

Voronoidiagramme

1. Aufgabenstellung

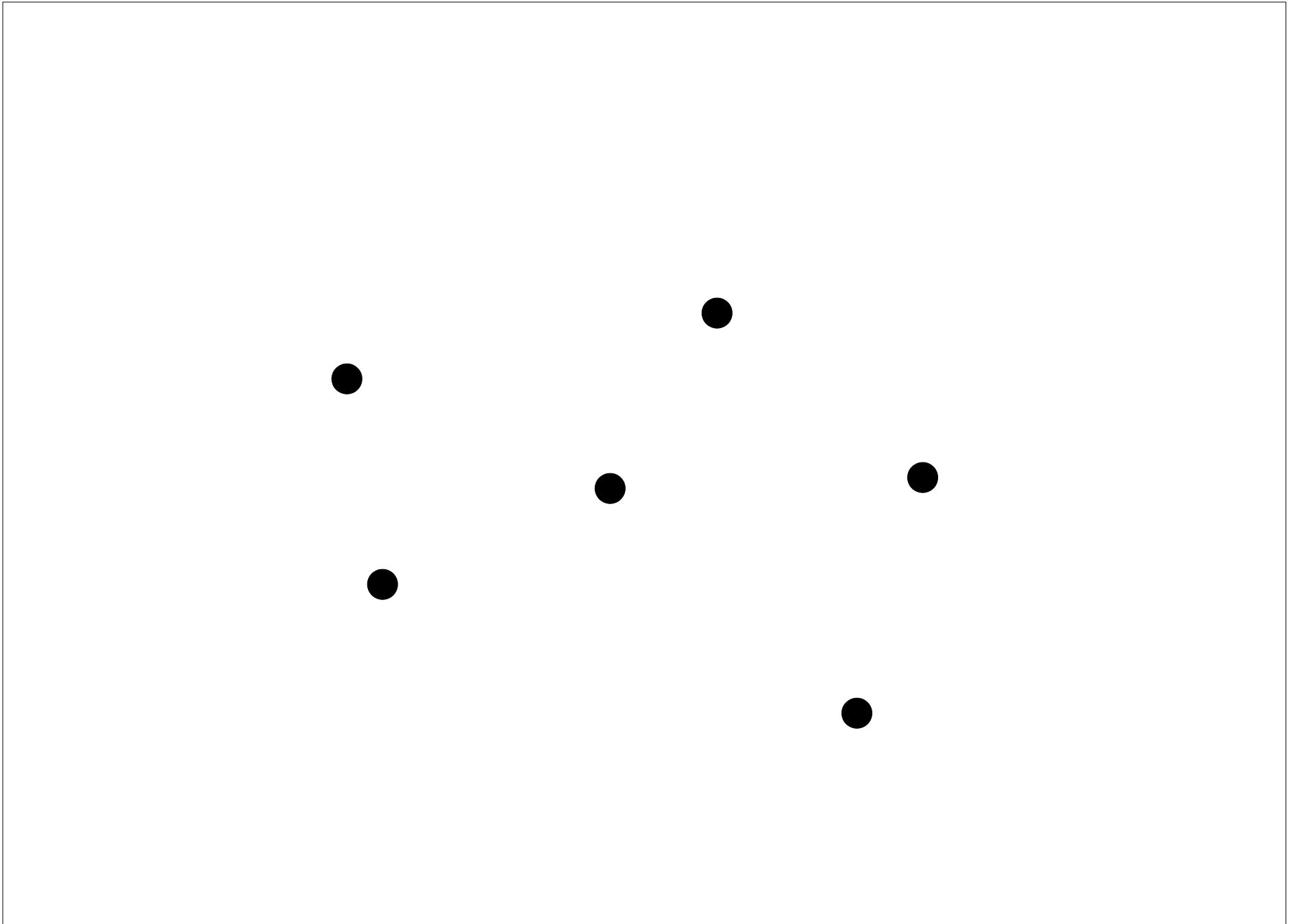
Sei P eine Menge von Punkten in der Ebene und q ein Punkt in P .

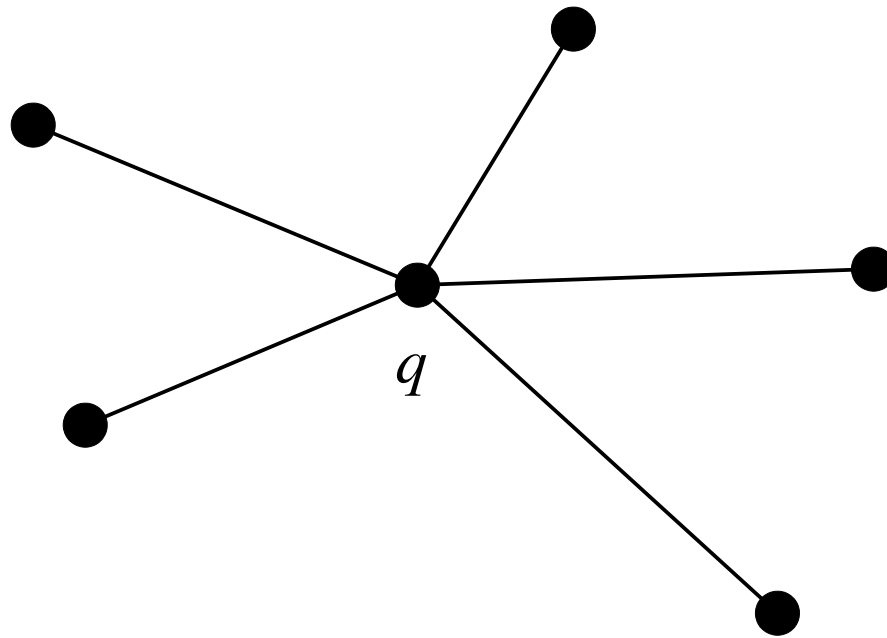
Wir definieren die **Voronioiregion** von q :

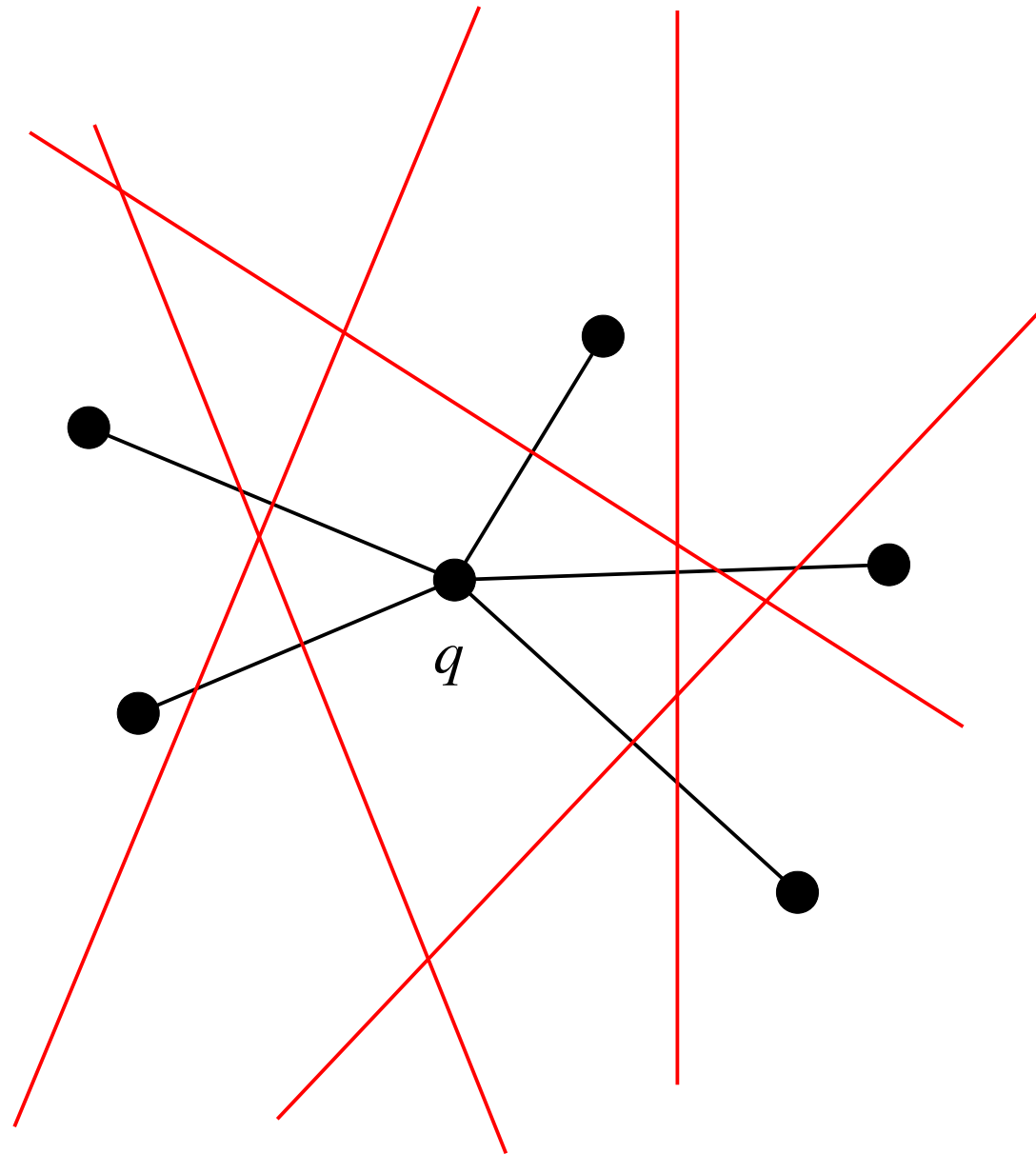
$$\text{vor}(q, P) = \{x \in \mathbb{R}^2 \mid \forall r \in P \setminus \{q\} (d(x, q) < d(x, r))\}$$

