma proposta para as componentes mais importantes.

6.1 Definição do espaço de busca

Seja $G = (V, E)$, $n = |V|$, $m = |E|$ um grafo geométrico completo para o qual se deseja resolver o MDSTP. Assumimos que os vértices foram previamente rotulados com inteiros de 1 a n . A representação escolhida para uma solução parcial na árvore de Branch&Bound é uma lista $L = ((v_1, v_2), (v_3, v_4), \dots, (v_k, v_{k+1}))$ de pares ordenados correspondentes a arestas do grafo. Essa lista tem a propriedade de que o subgrafo induzido pelas arestas da lista é uma árvore.

Uma lista L é uma representação válida sse pode ser construída utilizando o algoritmo abaixo:

1. $V_0 = \{1\}$ // O conjunto de vértices na árvore induzida por L
2. $C_0 = \{1\}$ // O conjunto de vértices candidatos de V aos quais ligar uma aresta
3. $L = ()$
4. For $i = 0$ to $n - 1$
 - Escolha aresta (v, u) , $v \in C_i$, $u \in V \setminus V_i$
 - $L = L + (v, u)$
 - $V_{i+1} = V_i \cup \{u\}$
 - $C_{i+1} = (C_i \setminus \{x \in C_i | x < v\}) \cup \{u\}$

Propriedade 1: A lista L produzida a cada iteração do algoritmo acima produz uma árvore.

Propriedade 2: A representação de uma árvore é única.

Prova: Sejam L_1 e L_2 duas representações diferentes para uma árvore T . Seja i o menor índice em que uma aresta $e_{i1} = (v_1, u_1) \in L_1$ difere de $e_{i2} = (v_2, u_2) \in L_2$.

Caso 1: $v_1 = v_2, u_1 \neq u_2$

S.p.g., assumamos $u_1 > u_2$. Então a aresta $(v_2, u_2) \notin L_1$ pois após inserção da aresta (v_1, u_1) , nenhuma aresta (v_1, x) , $x \leq u_1$ pode ser adicionada a L_1 . Logo, a árvore induzida por L_1 é diferente da induzida por L_2 .

Caso 2: $v_1 \neq v_2, u_1 = u_2$

Como u_1 pode se ser adjacente a apenas um vértice da subárvore induzida por V_{i-1} , a aresta $(v_1, u_1) \notin L_2$. Contradição.

Caso 3: $v_1 \neq v_2, u_1 \neq u_2$

S.p.g., assumamos $v_1 > v_2$. Pela escolha de v_1 , nenhum vértice $x < v_1$ pode se conectar a vértices de fora da árvore a partir do passo i . Logo $(v_2, u_2) \notin L_1$. Contradição.

A escolha da representação acima é motivada pela propriedade 2. Deseja-se evitar a repetição de soluções parciais, que acarretarão o recálculo de subárvores inteiras do espaço de busca. A operação de ramificação fica definida pelo passo 4 do algoritmo.

6.2 Cálculo dos limitantes

Um limitante inferior para cada solução parcial pode ser dada pela dilatação da própria solução. Um limitante superior, por sua vez, pode ser obtido a partir de uma solução viável para o MDSTP. Essa solução pode ser proveniente do uso da meta-heurística GRASP utilizada em [3]. Um novo limitante primal também pode ser obtido completando-se as árvores geradoras das soluções parciais. A decisão de como fazê-lo deve levar em consideração o tradeoff entre o custo de completar a árvore e a qualidade da solução obtida. Algumas possibilidades são:

- Utilizar a heurística GRASP em cada nó da árvore de branch and bound para completar a solução parcial. Esse método pode produzir soluções de alta qualidade, porém a um custo computacional possivelmente alto, dependendo dos parâmetros com os quais o GRASP é utilizado.
- Completar soluções parciais adicionando arestas que aumentem a dilatação ao mínimo. Essa abordagem gulosa parece promissora, mas é computacionalmente cara, visto que a adição de cada aresta pode afetar a dilatação entre $O(n^2)$ pares de vértices.
- Completar soluções parciais utilizando arestas aleatórias ou de menor custo. Essa abordagem tem baixo custo computacional, mas produz soluções de baixa qualidade.
- Calcular, em cada nó, a árvore resultante da adição de arestas da solução inicial produzida pelo GRASP à solução parcial, removendo arestas dos ciclos formados. A escolha das arestas a serem removidas também apresenta um tradeoff entre qualidade da solução e tempo: podemos escolher remover as arestas que mais contribuem com o valor da dilatação, a um custo computacional maior, ou remover arestas aleatórias, a um baixo custo.

