GEOMETRIA COMPUTACIONAL

JOSÉ COELHO DE PINA

1. Diagrama de Delaunay

O que veremos a seguir estabelece uma ligação fundamental entre o diagrama de Voronoi e uma triangularização especial.

Observação. O material desta seção foi extraído de Figueiredo e Carvalho [2].

1.1. O diagrama.

Já estudamos o grafo dual de uma triangularização quando falamos de triangularização de polígonos. Consideraremos agora o grafo dual do diagrama de Voronoi de um conjunto S de pontos em E^2.

O grafo dual do diagrama de Voronoi $\mathrm{Vor}(S)$ é o grafo cujos vértices são os sítios de S e dois vértices são ligados por uma aresta se as regiões correspondentes compartilham uma aresta. Consideremos agora o diagrama obtido ao representarmos estas arestas por segmentos de retas. Chamaremos este diagrama de *diagrama de Delaunay* de S (cf. Figueiredo e Carvalho [2]) e o denotaremos por $\mathrm{Del}(S)$. Veja a figura 1.

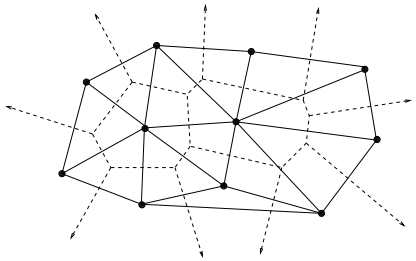

Figura 1. Os diagramas de Voronoi (tracejado) e de Delaunay de um conjunto de pontos.

1.2. Propriedades do diagrama de Delaunay.

Devido à dualidade entre o diagrama de Voronoi e o diagrama de Delaunay, muitas das propriedades do diagrama de Voronoi têm uma tradução imediata em termos do diagrama de Delaunay. Abaixo listaremos algumas dessas propriedades como teoremas.

Teorema 1 (Círculo Circunscrito)**.** *Três sítios p_1, p_2 e p_3 de um conjunto S definem uma face do diagrama de Delaunay de S se e somente se existe um círculo que contém p_1, p_2 e p_3 e não contém outro sítio de S.*

Date: novembro de 2007.

Teorema 2 (Círculo Vazio)**.** *Dois sítios p_1 e p_2 de um conjunto S são adjacentes no diagrama de Delaunay de S se e somente se existe algum círculo que contém p_1 e p_2 e não contém outro sítio de S.*

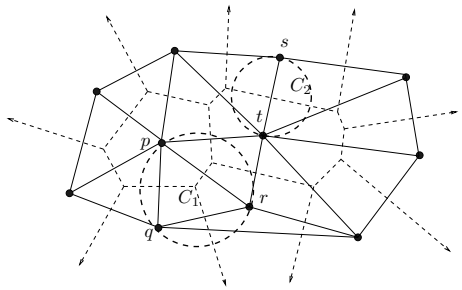

Figura 2. O círculo C_1 é um certificado de que $\triangle(p,q,r)$ é uma face de Delaunay. O círculo C_2 é um certificado de que st é uma aresta de Delaunay.

Teorema 3. *Todo sítio mais próximo de um sítio p de um conjunto S determina uma aresta do diagrama de Delaunay de S incidente a p.*

Teorema 4. *O diagrama de Delaunay de um conjunto S de sítios é a realização de um grafo planar.*

Construiremos um conjunto de triângulos

$$\mathcal{T}(S) := \{T(v) \mid v \text{ é um vértice de Voronoi de } \mathrm{Vor}(S)\}$$

tais que os seus interiores sejam disjuntos e cada ponto no interior de $\mathrm{conv}(S)$ pertença a um destes triângulos.

Seja v um vértice de Voronoi e sejam $p_{v,1}, p_{v,2}, p_{v,3}$ os sítios de S tais que $V(p_{v,1}), V(p_{v,2})$ e $V(p_{v,3})$ são as três regiões de $\mathrm{Vor}(S)$ que compartilham o vértice v. (Veja a figura 3.) Definimos $T(v)$ como sendo o triângulo cujos vértices são $p_{v,1}, p_{v,2}, p_{v,3}$. Temos que $T(v)$ é um triângulo do diagrama de Delaunay e $T(v) \subseteq \mathrm{conv}(S)$ para todo vértice de Voronoi v (a prova deste fato usa o Teorema 2).

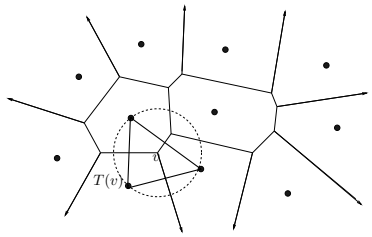

Figura 3. O triângulo $T(v)$ correspondente ao vértice de Voronoi v.

Teorema 5. *Sejam v e u vértices distintos de $\mathrm{Vor}(S)$. Então os interiores de $T(v)$ e $T(u)$ são disjuntos.*

Teorema 6. *Seja pq uma aresta do diagrama de Delaunay e seja $r \in S$. Então os sítios p, q e r formam um triângulo de $\mathcal{T}(S)$ se e somente se $\angle(prq)$ é máximo dentre todos os ângulos da forma $\angle(psq)$, onde $s \in S$ está no mesmo semiplano de r em relação a pq.*

Prova. (A ferramenta de geometria elementar que utilizamos na prova deste teorema é o Teorema do Ângulo Inscrito.) Suponhamos que $s \in S$ está no mesmo semiplano de r em relação a pq e que $\angle(psq) > \angle(prq)$. Então, pelo Teorema do Ângulo Inscrito, s é interior ao círculo contendo p, q e r na fronteira, o que contradiz o fato de $\triangle(p, q, r)$ ser um triângulo de $\mathcal{T}(S)$. (Veja a figura 4.) Reciprocamente, se p, q e r formam um triângulo de $\mathcal{T}(S)$ então por definição de $\mathcal{T}(S)$ e pelo Teorema 1 sabemos que existe um vértice de Voronoi v tal que $C(v)$ é um círculo vazio. Agora, pelo Teorema do Ângulo Inscrito (ou um corolário deste teorema) sabemos que o ângulo $\angle(prq)$ é máximo dentre todos os ângulos da forma $\angle(psq)$, onde $s \in S$ está no mesmo semiplano de r em relação a pq. ∎

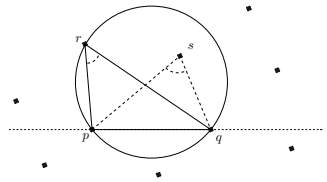

FIGURA 4. O triângulo $\triangle(p, q, r)$ pertence a $\mathcal{T}(S)$.

