Matemática Universitária Nº 13, junho de 1991, 83–93

Alguns Teoremas de Existência em Geometria Computacional

Luiz Henrique de Figueiredo

1. Introdução.

O que se entende geralmente por "teorema de existência" em matemática é uma afirmação que objetos de certa classe existem sem que no entanto seja dada qualquer pista concreta de como construir um tal objeto.

Em certos casos, isso é apenas um "defeito" da demonstração, pois podem existir outras demonstrações construtivas. Em outros casos, entretanto, não se conhece nenhum objeto da classe ainda que se possa demonstrar que a classe não é vazia. (e.g., bases de **R** como espaço vetorial sobre **Q**—as chamadas **bases de Hamel**). Existem correntes filosóficas que evitam esse problema exigindo que toda prova de existência seja construtiva [1].

Entre as demonstrações construtivas, temos aquelas que reduzem o problema a encontrar a solução numa lista finita de candidatos e temos aquelas que constroem efetivamente a solução. Estritamente falando, esta é uma falsa dicotomia. Na prática, entretanto, as construções tem eficiência variável. O estudo da eficiência de algoritmos chama-se **complexidade computacional** [4]; no caso específico de algoritmos geométricos, a área chama-se **geometria computacional** [2,6].

Neste artigo, mostraremos alguns exemplos de como questões de existência ocorrem em problemas construtivos de geometria computacional, podendo até mesmo sugerir algoritmos.

Como motivação e pano de fundo, considere o problema de **interpolação de dados esparsos:** são dados pontos em uma subvariedade de um espaço euclideano e deseja-se encontrar uma aproximação poligonal para esta subvariedade que interpole esses pontos. Este é um problema de modelagem geométrica que ocorre freqüentemente em ciências aplicadas (e.g., tomografia, geologia, topografia). Em geral, assume-se algo sobre a subvariedade, como, por exemplo, ser fechada, ser gráfico, ter gênero zero (ser topologicamente uma esfera).

Antes de tentar construir uma interpolação, é natural querer saber se tais aproximações poligonais existem.

2. Existência e algoritmos para polígonos simples.

Considere a versão do problema de interpolação de dados esparsos, na qual a subvariedade é uma curva plana fechada:

PROBLEMA 1. *Encontrar um polígono simples cujos vértices são pontos dados.*

Na forma como está enunciado, o problema assume a existência de um tal polígono. Uma prova elegante dessa existência é dada pelo seguinte teorema:

TEOREMA 1: *De todos os polígonos passando por um dado conjunto de pontos, o polígono de menor perímetro é simples.*

PROVA: Quatro aplicações da desigualdade triangular mostram que a soma dos comprimentos das diagonais de um quadrilátero convexo nunca é menor que a soma de dois lados opostos. Assim, um polígono não simples não pode ter perímetro mínimo (veja Figura 1). ∎

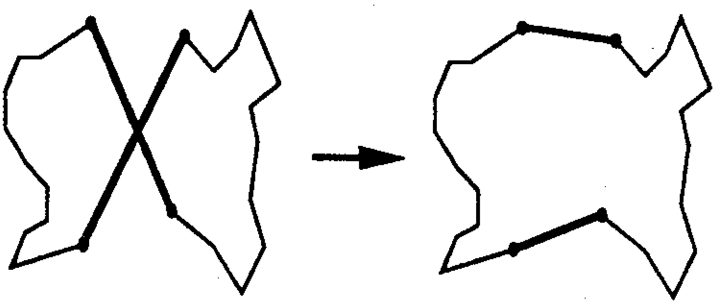

Figura 1

Como os polígonos passando por um dado conjunto de pontos podem ser identificados com as permutações circulares desse conjunto, a busca por um polígono simples fica restrita a um conjunto finito. Assim, o Teorema 1 mostra que o Problema 1 tem uma solução construtiva. Do ponto de vista de matemática clássica, o problema está totalmente resolvido. Do ponto de vista computacional, a situação é diferente: o algoritmo que o teorema fornece é exponencial e, portanto, somente útil na prática para conjuntos muito pequenos (e.g., com menos de 20 pontos). Além disso, não se conhece nenhum algoritmo que encontre o caminho poligonal mínimo em tempo polinomial (esse é o **problema do caixeiro viajante**, que desconfia-se ser exponencial já que é NP-completo [**4**]).