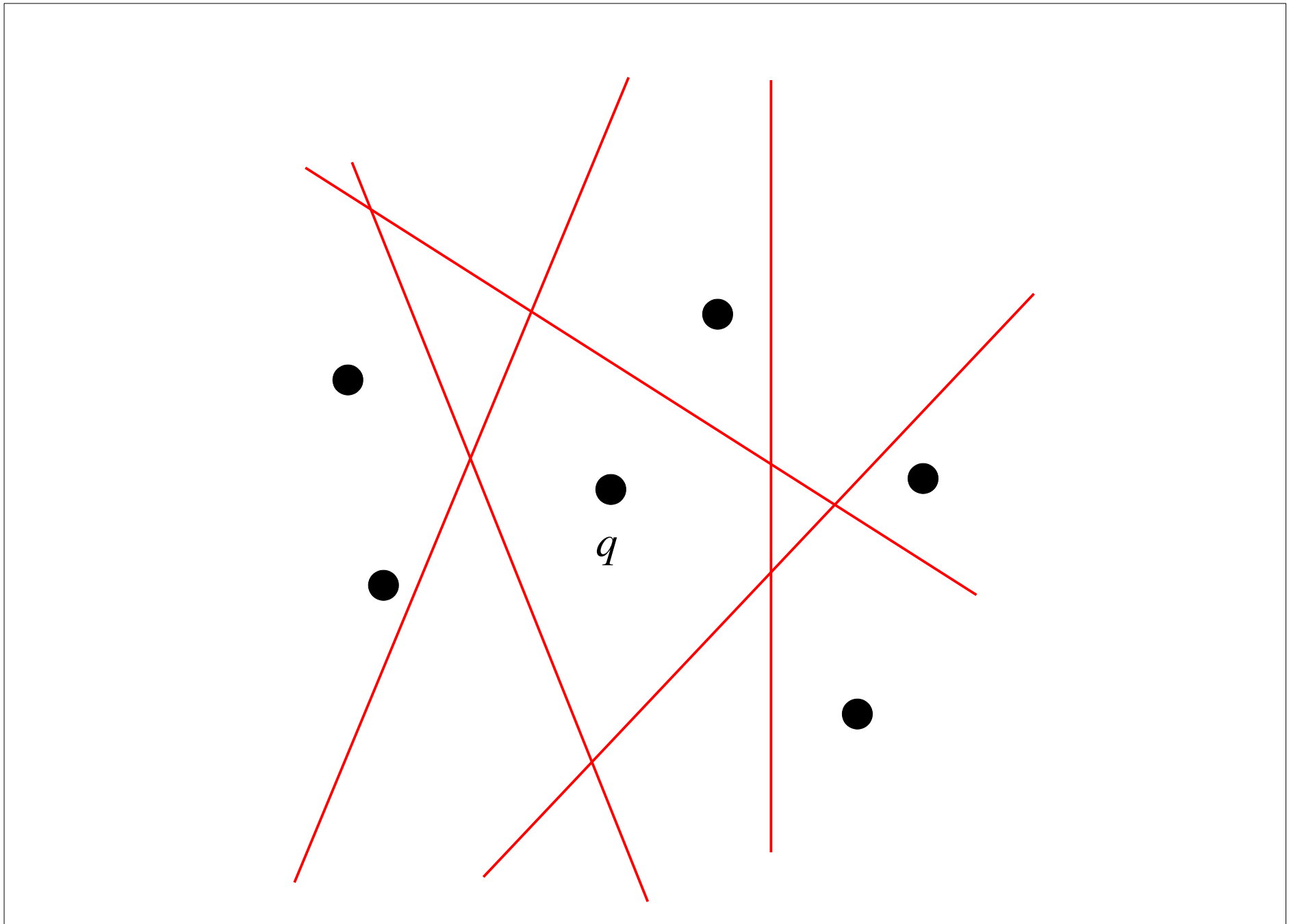
Dabei bezeichnet $d(a, b)$ den euklidischen Abstand von a und b .

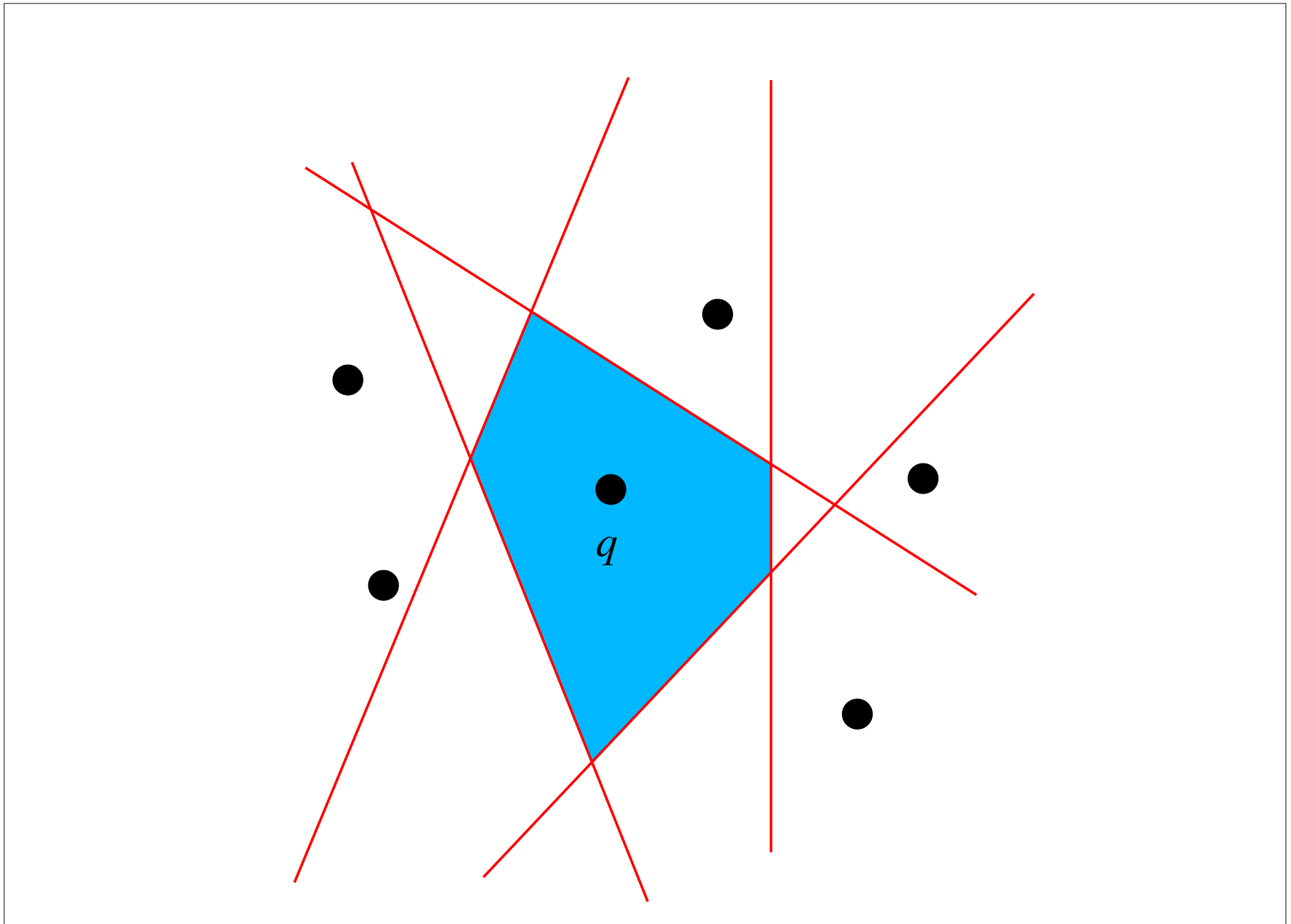
Das **Voronoidiagramm** $\text{vor}(P)$ von P besteht aus den Rändern der Voronoiregionen der Punkte in P .