Teorema 7. *Os triângulos em $\mathcal{T}(S)$ cobrem o fecho convexo dos sítios em S.*

Em 1934, Delaunay mostrou que o diagrama $\mathrm{Del}(S)$ produz uma triangularização do conjunto S de sítios. Esta triangularização é chamada de *triangularização de Delaunay* de S.

Teorema 8. *(Delaunay [1]) O diagrama de Delaunay é uma triangularização do conjunto de sítios S.*

Prova. Cada triângulo de $\mathcal{T}(S)$ é formado por arestas do diagrama de Delaunay de S. Pelo Teorema 5, a intersecção entre os interiores de dois triângulos distintos de $\mathcal{T}(S)$ é vazia e, pelo Teorema 7, os triângulos em $\mathcal{T}(S)$ cobrem $\mathrm{conv}(S)$. Logo as regiões (faces) limitadas do diagrama de Delaunay formam triângulos com interiores disjuntos que cobrem $\mathrm{conv}(S)$. ∎

Corolário 1. *O diagrama de Voronoi de um conjunto S de n sítios tem no máximo $2n-5$ vértices e $3n-6$ arestas.*

Prova. Cada aresta do diagrama de Delaunay $\mathrm{Del}(S)$ corresponde a uma única aresta de Voronoi de $\mathrm{Vor}(S)$. Como $\mathrm{Del}(S)$ é uma triangularização de S temos, pela Fórmula de Euler, que $\mathrm{Del}(S)$ tem no máximo $3n-6$ arestas e $2n-4$ faces. Logo o número de arestas de Voronoi de $\mathrm{Vor}(S)$ é no máximo $3n-6$ e o número de vértices de Voronoi é no máximo $2n-5$ (a face ilimitada de $\mathrm{del}(S)$ não corresponde a um vértice de Voronoi). ∎

2. Cota inferior

Discutiremos nesta seção a cota inferior de qualquer algoritmo que constrói um diagrama de Voronoi ou uma triangularização de um dado conjunto S com n pontos.

Dado o diagrama de Voronoi $\mathrm{Vor}(S)$ não é difícil encontrarmos o par de pontos mais próximos em tempo $O(n)$: basta examinarmos os sítios cujas regiões compartilham uma aresta de Voronoi—pelo Corolário 1 sabemos que teremos que examinar $O(n)$ pares de sítios; o número de arestas de um diagrama de Voronoi de n sítios é $O(n)$. Como sabemos que o Problema do Par Mais Próximo tem cota inferior $\Omega(n \log n)$ temos então o seguinte teorema.

Teorema 9. *Qualquer algoritmo para construir o diagrama de Voronoi de um dado conjunto com n sítios faz $\Omega(n \log n)$ operações elementares.* ∎

Como veremos, o diagrama de Delaunay $\mathrm{Del}(S)$ pode ser obtido em tempo linear a partir do diagrama de Voronoi $\mathrm{Vor}(S)$. O Teorema 9 já implica que o problema de construir o diagrama de Delaunay $\mathrm{Del}(S)$ de um conjunto com n sítios tem cota inferior $\Omega(n \log n)$. Entretanto, observaremos aqui que o Problema de Ordenação pode ser reduzido ao problema de construir uma triangularização (não necessariamente a de Delaunay) de um dado conjunto de pontos. Considere um conjunto $\{x_1, \ldots, x_n\}$ de n números. Dado este conjunto construiremos o conjunto S formado pelos $n+1$ pontos

$$\{(x_1, 1), (x_2, 1), \ldots, (x_n, 1), (0, 0)\}.$$

Esta transformação pode ser realizada em tempo linear. O conjunto S tem uma única triangularização (ilustrada na figura 5). Dada esta triangularização (ou seja, uma estrutura de dados, como a winged-edge, representando o grafo planar assoaciado à esta triangularização) podemos em tempo linear obter a ordenação dos números $x_1, \ldots, x_n$. Resumindo, obtemos o seguinte teorema.

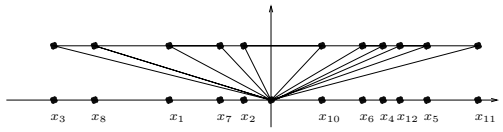

FIGURA 5. Ilustração da redução do Problema de Ordenação para o problema de construir uma triangularização.

Teorema 10. *Qualquer algoritmo para construir a triangularização de um dado conjunto com n pontos faz $\Omega(n \log n)$ operações elementares.* ∎

O teorema acima implica que, em particular, qualquer algoritmo que construa uma triangularização de Delaunay de um dado conjunto com n sítios faz $\Omega(n \log n)$ operações elementares.

3. A primitiva geométrica InCircle

Observação. O material desta seção foi extraído do artigo de Guibas e Stolfi [3].

Dados quatro pontos a, b, c e d, o predicado (primitiva) `InCircle`(a, b, c, d) é verdadeiro se o ponto d está na face esquerda da circunferência orientada pelo triângulo $\triangle(a, b, c)$. A face esquerda é a face interior (limitada) da circunferência se o triângulo $\triangle(a, b, c)$ estiver orientado positivamente, caso contrário a face esquerda é o exterior da circunferência. Logo, se tomamos a, b e c de tal forma que o triângulo $\triangle(a, b, c)$ esteja orientado positivamente então a primitiva `InCircle` decide se o ponto d está no círculo determinado pelos pontos a, b e c. Veja a figura 6.

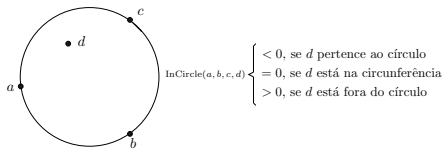

FIGURA 6. A primitiva `InCircle`.

Notemos que uma situação degenerada ocorre se d pertence a circunferência determinada por a, b e c. Curiosamente a situação não é degenerada se a, b e c são colineares e d esta fora da reta determinada por estes pontos.

Teorema 11. *O valor da primitiva* `InCircle`(a, b, c, d) *é dado pelo sinal do determinante*

$$\mathcal{D}(a,b,c,d) := \begin{vmatrix} x_a & y_a & x_a^2+y_a^2 & 1 \\ x_b & y_b & x_b^2+y_b^2 & 1 \\ x_c & y_c & x_c^2+y_c^2 & 1 \\ x_d & y_d & x_d^2+y_d^2 & 1 \end{vmatrix}.$$

4. Um algoritmo quadrático

Este algoritmo para construir o diagrama de Delaunay de um dado conjunto $S = \{p_1, \ldots, p_n\}$ de sítios imita o algoritmo `Embrulho-para-presente` para construir o fecho convexo de um conjunto de pontos em $\mathbb{R}^3$. Note que dado $\mathrm{Del}(S)$ podemos, em tempo linear, construir $\mathrm{Vor}(S)$.

