Geometria obliczeniowa

Krzysztof Gdawiec

UNIWERSYTET ŚLĄSKI
INSTYTUT INFORMATYKI

Diagramy Voronoi

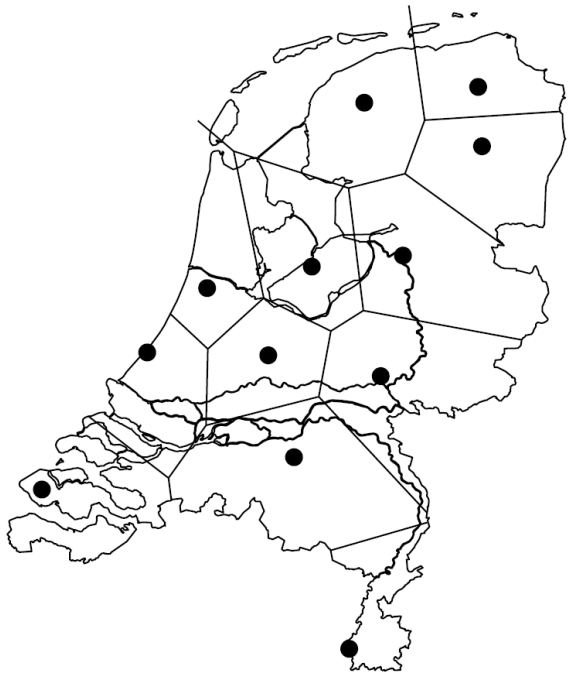
Powiedzmy, że mamy ustalony zbiór głównych miejsc zwanych centrami, w których oferuje się towary lub usługi. Chcemy wiedzieć dla każdego centrum, gdzie żyją ludzie, którzy nabywają w nim towary i realizują usługi.

W celu zbadania tej kwestii robimy pewne założenia:

- cena poszczególnych towarów i usług jest taka sama w każdym centrum,
- koszt nabycia towaru lub wykonania usługi jest równy cenie plus koszt transportu do danego centrum,
- koszt transportu jest równy odległości euklidesowej do centrum razy ustalona cena za jednostkową odległość,
- konsumenci, nabywając towary i realizując usługi, próbują minimalizować koszty.

Interesuje nas geometryczna interpretacja modelu. Nasze założenia powodują podział całej powierzchni na obszary (rejony zbytu danych centrów) takie, że ludzie, którzy żyją na tym samym obszarze udają się do tego samego centrum.

Zatem rejon zbytu dla danego centrum składa się ze wszystkich takich punktów, dla których jest ono bliżej niż każde inne.



Model, w którym każdy punkt jest przydzielony do najbliższego centrum nazywa się modelem przydziału Voronoi. Podział spowodowany przez ten model nazywa się diagramem Voronoi zbioru centrów.

Model, w którym każdy punkt jest przydzielony do najbliższego centrum nazywa się modelem przydziału Voronoi. Podział spowodowany przez ten model nazywa się diagramem Voronoi zbioru centrów.

Oznaczmy odległość euklidesową między punktami p i q przez $d(p, q)$

$$d(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}.$$

Niech $P = \{p_1, p_2, \ldots, p_n\}$ będzie zbiorem n różnych punktów na płaszczyźnie. Punkty te będą naszymi centrami.

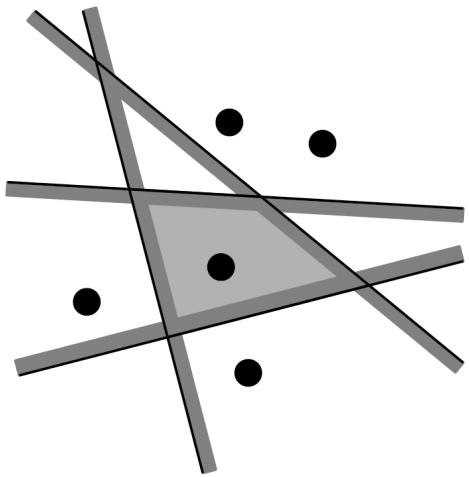
Diagram Voronoi dla P definiujemy jako podział płaszczyzny na n komórek, jedna dla każdego centrum, o tej własności, że punkt q leży w komórce odpowiadającej $p_i \iff$ $d(q, p_i) < d(q, p_j)$ dla każdego $p_j \in P$ takiego, że $j \neq i$.