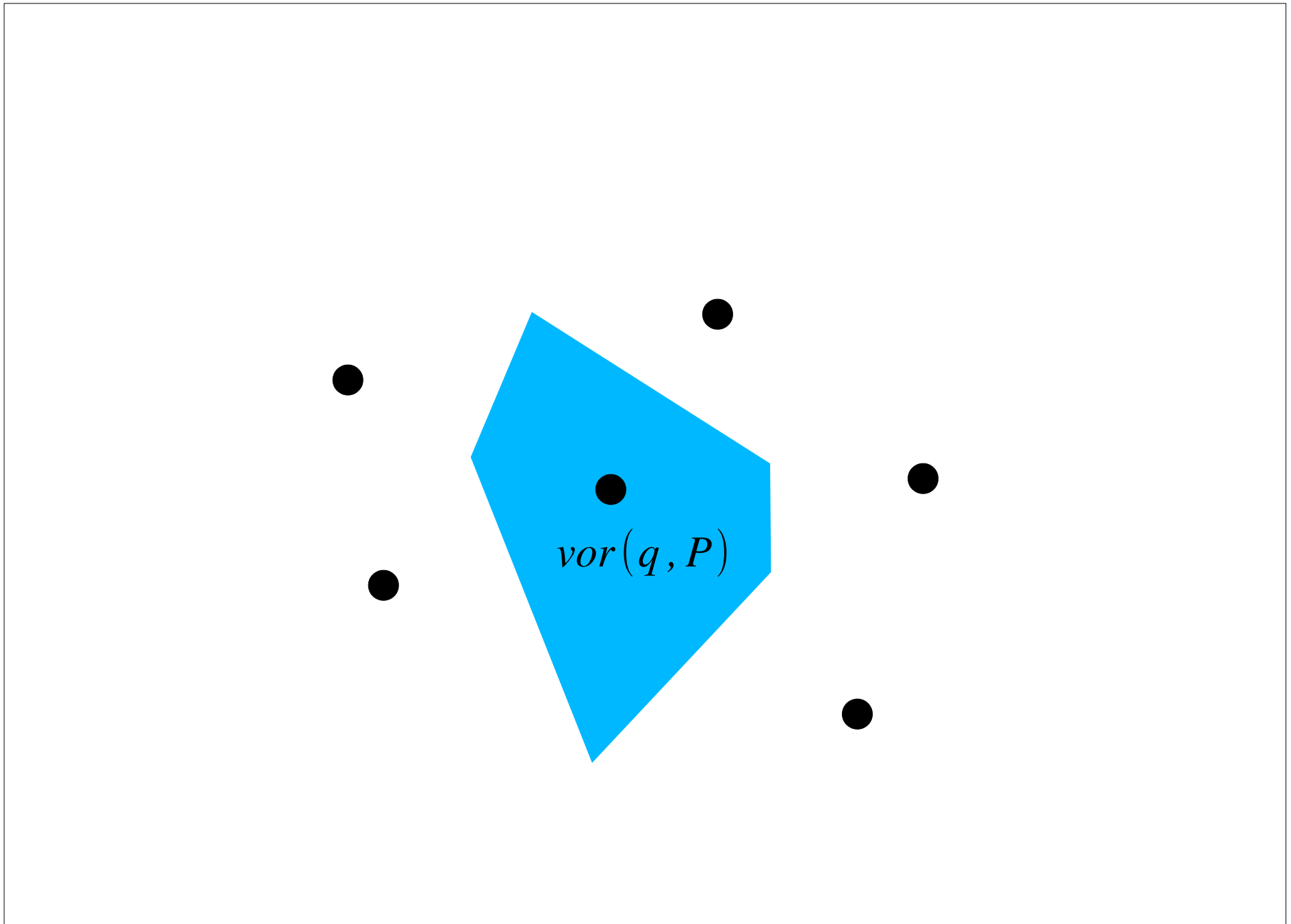




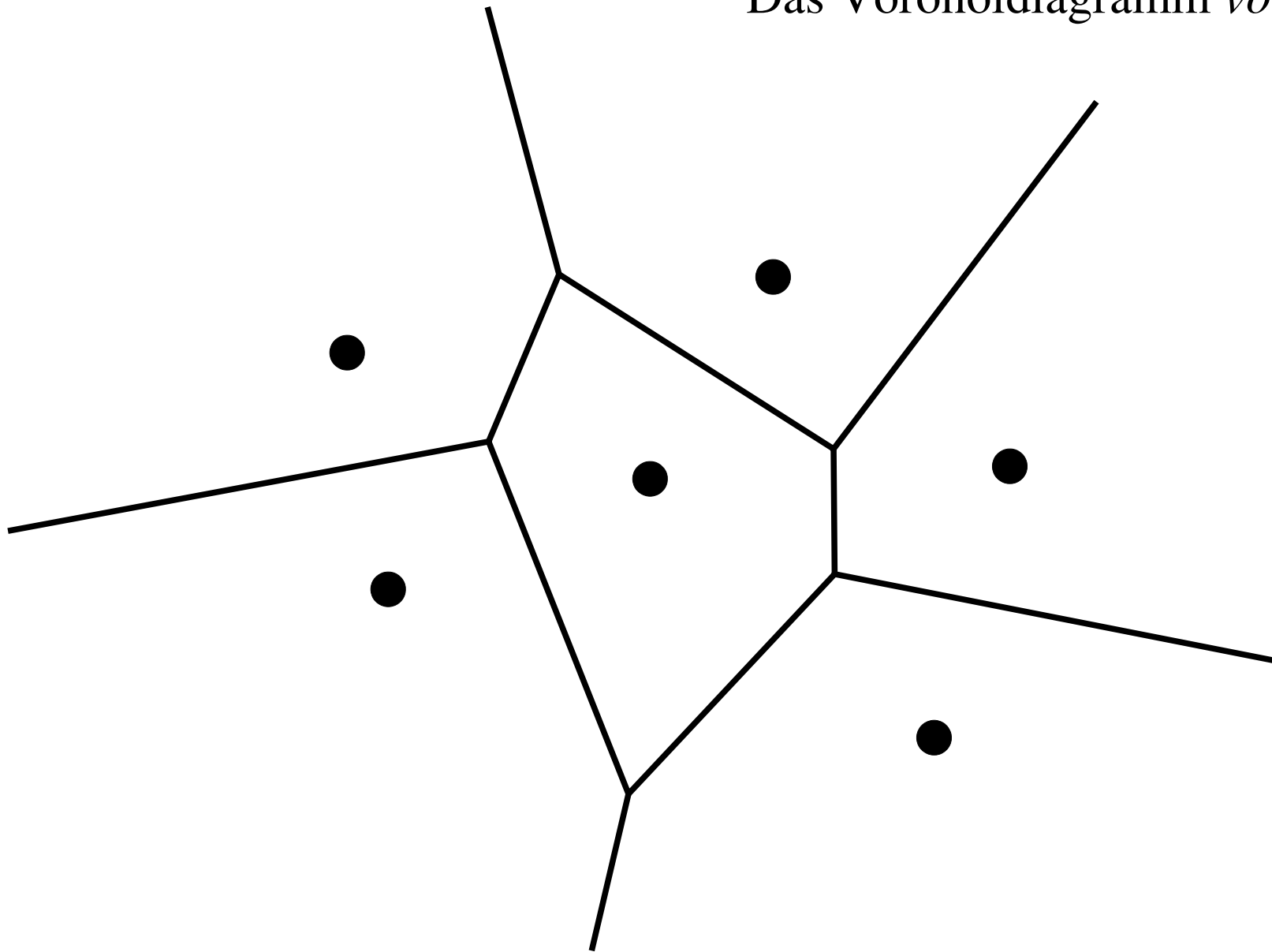






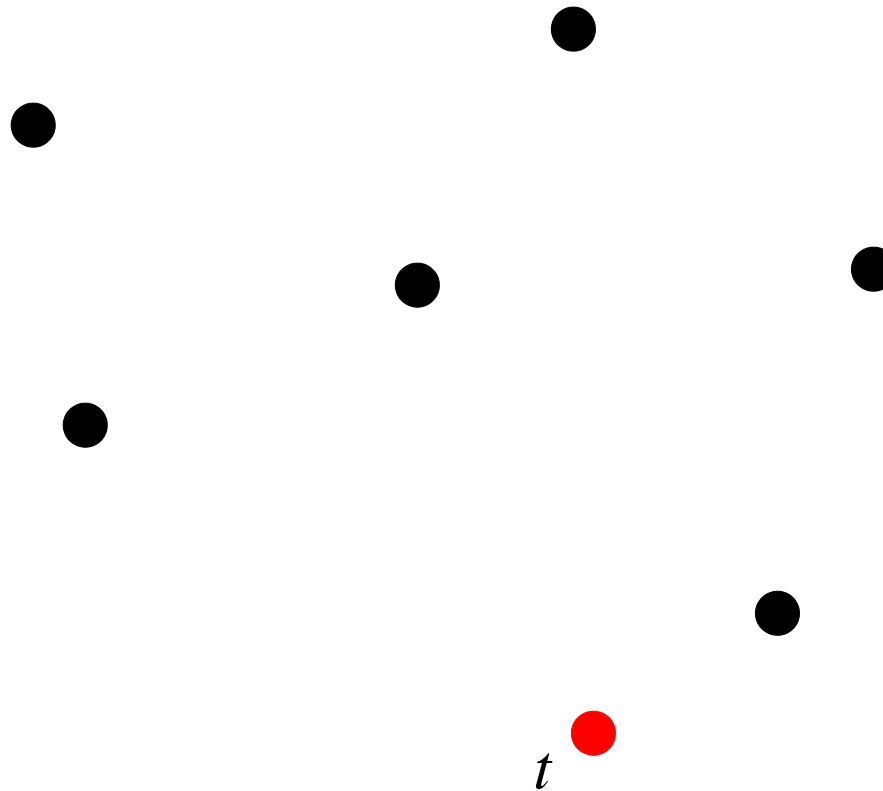


Das Voronoidiagramm $\text{vor}(P)$.