4.1. **A idéia do algoritmo.** A idéia básica, análoga a do Algoritmo `Embrulho-para-Presente`, é usar as arestas do diagrama de Delaunay (*arestas de Delaunay*) dos triângulos que já foram determinados para encontrar os outros triângulos. Cada aresta do diagrama de Delaunay é compartilhada por um ou dois triângulos (*triângulo de Delaunay*—triângulos da triangularização de Delaunay). Uma aresta de Delaunay tem apenas um triângulo do diagrama incidente a ela quando está na fronteira do fecho convexo de S. Chamaremos uma aresta de Delaunay de *livre* se algum dos triângulos que compartilham esta aresta ainda não foi determinado pelo algoritmo. Dada uma aresta livre p_ip_j de um triângulo de Delaunay, podemos determinar o triângulo que compartilha esta aresta da seguinte maneira: encontre o ponto p_k, do lado apropriado do segmento p_ip_j, para o qual o $\angle(p_ip_kp_j) > \angle(p_ip_sp_j)$ para todo ponto p_s que esta do mesmo lado de p_k (cf. Teorema 6). Este é o passo básico do algoritmo e pode ser feito usando-se as primitivas `Left` e `InCircle`.

4.2. **O Algoritmo.** Uma estrutura de adjacência (como *winged-edge*) pode ser utilizada para armazenarmos o diagrama de Delaunay. A medida que as faces são introduzidas no diagrama, suas arestas são examinadas para verificarmos se elas já fazem parte da estrutura (se são livres ou não). No caso da aresta examinada não fazer parte da estrutura (i.e., a aresta é livre) ela deverá ser explorada mais tarde.

Como começar o algoritmo já que no início não temos nenhuma face (triângulo) de $\mathrm{Del}(S)$? Basta procedermos de maneira análoga ao que fizemos no Algoritmo de Jarvis: determinar uma aresta de $\mathrm{conv}(S)$. Esta aresta é com certeza uma aresta de $\mathrm{Del}(S)$ (para a qual ainda resta construirmos o triângulo de $\mathrm{Del}(S)$ incidente a ela—i.e., esta aresta é livre). Com esta aresta podemos, usando o Teorema 6, encontrar uma face de $\mathrm{Del}(S)$.

O algoritmo empregará uma fila Q para armazenar as faces que vão sendo encontradas ao longo da execução.

Algoritmo `Delaunay`(S)
`Entrada:` Um conjunto finito S de pontos em E^2.
`Saída:` O diagrama de Delaunay dos pontos em S.
1. $Q \leftarrow \emptyset$; [Fila com as faces encontradas.]
2. $f \leftarrow$ uma face inicial de $\mathrm{del}(S)$;
3. Insira f na fila Q e na winged-edge estrutura T;
4. **while** $Q \neq \emptyset$ **do**
5. $\quad f \leftarrow$ primeira face na fila Q;
6. $\quad$ **for** cada aresta livre e de f **do**
$\quad\quad$ [Passo correspondente ao Teorema 6.]
7. $\quad\quad f' \leftarrow$ face do diagrama de Delaunay que compartilha e com f;
8. $\quad\quad$ Insira f' na fila Q;
9. $\quad\quad$ Insira f' na estrutura winged-edge T, conectando-a com as faces já geradas que lhe são adjacentes e determinando suas arestas livres.
10. **return** T.

4.3. **Análise do algoritmo.** O algoritmo pára somente quando toda a triangularização de Delaunay tiver sido obtida. Observe que se as faces geradas (inclusive a face externa) não cobrissem o plano, alguma aresta de uma das faces estaria livre, o que não ocorre já que o algoritmo só termina quando todas as arestas livres foram exploradas. O Passo 2 pode ser feito em tempo $O(n)$. Utilizando a primitiva `InCircle` podemos executar o Passo 7 em tempo $O(n)$. A complexidade de tempo do Passo 9 também é $O(n)$ (precisamos varrer toda a estrutura T para determinarmos se uma aresta esta presente ou não). Como os Passos 7 e 9 serão executados $O(n)$ vezes (cf. Corolário 1) concluímos então que a complexidade de tempo do Algoritmo `Delaunay` é $O(n^2)$.

5. Um algoritmo de divisão-e-conquista

Neste seção estudaremos um algoritmo de divisão-e-conquista que constrói o diagrama de Delaunay de um dado conjunto S de n pontos no plano em tempo $O(n \log n)$ (i.e., um algoritmo ótimo para o problema).

Observação. O material desta seção foi extraído do Capítulo 4 de Figueiredo e Carvalho [2] e do artigo de Guibas e Stolfi [3]. O algoritmo que veremos é devido a Lee e Schachter [4] e é dual do algoritmo de Shamos e Hoey [6] (veja também Preparata e Shamos [5] páginas 211–220).

5.1. **A descrição do algoritmo.** Para facilitar a nossa exposição do algoritmo, suporemos que qualquer subconjunto de quatro pontos de S são não-cocirculares e que qualquer dois pontos não têm a mesma x-coordenada. O algoritmo está descrito logo abaixo. Notemos que, para que o diagrama de Delaunay de S seja construído em tempo $O(n \log n)$, necessitamos que a fase `Combinar` seja feita em tempo $O(n)$.

Algoritmo `Delaunay2`(S)
`Entrada:` Um conjunto S de sítios em E^2.
`Saída:` O diagrama de Delaunay $\mathrm{Del}(S)$ de S.
1. **if** $|S| \leq 3$, construa $\mathrm{Del}(S)$ e **return**.
2. (`Dividir`) Particione S em S_1 com $\lceil n/2 \rceil$ pontos e S_2 com $\lfloor n/2 \rfloor$ pontos.
3. (`Conquistar`) Recursivamente construa os diagramas de $\mathrm{Del}(S_1)$ e $\mathrm{Del}(S_2)$.
4. (`Combinar` ou `Merge`) Construa $\mathrm{Del}(S_1 \cup S_2)$ usando $\mathrm{Del}(S_1)$ e $\mathrm{Del}(S_2)$.

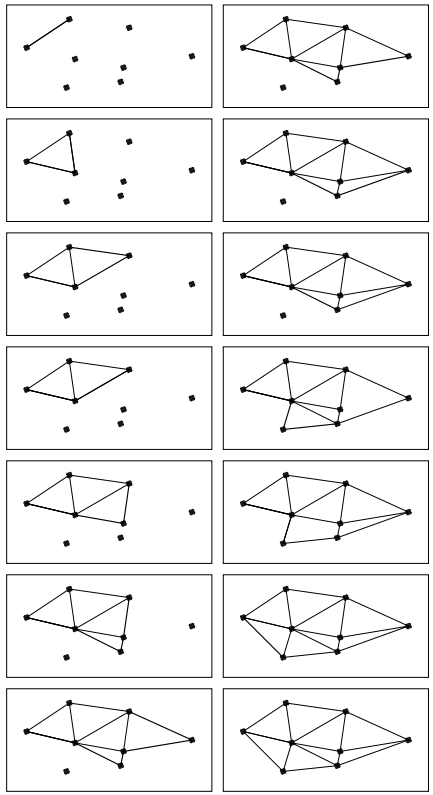

FIGURA 7. Ilustração do Algoritmo `Delaunay` em ação.