Diagram Voronoi dla P oznaczamy przez $Vor(P)$ lub $\mathcal{V}$. Komórkę z $Vor(P)$, która odpowiada centrum p_i oznaczamy $V(p_i)$ i nazywamy komórką Voronoi (ang. Voronoi cell) dla p_i.

Jeśli dane są dwa punkty p_i, p_j, to zbiór punktów bliższych p_i niż p_j jest półpłaszczyzną zawierającą p_i wyznaczoną przez prostą prostopadłą dzielącą na pół odcinek $\overline{p_i p_j}$. Oznaczamy tę półpłaszczyznę przez $H(p_i, p_j)$.

Zatem komórka Voronoi dla p_i dana jest wzorem:

$$V(p_i) = \bigcap_{i \neq j} H(p_i, p_j).$$



Wzór z poprzedniego slajdu daje nam naiwną metodę tworzenia diagramu Voronoi, tzn. tworzymy kolejne komórki Voronoi jako przecięcie odpowiednich półpłaszczyzn.

Czas takiego algorytmu wynosi $\mathcal{O}(n^2 \log n)$. Za chwilę podamy algorytm o czasie $\theta(n \log n)$.

Wzór z poprzedniego slajdu daje nam naiwną metodę tworzenia diagramu Voronoi, tzn. tworzymy kolejne komórki Voronoi jako przecięcie odpowiednich półpłaszczyzn.

Czas takiego algorytmu wynosi $\mathcal{O}(n^2 \log n)$. Za chwilę podamy algorytm o czasie $\theta(n \log n)$.

Własności diagramu Voronoi

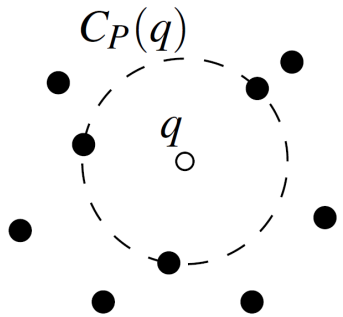
Niech P będzie zbiorem n centrów na płaszczyźnie. Jeśli wszystkie lokalizacje są współliniowe, to $Vor(P)$ składa się z $n-1$ równoległych prostych. W przeciwnym razie $Vor(P)$ jest spójny i jego krawędzie są albo odcinkami albo półprostymi.

Dla $n \geqslant 3$ liczba wierzchołków w diagramie Voronoi dla zbioru n centrów na płaszczyźnie wynosi co najwyżej $2n - 5$, a liczba krawędzi wynosi co najwyżej $3n - 6$.

Załóżmy, że żadne cztery punkty ze zbioru P nie leżą na obwodzie jednego koła. Wówczas:

- Każdy wierzchołek diagramu Voronoi jest wspólnym przecięciem dokładnie trzech krawędzi diagramu.
- Każdy najbliższy sąsiad $p_i \in P$ definiuje krawędź komórki Voronoi $V(p_i)$.
- Komórka $V(p_i)$ jest nieograniczona $\iff$ p_i jest punktem na brzegu otoczki wypukłej zbioru P.

Niech q będzie punktem. Definiujemy największy pusty okrąg dla q względem P jako największy okrąg z q będącym jego środkiem, który nie zawiera żadnego centrum z P w swoim wnętrzu. Oznaczamy go $C_P(q)$.



Punkt q jest wierzchołkiem $Vor(P) \iff C_P(q)$ zawiera na swoim brzegu trzy lub więcej centrów.
Symetralna między centrami p_i, p_j określa krawędź $Vor(P) \iff$ istnieje taki punkt q na symetralnej, że $C_P(q)$ zawiera na swoim brzegu p_i i p_j, ale nie zawiera żadnego innego centrum.

Obliczanie diagramu Voronoi

Jeśli dany jest podział $\{P_1, P_2\}$ zbioru P, to niech $\sigma(P_1, P_2)$ oznacza zbiór krawędzi Voronoi wspólnych dla par komórek $V(p_i)$ i $V(p_j)$ diagramu $Vor(P)$ dla $p_i \in P_1$ i $p_j \in P_2$.

