Estruturas de dados cinéticas

Marcos Siolin Martins **Orientadora: Profª. Drª. Cristina Gomes Fernandes**

Departamento de Ciência da Computação, Instituto de Matemática e Estatística, Universidade de São Paulo

Introdução

Este trabalho discute algumas estruturas de dados cinéticas, propostas por Basch, Guibas e Hershberger [1] para a utilização em algoritmos que resolvem os chamados problemas *cinéticos*. Problemas *cinéticos* são problemas em que deseja-se manter um determinado atributo sobre objetos que estão em movimento contínuo. Os problemas estudados neste trabalho envolvem objetos se movendo apenas em trajetórias lineares.

Ordenação cinética

Para entender a interface geral das estruturas de dados cinéticas precisamos entender o que são *certificados*, *prazos de validade*, eventos e *plano de vôo*. Vamos utilizar um problema e uma estrutura simples para exemplificar essas ideias: a *ordenação cinética* e a *lista ordenada cinética*.

Considere o seguinte problema cinético. Dada uma coleção de n pontos em movimento retilíneo, o objetivo é responder eficientemente consultas do tipo: para um certo i, com $1 \leq i \leq n$, quem é o i-ésimo maior valor da coleção no instante corrente.

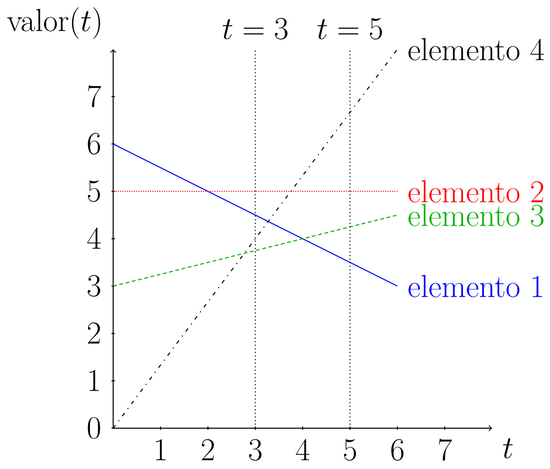

Figura 1: O segundo maior valor no instante $t = 0$ é do elemento 2, e no instante $t = 3$ é do elemento 1. O menor valor da coleção no instante $t = 3$ é do elemento 3, e no instante $t = 5$ é do elemento 1.

Queremos dar suporte às seguintes operações:

- ADVANCE$(t) \rightarrow$ avança o tempo corrente para t;
- CHANGE$(j, v) \rightarrow$ altera a velocidade do elemento j para v;
- QUERY_KTH$(i) \rightarrow$ devolve o elemento cujo valor é o i-ésimo maior no instante atual.

Para implementar estas operações vamos manter um vetor com os elementos dados em ordem decrescente do valor no instante atual. Ele será utilizado para responder as consultas de quem é o i-ésimo elemento. O vetor começa ordenado com os valores iniciais dos elementos.

Uma vez de posse do vetor ordenado com os valores iniciais decrescentemente, construímos um certificado para cada par de elementos consecutivos no vetor.

Os certificados estabelecem que uma relação entre um objeto da estrutura e outro se mantém verdadeira até o seu vencimento. No caso da lista, o vencimento é o instante de tempo em que um elemento ultrapassa o valor do elemento seguinte na lista. Chamaremos esse instante de tempo de valor ou *prazo de validade* do certificado.

Esses prazos de validade determinam os *eventos* que potencialmente causarão modificações na nossa estrutura e consequentemente em alguns certificados. Quando avançarmos no tempo trataremos dos eventos que surgem antes do momento em que queremos avançar.

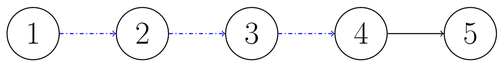

Figura 2: Ao avançar no tempo passamos por um evento gerado pelo vencimento do certificado do elemento 2. O tratamento desse evento é a troca entre 2 e 3, e resulta na atualização da estrutura e dos certificados afetados, tracejados em azul.

Para calcular o prazo de validade dos certificados, utilizaremos o chamado *plano de vôo* dos objetos. O plano de vôo de um objeto é uma função que determina sua trajetória.

O plano de vôo pode sofrer mudanças. Essas mudanças serão feitas através da operação CHANGE e geram a necessidade de atualização de certificados e de ajustes nas estruturas.

Basch, Guibas e Hershberger propuseram alguns critérios de desempenho como forma de analisar e medir as chamadas estruturas cinéticas. São eles: *responsividade*, *eficiência*, *compacidade* e *localidade*.

Referências

[1] Basch, J., Guibas, L., Hershberger, J., "**Data structures for mobile data**," in *Journal of Algorithms*, 1999, 31 (1), pp. 1–28

[2] Freitas, E. G., "**Problemas Cinéticos em Geometria Computacional**," *Dissertação de mestrado*, Universidade de São Paulo, 2000. https://www.teses.usp.br

[3] Basch, J., "**Kinetic data structures**," *PhD Thesis*, Stanford University, 1999. http://www.basch.org/phdthesis

Par mais próximo cinético

Vamos agora falar de um outro problema cinético interessante: o problema do par mais próximo cinético. Num conjunto dado de n pontos em movimento, determinar qual par de pontos possui distância mínima. Explicaremos as principais ideias de um algoritmo proposto por Basch, Guibas e Hershberger que utiliza a técnica de linha de varredura para resolver o problema estático e depois mostraremos algumas ideias de como tratar os eventos na sua versão cinética.