Existem algoritmos que produzem um polígono simples muito mais eficientemente; por exemplo, ligar os pontos em ordem polar com centro em um ponto do fecho convexo constrói um polígono simples em tempo $O(n \log n)$ [**6**,**7**] (veja Figura 2).

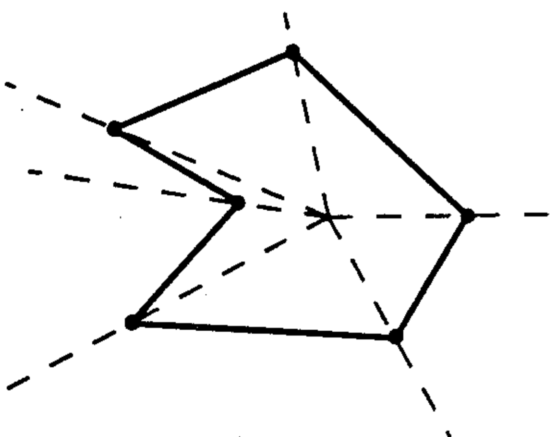

Figura 2

Considere agora o análogo espacial do Problema 1:

PROBLEMA 2. *Encontrar um poliedro de gênero zero cujos vértices são pontos dados.*

A exigência de gênero zero não é essencial e foi feita somente para reforçar a analogia topológica com o Problema 1.

Infelizmente, os algoritmos que demos para o Problema 1 não se generalizam para 3 dimensões: não temos coordenadas polares nem conhecemos nenhum resultado análogo àquele sobre as diagonais de um quadrilátero usado na prova do Teorema 1 que nos permita concluir que poliedros de área mínima não têm auto-interseções.

A procura por um teorema de existência para o Problema 2 (e talvez um algoritmo) pode começar então com algoritmos para o Problema 1, não necessariamente ótimos, mas que sejam generalizáveis para 3 dimensões.

Uma técnica que está diretamente ligada a provas por indução é a procura de algoritmos **incrementais**: tenta-se modificar uma solução parcial para atender a "chegada" de mais um ponto. No nosso caso, queremos modificar um polígono simples para incluir mais um ponto. Ordenando os pontos pelas suas abcissas, podemos organizar a chegada dos pontos de modo que o novo ponto esteja sempre fora da solução parcial. A modificação seria fácil se pudéssemos substituir exatamente uma aresta do polígono por um desvio até o novo ponto sem cortar o polígono (veja Figura 3).

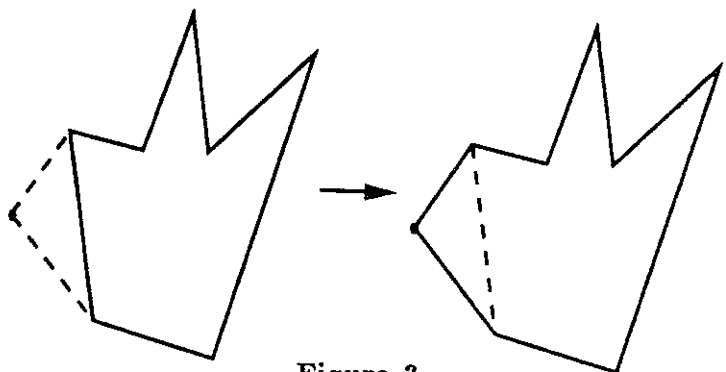

Figura 3

Isso nos leva a um novo problema de existência, que trataremos na próxima seção. Por enquanto, note que esse algoritmo tem uma chance de ser generalizado, ainda que tenhamos que provar resultados análogos para 3 dimensões.

