An $O(n^4)$ algorithm to construct all Voronoi diagrams for k-nearest neighbor searching in the Euclidean plane

by

Frank Dehne

ABSTRACT

This paper presents an algorithm, that constructs all Voronoi diagrams for k nearest neighbor searching simultaneously. Its space an time complexity of $O(n^4)$ is shown to be optimal.

F. Dehne, "An $O(n^4)$ algorithm to construct all voronoi diagrams for k nearest neighbor searching in the euclidean plane," in Proc. *International Colloquium on Automata, Languages and Programming*, Barcelona (Spain), 1983, Springer Verlag, Lecture Notes in Computer Science, Vol. 154.

1. INTRODUCTION

In /ShHo75/ Shamos and Hoey introduce the idea of generalized Voronoi diagrams to get an optimal solution of the k nearest neighbor problem and give an $O(N \log N)$ algorithm to construct the order one diagram.
Lee /Le81/ extends this to an algorithm, that computes an order k diagram in $O(k^2 N \log N)$.

To answer k nearest neighbor queries with arbitrary k we now want to construct all Voronoi diagrams.
This paper presents a simple solution of this problem.
The given algorithm has time and space complexity $O(N^4)$ and is shown to be optimal. Its implementation is not very difficult and the constant factors for the complexity are expected to be quite good.

2. K NEAREST NEIGHBOR SEARCHING AND GENERALIZED VORONOI DIAGRAMS

Let $S:=\{s_1,\ldots,s_N\}$ be a set of $N\geq 3$ points in the Euclidean plane $\mathbb{E}^2$ (with distance measure d).
We shall assume that no more than three of these points lie on a circle and that they are not all collinear.
To answer a query for the k nearest neighbors of a point $q\in\mathbb{E}^2$, we have to find a subset $A\subset S$ with $|A|=k$ and $(\forall x\in A, y\in S-A): d(q,x)\leqq d(q,y)$.
With $B(x,y):=\{z\in\mathbb{E}^2 / d(x,z)=d(y,z)\}$ and $h(x,y):=\{z\in\mathbb{E}^2 / d(x,z)\leqq d(y,z)\}$ we call

$$v(A):=\bigcap_{\substack{x\in A\\ y\in S-A}} h(x,y)$$

the Voronoi polygon of $A\subset S$ and
$V_k(S):=\{v(A) / A\subset S \text{ and } |A|=k\}-\{\phi\}$ the (generalized) Voronoi diagram of order k.

It is easy to see that $V_k(S)$ can be described by a straight line graph, that divides the Euclidean plane into a finite number of convex polygons* (the Voronoi polygons) and that $A \subset S$ is a set of k nearest neighbors of all query points $q \in v(A) \in V_k(S)$.
With these postulates we can solve the k nearest neighbor problem in the following way:

A. Construct all $V_k(S)$ for $1 \leq k \leq N-1$ (preprocessing)

B. For every query $(q \in \mathbb{E}^2, k \in \{1,\ldots,N-1\})$ find a $v(A) \in V_k(S)$ with $q \in v(A)$.

For part B Kirkpatrick (/Ki81/) has already found an optimal algorithm that answers a query in $O(k+\log N)$ steps.
The next two sections of this paper will describe an optimal solution of part A.

3. PROPERTIES OF GENERALIZED VORONOI DIAGRAMS

Every Voronoi edge (edge of a Voronoi diagram) is part of a bisector $B(x,y)$ with $x,y \in S$.
So let $\overline{B}_k(x,y)$ be the part of $B(x,y)$ that is Voronoi edge of $V_k(S)$.
With every Voronoi polygon being convex and only up to three points of S lying on a circle, every Voronoi point (point of a Voronoi diagram) has degree three and is the center of exactly three points of S.