O algoritmo é baseado na ideia de dividir o plano, para cada ponto, em seis cones iguais. Os cones são delimitados pela reta paralela ao eixo y que passa pelo ponto e pelas retas $x \pm 30°$, isto é, as retas que passam pelo ponto e formam $\pm 30°$ com o eixo x.

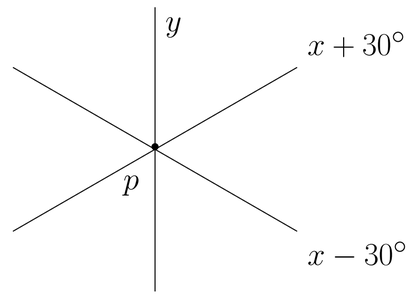

Figura 3: A reta paralela ao eixo y que passa por p e as retas $x \pm 30°$.

Tendo dividido o plano em cones, a ideia é achar o ponto mais próximo de p dentro de cada um desses cones. Se assim o fizermos para todos os pontos, um desses pares possui a menor distância entre si e será um par mais próximo que buscamos. Na prática, utilizaremos apenas os cones à direita do ponto.

Precisamos também definir alguns conjuntos que serão utilizados no algoritmo:

- $Dom(p)$: *dominância de p*. É o cone à direita de p cujo eixo central é paralelo ao eixo x;
- $Maxima(p)$: o conjunto dos pontos à direita de p que não pertencem à *dominância* de nenhum ponto à direita de p;
- $Cands(p)$: os *candidatos* de p são aqueles pontos na *dominância* de p que não pertencem a *dominância* de nenhum ponto à direita de p.

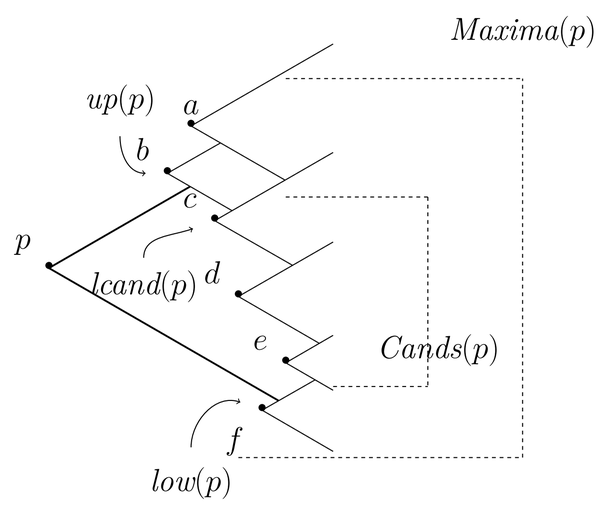

Figura 4: Os pontos c, d e e pertencem a $Cands(p)$, e todos os pontos exceto p pertencem a $Maxima(p)$. O ponto b é $up(p)$ e o ponto f é $low(p)$. O ponto c é $lcand(p)$.

Chamaremos o ponto de $Maxima(p)$ de menor ordenada que está acima de $Dom(p)$ de $up(p)$ e o ponto de $Maxima(p)$ de maior ordenada que está abaixo de $Dom(p)$ de $low(p)$. Os pontos de $Maxima(p)$ estritamente entre $low(p)$ e $up(p)$ são justamente os de $Cands(p)$.

Dentre os *candidatos* de p, chamaremos o ponto com menor coordenada x de $lcand(p)$. Consideraremos apenas os pares $(p, lcand(p))$ como candidatos a par mais próximo.

O algoritmo varre o plano da direita para a esquerda três vezes: com o sistema de coordenadas rotacionado 60°, 0° e −60°. Em cada varredura, computa as estruturas necessárias para comparar apenas os pares $(p, lcand(p))$, para determinar o par mais próximo em tempo $O(n \lg n)$.

"Cinetização" do algoritmo estático

Para "cinetizar" o algoritmo estático utilizaremos duas estruturas de dados cinéticas. Primeiramente, teremos os certificados das três *listas ordenadas cinéticas*, que guardarão a ordem dos pontos de acordo com os eixos x, $x+60°$ e $x-60°$.

Para garantir qual, dentre os pares $(p, lcand(p))$, é o par mais próximo, usaremos um *torneio cinético*, estrutura de dados cinética utilizada em algoritmos para resolver o problema do máximo (ou mínimo) cinético.

Também precisaremos manter informação guardada para atualizar com eficiência mudanças provocadas por trocas na ordem dos pontos em relação a um dos três eixos. Por exemplo, uma troca na ordem dos pontos p e q pode acarretar em mudanças nos conjuntos $Cands(p)$ e $Cands(q)$.