Apresentamos acima uma proposta para abordagem do MDSTP baseada em uma intuição teórica. A modelagem do problema e escolha de métodos de cálculo de limitantes, ordem de busca do branch and bound e pré-processamento, por sua vez, devem ser guiadas por experimentos computacionais e pela comparação com os resultados previamente obtidos na literatura.

Referências

- [1] Boris Aronov, Mark de Berg, Otfried Cheong, Joachim Gudmundsson, Herman Haverkort, Michiel Smid, and Antoine Vigneron. Sparse geometric graphs with small dilation. *Computational Geometry*, 40(3):207 – 219, 2008.
- [2] Prosenjit Bose and Michiel Smid. On plane geometric spanners: A survey and open problems. *Computational Geometry*, 46(7):818–830, 2013.
- [3] Aléx Fernando Brandt. Algoritmos exatos para problemas de dilatação mínima em grafos geométricos. Master’s thesis, Universidade Estadual de Campinas, Brazil, 2014.
- [4] Leizhen Cai. Np-completeness of minimum spanner problems. *Discrete Appl. Math.*, 48(2):187–194, January 1994.
- [5] Paz Carmi and Lilach Chaitman-Yerushalmi. Minimum weight euclidean t-spanner is np-hard. *J. of Discrete Algorithms*, 22:30–42, September 2013.
- [6] Bernard Chazelle. A minimum spanning tree algorithm with inverse-ackermann type complexity. *J. ACM*, 47(6):1028–1047, November 2000.
- [7] Otfried Cheong, Herman Haverkort, and Mira Lee. Computing a minimum-dilation spanning tree is NP-hard. *Computational Geometry*, 41(3):188–205, 2008.
- [8] L. P. Chew. There is a planar graph almost as good as the complete graph. In *Proceedings of the Second Annual Symposium on Computational Geometry*, SCG ’86, pages 169–177. ACM, 1986.
- [9] Miguel Francisco Alves de Mattos Gaiowski and Cid Carvalho de Souza. Minimum dilation geometric spanning trees. In *Proceedings of XLIII Simpósio Brasileiro de Pesquisa Operacional*, pages 1824–1835, Ubatuba, SP, Brasil, 2011. SOBRAPO.
- [10] Mohammad Farshi and Joachim Gudmundsson. Experimental study of geometric t-spanners. *Journal of Experimental Algorithmics*, 14:3:1.3–3:1.39, Jan 2009.
- [11] Giri Narasimhan and Michiel Smid. *Geometric Spanner Networks*. Cambridge University Press, New York, NY, USA, 2007.
- [12] David Peleg and Jeffrey D. Ullman. An optimal synchronizer for the hypercube. In *Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing*, PODC ’87, pages 77–85, New York, NY, USA, 1987. ACM.
- [13] Schäffer A.A. Peleg, D. Graph spanners. *J. of Graph Theory*, 13(1):99–116, 1989.
- [14] Michael Ian Shamos and Dan Hoey. Closest-point problems. In *Proceedings of the 16th Annual Symposium on Foundations of Computer Science*, SFCS ’75, pages 151–162, Washington, DC, USA, 1975. IEEE Computer Society.

- [15] Mikkel Sigurd and Martin Zachariasen. Construction of minimum-weight spanners. In Susanne Albers and Tomasz Radzik, editors, *Algorithms - ESA 2004*, volume 3221 of *Lecture Notes in Computer Science*, pages 797–808. Springer Berlin / Heidelberg, 2004.