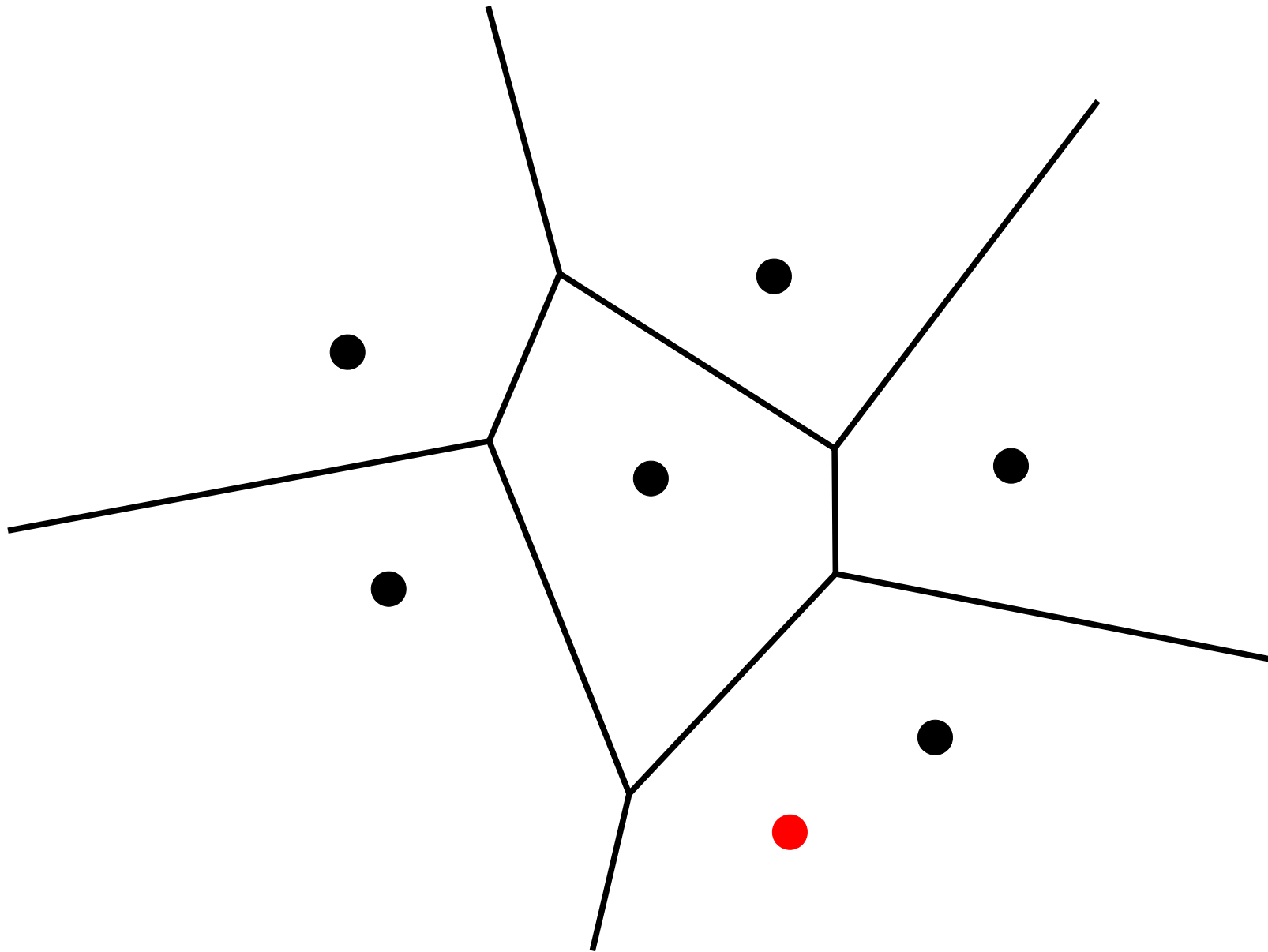


2. Anwendungen des Voronoidiagramms

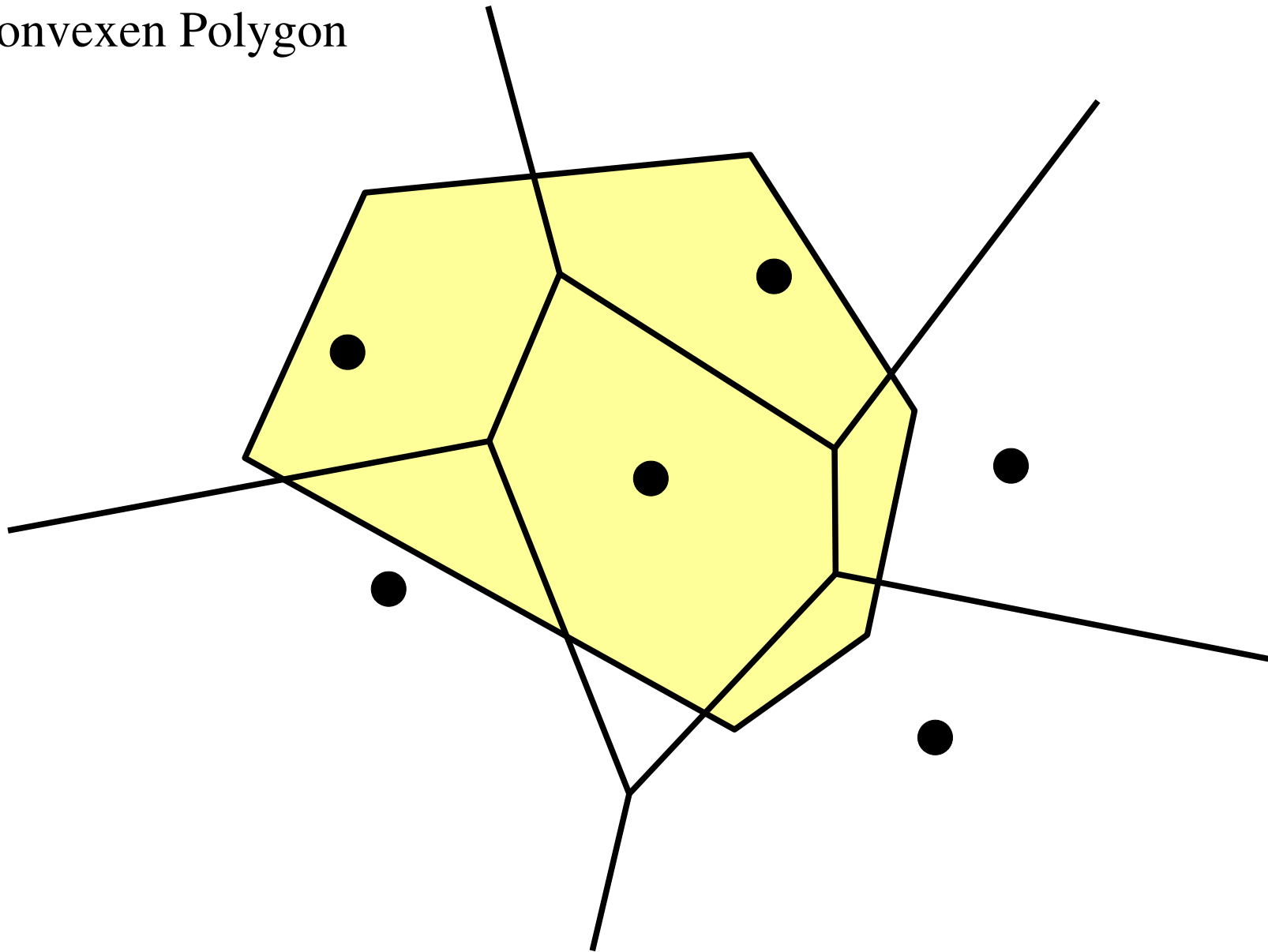
Bestimmung des **nächsten Nachbarn** eines Punktes t in $P...$

