
Modelos Matemáticos para o Problema da Poligonização de Área Máxima

Raí Caetano de Jesus¹

Fábio L. Usberti²

Instituto de Computação, UNICAMP, Campinas, SP

Resumo. O problema do caixeiro viajante, do ponto de vista geométrico, tem por objetivo encontrar um polígono simples sobre um dado conjunto de vértices cujo perímetro é mínimo. É possível derivar o problema de modo que o objetivo seja encontrar um polígono simples cuja área interna seja máxima: trata-se do problema de poligonização de área máxima. Este é um problema de otimização combinatória NP-difícil com aplicações práticas em segmentos como reconhecimento de padrões, reconstrução de imagens, clusterização e robótica. Este trabalho apresenta dois modelos matemáticos de programação linear inteira. Experimentos computacionais foram executados para comparar as metodologias propostas com um algoritmo $\frac{1}{2}$ -aproximado da literatura. Os modelos matemáticos propostos mostram a dificuldade de resolver de maneira exata o problema de poligonização de área máxima, encontrando soluções ótimas em uma hora somente para as instâncias de 10 pontos, em um conjunto de instâncias de até 50 pontos.

Palavras-chave. Poligonização, Área Máxima, Programação Linear Inteira, Matheuristic

1 Introdução

Dentre os problemas de otimização combinatória, o Problema do Caixeiro Viajante (*Traveling Salesman Problem - TSP*) é um dos mais investigados pela comunidade acadêmica [4]. Neste problema, dado um conjunto de cidades e o custo de viagem entre cada par de cidades, o objetivo é encontrar uma rota de comprimento mínimo que visita todas as cidades [1]. Do ponto de vista geométrico, o objetivo do TSP é encontrar um polígono simples, a partir de um dado conjunto de vértices, cujo perímetro é mínimo. Por outro lado, é possível avaliar um polígono por outra métrica: sua área interna [4]. O problema de maximizar a área de um polígono é conhecido na literatura como *Maximum Area Polygonization* (MAXAP).

O problema MAXAP apresenta aplicações no contexto de reconhecimento de padrões, reconstrução de imagens e clusterização [5]. Em particular, trata-se de um problema muito relacionado com a Triangulação de Área Máxima, que é relevante no contexto da robótica [3, 7, 8] para explorar e triangularizar uma região com um enxame de robôs.

¹raicaetanodejesus@gmail.com

²fusberti@ic.unicamp.br

São poucos trabalhos na literatura que abordam a geração de polígonos simples com área máxima. O trabalho de Taranilla, Gagliardi e Hernández [13] aborda a versão de minimização do MAXAP. Os autores apresentam um algoritmo para construir um polígono simples, baseado na heurística *Steady Growth* proposta por Auer e Held [2] para geração de polígonos simples aleatórios. Esses autores propõem três algoritmos para encontrar soluções heurísticas para o MINAP, sendo um deles a metaheurística *Ant Colony Optimization* - ACO. O trabalho mais recente é dos autores Peethambaran, Parakkat e Muthuganapathy [12], que propõem uma heurística construtiva gulosa para encontrar soluções para o MAXAP.

Este artigo apresenta duas formulações matemáticas de programação linear inteira para o MAXAP, originalmente propostas na dissertação de mestrado do autor deste trabalho [11]. Além disso, é apresentado um estudo computacional quanto à eficácia da solução exata desses modelos sobre um conjunto de instâncias. Esse estudo também compara os resultados dos modelos com os obtidos por um algoritmo $\frac{1}{2}$ -aproximado para o MAXAP [4].

2 Definições e Descrição do Problema

Um *polígono* é uma sequência ordenada de pontos $\langle p_0, p_1, \dots, p_{n-1} \rangle$ com $n \geq 3$ juntamente com o conjunto de segmentos $\langle \overline{p_0p_1}, \overline{p_1p_2}, \dots, \overline{p_{n-2}p_{n-1}} \rangle$ denominados *arestas* [13]. O ponto onde duas arestas se encontram é denominado *vértice* [10]. Um polígono é *fechado* se o primeiro e o último ponto estão conectados por um segmento de linha. Um polígono *simples* é um polígono fechado sem interseções entre arestas, exceto aquelas em pontos finais comuns de arestas consecutivas [13].