5.2. **Pré-processamento: 'sorting step'.** Para facilitar os passos `Dividir` e `Conquistar`, antes de aplicarmos o algoritmo `Delaunay2`, ordenaremos o conjunto de pontos dado em relação as suas x-coordenadas.

5.3. **Dividir.** O conjunto de pontos S será particionado em dois subconjunto em relação a mediana das x-coordenadas. Graças ao pré-processamento este passo pode ser feito rapidamente. Em outras palavras, o conjunto S será divido com relação a uma reta vertical: seja L o conjunto formado pelos sítios de S que têm x-coordenada menor ou igual a mediana das x-coordenadas dos pontos e seja R o conjunto formado pelos demais pontos de S (pontos à direita da reta vertical passando pela mediana das x-coordenadas). (Escreveremos L (*left*) e R (*right*) em vez de S_1 e S_2, respectivamente.) Notemos que esta escolha de conjuntos L e R, juntamente com a nossa hipótese de que S não tem dois pontos com a mesma x-coordenada, garante que os diagramas $\mathrm{Del}(L)$ e $\mathrm{Del}(R)$ são disjuntos — se permitíssemos vários pontos com a mesmo coordenada teríamos apenas que tormar um certo cuidado para definirmos L e R, pois não queremos que um conjunto fique com todos os sítios e o outro fique vazio.

5.4. **Conquistar.** O passo conquistar corresponde às chamadas recursivas onde cada subproblema será resolvido independentemente. A base da recursão é atingida quando o conjunto de sítios em questão tem no máximo três sítios. Na base da recursão o diagrama de Delaunay desses (no máximo três) sítios pode ser facilmente construído. Por exemplo, o diagrama de Delaunay de três sítios corresponde a um triângulo que tem esses sítios como vértices ou a dois segmentos ligando os sítios (no caso dos sítios serem colineares).

5.5. **Combinar: 'the merging step'.** A maior parte do trabalho deste algoritmo de divisão-e-conquista está neste passo (o mesmo fenômeno se verificou quando estudamos outros algoritmos de divisão-e-conquista). Precisamos ser cuidadosos para realizarmos este passo em tempo linear: um algoritmo ingênuo pode atingir a complexidade de tempo $O(n^2)$ para este passo, o que não nos permitiria obter a complexidade de tempo de $O(n \log n)$ para todo o algoritmo.

Chamaremos as arestas de $\mathrm{Del}(L)$ e $\mathrm{Del}(R)$ de L-L-arestas e R-R-arestas, respectivamente. As arestas de $\mathrm{Del}(S)$ que têm um extremo em um sítio de L e o outro extremo em um sítio de R serão ditas L-R-arestas (ou *crossing edges*). O passo `Combinar` consistirá em removermos algumas L-L e R-R-arestas e incluirmos algumas L-R-arestas.

Todas as L-R-arestas cruzam uma reta vértical l que separa o conjunto L e o conjunto R. Estes pontos de interseção de l com as L-R-arestas de $\mathrm{Del}(S)$ induzem uma ordem total no conjunto das L-R-arestas. Chamaremos esta ordem total de y-ordem. A L-R-aresta mais 'baixa' (menor na y-ordem) e mais 'alta' (maior na y-ordem) corresponde a aresta tangente inferior e superior a $\mathrm{Conv}(L)$ e $\mathrm{Conv}(R)$, respectivamente. (Veja a figura 8.) O seguinte teorema nos fornece mais informações sobre a estrutura das L-R-arestas.

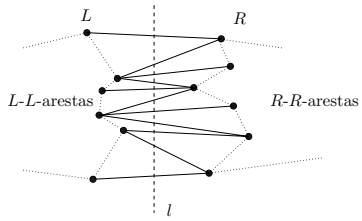

FIGURA 8. As L-R-arestas de $\mathrm{Del}(S)$.

Teorema 12. *Qualquer duas L-R-arestas consecutivas na y-ordem têm um vértice em comum. Ademais, o terceiro lado do triângulo que estas arestas determinam é uma L-L-aresta ou uma R-R–aresta.*

Suponha que tenhamos encontrado todas as L-R-arestas na y-ordem entre a tangente inferior a $\mathrm{conv}(L)$ e $\mathrm{conv}(R)$ até uma aresta $p_l p_r$, $p_l \in L$ e $p_r \in R$. (Veja a figura 9.) Pelo Teorema 12 sabemos que a próxima L-R-aresta na y-ordem é incidente a p_l ou p_r. Para determinarmos esta aresta basta examinarmos os sítios de L adjacentes a p_l em $\mathrm{Del}(L)$ e os sítios de R adjacentes a p_r em R. Por outro lado, pelo Teorema 6, sabemos que a próxima L-R-aresta na y-ordem tem uma ponta em p_l ou p_r e a outra ponta em um sítio p tal que o ângulo $\angle(p_l p p_r)$ é máximo entre todos os sítios que estão à esquerda do segmento (orientado) $p_l p_r$.

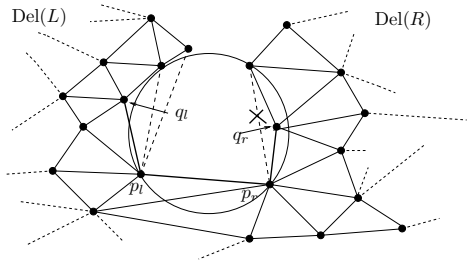

FIGURA 9. Determinação da próxima L-R-aresta de Del($L \cup R$).

Abaixo encontra-se uma descrição do Algoritmo `MergeDelaunay` que encontra as L-R-arestas e remove as L-L e R-R-arestas que deixam de ser arestas do diagrama Del($L \cup R$). Em cada iteração o algoritmo começa com uma L-R-aresta $p_l p_r$ e trabalha no sentido de encontrar a próxima L-R-aresta (próxima em relação a y-ordem). Dentre os sítios adjacentes a p_l (p_r, respectivamente) o algoritmo determina um sítio q_l (q_r, respectivamente) candidato a ponta da próxima L-R-aresta. (Veja a figura 9.) No final da iteração, o algoritmo determina, entre q_l e q_r qual sítio será a ponta da próxima L-R-aresta (a próxima L-R-arestas será $p_l q_r$ ou $p_r q_l$).