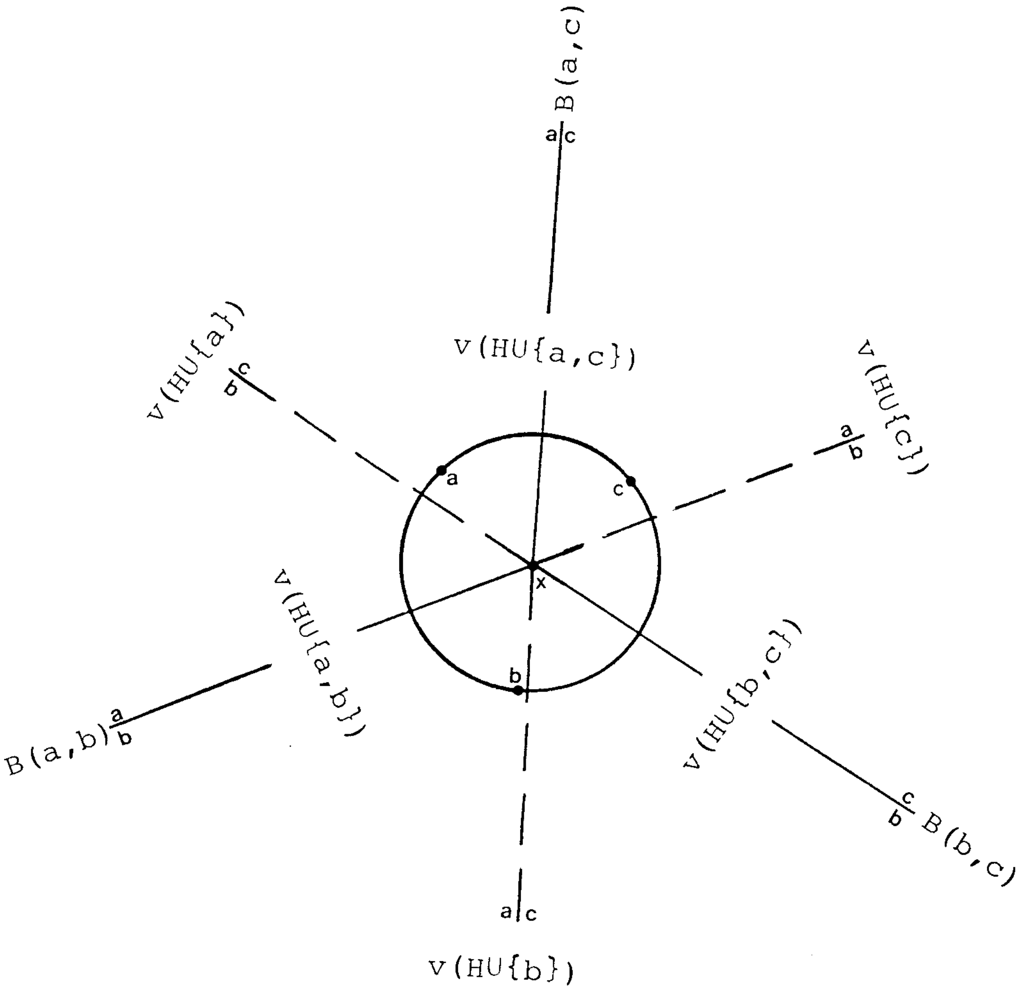
---

*In the following this paper only will operate with this diagram also called $V_k(S)$.

Now we can prove the following theorem.

<u>Theorem 1:</u>

Let $x \in \mathbb{E}^2$ be the center of $a,b,c \in S$ and $H := \{z \in S / d(x,z) < d(x,a)\}$ with $|H| =: k \leq N-3$, then x is Voronoi point of $V_{k+1}(S)$ and $V_{k+2}(S)$.
The Voronoi edges and polygons that are incident upon x are given by the following diagram.