Zbiór $\sigma(P_1, P_2)$ jest zbiorem krawędzi podgrafu grafu $Vor(P)$ i ma własności:

- $\sigma(P_1, P_2)$ składa się z cykli i łańcuchów rozłącznych krawędzi. Jeśli łańcuch zawiera tylko jedną krawędź, to jest to prosta; w przeciwnym razie jego dwie krawędzie skrajne są półprostymi.
- Jeśli P_1 i P_2 są oddzielone liniowo, to $\sigma(P_1, P_2)$ składa się z pojedynczego łańcucha monotonicznego (przecięcie z prostą prostopadłą do prostej oddzielającej jest punktem).

Twierdzenie. Jeśli P_1 i P_2 są rozdzielone liniowo pionową prostą tak, że P_1 jest na lewo od P_2, to diagram Voronoi $Vor(P)$ jest sumą $Vor(P_1) \cap \pi_L$ i $Vor(P_2) \cap \pi_R$, gdzie π_L jest lewą częścią a π_R prawą częścią płaszczyzny względem $\sigma(P_1, P_2)$.

Twierdzenie daje nam pomysł na algorytm typu „dziel i rządź" obliczania diagramu Voronoi.

Załóżmy, że punkty z $P = \{p_1, p_2, \ldots, p_n\}$ są ułożone w rosnącym porządku względem współrzędnej x.

Algorithm 4.4.1 (Divide-and-conquer method)

Input: Number n of generators and list $P = (p_1, p_2, \ldots, p_n)$ of the generators arranged in increasing order of the x coordinates.

Output: Voronoi diagram $\mathcal{V}$ for P.

Procedure:

Step 1. If $n \leq 3$, then construct the Voronoi diagram $\mathcal{V}$ for P directly and go to Step 3.

Step 2. Otherwise do the following.

2.1. Let t be the integral part of $n/2$, and divide P into $P_L = (p_1, \ldots, p_t)$ and $P_R = (p_{t+1}, \ldots, P_n)$.

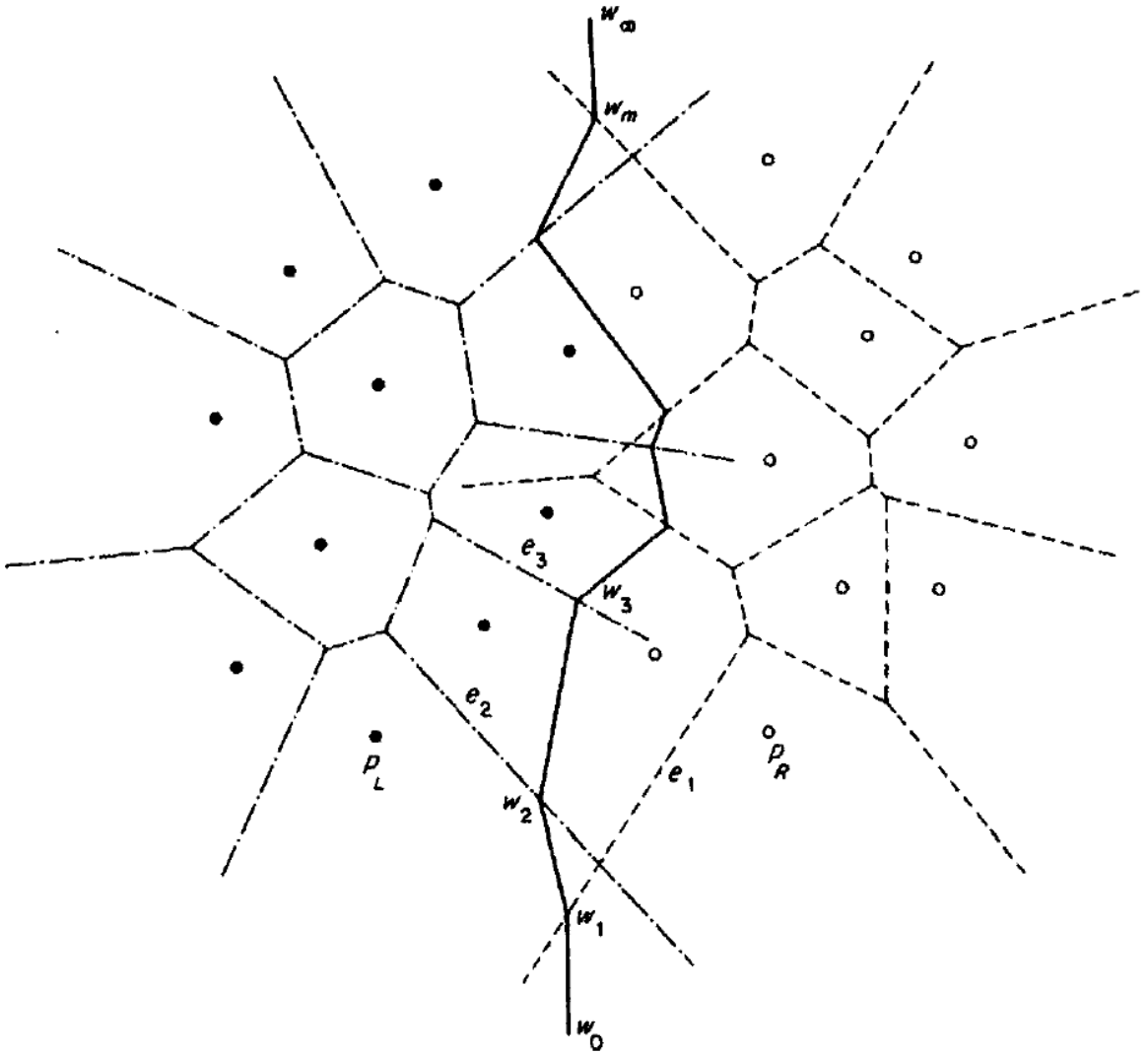
2.2. Construct the Voronoi diagram $\mathcal{V}_L$ for P_L by Algorithm 4.4.1.