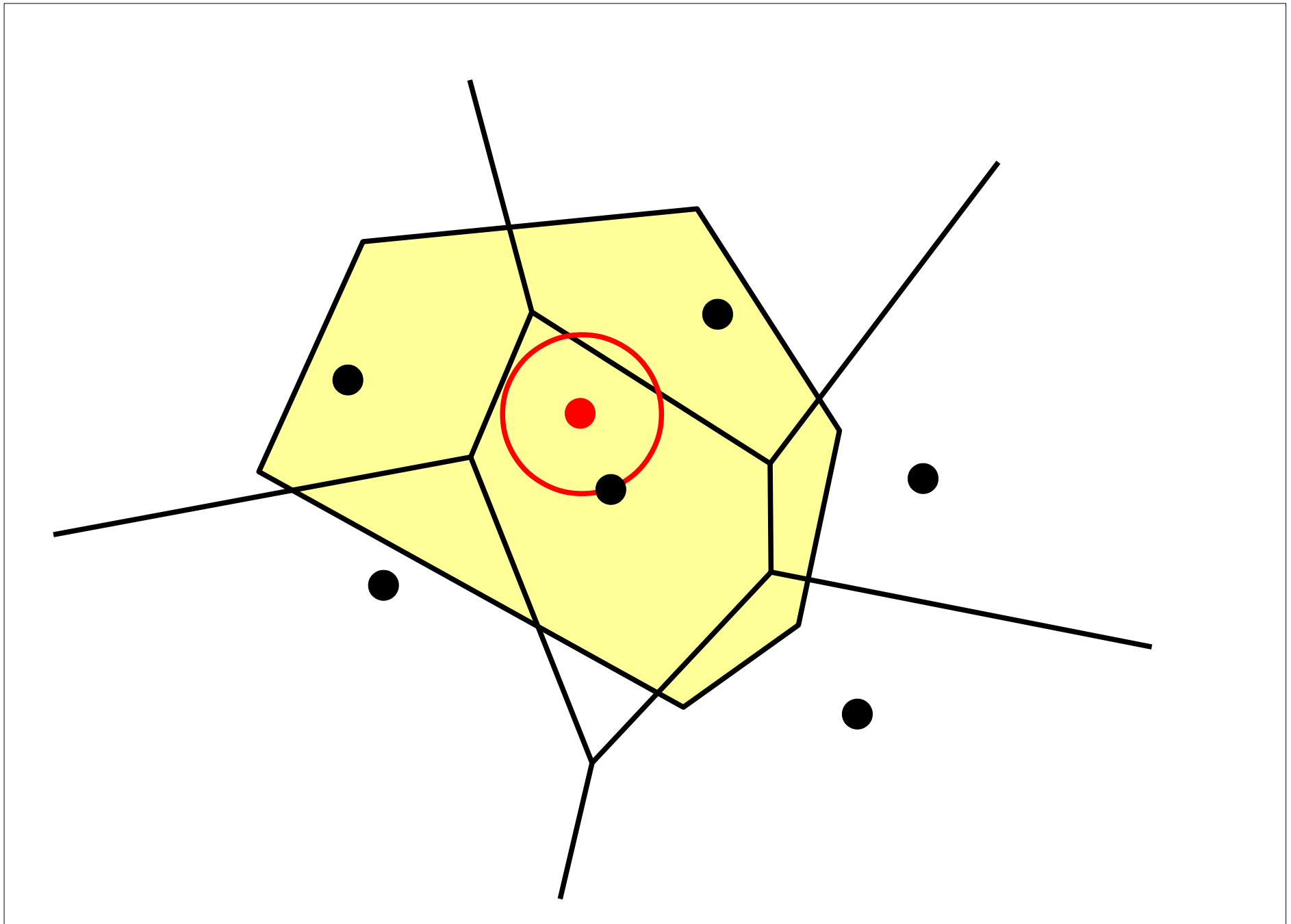


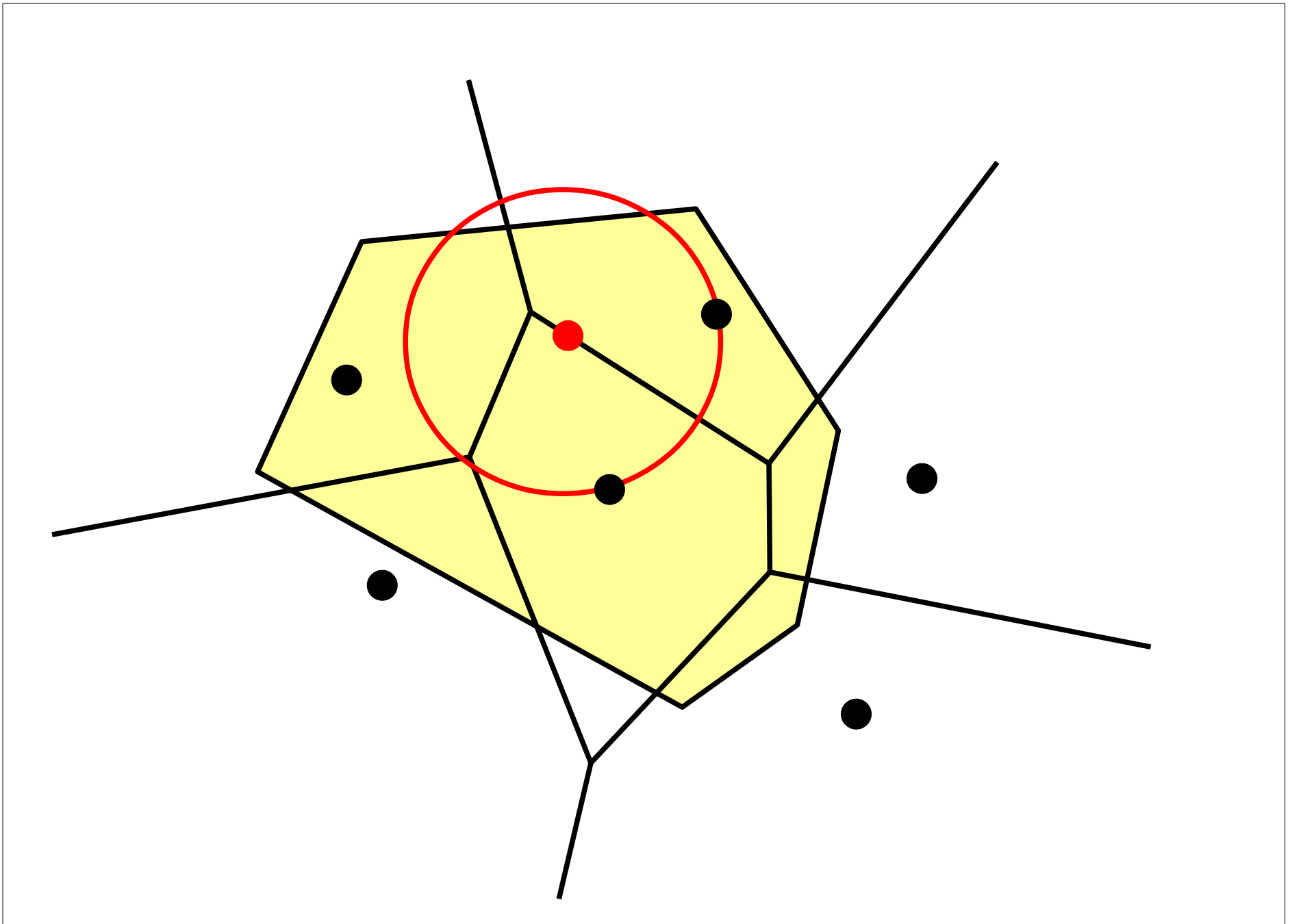
...als **Punktlokalisierung** im Voronoidiagramm von P .