Seja P um conjunto de pontos no plano Euclidiano. Uma *poligonização simples* de P é um polígono simples cujos vértices são compostos exclusivamente por P . A menos que todos os pontos sejam colineares, um conjunto de pontos no plano sempre permite uma poligonização simples [13]. É conhecido que a quantidade de poligonizações simples para um conjunto de n pontos é exponencial em n [9]. Se os pontos estão em posição convexa, o conjunto permite uma única poligonização, a *envoltória convexa* (do inglês *convex hull*) [13]. A envoltória convexa de um conjunto de pontos P é o menor polígono convexo $CH(P)$ no qual cada ponto em P está na borda de $CH(P)$ ou em seu interior [10].

Definition 2.1. MAXAP (Maximum Area Polygonization): *Seja P um conjunto finito de pontos no plano Euclidiano. Dentre todos os polígonos simples com um conjunto de vértices P , encontre um cuja área interna seja máxima.*

Fekete [4] mostrou que o MAXAP é NP-difícil e também mostrou ser NP-difícil encontrar uma $(\frac{2}{3} + \epsilon)$ -aproximação, para qualquer $\epsilon \in (0, \frac{1}{3})$. Contudo, Fekete apresentou um algoritmo para gerar uma poligonização de um conjunto de pontos P cuja área é maior ou igual à metade da área da envoltória convexa de P . Uma vez que a área da envoltória convexa é um limitante superior para qualquer solução do MAXAP, isto produz uma $\frac{1}{2}$ -aproximação para o problema. Como a literatura não apresentou resultados experimentais referentes ao desempenho desse algoritmo aproximado, um dos objetivos deste trabalho consistiu no estudo computacional do mesmo.

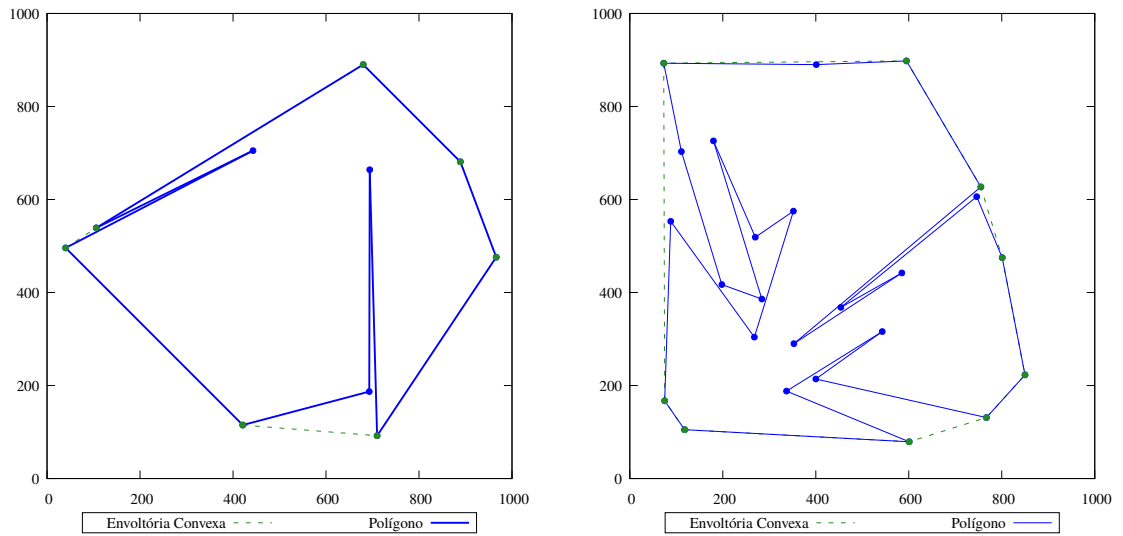


Figura 1: Exemplos de poligonizações para um conjunto de 10 pontos (esquerda) e 25 pontos (direita).