----: Voronoi edge of $V_{k+2}(S)$

——: Voronoi edge of $V_{k+1}(S)$

<u>Proof:</u>

$H \cup \{a\}$, $H \cup \{b\}$ and $H \cup \{c\}$ are sets of $k+1$ nearest neighbors of x. So $x \in v(H \cup \{a\}) \cap v(H \cup \{b\}) \cap v(H \cup \{c\})$ is a Voronoi point of $V_{k+1}(S)$. Because $H \cup \{a,b\}$, $H \cup \{b,c\}$ and $H \cup \{a,c\}$ are sets of $k+2$ nearest neighbors of x, it is also a Voronoi point of $V_{k+2}(S)$. With this the construction of the above diagram is trivial.

□

The next theorem will demonstrate, that every Voronoi point can be constructed as described in theorem 1.

<u>Theorem 2:</u>

Let a Voronoi point $x \in v(A) \cap v(B) \cap v(C)$ with $v(A), v(B), v(C) \in V_i(S)$ be the center of $a,b,c \in S$ and $H := \{z \in S / d(x,z) < d(x,a)\}$ with $|H| =: k \leq N-3$, then

$(\{A,B,C\} = \{H \cup \{a\}, H \cup \{b\}, H \cup \{c\}\}$ and $i = k+1)$

or

$(\{A,B,C\} = \{H \cup \{a,b\}, H \cup \{b,c\}, H \cup \{a,c\}\}$ and $i = k+2)$.

<u>Proof:</u>

Let without loss of generality $v(A) \cap v(B) = \overline{B}_i(a,b)$, $v(B) \cap v(C) = \overline{B}_i(b,c)$ and $v(A) \cap v(C) = \overline{B}_i(a,c)$, then theorem 2 follows from the next three statements.

(1) Because $v(A) = \bigcap_{\substack{x \in A \\ y \in S-A}} h(x,y)$ and $\overline{B}_i(a,b)$ borders $v(A)$,
there exists a $x \in A$ and $y \in S-A$ with $\{x,y\} = \{a,b\}$, getting $A \cap \{a,b,c\} \neq \phi$ and $\{a,b,c\} \not\subset A$.
In the same way we get $B \cap \{a,b,c\} \neq \phi$, $\{a,b,c\} \not\subset B$, $C \cap \{a,b,c\} \neq \phi$ and $\{a,b,c\} \not\subset C$.

(2) $H \subset A$, $H \subset B$, $H \subset C$.
Proof of $H \subset A$ with $a \in A$ and $b \notin A$ (see (1)):
If there would be a $z \in H$ with $z \notin A$, this would be a contradiction to $x \in v(A)$ because of $d(x,z) < d(x,a)$.

(3) $A,B,C \subset H \cup \{a,b,c\}$.
Proof of $A \subset H \cup \{a,b,c\}$ with $a \in A$ and $b \notin A$ (see (1)):
If there would be a $z \in A$ with $z \notin H \cup \{a,b,c\}$, we would have $d(x,z) > d(x,a) = d(x,b)$ with $z \in A$ and $b \notin A$; a contradiction to $x \in v(A)$.

□

With theorem 2 we know theorem 1 describing all Voronoi points, edges and polygons of all $V_k(S)$ $(1 \leq k \leq N-1)$.
This is the main idea for the algorithm in section 4.

For the analysis of this algorithm the next theorem gives us the number of Voronoi points, edges and polygons we have to compute.

Theorem 3:

Let I_k be the number of Voronoi points, E_k the number of edges and N_k the number of polygons of $V_k(S)$ $(1\leq k\leq N-1)$, then we get

(i) $(\forall 1\leq k\leq N-1): \Theta(I_k)=\Theta(E_k)=\Theta(N_k)$ *

(ii) $(\forall 1\leq k\leq N-1): N_k\in O(k(N-k))\subset O(kN)$

(iii) $\sum_{k=1}^{N-1} N_k\in\Theta(N^3)$

(iv) $\sum_{k=1}^{N-1} kN_k\in\Theta(N^4)$.

Proof:

(i),(ii),(iii): see /ShHo75/ and /Le81/.