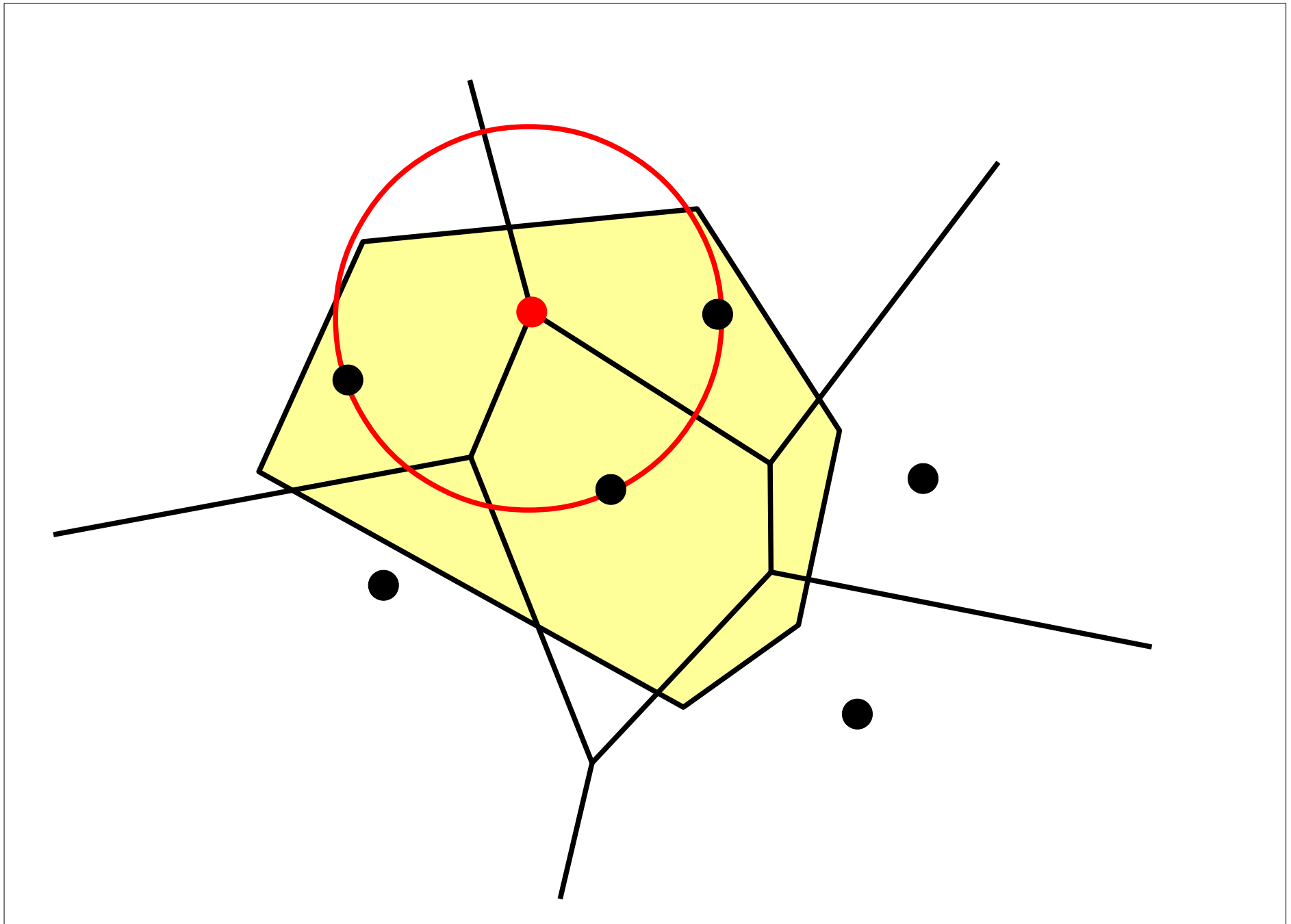


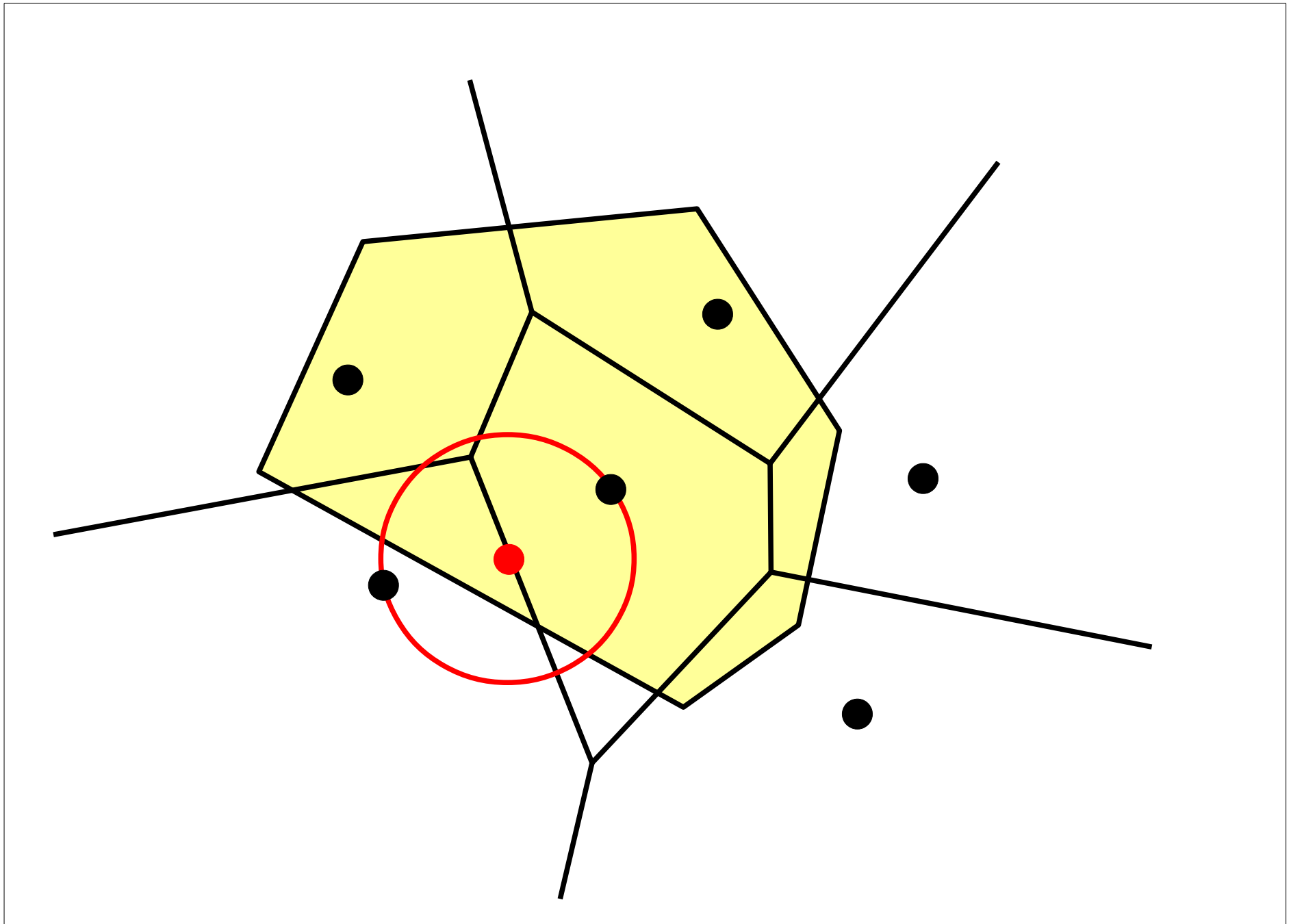
Bestimmung des **größten leeren Kreises** mit Mittelpunkt in einem konvexen Polygon