A Figura 1 apresenta soluções viáveis para o problema MAXAP para dois polígonos, de 10 e 25 vértices.

3 Modelos matemáticos

Este trabalho propõe duas formulações PLI para o MAXAP, a primeira utiliza-se da fórmula de cálculo de área de um polígono simples, e a segunda é fundamentada no cálculo de área a partir de uma triangulação de polígonos.

Modelo baseado no cálculo de área de um polígono simples – Modelo 1

$$MAX \quad \frac{1}{2} \sum_{ij \in E} (a_i b_j - a_j b_i) x_{ij} \tag{1}$$

s.a.

$$x_{ij} + x_{ji} \leq 1 \quad ij, ji \in E \tag{2}$$

$$\sum_{j=1}^{|V|} x_{ij} = 1 \quad i \in V \tag{3}$$

$$\sum_{i=1}^{|V|} x_{ij} = 1 \quad j \in V \tag{4}$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset V \tag{5}$$

$$x_{ij} + x_{uv} \leq 1 \quad ((i, j), (u, v)) \in \mathcal{L} \tag{6}$$

$$x_{ij} \in \{0, 1\} \quad ij \in E \tag{7}$$

No Modelo 1, tem-se que $a_i b_i$ e $a_j b_j$ são as coordenadas dos pontos que compõem o arco ij ; $x_{ij} = 1$ se o arco (i, j) pertence à solução e $x_{ij} = 0$, caso contrário; \mathcal{L} é a família de pares de arcos que se interceptam. A função objetivo (1) determina a área interna do polígono (solução). A restrição (2) garante a unidirecionalidade da rota, ou seja, os arcos escolhidos para compor a solução devem seguir um único sentido (horário ou anti-horário). As restrições (3) e (4) asseguram a continuidade da rota, isto é, cada vértice deve possuir apenas um arco de entrada e um arco de saída. A restrição (5) garante a eliminação de subciclos, e a restrição (6) define que não há cruzamento de arcos (polígono simples).

Em 2015, Fekete *et al.* [6] propuseram um modelo semelhante, apresentando resultados computacionais e discutindo, inclusive, melhorias que evidenciaram ganhos práticos. Vale ressaltar que, apesar dos autores apresentarem esta formulação recentemente, o modelo proposto neste trabalho foi desenvolvido de maneira independente.

Modelo baseado em triangulação – Modelo 2

Um triângulo vazio é um triângulo, representado pela tripla ijk , com vértices em P e nenhum outro ponto de P em seu interior. Seja P um conjunto de pontos no plano, $\Delta(P)$ o conjunto de triângulos vazios sobre P , $L^+(ij)$ e $L^-(ij)$ são os dois semi-planos definidos pela reta contendo o segmento ij .

$$MAX \quad \sum_{ijk \in \Delta(P)} a_{ijk} t_{ijk} \tag{8}$$

s.a.

$$x_{ij} \geq \sum_{\substack{ijk \in \Delta(P) \\ ijk \subset L^+(ij)}} t_{ijk} - \sum_{\substack{ijk \in \Delta(P) \\ ijk \subset L^-(ij)}} t_{ijk} \quad ij \in E \setminus E_H \tag{9}$$

$$x_{ij} \geq \sum_{\substack{ijk \in \Delta(P) \\ ijk \subset L^-(ij)}} t_{ijk} - \sum_{\substack{ijk \in \Delta(P) \\ ijk \subset L^+(ij)}} t_{ijk} \quad ij \in E \setminus E_H \tag{10}$$

$$x_{ij} \leq \sum_{ijk \in \Delta(P)} t_{ijk} \leq 2 - x_{ij} \quad ij \in E \setminus E_H \tag{11}$$

$$\sum_{ijk \in \Delta(P)} t_{ijk} = x_{ij} \quad ij \in E_H \tag{12}$$

$$t_{ijk} + t_{i'j'k'} \leq 1 \quad (ijk, i'j'k') \in T \tag{13}$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset P \tag{14}$$