(iv):

From (ii) and $k\leq N$ we get $N_k\in O(N^2)$. So let us take an $a\in\mathbb{N}$ with $N_k\leq aN^2$ (for large N), then we get

$$\sum_{k=1}^{N-1} kN_k \leq \sum_{k=1}^{N-1} kaN^2 = \frac{a}{2}(N-1)N^3 \in O(N^4).$$

To prove $\sum_{k=1}^{N-1} kN_k\in\Omega(N^4)$ we define $I:=\{1\leq k\leq N-1/N_k\in\Omega(N^2)\}$ and $J:=\{1\leq k\leq N-1/N_k\notin\Omega(N^2)\}$ and show $|I|\in\Omega(N)$.

Assuming $|I|\notin\Omega(N)$ we would get $\sum_{k\in I} N_k\notin\Omega(N^3)$ from $N_k\in O(N^2)$.

---

*Having two functions $f,g:\mathbb{N}\to\mathbb{R}$ we say

$f\in O(g) \;:\Leftrightarrow (\exists m\in\mathbb{N}, c\in\mathbb{N})(\forall n\geq m): f(n)\leq c\cdot g(n)$,

$f\in\Omega(g) \;:\Leftrightarrow (\exists m\in\mathbb{N}, c\in\mathbb{N})(\forall n\geq m): f(n)\geq c\cdot g(n)$ and

$f\in\Theta(g) \;:\Leftrightarrow (f\in O(g)\wedge f\in\Omega(g))$.

By definition of J we get $\sum_{k\in J} N_k \notin \Omega(N^3)$ and further more

$$\sum_{k=1}^{N-1} N_k = \sum_{k\in I} N_k + \sum_{k\in J} N_k \notin \Omega(N^3);$$ a contradiction to (iii).

Let b and c be two numbers with $|I| \geqq bN$ and $N_k \geqq cN^2$ for $k\in I$ and large N, then we get

$$\sum_{k=1}^{N-1} kN_k \geqq \sum_{k\in I} kN_k \geqq cN^2 \sum_{k\in I} k \geqq cN^2 \sum_{k=1}^{|I|} k \geqq cN^2 \sum_{k=1}^{bN} k$$

$$= \frac{cb}{2} N^3 (bN+1) \in \Omega(N^4).$$

$\square$

4. THE ALGORITHM

This section will give a description of the algorithm to construct all $V_k(S)$.
It needs $\theta(N^4)$ time and $\theta(N^4)$ storage. Theorem 3 No.(iv) showed that all Voronoi diagrams have a space complexity of at least $\theta(N^4)$ and therefore we need at least $\theta(N^4)$ steps to construct them. So the algorithm has optimal time and space complexity.
The data structure for the $V_k(S)$ is the same Kirkpatrick defines for the input data of his region location algorithm (see /Ki81/). Every $V_k(S)$ is stored by a list of its Voronoi points, each of wich contains the information about the incident Voronoi edges and polygons.

The basic idea is, to take all triples of points $a,b,c\in S$ and compute all Voronoi points in the way theorem 1 describes. Theorem 2 makes sure, that we get all of them. In a second step we have to link the points of each $V_k(S)$ together.
Before coming to a detailed description of the algorithm, we need some more definitions.

Definition:

For every $(s_a,s_b,s_c)\in S^3$ let

(i) $M(a,b,c)$ be the center of s_a,s_b,s_c (if exists)

(ii) $H(a,b,c):=\{y\in S-\{s_a,s_b,s_c\}/d(M(a,b,c),y)<d(M(a,b,c),s_a)\}$

(iii) $\nabla:=\{(u,v,w)\in\{1,\ldots,N\}^3/u>v>w\}$.

With this we can construct all $V_k(S)$ as follows:

(1) Construct an array L of all $M(a,b,c)$ with $(a,b,c)\in\nabla$ in wich every $M(a,b,c)$ can be found in $O(1)$ steps (store $M(a,b,c)$ with $(a,b,c)\in\nabla$ at the adress $\binom{a-1}{3}+\binom{b-1}{2}+c)$.