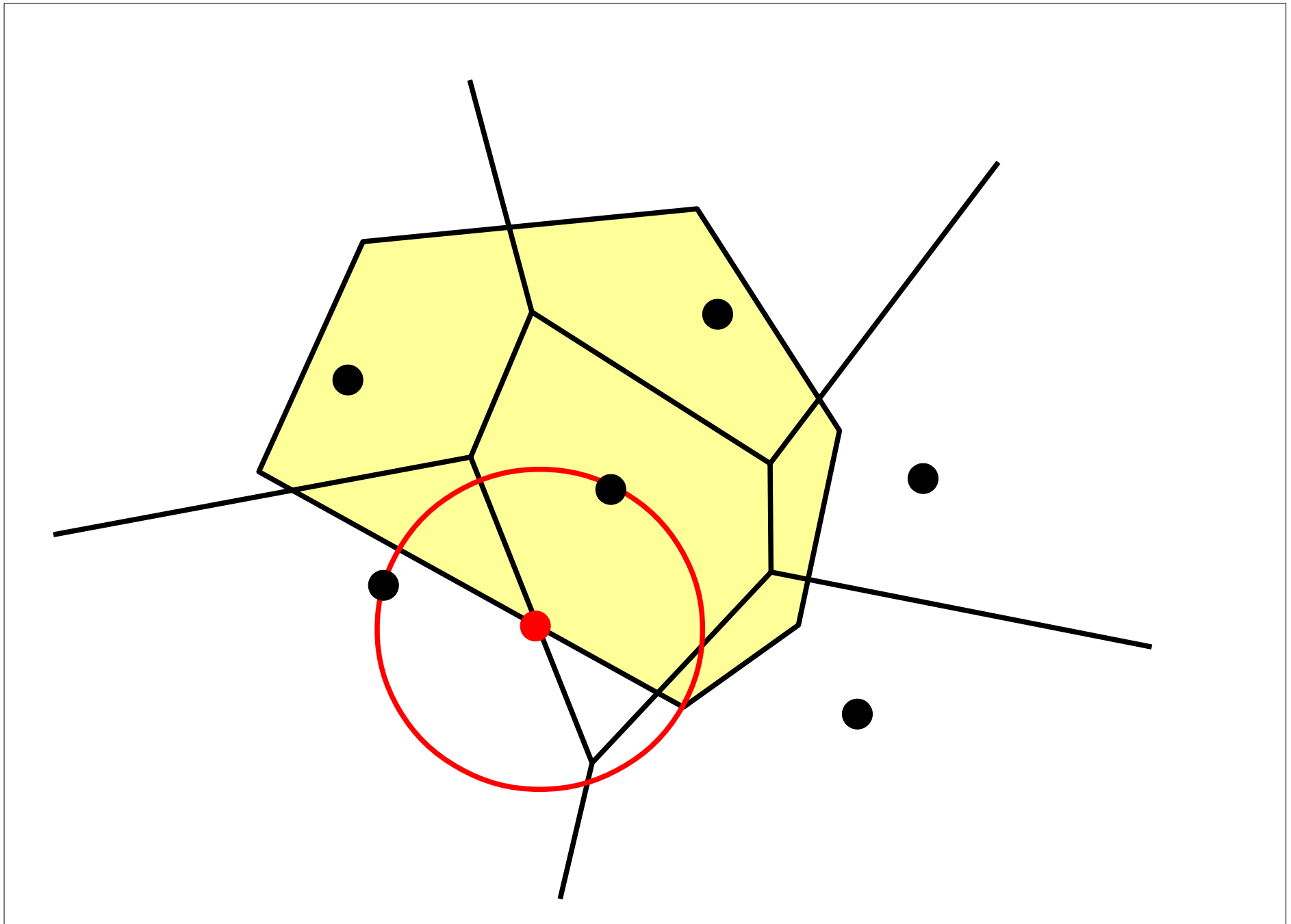


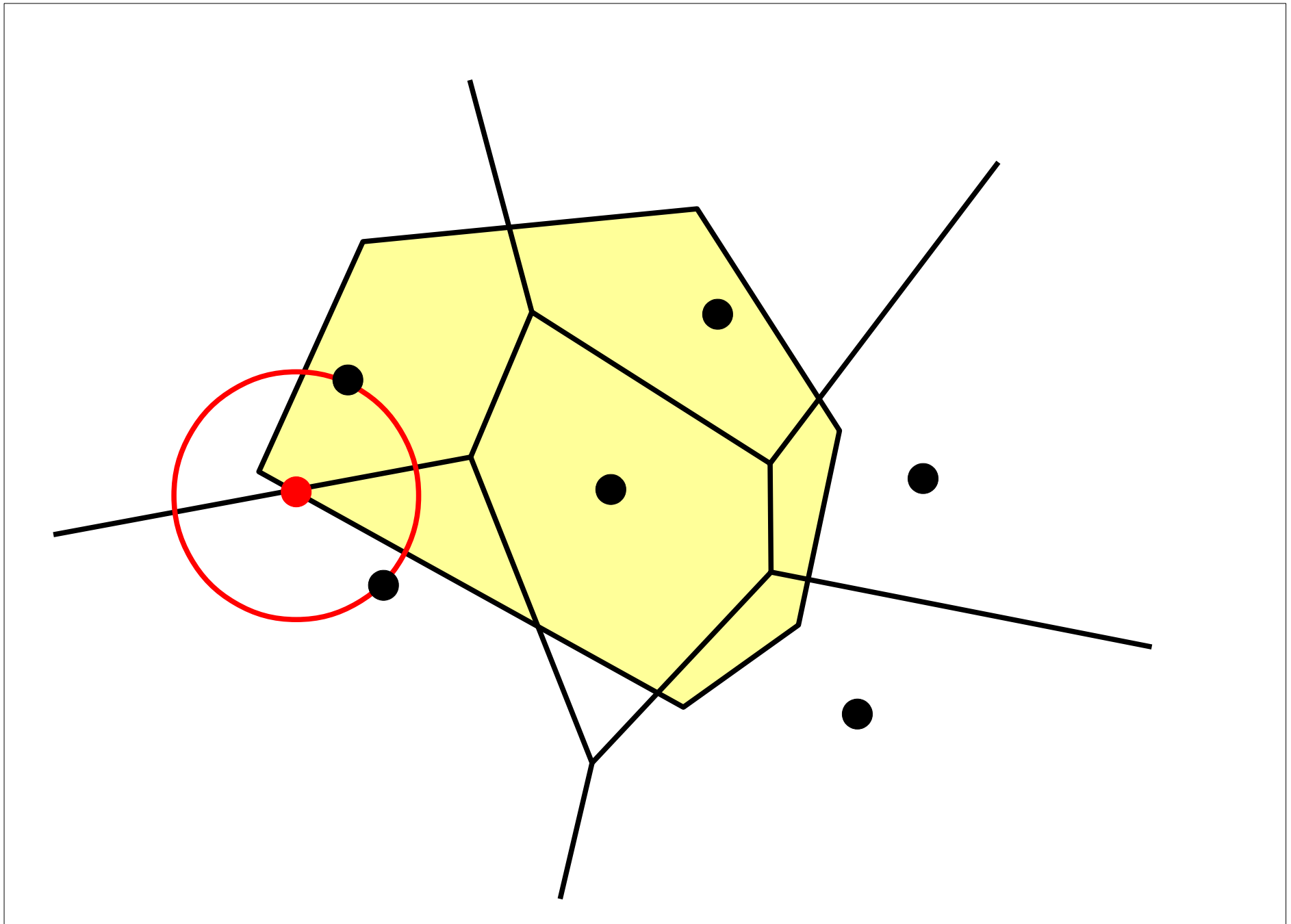


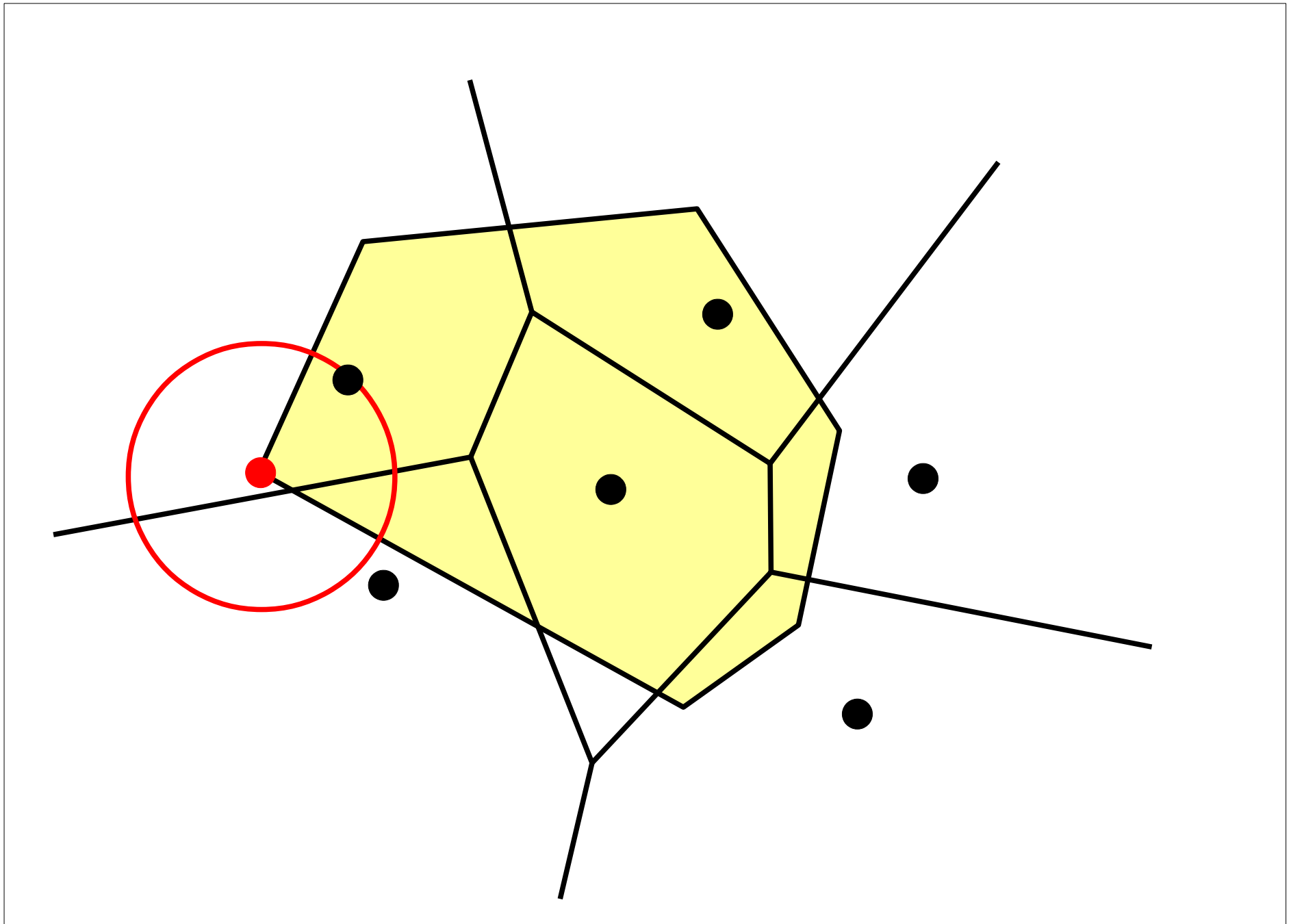












Zusammenfassung zur Bestimmung eines größten leeren Kreises:

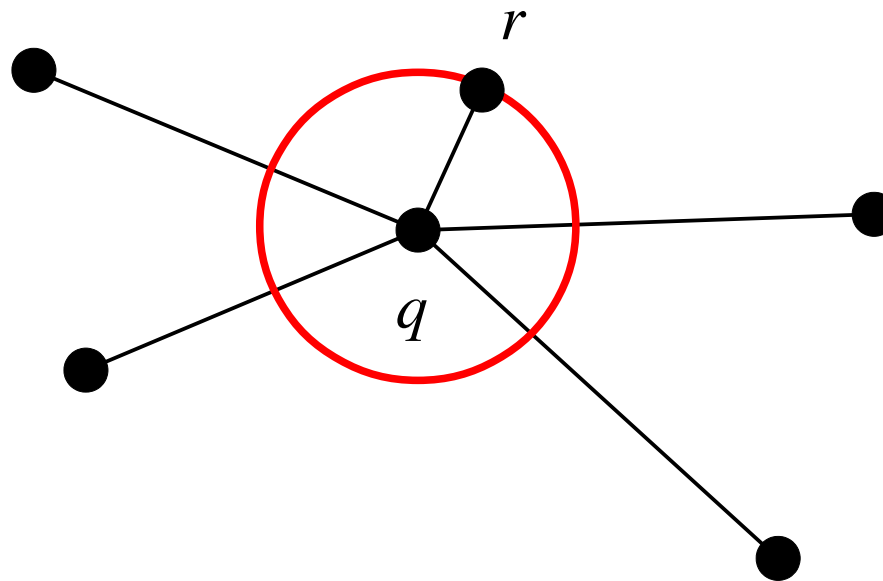
Es müssen als **Kandidaten** für den Mittelpunkt eines größten leeren Kreises nur untersucht werden:

1. Die Ecken des Voronoidiagramms, die im konvexen Polygon liegen.
2. Die Schnittpunkte von Kanten des Voronoidiagramms mit Kanten des konvexen Polygons.
3. Die Ecken des konvexen Polygons.