3. Um problema de visibilidade.

O problema de existência a que chegamos na seção anterior pode ser descrito em termos de visibilidade:

PROBLEMA 3. *Dado um polígono simples e um ponto exterior, encontrar uma aresta do polígono que seja totalmente visível a partir desse ponto.*

Na generalidade com que está enunciado, o Problema 3 não tem solução pois existem casos nos quais nenhuma aresta é totalmente visível (veja Figura 4).

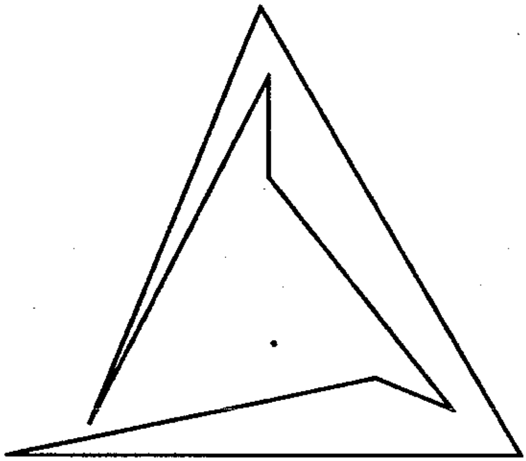

Figura 4

Entretanto, o candidato a algoritmo proposto na seção anterior não precisa de toda essa generalidade pois podemos organizar a chegada dos pontos de modo que o novo ponto esteja sempre à esquerda da solução parcial. Neste caso, o novo ponto está **longe** do polígono, no sentido que existe uma reta que separa o ponto do polígono.

O problema que realmente precisamos resolver é:

PROBLEMA 4. *Encontrar uma aresta que seja totalmente visível a partir de um ponto longe de um polígono simples.*

Nesse momento, a hipótese de termos um *polígono* parece supérflua e tentamos resolver um problema mais geral:

PROBLEMA 5. *Dado um conjunto de segmentos de reta que não se interceptam a não ser possivelmente nos extremos e um ponto longe desse conjunto, encontrar um segmento que seja totalmente visível a partir desse ponto.*

Para resolver esse problema, assim como outros problemas de visibilidade, é útil considerar o **grafo de visibilidade** associado

[5]. Em contraste com esses outros problemas, no nosso caso temos um grafo orientado. Explicitamente, temos um grafo G cujos vértices são os segmentos dados e cujas arestas são de A para B sse A bloqueia B. Em termos desse grafo, um segmento totalmente visível corresponde a um vértice ao qual não chega nenhuma aresta. Tais vértices são chamados **fontes** e sempre existem em grafos **acíclicos** (sem ciclos) [3].

Dado um segmento A e um ponto p na projeção $\overline{A}$ de A na reta separatriz, denotaremos por $p(A)$ a distância do ponto de visão O ao ponto onde a semi-reta $\overrightarrow{Op}$ encontra A. Então, A bloqueia B e temos uma aresta $A \to B$ em G se e somente se as projeções $\overline{A}$ e $\overline{B}$ se interceptam e $p(A) \leq p(B)$ para algum (e portanto todo) ponto $p \in \overline{A} \cap \overline{B}$ (veja Figura 5).