(2) Traverse L.
For every $M(a,b,c)$ calculate $H(a,b,c)$ and the incident rays and polygons as described by theorem 1 and add them to the lists of $V_{|H(a,b,c)|+1}(S)$ and $V_{|H(a,b,c)|+2}(S)$. Note the two adresses in L.

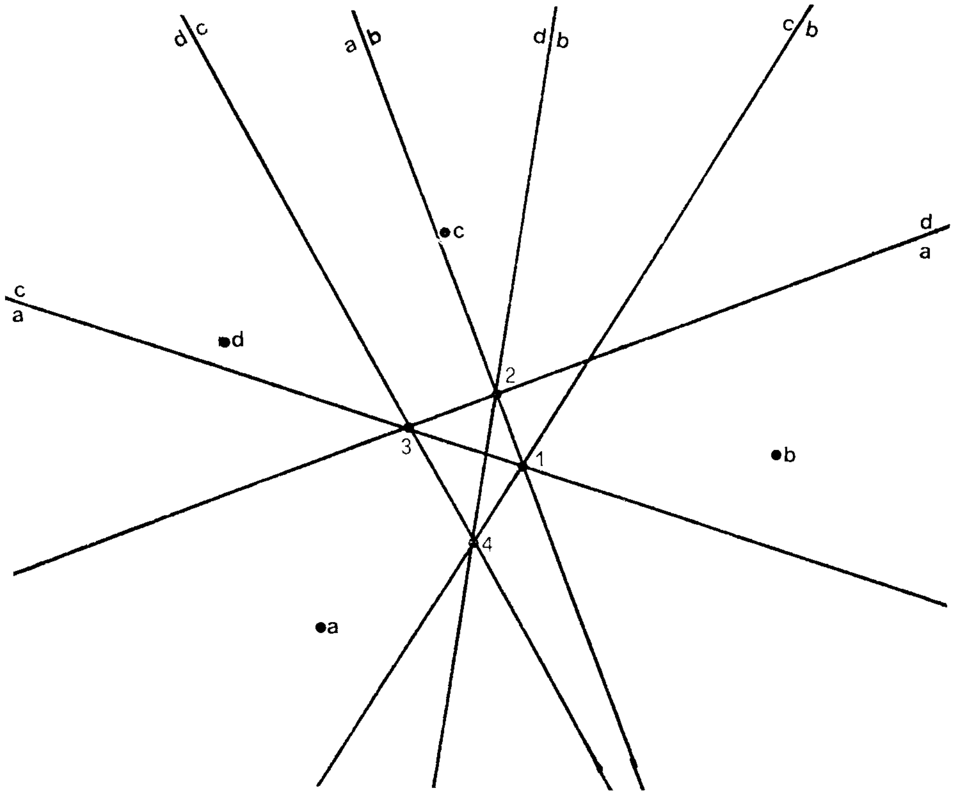
(3) Traverse L again.
Every $M(a,b,c)$ is a Voronoi point in two lists $V_i(S)$ and $V_{i+1}(S)$ with at most 6 incident rays.
With each ray r do the following steps:
Let r be in $V_j(S)$ and be part of $B(s_a,s_b)$.
Take all $M(a,b,x)$ with $s_x\in S-\{s_a,s_b,s_c\}$ and check, whether $M(a,b,x)$ is a Voronoi Point of $V_j(S)$ and lies on r. If there are more such points, take the $M(a,b,x_0)$ with minimum distance from $M(a,b,c)$.
Reduce r to an edge $(M(a,b,c),M(a,b,x_0))$ and the corresponding ray of $M(a,b,x_0)$ to an edge $(M(a,b,x_0),M(a,b,c))$.