Para encontrar q_l o algoritmo procederá como segue. Sejam $q_1, q_2, \ldots, q_k$ os sítios de L adjacentes a p_l em Del(L) listados no sentido anti-horário. Se o círculo determinado pelo triângulo $\triangle(p_l, p_r, q_1)$ contém o ponto q_2 concluímos imediatamente que $p_l q_1$ não pode ser uma aresta de Del($L \cup R$). Neste caso, eliminamos a L-L-aresta $p_l q_1$ e repetimos o mesmo teste com os sítios q_2 e q_3 no lugar de q_1 e q_2, respectivamente. O sítio procurado q_l é o primeiro sítio q_t tal que o círculo determinado pelo triângulo $\triangle(p_l, p_r, p_t)$ não contém p_{t+1}. (Por que isto funciona?) A determinação de q_r é análoga. A única diferença é que examinamos os sítios adjacentes a p_r em Del(R) no sentido horário ao ínves de anti-horário.

Para garantirmos que a determinação de q_l e q_r seja feita em tempo constante por aresta incidente a p_l e p_r, suporemos que Del(L) e Del(R) sejam dadas através de suas estruturas *winged-edge* (ou qualquer outra estrutura que represente um grafo planar de uma maneira 'decente' — ou seja, em que as operações envolvidas sejam rápidas).

Notação. Se u e v são vértices adjacentes de um diagrama planar H então denotaremos por `esq`(u, v) o vértice de H adjacente a u que está imediatamente à esquerda do segmento (orientado) uv no sentido anti-horário. Analogamente, denotaremos `dir`(u, v) o vértice de H adjacente a u que está imediatamente à direita do segmento uv no sentido horário. (Veja a figura 10.) Notemos que se H é representado por uma estrutura *winged-edge* então as operações `esq` e `dir` podem ser feitas em tempo constante (i.e., $O(1)$).

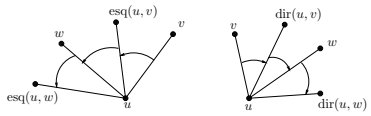

FIGURA 10. Ilustração da definição de `esq`(u, v) e `dir`(u, v).

Algoritmo `MergeDelaunay`(Del(L),Del(R))
`Entrada:` Dois diagramas de Delaunay Del(L) e Del(R) tais que os sítios de L estão à esquerda dos sítios de R.
`Saída:` O diagrama de Delaunay Del($L \cup R$) de $L \cup R$.

1. $T \leftarrow$ Del(L) $\cup$ Del(R); [T é uma estrutura winged-edge.]
2. Determine o segmento $p_l p_r$ tangente inferior a conv(L) e conv(R).
3. **while** $p_l p_r$ não é a tangente superior a conv(L) e conv(R) **do**
   [**Determinação do candidato** q_l**.**]
4. $q_l \leftarrow$ `esq`(p_l, p_r); $q \leftarrow$ `esq`(p_l, q_l);
5. **while** `Left`(p_l, p_r, q_l) e `InCircle`(p_l, p_r, q_l, q) **do**
6. Remova a L-L-aresta $p_l q_l$ de T;
7. $q_l \leftarrow q$; $q \leftarrow$ `esq`(p_l, q);
8. **if** `Left`(p_r, p_l, q_l)
9. **then** não temos candidado a q_l.
   [**Determinação do candidato** q_r**.**]
10. $q_r \leftarrow$ `dir`(p_r, p_l); $q \leftarrow$ `dir`(p_r, q_r);
11. **while** `Left`(p_l, p_r, q_r) e `InCircle`(p_l, p_r, q_l, q) **do**
12. Remova a R-R-aresta $p_r q_r$ de T;
13. $q_r \leftarrow q$; $q \leftarrow$ `dir`(p_r, q);
14. **if** `Left`(p_r, p_l, q_r)
15. **then** não temos candidado a q_r.
   [**Determinação da próxima** L-R**-aresta.**]
16. Seja p o sítio entre os candidatos cujo ângulo $\angle(p_l p p_r)$ é máximo.
   [Se não existirem candidatos, então $p_l p_r$ é tangente superior.]
17. **if** $p \in L$
18. **then** insira $p_r p$ em T;
19. **else** insira $p_l p$ em T;
20. **return** T.

As operações primitivas usadas pelo algoritmo são `Lefton` e `InCircle`. Podemos intuitivamente ver o algoritmo como uma bolha (bidimensional) que vai subindo entre Del(L) e Del(R) e que a cada instante ocupa a maior área possível. Cada vez que a bolha atinge um novo sítio a próxima L-R-aresta é determinada (e talvez algumas L-L- ou R-R-arestas tenham que ser removidas).

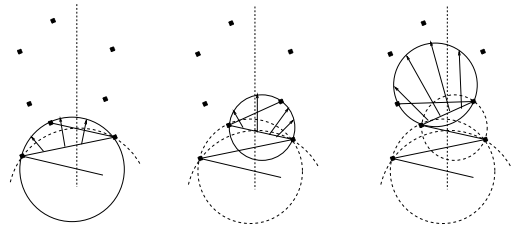

FIGURA 11. Ilustração do algoritmo `MergeDelaunay` através de 'bolhas subindo.'

5.6. **Correção do Algoritmo** `MergeDelaunay`**.** Para mostrarmos a correção do algoritmo precisamos verificar que:

(1) o algoritmo produz todas as L-R-arestas de Del($L \cup R$);
(2) as L-L e R-R-arestas removidas pelo algoritmo não são arestas de Del($L \cup R$); e
(3) ao final do Algoritmo `MergeDelaunay` o diagrama resultante corresponde à triangularização de Delaunay de $L \cup R$.

A primeira L-R-aresta determinada pelo algoritmo na linha 2 é certamente uma aresta de $\mathrm{Del}(L \cup R)$ (cf. Teorema 2). Os Teoremas 12 e 15 implicam que as linhas 5–9 do Algoritmo `MergeDelaunay` encontra o sítio $q_l \in L$ tal que o ângulo $\angle(p_l q_l p_r)$ é máximo. Analogamente, as linhas 10–15 determinam o sítio $q_r \in R$ tal que o ângulo $\angle(p_l q_r p_r)$ é máximo. A Linha 16 decide qual sítio $p \in \{q_l, q_r\}$ maximiza o ângulo $\angle(p_l p p_r)$. Logo, pelo Teorema 6, a L-R-aresta inserida na estrutura T pelas linhas 17–19 é a aresta de $\mathrm{Del}(L \cup R)$ sucessora da L-R-aresta $p_l p_r$ na y-ordem. Isto implica que o Algoritmo `MergeDelaunay` encontra todas as L-R-arestas de $L \cup R$.