$$\sum_{ij \in \delta(i)} x_{ij} = 2 \quad i \in P \tag{15}$$

$$t_{ijk} \in \{0, 1\} \quad ijk \in \Delta(P) \tag{16}$$

$$x_{ij} \in \{0, 1\} \quad ij \in E \tag{17}$$

No Modelo 2, tem-se que a_{ijk} é a área do triângulo ijk ; $t_{ijk} = 1$ se ijk pertence à solução, $t_{ijk} = 0$ caso contrário; $x_{ij} = 1$ se a aresta (i, j) pertence à solução, $x_{ij} = 0$ caso contrário; T é a família de pares de triângulos vazios que se interceptam; $\delta(i)$ é o conjunto de arestas não orientadas incidentes ao vértice i . A função objetivo (8) determina a área do polígono somando as áreas dos triângulos vazios que compõem sua triangulação. As

restrições (9) e (10) garantem que, para cada aresta (i, j) não pertencente à envoltória convexa, se $x_{ij} = 0$, a quantidade de triângulos no semi-plano $L^+(ij)$ presentes na solução é igual à quantidade de triângulos no semi-plano $L^-(ij)$. Caso contrário, se $x_{ij} = 1$, a quantidade destes triângulos nos dois semi-planos pode diferir em uma unidade. A restrição (11) garante que, para cada aresta (i, j) não pertencente à envoltória convexa, se $x_{ij} = 1$, então a solução deve conter estritamente um triângulo $ijk \in \Delta(P)$. Caso contrário, se $x_{ij} = 0$, então a solução pode conter no máximo dois triângulos $ijk \in \Delta(P)$ e $ijk' \in \Delta(P)$ tal que $k \neq k'$. A restrição (12) assegura que se a aresta (i, j) da envoltória convexa pertence à solução, então a solução deve conter estritamente um triângulo $ijk \in \Delta(P)$. A restrição (13) garante que, para cada par de triângulos vazios que se interceptam, no máximo um deles pode estar na solução. A restrição (14) define que a solução deve conter apenas um polígono simples (restrição para eliminação de sub-ciclos). Finalmente, a restrição (15) garante que o polígono gera todos os vértices, ou seja, cada vértice deve ter estritamente duas arestas.

4 Experimentos computacionais

Todos os algoritmos foram implementados na linguagem C++. A máquina utilizada nos experimentos computacionais é composta por Processador Intel® Core i7-4771 CPU 3.5GHz, 8GB de memória RAM e sistema operacional Windows 10 64 bits. A solução dos modelos matemáticos foi realizada pelo *solver* Gurobi versão 7.5. Um conjunto de 30 instâncias, divididas em subconjuntos de 10 instâncias com 10, 25 e 50 pontos, respectivamente, com coordenadas inteiras aleatórias no intervalo $[0, 1000]$ em posição geral.

Na Tabela 1 estão sumarizados os resultados dos experimentos computacionais com os modelos matemáticos descritos propostos. A coluna 3 apresenta as áreas obtidas a partir do algoritmo $\frac{1}{2}$ -aproximado de Fekete [4]. As colunas 4-7 exibem os resultados para o modelo baseado no cálculo de área de um polígono simples (Modelo 1), e as colunas 8-11 mostram os resultados do modelo baseado em triangulação (Modelo 2). Os valores em negrito destacam as soluções ótimas.

Os resultados mostram que o algoritmo aproximado encontrou soluções primais factíveis com tempos de execução inferiores a 0,01 segundo para todas as instâncias (não reportados na Tabela 1). Os fatores de aproximação empíricos obtidos pelo algoritmo aproximado foram em média 81,6% e no mínimo 60,5%, o que representam desempenhos substancialmente melhores do fator de pior $\frac{1}{2}$.