With theorem 3 the analysis of the algorithm is easy.
The space complexity is

$$\theta\Big(\sum_{k=1}^{N-1}(I_k+kE_k)\Big) = \theta\Big(\sum_{k=1}^{N-1}kN_k\Big) = \theta(N^4).$$

L contains $\theta(N^3)$ points. So part (1) of the algorithm needs time $\theta(N^3)$. In part (2) for each of these points we need $\theta(N)$ steps and so the whole part takes $\theta(N^4)$ steps. In the same way you see evidently part (3) needing time $\theta(N^4)$ too. So the time complexity of the whole algorithm is $\theta(N^4)$.
This short description of the algorithm yields already so many details, that it is easy to be implemented. We don´t need very much overhead (compared with the algorithms in /Le81/ and /ShHo75/) and the multiplicative constants for the complexity are expected to be quite good.

5. AN EXAMPLE



This picture shows a set $S:=\{a,b,c,d\}\subset\mathbb{E}^2$ and all possible bisectors.

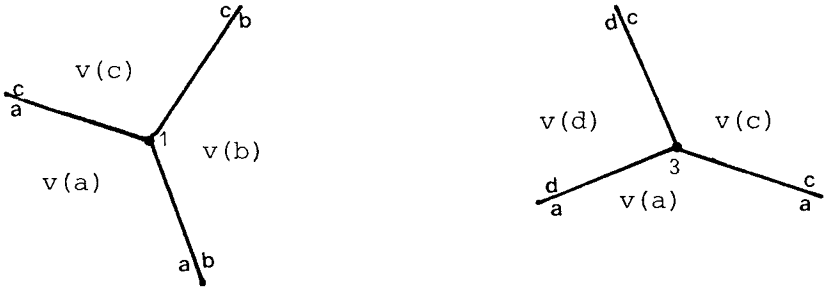
With these points the array L looks as follows:

Voronoi point No.	center of	H	$\|H\|+1$	$\|H\|+2$
1	{a,b,c}	ϕ	1	2
2	{a,b,d}	{c}	2	3
3	{a,c,d}	ϕ	1	2
4	{b,c,d}	{a}	2	3

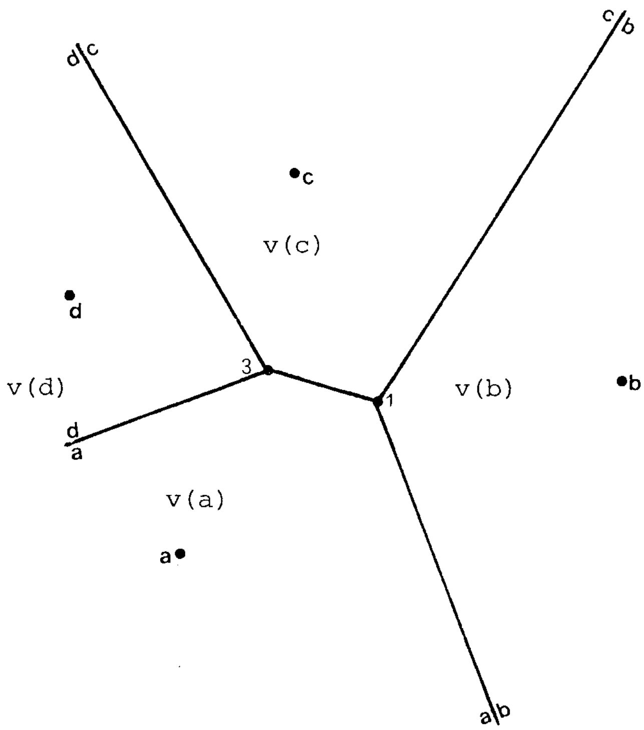
Now we can construct all Voronoi points and diagrams at once.