2.3. Construct the Voronoi diagram $\mathcal{V}_R$ for P_R by Algorithm 4.4.1.

2.4. Merge $\mathcal{V}_L$ and $\mathcal{V}_R$ into the Voronoi diagram $\mathcal{V}$ for P (this can be done by Algorithm 4.4.3.).

Step 3. Return $\mathcal{V}$.

Najważniejszym krokiem w algorytmie jest krok łączenia dwóch diagramów Voronoi (krok 2.4 w algorytmie).



Dla danego punktu q niech $d(q, P)$ oznacza minimalną odległość q od zbioru $P = \{p_1, \ldots, p_m\}$, tzn.

$$d(q, P) = \min_{i \in \{1,\ldots,m\}} d(q, p_i).$$

Rozważmy punkt q poruszający się od lewej do prawej po dowolnej prostej poziomej. Ponieważ P_L jest po lewej względem P_R, więc istnieje dokładnie jeden punkt q^* taki, że $d(q^*, P_L) = d(q^*, P_R)$ i $d(q, P_L) < d(q, P_R)$ zanim q osiągnie q^* i $d(q, P_L) > d(q, P_R)$ po przekroczeniu q^* przez q.

W miarę jak prosta pozioma porusza się od dołu w górę punkt q^* porusza się w sposób ciągły śledząc łamaną pomiędzy P_L i P_R. Naszym zadaniem jest znalezienie tej łamanej i usunięcie krawędzi z $\mathcal{V}_L$ i $\mathcal{V}_R$, które leżą na prawo i lewo (odpowiednio) względem tej łamanej.

Algorytm śledzący łamaną musi zostać zainicjowany wartością startową. Do tego celu będziemy musieli znaleźć tzw. dolny odcinek wspierający (ang. lower common support).

Niech $L(p, q)$ oznacza skierowaną prostą przechodzącą przez p i q oraz niech $\overrightarrow{pq}$ oznacza skierowany odcinek łączący p i q.

Niech U będzie wielokątem wypukłym danym przez wierzchołki w porządku przeciwnym do ruchu wskazówek zegara $(u_1, u_2, \ldots, u_s, u_1)$. Dla wierzchołka $u \in U$ niech $cnext[u]$ i $ccnext[u]$ oznacza poprzednik i następnik (odpowiednio).

Niech U_L i U_R będą wielokątami wypukłymi takimi, że U_L leży na lewo od U_R, tzn. współrzędne x wierzchołków z U_L są mniejsze od współrzędnych x wierzchołków z U_R.

Dla wierzchołka $u \in U_L$ i wierzchołka $w \in U_R$ prostą $L(u, w)$ nazywamy odcinkiem wspierającym (ang. common support) U_L i U_R jeśli wszystkie wierzchołki w U_L i U_R leżą po tej samej stronie $L(u, w)$.