Os resultados apresentados pelos modelos mostram a dificuldade em resolver de maneira exata o MAXAP. Foram obtidas soluções ótimas no período de uma hora somente para instâncias de 10 pontos, em um conjunto de instâncias com até 50 pontos. Para as instâncias com 25 pontos, as soluções do modelo 2 obtiveram melhores desvios de otimalidade: 8,7% em média, valor muito abaixo dos 81,37% obtidos pelo primeiro modelo (1). Ainda com relação às instâncias de 25 pontos, o modelo 2 obteve desvios de otimalidade de no máximo 12%, enquanto que o modelo 1 obteve desvios acima de 50%, ultrapassando 100% em alguns casos. Nas instâncias com 50 pontos os modelos não encontraram nenhum limitante primal ou dual dentro do limite de uma hora. Para as instâncias de 10

Tabela 1: Resultados computacionais.

Instância	A_{CH}	$\frac{1}{3}$ -Approx.	Modelo 1				Modelo 2			
		L_P	L_P	L_D	GAP (%)	CPU (s)	L_P	L_D	GAP (%)	CPU (s)
10.1	433981,0	368327,0	414579,5	414579,5	0,0	0,1	414579,5	414579,5	0,0	0,1
10.2	342695,5	196557,0	288219,5	288219,5	0,0	0,0	288219,5	288219,5	0,0	0,2
10.3	348374,0	284373,0	341230,5	341230,5	0,0	0,0	341230,5	341230,5	0,0	0,0
10.4	490496,5	448410,0	465488,0	465488,0	0,0	0,0	465488,0	465488,0	0,0	0,0
10.5	493022,0	415376,5	470497,0	470497,0	0,0	0,0	470497,0	470497,0	0,0	0,0
10.6	318446,5	229753,5	265929,0	265929,0	0,0	0,0	265929,0	265929,0	0,0	0,1
10.7	485913,00	319993,0	406472,5	406472,5	0,0	0,0	406472,5	406472,5	0,0	0,1
10.8	407827,5	242632,0	401340,5	401340,5	0,0	0,0	401340,5	401340,5	0,0	0,0
10.9	445677,0	284348,5	398426,0	398426,0	0,0	0,0	398426,0	398426,0	0,0	0,0
10.10	246394,5	225369,0	238795,0	238795,0	0,0	0,0	238795,0	238795,0	0,0	0,1
25.1	548433,0	381291,0	476788,0	812348,5	70,3	3600,0	480624,5	538177,0	12,0	3600,0
25.2	821460,50	456812,5	687706,5	1246424,3	81,2	3600,0	743503,5	811004,1	9,0	3600,0
25.3	731502,00	522522,5	651016,0	1248515,2	91,7	3600,0	680270,0	727763,2	7,0	3600,0
25.4	669933,50	416112,0	526856,5	1083443,1	105,6	3600,0	590171,5	652797,6	11,0	3600,0
25.5	739425,50	411559,0	640466,0	1010784,4	57,8	3600,0	649732,0	716571,3	10,0	3600,0
25.6	767510,00	564014,0	679129,5	1103089,4	62,4	3600,0	711222,5	749342,3	5,0	3600,0
25.7	759767,00	541152,0	696951,5	1128256,1	61,8	3600,0	701971,0	748215,4	7,0	3600,0
25.8	668515,00	487048,0	565966,0	1045346,5	84,7	3600,0	608166,0	650999,7	7,0	3600,0
25.9	748495,50	398391,5	643627,5	1388115,8	115,6	3600,0	681174,0	742610,0	9,0	3600,0
25.10	639718,50	423645,5	539330,5	982916,6	82,2	3600,0	565254,5	620736,0	10,0	3600,0
50.1	842095,0	494204,0	—	—	—	3600,0	—	—	—	3600,0
50.2	772040,5	420149,5	—	—	—	3600,0	—	—	—	3600,0
50.3	861561,5	521540,0	—	—	—	3600,0	—	—	—	3600,0
50.4	786575,5	551868,5	—	—	—	3600,0	—	—	—	3600,0
50.5	671751,5	413532,5	—	—	—	3600,0	—	—	—	3600,0
50.6	832342,0	433834,5	—	—	—	3600,0	—	—	—	3600,0
50.7	841551,5	481641,0	—	—	—	3600,0	—	—	—	3600,0
50.8	804230,5	450032,0	—	—	—	3600,0	—	—	—	3600,0
50.9	761448,0	419313,5	—	—	—	3600,0	—	—	—	3600,0
50.10	857377,0	480593,5	—	—	—	3600,0	—	—	—	3600,0

A_{CH} : área da envoltória convexa, L_P : limitante primal, L_D : limitante dual, $GAP = 100 * \frac{(L_D - L_P)}{L_P}$: desvio de otimalidade, CPU: tempo em segundos gasto

pontos, para as quais soluções ótimas foram determinadas, é possível verificar que as áreas máximas utilizaram em média 92,0% e no mínimo 83,5% da área da envoltória convexa.