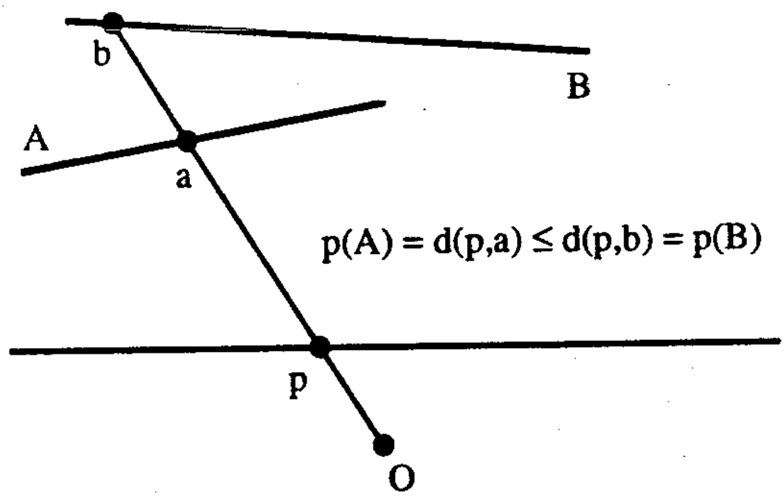

Figura 5

Temos assim um grafo não orientado $\overline{G}$ cujos vértices são as projeções dos segmentos dados e cujas arestas são $\overline{AB}$ sse $\overline{A} \cap \overline{B} \neq \phi$. O grafo $\overline{G}$ é a versão não orientada de G; além disso, $\overline{G}$ é um grafo de intervalos da reta e, felizmente, os ciclos neste tipo de grafos são especiais, como veremos nos teoremas a seguir. Antes, entretanto, algumas definições: um **triângulo** num grafo é um ciclo de 3 vértices; uma **corda** num ciclo é uma aresta entre dois vértices não consecutivos; um grafo é dito **de intervalo** quando seus vértices correspondem a intervalos da reta e suas arestas correspondem aos pares de intervalos que se interceptam.

TEOREMA 2: *Num grafo de intervalo, os únicos ciclos que não tem cordas são os triângulos.*

PROVA: Seja $V_1, V_2, \ldots, V_n$ um ciclo com $n > 3$. Sem perda de generalidade, podemos supor que V_1 é o intervalo mais à esquerda. Se $V_2 \subseteq V_1$ então V_1V_3 é uma corda. Caso contrário, se $V_n \subseteq V_1$ então V_1V_{n-1} é uma corda. Caso contrário, V_2V_n é uma corda. ∎

TEOREMA 3: *Se A, B, C formam um triângulo num grafo de intervalo, então $A \cap B \cap C \neq \phi$.*

PROVA: Sem perda de generalidade, podemos supor que $\inf A \leq \inf B \leq \inf C$. Mas então $\inf C \in A \cap B \cap C$. ∎

Estas propriedades dos ciclos de $\overline{G}$ nos permitem mostrar que G não tem ciclos:

TEOREMA 4: *O grafo G é acíclico.*

PROVA: Como a relação "bloquear visão" é não-reflexiva e anti-simétrica, G não tem ciclos de tamanho 1 ou 2. Passemos, então, aos ciclos de tamanho 3, os triângulos:

Seja $A \to B \to C \to A$ um triângulo em G. Então $\overline{A}$, $\overline{B}$, $\overline{C}$ formam um triângulo em $\overline{G}$ e, portanto, existe um ponto $p \in \overline{A} \cap \overline{B} \cap \overline{C}$. Então $p(A) \leq p(B) \leq p(C)$ e, portanto, $p(A) \leq p(C)$. Como $\overline{A} \cap \overline{C} \neq \phi$, isso quer dizer que $A \to C$, o que contradiz $C \to A$.

No caso geral, se C um ciclo em G então $\overline{C}$ é um ciclo em $\overline{G}$ e, logo, tem uma corda. Esta corda induz um subciclo em C. Continuando por indução, concluímos que se G contém um ciclo então G contém um triângulo. Como G não contém triângulos, concluímos que G não tem ciclos. ∎

4. Algoritmos para o problema de visibilidade.

O Teorema 4 garante que o Problema 5 tem solução. Na verdade, uma vez que se sabe que o grafo G é acíclico, é possível usar algoritmos clássicos de grafos para encontrar uma fonte e assim um segmento totalmente visível.

A dificuldade de aplicar algoritmos genéricos de grafos a problemas geométricos é que, em geral, as arestas estão implícitas e podem requerer um tempo quadrático para serem totalmente explicitadas. Neste caso, algoritmos que são lineares no número de arestas do grafo tornam-se quadráticos ao serem aplicados ao problema geométrico, já que os dados geométricos correspondem aos vértices do grafo e não às arestas. O desafio é explorar a geometria para que somente algumas arestas precisem ser explicitadas.