Para que consigamos manter $lcand(p)$ de maneira eficiente, cada ponto p terá três árvores binárias de busca associadas a ele, com os conjuntos $Cands(p)$, $Hits_{up}(p)$ e $Hits_{low}(p)$. A árvore $Hits_{up}(p)$ guarda os pontos q tais que $up(q) = p$, enquanto a árvore $Hits_{low}(p)$ guarda os pontos q tais que $low(q) = p$.

Cada uma das três árvores tem sua raiz apontando para o nó p, e cada nó das árvores aponta para o seu nó pai. Na árvore $Cands(p)$, cada nó deve apontar para o descendente que contém o ponto mais à esquerda na ordenação horizontal. Ademais, cada ponto tem um ponteiro para os nós que o tem como chave.

Trocas na ordem dos pontos geram mudanças em três casos:

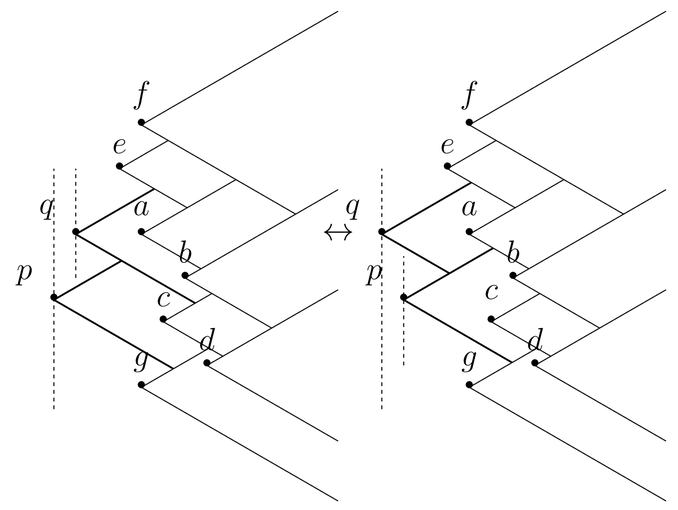

Figura 5: Troca na ordem horizontal. Da esquerda para a direita, o caso em que p está em $Hits_{up}(q)$. Da direita para a esquerda, o caso em que q está em $Hits_{low}(p)$.

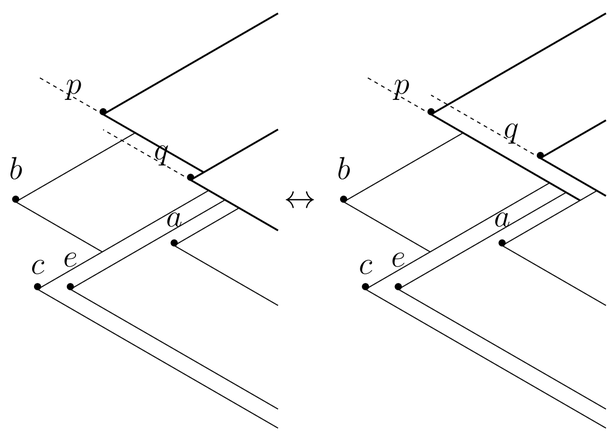

Figura 6: Troca na ordem 60°. Da esquerda para a direita, o caso em que p está em $Hits_{low}(q)$, ou seja, q está entrando em $Dom(p)$. Da direita para a esquerda, o caso em que q está em $Cands(p)$, saindo de $Dom(p)$.

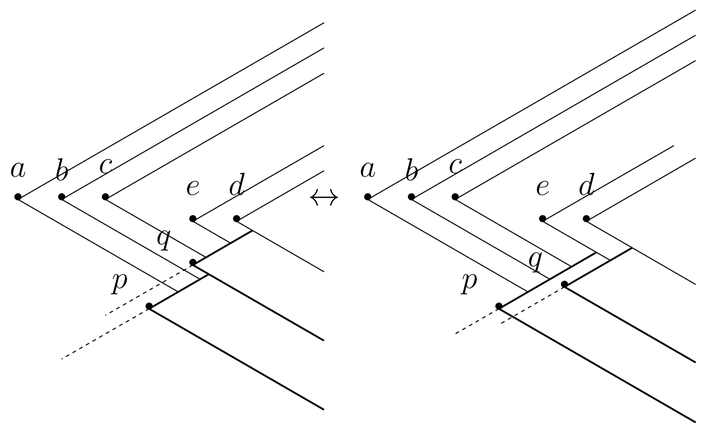

Figura 7: Troca na ordem −60°. Da esquerda para a direita, o caso em que p está em $Hits_{up}(q)$, ou seja, q está entrando em $Dom(p)$. Da direita para a esquerda, o caso em que q está em $Cands(p)$, saindo de $Dom(p)$.

Mais informações

Além dos problemas da ordenação cinética e do par mais próximo cinético, também foram estudadas estruturas para resolver o problema do máximo cinético e da triangulação de Delaunay cinética. Algumas dessas estruturas foram implementadas e encontram-se em: `https://github.com/siolinm/mac0499`.
Para mais informações, acesse: `https://www.linux.ime.usp.br/~siolinm/mac0499`.