Teorema 13. *As L-L e R-R-arestas removidas pelo algoritmo não são arestas do diagrama de Delaunay de $(L \cup R)$.*

Prova. Uma L-L-aresta $p_l q_l$ é removida pelo algoritmo na Linha 6 somente se o círculo determinado por q_l está à esquerda do segmento $p_l p_r$ e o sítio q imediatamente à esquerda do segmento $p_l q_l$ no sentido anti-horário pertence ao círculo determinado por $\triangle(p_l, p_r, q_l)$. Como $p_l p_r$ é uma aresta da triangularização de Delaunay de $L \cup R$ temos que se $p_l q_l$ é uma aresta de $\mathrm{Del}(L \cup R)$ então $\triangle(p_l, p_r, q_l)$ é um triângulo de $\mathrm{Del}(L \cup R)$. Isto contradiz o Teorema 1, já que q é um sítio contido no interior do $C(v)$ onde v é o vértice de Voronoi correspondente ao triângulo $\triangle(p_l, p_r, q_l)$.

Analogamente mostra-se que as L-L-arestas removidas pela Linha 12 do algoritmo não são arestas do diagrama de Delaunay de $L \cup R$. ∎

Teorema 14. *Ao final do Algoritmo `MergeDelaunay` o diagrama resultante corresponde a triangularização de Delaunay de $L \cup R$*

Prova. Primeiro mostraremos que o diagrama obtido é uma triangularização de $L \cup R$. No ínicio do algoritmo $p_l p_r$ corresponde a tangente inferior a $\mathrm{conv}(L)$ e $\mathrm{conv}(R)$ e depois de cada iteração, o segmento $p_l p_r$ terá uma de suas extremidades mais alta (maior y-coordenada). O laço das linhas 3–19 termina quando $p_l p_r$ alcança a tangente superior a $\mathrm{conv}(L)$ e $\mathrm{conv}(R)$. As faces interiores do diagrama serão ou faces de $\mathrm{Del}(L)$ ou $\mathrm{Del}(R)$ ou faces formadas por duas L-R-arestas e uma outra L-L- ou R-R-aresta. Em ambos os casos as faces do novo diagrama formam triângulos.

Precisamos verificar agora que a triangularização obtida de $L \cup R$ é de fato uma triangularização de Delaunay, ou seja, que as arestas do diagrama resultante são arestas de $\mathrm{Del}(L \cup R)$. Devido aos Teoremas 15 e 13 precisamos apenas verificar que arestas do diagrama resultante que são L-L- ou R-R-arestas são arestas de $\mathrm{Del}(L \cup R)$. Seja pq uma L-L-aresta que não foi removida pelo algoritmo. Se pq não é uma aresta de $\mathrm{Del}(L \cup R)$ então existe uma aresta rs de $\mathrm{Del}(L \cup R)$ que tem intersecção não vazia com pq. Já que pq é uma L-L-aresta então um dos sítios r ou s deve ser um sítio de R. Mas isto implica que rs é uma L-R-aresta e portanto foi corretamente encontrada pelo algoritmo. Logo, se pq não é uma aresta de $\mathrm{Del}(L \cup R)$ ela deve ter sido removida pelas linhas 5–7 do algoritmo. Analogamente se verifica que toda R-R-aresta do diagrama resultante ao final do algoritmo é uma aresta de $\mathrm{Del}(L \cup R)$. ∎

Estes teoremas completam a prova da correção do Algoritmo `MergeDelaunay`—o que implica a correção do Algoritmo `Delaunay2`.

5.7. **Análise do algoritmo.** Seja n o número de sítios de S. A complexidade de tempo do pré-processamento (ordenação dos sítios) é $O(n \log n)$.

No Algoritmo `MergeDelaunay` (passo `Combinar`) a Linha 2 pode ser executada em tempo $O(n)$ (veja o Algoritmo `Tangente-Inferior`). O laço entre as linhas 3–19 é executado $O(n)$ vezes. Em cada iteração do laço das linhas 5–7 uma L-L-aresta é removida. Logo, esse laço será executado $O(n)$ vezes. Analogamente, o laço das linhas 11–13 também será executado $O(n)$ vezes. Como a execução de cada Linha i, $(i = 3, \ldots, 19)$ gasta tempo $O(1)$ concluímos que a complexidade de tempo do Algoritmo `MergeDelaunay` é $O(n)$.

Se $T(n)$ é a função complexidade de tempo do Algoritmo `Delaunay2`, sem contar o custo do pré-processamento, temos que

$$T(n) \leq T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + cn = O(n \log n),$$

onde c é uma constante. Logo, a complexidade de tempo do Algoritmo `Delaunay2` é $O(n \log n)$.

Referências

[1] B. Delaunay, Sur la sphère vide, *Bull. Acad. Sci. USSR(VII)* (1934), 793–800, Classe Sci. Mat. Nat.

[2] L.H. Figueiredo e P.C.P. Carvalho, *Introdução à geometria computacional*, 18º Colóquio Brasileiro de Matemática, IMPA, 1991, QA758 F475i.

[3] L.J. Guibas e J. Stolfi, Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams, *ACM Transactions on Graphics* **4** (1985), no. 2, 74–123.

[4] D.T. Lee e B.J. Schachter, Two algorithms for constructing the Delaunay triangulation, *Int. J. Comput. Inf. Sci.* **9** (1980), no. 3, 219–242.

[5] F.P. Preparata e M.I. Shamos, *Computational geometry: An introduction*, Texts and Monographs in Computer Science, Springer-Verlag, New York, 1985, QA758 P927c.

[6] M.I. Shamos e D. Hoey, Closest point problems, *Proc. 16th Annual IEEE Symposium in Foundations of Computer Science*, 1975, pp. 151–162.

[7] M.I. Shamos e D. Hoey, Geometric intersection problems, *Proceedings of the 17th Annual IEEE Symposium on Foundations of Computer Science*, 1976, pp. 208–215.

Para provarmos o Teorema 4 precisaremos do seguinte lema:

Lema 1. *Sejam pq e rs segmentos que se intersectam em um único ponto o. Existe um círculo que contém p e q e não contém r e s se e somente se $\angle(qrp) + \angle(qsp) < \pi$ (equivalentemente, $\angle(rqs) + \angle(rps) > \pi$).*

Prova. Suponhamos que exista um círculo C que contém p e q e não contém r e s. Podemos assumir que p e q estão na fronteira de C. Sejam r' e s' as intersecções de rs com C (veja a figura 12). Então $\angle(qrp) + \angle(qsp) < \angle(qr'p) + \angle(qs'p) = \pi$. Reciprocamente, se $\angle(qrp) + \angle(qsp) < \pi$ então existem pontos r' e s' sobre o segmento rs tais que $\angle(qr'p)+\angle(qs'p) = \pi$. Estes pontos, juntamente com p e q determinam um círculo que não contém r e s. ∎

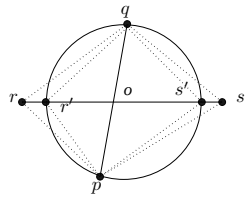

Figura 12. Círculo contendo p e q e excluindo r e s.