Isto é exatamente o que ocorre na procura de fontes em grafos orientados acíclicos: busca em profundidade encontra uma fonte em grafos acíclicos genéricos em tempo linear no tamanho do grafo (número de vértices + número de arestas) **[3,7]**. Para usar estes algoritmos no nosso problema de visibilidade, teríamos que usar grafos completos, que têm tamanho quadrático no número de pontos (vértices). Obteríamos, então, algoritmos quadráticos e não lineares. Na prática, o algoritmo quadrático mais apropriado seria mesmo o mais simples: comparar a visibilidade de cada segmento com a de cada outro segmento.

Daremos agora um algoritmo que encontra um segmento totalmente visível em tempo $O(n \log n)$. Esse algoritmo utiliza **varredura plana**, uma técnica muito comum em geometria computacional [**6**]. Conceitualmente, varre-se o plano com uma reta, atualizando uma estrutura de dados cada vez que esta reta encontrar um objeto geométrico. Esta estrutura de dados deve descrever os aspectos da distribuição espacial dos objetos que são relevantes para o problema em questão e é chamada o **estado** da varredura. Como algoritmos só são capazes de descrever um número finito de posições da reta de varredura, o estado só é alterado quando certos **eventos** acontecem. Estes eventos são geralmente o início ou o fim da interseção da reta de varredura com um objeto geométrico.

No caso do Problema 5, organizamos a varredura da seguinte forma. Ordenam-se as projeções dos extremos dos segmentos na reta separatriz. A seguir, percorre-se esta lista em ordem, atualizando o estado a cada extremo encontrado. O estado é muito simples: somente um segmento cujo trecho até a posição da reta de varredura é totalmente visível, chamado o **segmento visível**. A atualização do estado é igualmente simples: quando encontramos o extremo inicial de um segmento, este segmento passa a ser o novo segmento visível caso ele bloqueie a visão do segmento visível; quando encontramos o extremo final do segmento visível, paramos, pois este é o segmento que procuramos (veja Figura 6).

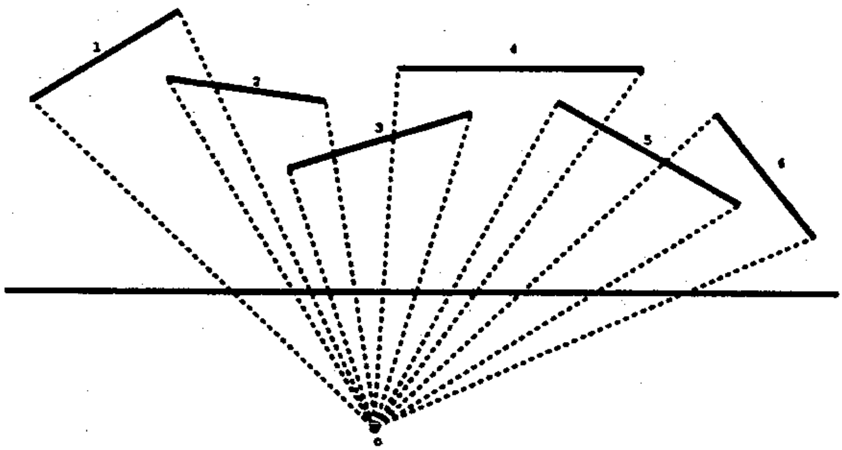

Figura 6

O Teorema 4 garante que a condição de parada sempre ocorre. Portanto, a varredura descrita acima sempre termina e dá um algoritmo que encontra um segmento totalmente visível. Como temos um número linear de extremos e podemos testar bloqueio de visão em tempo constante, o custo computacional deste algoritmo é dominado pela ordenação, o que mostra que a sua complexidade é $O(n \log n)$.

Como não foi encontrada nenhuma referência na literatura sobre a complexidade do Problema 5, resta, então, a questão de saber se existem algoritmos que encontrem um segmento visível em tempo linear. De qualquer modo, é possível que o Problema 4 tenha complexidade menor que a do Problema 5, já que tem mais restrições geométricas.

Conforme descrito anteriormente, segmentos totalmente visíveis permitem construir polígonos simples incrementalmente. Usando o algoritmo acima, obtemos um algoritmo incremental que resolve o Problema 1 em tempo $O(n^2 \log n)$.