Wyróżniamy dwa rodzaje odcinków wspierających: dolny (ang. lower) i górny (ang. upper). Dolny odcinek wspierający to odcinek wspierający, dla którego wszystkie wierzchołki leżą powyżej niego. Jeśli wszystkie wierzchołki leżą poniżej odcinka wspierającego, to nazywamy go górnym odcinkiem wspierającym.

Algorithm 4.4.2 (Lower common support)

Input: Two convex polygons U_L and U_R such that the maximum x coordinate over all vertices in U_L is smaller than the minimum x coordinate over all vertices in U_R.

Output: Pair consisting of the vertex u in U_L and the vertex w in U_R such that $L(u, w)$ forms the lower common support of U_L and U_R.

Procedure:

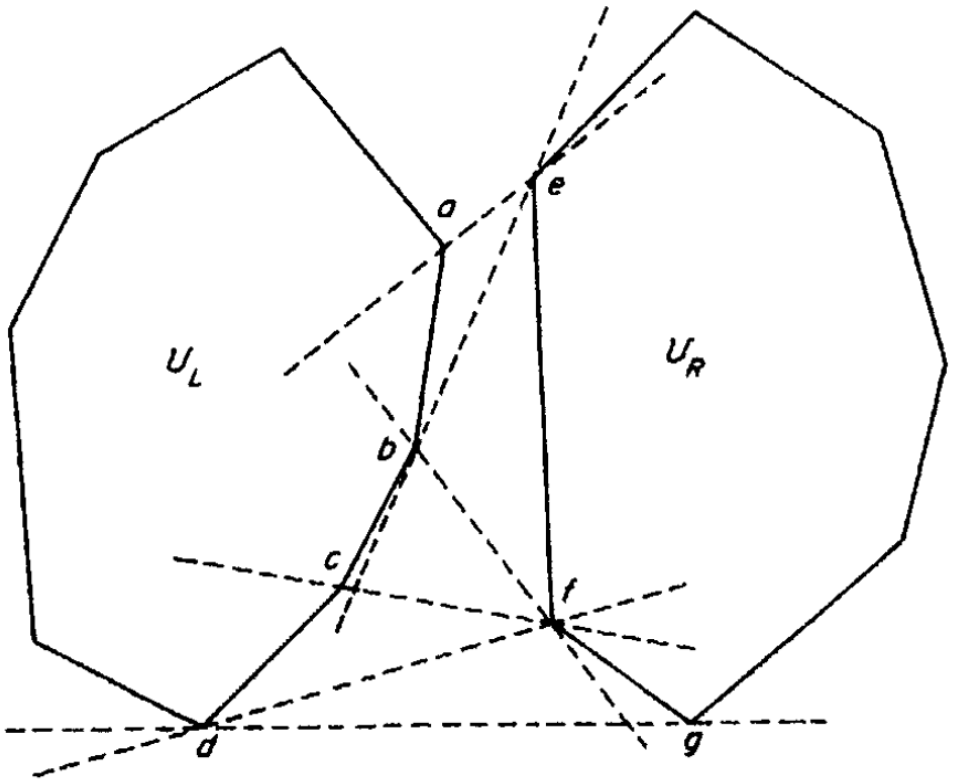
Step 1. Find the vertex u in U_L with the largest x coordinate, and the vertex w in U_R with the smallest x coordinate.

Step 2. Do 2.1 and 2.2 alternately until u and w are not changed any more.

2.1. While vertex cnext[u] is lower than $L(u, w)$, repeat $u \leftarrow$ cnext[u].

2.2. While vertex ccnext[w] is lower than $L(u, w)$, repeat $w \leftarrow$ ccnext[w].

Step 3. Return $L(u, w)$.



Algorytm 4.4.2 działa w czasie $\mathcal{O}(n)$, gdzie n oznacza łączną liczbę wierzchołków w U_L i U_R.

Niech $p_L \in P_L$ i $p_R \in P_R$ będzie parą wierzchołków wyznaczających dolny odcinek wspierający $L(p_L, p_R)$ otoczki wypukłej zbioru P_L i P_R. Oznaczmy przez $b(p_L, p_R)$ symetralną odcinka $\overline{p_L p_R}$. Wyznacza ona najniższą krawędź szukanej łamanej.