5 Comentários Finais

A contribuição principal deste trabalho consiste no avanço do estado-da-arte no que se refere à formulação matemática do MAXAP e de metodologias exatas para sua solução. As metodologias propostas neste trabalho têm o potencial de serem aplicadas no aperfeiçoamento de soluções envolvendo reconhecimento de padrões, reconstrução de imagens, clusterização e robótica. Além disso, o conjunto de instâncias gerado neste trabalho³ torna-se um *benchmark* para avaliar e comparar outras metodologias para o MAXAP. Sugere-se como trabalho futuro realizar um estudo poliédrico dos dois modelos propostos, propondo desigualdades válidas fortes. Além disso, uma lacuna no estudo do MAXAP diz respeito às meta-heurísticas para a solução de instâncias de maior porte.

Agradecimentos

Os autores agradecem o apoio do FAEPEX e CNPq (proc. 307472/2015-9).

³Disponível em: <http://www.ic.unicamp.br/~fusberti/problems/maxarea/>.

Referências

- [1] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook. *The Traveling Salesman Problem: A Computational Study (Princeton Series in Applied Mathematics)*. Princeton University Press, 2007.
- [2] T. Auer and M. Held. Heuristics for the generation of random polygons. In *Proceedings of the 8th Canadian Conference on Computational Geometry (CCCG'96)*, pages 38–43. CCCG, 1996.
- [3] A. Becker, S. P. Fekete, A. Krölller, S. K. Lee, J. McLurkin, and C. Schmidt. Triangulating unknown environments using robot swarms. In *Proceedings of the twenty-ninth annual symposium on Computational geometry*, pages 345–346. ACM, 2013.
- [4] S. P. Fekete. Geometry and travelling salesman problem. Tese de Doutorado, Universidade de Waterloo, 1992.
- [5] S. P. Fekete. On simple polygonizations with optimal area. *Discrete & Computational Geometry*, 23(1):73–110, 2000.
- [6] S. P. Fekete, S. Friedrichs, M. Hemmer, M. Papenberg, A. Schmidt, and J. Troegel. Area- and boundary-optimal polygonalization of planar point sets. In *31st European Workshop on Computational Geometry*, pages 133–136, 2015.
- [7] S. P. Fekete, T. Kamphans, A. Krölller, J. S. B. Mitchell, and C. Schmidt. Exploring and triangulating a region by a swarm of robots. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 206–217. Springer, 2011.
- [8] S. P. Fekete, S. Rex, and C. Schmidt. Online exploration and triangulation in orthogonal polygonal regions. In *WALCOM: Algorithms and Computation*, pages 29–40. Springer, 2013.
- [9] A. García and J. Tejel. A lower bound for the number of polygonizations of n points in the plane. *Ars Combinatoria*, 49:3–19, 1998.
- [10] S. Halim, F. Halim, S. S. Skiena, and M. A. Revilla. *Competitive Programming 3*. Lulu Independent Publish, 2013.
- [11] R. C. Jesus. Formulações e algoritmos para o problema da poligonização de área máxima. Dissertação de Mestrado, Unicamp, 2018.
- [12] J. Peethambaran, A. D. Parakkat, and R. Muthuganapathy. An empirical study on randomized optimal area polygonization of planar point sets. *Journal of Experimental Algorithmics (JEA)*, 21(1):1–10, 2016.
- [13] M. T. Taranilla, E. O. Gagliardi, and G. Hernández Peñalver. Approaching minimum area polygonization. In *XVII Congreso Argentino de Ciencias de la Computación*. Universidad Nacional de La Plata, 2011.