É possível adaptar o algoritmo dado para encontrar não só um mas *todos* os segmentos totalmente visíveis. Para isso, é necessário manter uma lista de todos os segmentos que interceptam a reta de varredura, ordenados pela distância ao ponto de visão — lembrar somente o segmento mais perto não é mais suficiente. Para que a complexidade continue a mesma, é preciso usar estruturas de dados mais sofisticadas que possam ser pesquisadas e atualizadas em tempo logarítimico.

5. Conclusão.

A procura de soluções construtivas para o Problema 1 nos levou a uma seqüência de problemas puramente existenciais, cuja solução indicou algoritmos.

Esta é uma situação típica em geometria computacional, na qual técnicas clássicas colaboram com motivações computacionais — uma colaboração que reacendeu o interesse em geometria "elementar".

O Problema 2, apesar de ter sido usado como motivação, não foi resolvido pois, em contraste com a situação no plano, o grafo de visibilidade de um conjunto de faces pode ter ciclos (veja Figura 7). Deste modo, não é possível uma solução para o Problema 2 na mesma linha da que demos para os Problemas 4 e 5.

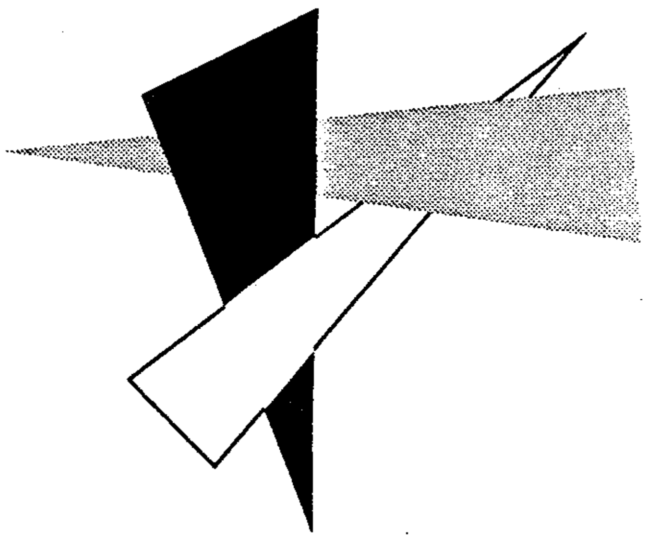

Figura 7

Existem, entretanto, soluções baseadas em soluções alternativas para o Problema 1; por exemplo, usando uma triangulação das projeções dos pontos numa face do fecho convexo.

Um outro análogo espacial para o Problema 1 seria continuar procurando polígonos simples mas agora com os pontos no espaço. Este problema também pode ser resolvido via triangulações: basta encontrar uma triangulação que contenha um polígono simples passando pelas projeções dos pontos. Triangulações deste tipo são chamadas **hamiltonianas** e podem ser construídas em tempo $O(n \log n)$, mas esta é uma outra estória.

6. Agradecimentos.

Gostaria de agradecer a Paulo Cezar Pinto Carvalho pelas várias conversas sobre os problemas apresentados neste artigo, numa das quais surgiu a idéia de considerar grafos de visibilidade.

BIBLIOGRAFIA

1. E. Bishop, "Constructive Analysis," McGraw-Hill, 1967.
2. P. C. P. Carvalho e L. H. de Figueiredo, "Introdução à Geometria Computacional," 18º Colóquio Brasileiro de Matemática, IMPA,, 1991.
3. M. C. Golumbic, "Algorithmic Graph Theory and Perfect Graphs," Academic Press, 1980.
4. M. R. Garey e D.S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness," Freeman, 1979.
5. J. O'Rourke, "Art Gallery Theorems and Algorithms," Oxford University Press, 1987.
6. F. Preparata e M. Shamos, "Computational Geometry: An Introduction," Springer-Verlag, 1985.
7. R. Sedgewick, "Algorithms," Addison-Wesley, 1988.

IMPA/CNPq
Estrada Dona Castorina, 110
22460 Rio de Janeiro, RJ