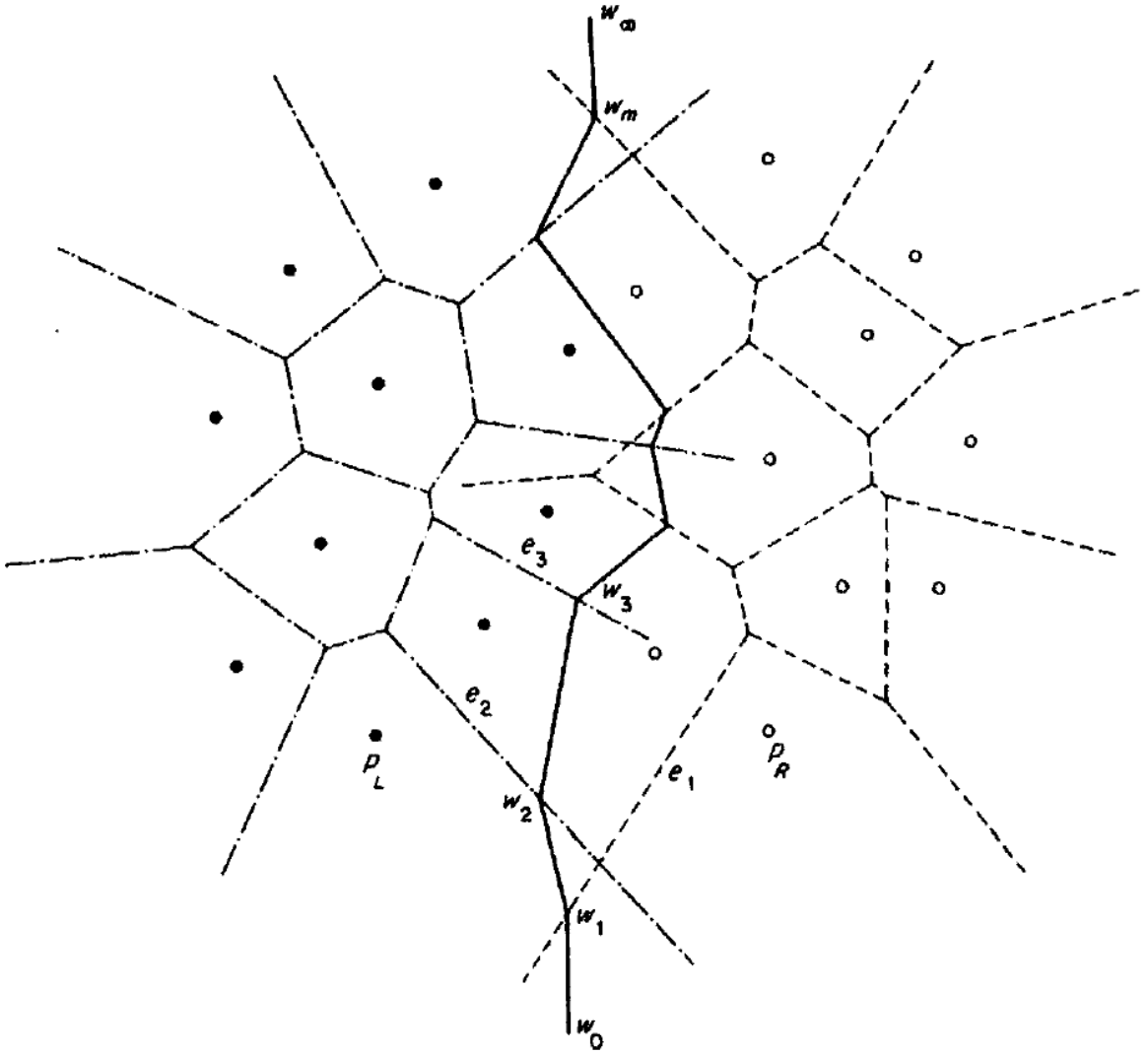
Rozważmy punkt w poruszający się po $b(p_L, p_R)$ od dołu w górę. Jeśli współrzędna y punktu w jest wystarczająco mała, to w należy do $V(p_L)$ i $V(p_R)$. Niech w_0 będzie punktem w nieskończoności w kierunku ujemnej osi y wzdłuż $b(p_L, p_R)$.

Niech w_1 będzie punktem, który przecina po raz pierwszy krawędź Voronoi e_1 diagramu $\mathcal{V}_L$ lub $\mathcal{V}_R$. Wówczas półprosta $\overline{w_0 w_1}$ jest pierwszą krawędzią Voronoi, a w_1 jest pierwszym wierzchołkiem Voronoi.