Prova do Teorema 4. Precisamos provar que as arestas de $\mathrm{Del}(S)$ não se intersectam a não ser nos vértices. Suponhamos que pq e rs sejam duas arestas de $\mathrm{Del}(S)$ que se intersectam em um ponto o. Como $V(p)$ e $V(q)$ compartilham uma aresta de Voronoi então existe um círculo contendo p e q que exclui r e s (cf. Teorema 2). Pelo lema anterior isto implica que $\angle(qrp) + \angle(qsp) < \pi$. Logo, $\angle(rqs) + \angle(rps) > \pi$ o que implica (ainda pelo lema anterior) que não existe um círculo contendo r e s que exclua p e q. Portanto, pelo Teorema 2, $V(r)$ e $V(s)$ não possuem uma aresta em comum. ∎

Prova do Teorema 5. Considere agora dois triângulos $T(v_1)$ e $T(v_2)$ de $\mathcal{T}(S)$, $v_1 \neq v_2$. Mostraremos que os interiores de $T(v_1)$ e $T(v_2)$ não se intersectam. Se $C(v_1)$ e $C(v_2)$ são disjuntos então $T(v_1)$ e $T(v_2)$ também são disjuntos. (Lembremos que $C(v)$ denota o círculo que tem $p_{v,1}, p_{v,2}$ e $p_{v,3}$ na fronteira.) Assuma então que $C(v_1)$ e $C(v_2)$ se intersectam. Pelo Teorema 1 temos que $C(v_1) \not\subset C(v_2)$ e $C(v_2) \not\subset C(v_1)$. Assim as fronteiras de $C(v_1)$ e $C(v_2)$ se intersectam em dois pontos q_1 e q_2. Seja l a reta determinada pelo segmento q_1q_2; esta reta também separa v_1 de v_2. (Veja a figura 13.) Afirmamos que l também separa o interior de $T(v_1)$ do interior de $T(v_2)$. De fato, assuma, por contradição, que existe um ponto de $T(v_1)$ no lado de l onde está v_2. (Veja a área sombreada da figura 13.) Como $T(v_1) \subset C(v_1)$ temos então que um vértice de $T(v_1)$ está no interior de $C(v_2)$ o que contradiz o Teorema 1. Logo, os interiores de $T(v_1)$ e $T(v_2)$ são disjuntos. ∎

Prova do Teorema 7. Notemos primeiro que $T(v) \subset \mathrm{conv}(S)$, o que implica que a união dos triângulos em $\mathcal{T}(S)$ está contida em $\mathrm{conv}(S)$.

Se p_ip_j é segmento na fronteira de $\mathrm{conv}(S)$ então p_ip_j é uma aresta do diagrama de Delaunay (isto segue do Teorema 2).

Toda aresta p_ip_j do diagrama de Delaunay $\mathrm{Del}(S)$ delimita uma ou duas regiões. Se p_ip_j delimita uma região do diagrama então existe um triângulo em $\mathcal{T}(S)$ incidente a p_ip_j: o triângulo formado por p_i, p_j e pelo sítio p_k que maximiza o ângulo $\angle(p_ip_kp_j)$ (cf. Teorema 6). Se p_ip_j não fizer parte da fronteira de $\mathrm{conv}(S)$ então existem dois triângulos de $\mathcal{T}(S)$ incidentes a p_ip_j, novamente determinados pelos sítios que formam ângulos máximos com p_ip_j; um em cada semiplano determinado pela reta que contém p_ip_j.

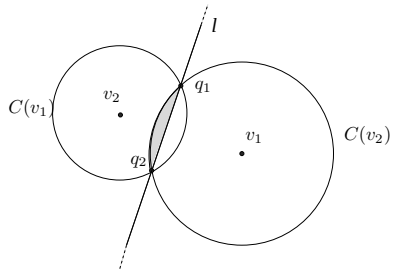

Figura 13. Não existe ponto de $T(v_1)$ no interior da região sombreada.

Seja agora p um ponto arbitrário de $\mathrm{conv}(S)$. Pelo que foi exposto acima sabemos que se p pertence a alguma aresta de $\mathrm{Del}(S)$ então p pertence a algum triângulo de $\mathcal{T}(S)$. Assim podemos supor que p não pertence a uma aresta de $\mathrm{Del}(S)$. Seja l uma semi-reta que tenha a sua extremidade em p e não contenha nenhum outro sítio de S. Seja p_ip_j a 'primeira' aresta de $\mathrm{Del}(S)$ que tem intersecção não vazia com l. (Veja a figura 14.) Afirmamos que p pertence ao triângulo T de $\mathcal{T}(S)$ formado pela aresta p_ip_j e pelo sítio p_k tal que $\angle(p_ip_kp_j)$ seja máximo. De fato, se p não pertence a T então a semi-reta l deveria intersectar alguma outra aresta de $T4$ e não somente p_ip_j. Este ponto de intersecção seria necessariamente anterior ao ponto de intersecção com p_ip_j, o que contradiz a escolha da aresta p_ip_j. Desta forma temos que os triângulos em $\mathcal{T}(S)$ cobrem $\mathrm{conv}(S)$. ∎

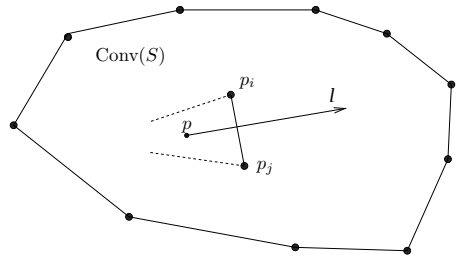

Figura 14. Ilustração da prova do Teorema 7.

Prova do Teorema 11. Considere a seguinte função que leva pontos do plano xy no parabolóide $z = x^2 + y^2$ (A figura 15 ilustra esta função.)

$$\lambda : (x, y) \mapsto (x, y, x^2 + y^2).$$

Primeiro mostraremos que a, b, c e d são cocirculares se e somente se $\lambda(a)$, $\lambda(b)$, $\lambda(c)$ e $\lambda(d)$ são coplanares.