$V_1(S)$:

Voronoi points

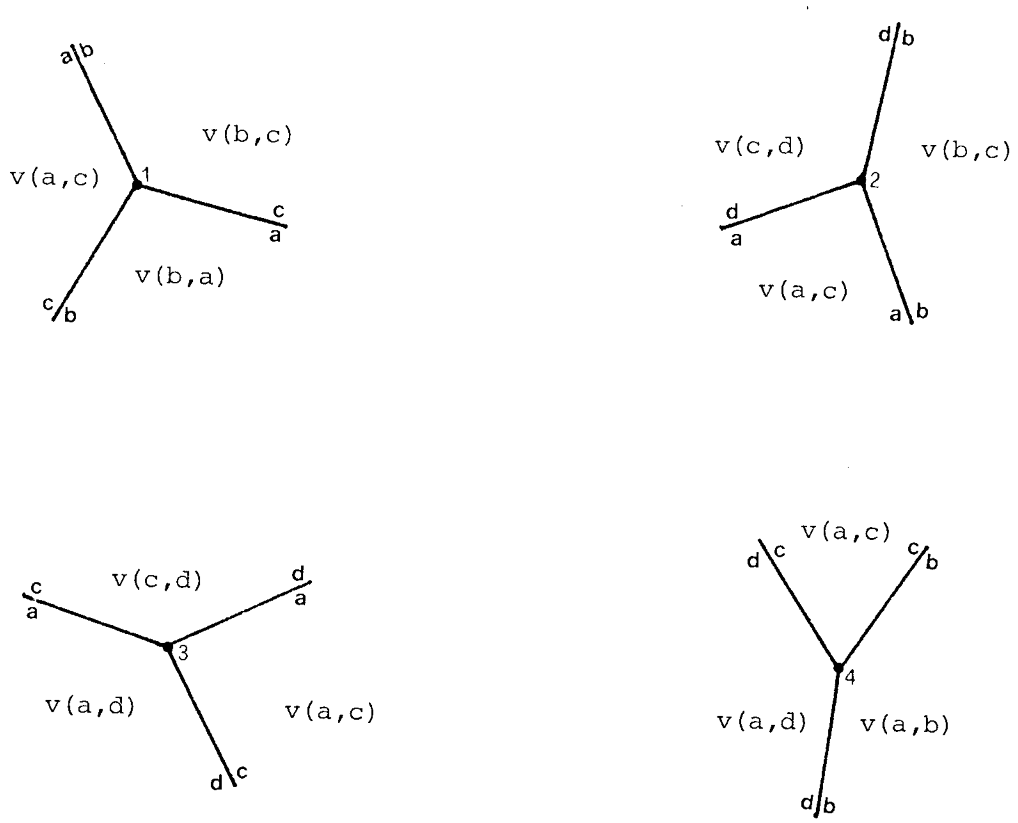


diagram

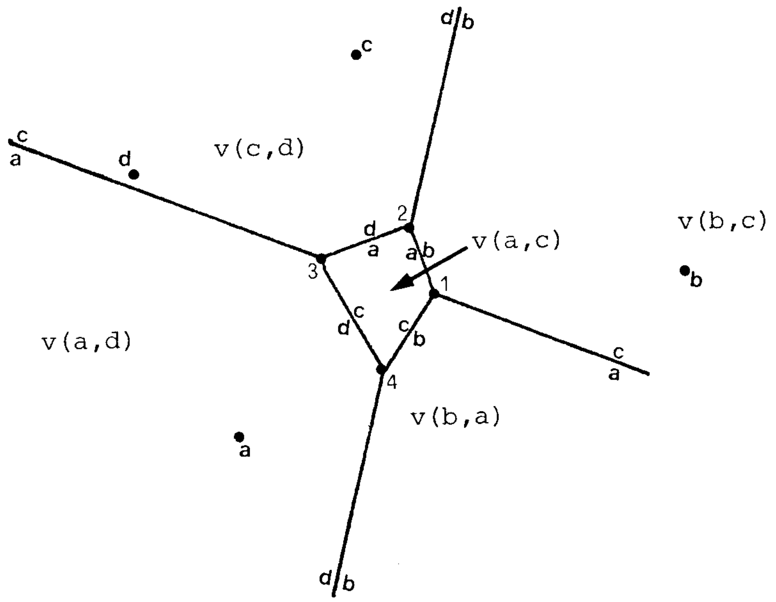


$V_2(S)$:

Voronoi points

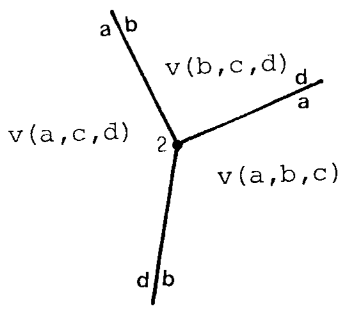


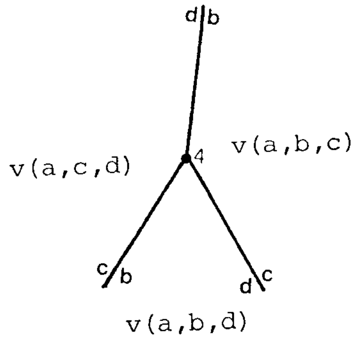
diagram



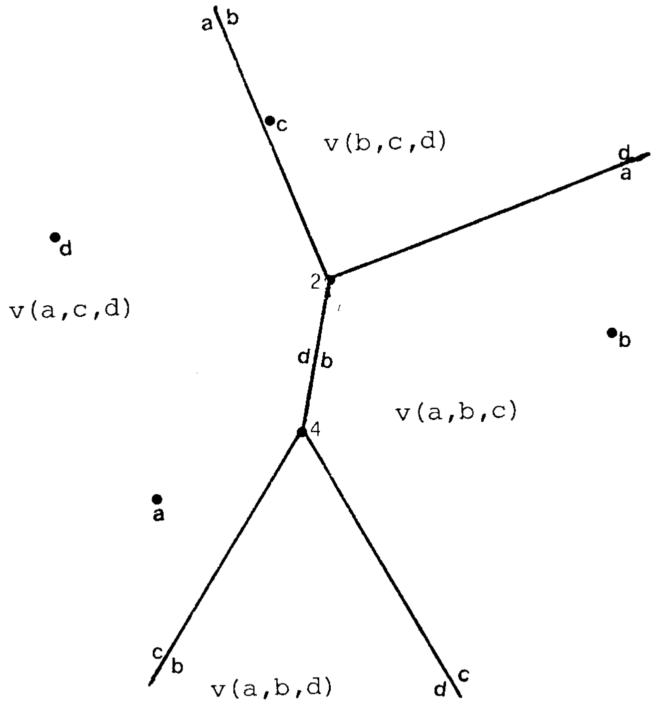
$V_3(S)$

Voronoi points





diagram



6. ACKNOWLEDGMENT

The author wishes to thank Prof. Dr. H. Noltemeier for his encouragement and helpfull discussions.

7. REFERENCES

/Ki81/ Kirkpatrick: OPTIMAL SEARCH IN PLANAR SUBDIVISIONS, Department of Computer Science, Univ. of British Columbia, Vancouver, B.C., Canada, Report 81-13, 1981

/Le81/ Lee: AN APPROACH TO FINDING THE K-NEAREST NEIGHBOR IN THE EUCLIDEAN PLANE, Department of Electrical Engineering and Computer Science, Northwestern Univ., Evanston, Il. 60201, USA, Report 1981

/Ma80/ Maurer: THE POST-OFFICE PROBLEM AND RELATED QUESTIONS, in Noltemeier (ed.): GRAPHTHEORETIC CONCEPTS IN COMPUTER SCIENCE, Proceedings of the International Workshop WG80, Bad Honnef, June 1980, Lecture Notes in Computer Science 100, Springer 1981

/Sh75a/ Shamos: COMPUTATIONAL GEOMETRY, Yale University, New Haven, Connecticut,USA, Ph.D.Thesis, 1975

/Sh75b/ Shamos: GEOMETRIC COMPLEXITY, Proceedings of the 7th Annual ACM Symposium on Theory of Computing, 1975, pp.224-233

/ShHo75/ Shamos,Hoey: CLOSEST-POINT PROBLEMS, Proceedings of the 16th IEEE Symposium on Foundations of Computer Science, 1975, pp.151-162