W punkcie w_1 poruszający się punkt w zmienia komórkę w $\mathcal{V}_L$ lub $\mathcal{V}_R$ zależnie czy e_1 jest krawędzią $\mathcal{V}_L$ lub $\mathcal{V}_R$. Jeśli e_1 jest krawędzią $\mathcal{V}_L$, to zamieniamy p_L na centrum komórki znajdującej się po drugiej stornie e_1. Jeśli e_1 jest krawędzią $\mathcal{V}_R$, to zamieniamy p_R na centrum komórki znajdującej się po drugiej stornie e_1.

Następnie tworzymy nową symetralną $b(p_L, p_R)$. Punkt w_1 należy do tej symetralnej. Ponownie poruszamy się po symetralnej w górę aż napotkamy krawędź $\mathcal{V}_L$ lub $\mathcal{V}_R$. Oznaczmy punkt przecięcia przez w_2, który jest drugim wierzchołkiem Voronoi, a $\overline{w_1 w_2}$ jest drugą krawędzią diagramu Voronoi.

Powtarzamy to wszystko aż para punktów (p_L, p_R) nie stanie się parą punktów definiujących górny odcinek wspierający $L(p_L, p_R)$.


w_∞
w_m
e_3
w_3
e_2
p_L
p_R
e_1
w_2
w_1
w_0

Algorithm 4.4.3 (Merger of two Voronoi diagrams)

Input: Two Voronoi diagrams $\mathcal{V}_L$ and $\mathcal{V}_R$ for generator sets P_L and P_R, respectively, such that the generators in $\mathcal{V}_L$ have smaller x coordinates than those in $\mathcal{V}_R$.

Output: Voronoi diagram for $P_L \cup P_R$.

Procedure:

Step 1. Construct the convex hull of P_L and that of P_R.

Step 2. Find the lower common support $L(P_L, P_R)$ by Algorithm 4.4.2.

Step 3. $w_0 \leftarrow$ the point at infinity downward on $b(p_L, p_R)$, and $i \leftarrow 0$.

Step 4. While $L(p_L, p_R)$ is not the upper common support, repeat 4.1, ..., 4.4.

4.1. $i \leftarrow i + 1$.

4.2. Find the point a_L (other than w_{i-1}) of the intersection of $b(p_L, p_R)$ with the boundary of $V(p_L)$.

4.3. Find the point a_R (other than w_{i-1}) of the intersection of $b(p_L, p_R)$ with the boundary of $V(p_R)$.

4.4. If a_L has a smaller y coordinate than a_R,

$w_i \leftarrow a_L$, and

$p_L \leftarrow$ the generator on the other side of the Voronoi edge containing a_L.

Otherwise

$w_i \leftarrow a_R$, and

$p_R \leftarrow$ the generator on the other side of the Voronoi edge containing a_R.

Step 5. $m \leftarrow i$.

$w_{m+1} \leftarrow$ the point at infinity upward on $b(p_L, p_R)$.

Step 6. Add the polygonal line $(\overline{w_0 w_1}, \overline{w_1 w_2}, \ldots, \overline{w_m w_{m+1}})$, and delete from $\mathcal{V}_L$ the part to the right of the polygonal line and delete from $\mathcal{V}_R$ the part to the left of the polygonal line. Return the resultant diagram.

Algorytm łączenia diagramów Voronoi (algorytm 4.4.3) działa w czasie $\mathcal{O}(n)$.

Zatem algorytm wyznaczania diagramu Voronoi (algorytm 4.4.1) działa w czasie $\theta(n \log n)$.

Algorytm łączenia diagramów Voronoi (algorytm 4.4.3) działa w czasie $\mathcal{O}(n)$.

Zatem algorytm wyznaczania diagramu Voronoi (algorytm 4.4.1) działa w czasie $\theta(n \log n)$.

Diagram, który omawialiśmy dotyczył odległości euklidesowej. Prowadzone są również badania nad diagramami Voronoi z użyciem metryki L_p dla $1 \leqslant p \leqslant \infty$. Metryka ta dana jest wzorem:

$$d_p(u, v) = (|u_x - v_x|^p + |u_y - v_y|^p)^{\frac{1}{p}}$$