Suponha primeiro que a, b, c e d sejam cocirculares. Se nós temos o caso degenerado onde os pontos são colineares então o determinante é zero como podemos verificar através da expansão da terceira coluna. Mas o valor do determinante também é igual ao volume sinalizado do tetraedro definido por $\lambda(a)$, $\lambda(b)$, $\lambda(c)$ e $\lambda(d)$. Como o volume é zero então os pontos devem ser coplanares. Caso contrário seja (p, q) o centro da circunferência de raio r que contém os pontos a, b, c e d. Nos

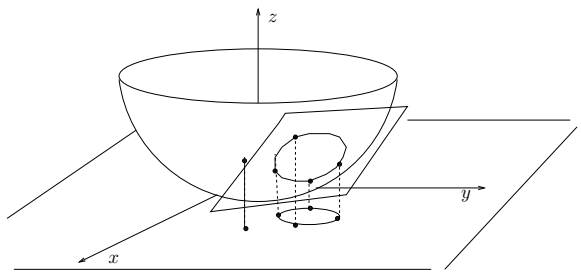

FIGURA 15. O parabolóide usado para calcular a primitiva `InCircle`.

temos que

$$(x_a - p)^2 + (y_a - q)^2 = r^2,$$

ou equivalentemente

$$-2p \cdot x_a - 2q \cdot y_a + 1 \cdot (x_a^2 + y_a^2) + (p^2 + q^2 - r^2) \cdot 1 = 0. \tag{1}$$

Relações análogas se verificam para os pontos b, c e d, e portanto nós temos uma dependência linear entre as colunas da matriz acima, o que implica que $\mathcal{D}(a, b, c, d) = 0$. Assim, concluímos novamente que $\lambda(a), \lambda(b), \lambda(c)$ e $\lambda(d)$ são coplanares.

Agora, reciprocamente, suponha que $\lambda(a), \lambda(b), \lambda(c)$ e $\lambda(d)$ são coplanares. Se os pontos a, b, c e d são colineares não há o que provar. Logo suponha que (digamos) a, b, c são não-colineares. Seja (p, q) o centro da circunferência de raio r determinada pelo triângulo $\triangle(a, b, c)$. Então a, b e c satisfazem a Equação 1 (as respectivas versões). Como a, b e c são não-colineares então as três linhas correspondentes da matriz acima são linearmente independentes. Como o determinante é nulo então a quarta linha da matriz (correspondente ao ponto d) deve ser combinação linear das três primeiras linhas, e portanto d também satisfaz a Equação 1, ou seja, d está na circunferência determinada por a, b e c.

O que acabamos de provar mostra que a projeção de seções planares do parabolóide $z = x^2 + y^2$ no plano xy determinam circunferências. O parabolóide é uma superfície convexa (para 'cima') então em uma seção de um plano com o parabolóide a parte do parabolóide abaixo do plano tem sua projeção no plano xy no interior do círculo determinado por a, b e c e os pontos do parabolóide tem sua projeção fora deste círculo. A partir disto vemos que para decidir se um ponto está dentro ou fora do círculo determinado pelos pontos a, b, c e d basta aplicarmos a primitiva `Teste-de-Orientação`$(\lambda(a), \lambda(b), \lambda(c), \lambda(d))$, que corresponde ao volume sinalizado do tetraedro determinado pelos pontos $\lambda(a), \lambda(b), \lambda(c)$ e $\lambda(d)$. O resultado da primitiva `Teste-de-Orientação`$(\lambda(a), \lambda(b), \lambda(c), \lambda(d))$ é dado através do sinal do determinante $\mathcal{D}(a, b, c, d)$. ∎

Prova do Teorema 12. A primeira parte do teorema segue do fato de Del(S) ser uma triangularização dos pontos em S (cf. Teorema 8). Logo, o terceiro lado do triângulo tem ambos os seus vértices em L ou em R. Sejam p, q e r os sítios do triângulo em questão e suponhamos, sem perda de generalidade, que $p \in L$, $q, r \in R$ (ou seja, pq e pr são as L-R-arestas consecutivas na y-ordem). Precisamos mostrar que qr é uma aresta de Del(L). Como o triângulo $\triangle(p, q, r)$ é um triângulo de Del$(L \cup R)$ temos então que qr é uma aresta de Del$(L \cup R)$. Pelo Teorema 2 existe um círculo C contendo q e r tal que todos os demais sítios de S são exteriores a C. Logo, em particular, C é um círculo que contém q e r e não contém nenhum outro sítio de R. Portanto, novamente pelo Teorema 2, temos que qr é uma aresta de Del(R). ∎

O próximo teorema é parte da análise do algoritmo de divisão e conquista.

Teorema 15. *Seja S um conjunto de sítios em E^2 e considere um sítio particular p e uma reta l passando por p. Assuma por conveniência que l é uma reta horizontal. Sejam $p_1, p_2, \ldots, p_k$ $(k \geq 1)$ os sítios de S acima de l que são vizinhos de p em Del(S) listados em ordem anti-horária. Seja q um ponto em l à direita de p e seja $\alpha_i := \angle(pp_iq)$ $(i = 1, \ldots, k)$. Então existe um índice j, $1 \leq j \leq k$ tal que*

(1) *$\alpha_i < \alpha_{i+1}$ para $i = 1, \ldots, j-1$; e*
(2) *$\alpha_i > \alpha_{i+1}$ para $i = j, \ldots, k$.*

Prova. Primeiro mostraremos que se q_i denota a intersecção da circunferência determinada pelo triângulo $\triangle(p, p_i, p_{i+1})$ $(i = 1, \ldots, k-1)$ com a reta l então a seqüência $q_1, q_2, \ldots, q_{k-1}$ se move monotonicamente para a esquerda. (Veja a figura 16.)

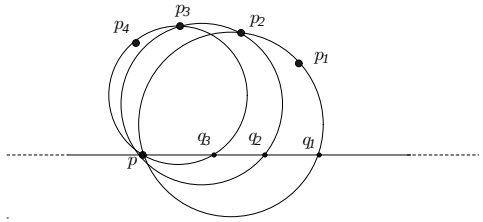

FIGURA 16. A seqüência $q_1, \ldots, q_{k-1}$ se move monotonicamente para a esquerda.

Para $i = 1, \ldots, k-3$ temos que o sítio p_{i+2} não pertence ao círculo determinado por $\triangle(p, p_i, p_{i+1})$. Logo, $\angle(pp_{i+2}p_{i+1}) + \angle(pq_ip_{i+1}) < \pi$ Portanto, q_i não pertence ao círculo determinado por $\triangle(p, p_{i+1}, p_{i+2})$, o que implica que q_{i+1} está à esquerda de q_i.

Para terminarmos a prova do teorema, basta notarmos que se q_i está à direita de q então q pertence ao círculo determinado por $\triangle(p, p_i, p_{i+1})$ (o que implica que $\angle(pqp_i) + \angle(pp_{i+1}p_i) > \pi$, ou, equivalentemente, $\angle(p_{i+1}pq) + \angle(p_{i+1}p_iq) < \pi$). Logo, p_{i+1} pertence ao círculo determinado por $\triangle(p, q, p_i)$. Analogamente, se q_i esta à esquerda de q então p_{i+1} não pertence ao círculo determinado por $\triangle(p, p_i, q)$. Com isto completamos a prova do teorema. ∎