Bestimmung des nächsten Nachbarn **zu jedem Punkt** in P .

Grundidee:

Die Voronoiregion des Punktes r in P , der den kleinsten Abstand zum Punkt q in P hat, hat eine Kante gemeinsam mit der Voronoiregion von q .



Wir müssen also nur für alle Punkte q aus P die **Kanten** der Voronoiregion $vor(q,P)$ **durchlaufen**, um für jeden Punkt in P seinen nächsten Nachbarn in P zu bestimmen.

Damit finden wir insbesondere das Paar von Punkten in P mit kleinstem Abstand.

Da für das Auffinden des Punktepaares mit kleinstem Abstand eine **untere Schranke für die Laufzeit** jedes Algorithmus im worst case von $\Omega(n \log n)$ gilt, gilt dieselbe untere Schranke auch für jeden Algorithmus, der das Voronoidiagramm berechnet.

Ziel ist also ein $O(n \log n)$ Algorithmus.

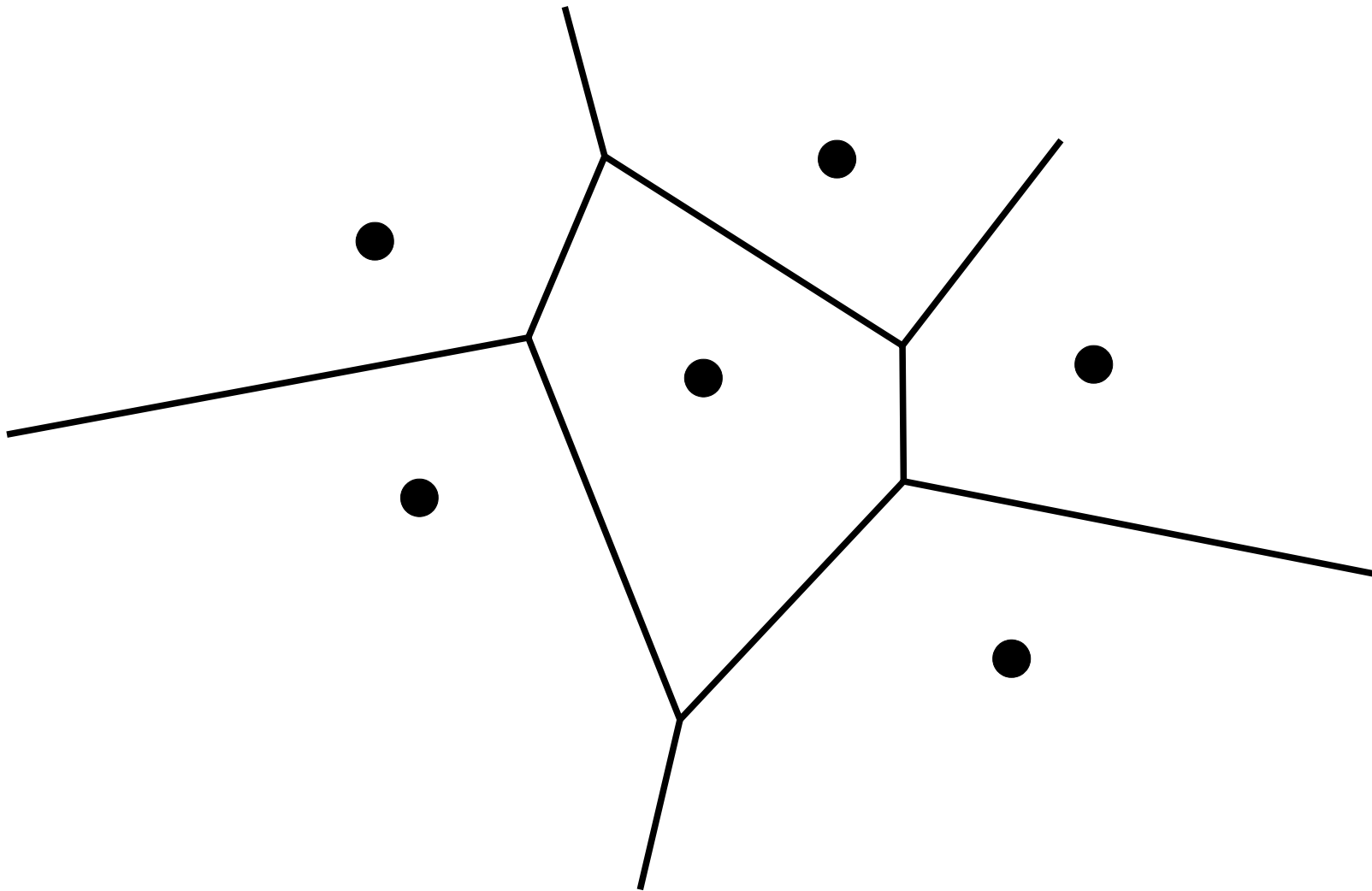
Wir bemerken an dieser Stelle noch, dass man ganz leicht einen $O(n^2 \log n)$ Algorithmus angeben kann:

Wir berechnen einfach für jeden Punkt aus P die Voronoiregion als **Durchschnitt** von $n - 1$ **Halbebenen**.

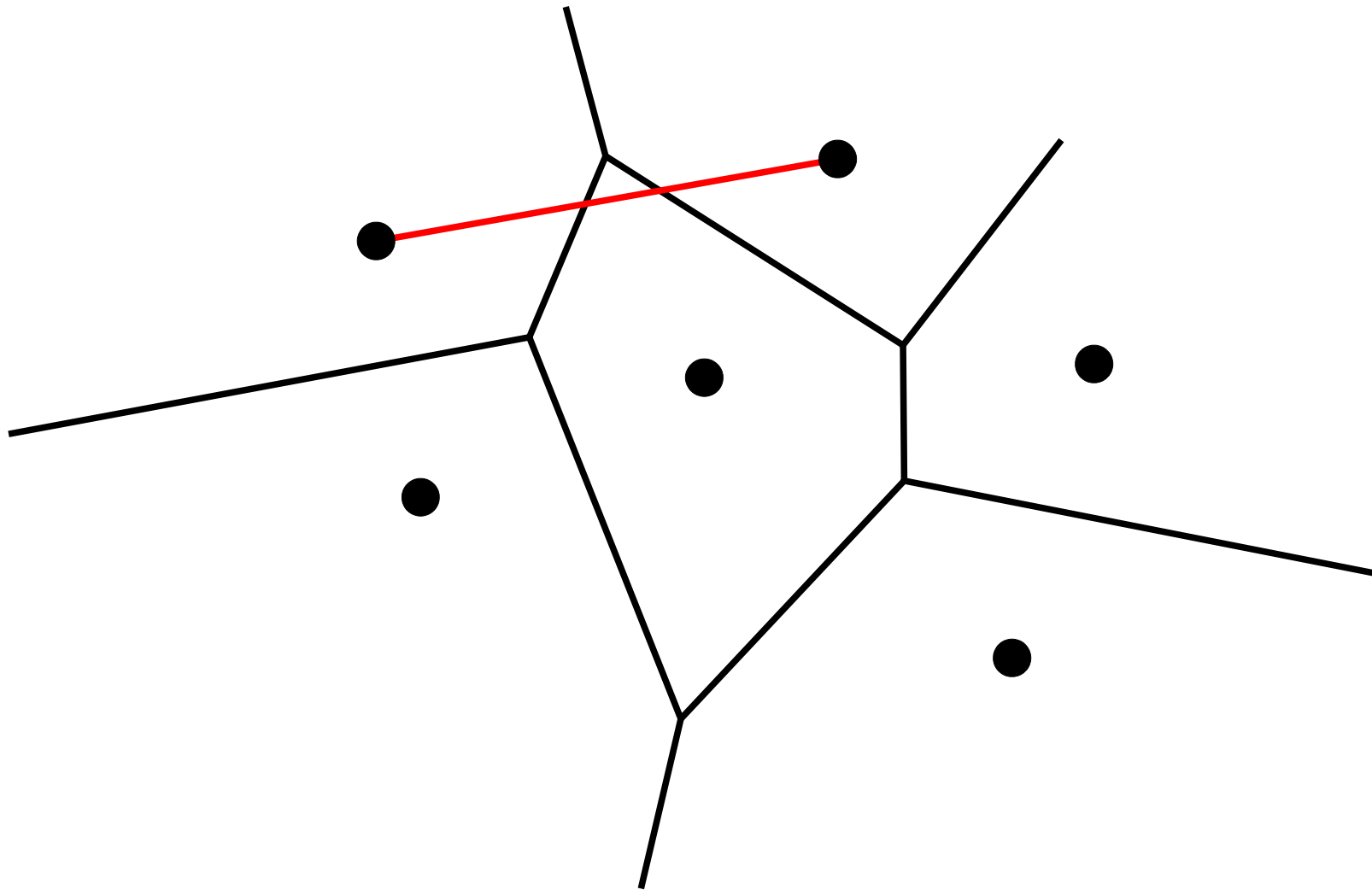
Wir wissen schon, wie jeder einzelne dieser Durchschnitte in $O(n \log n)$ Zeit berechnet werden kann.

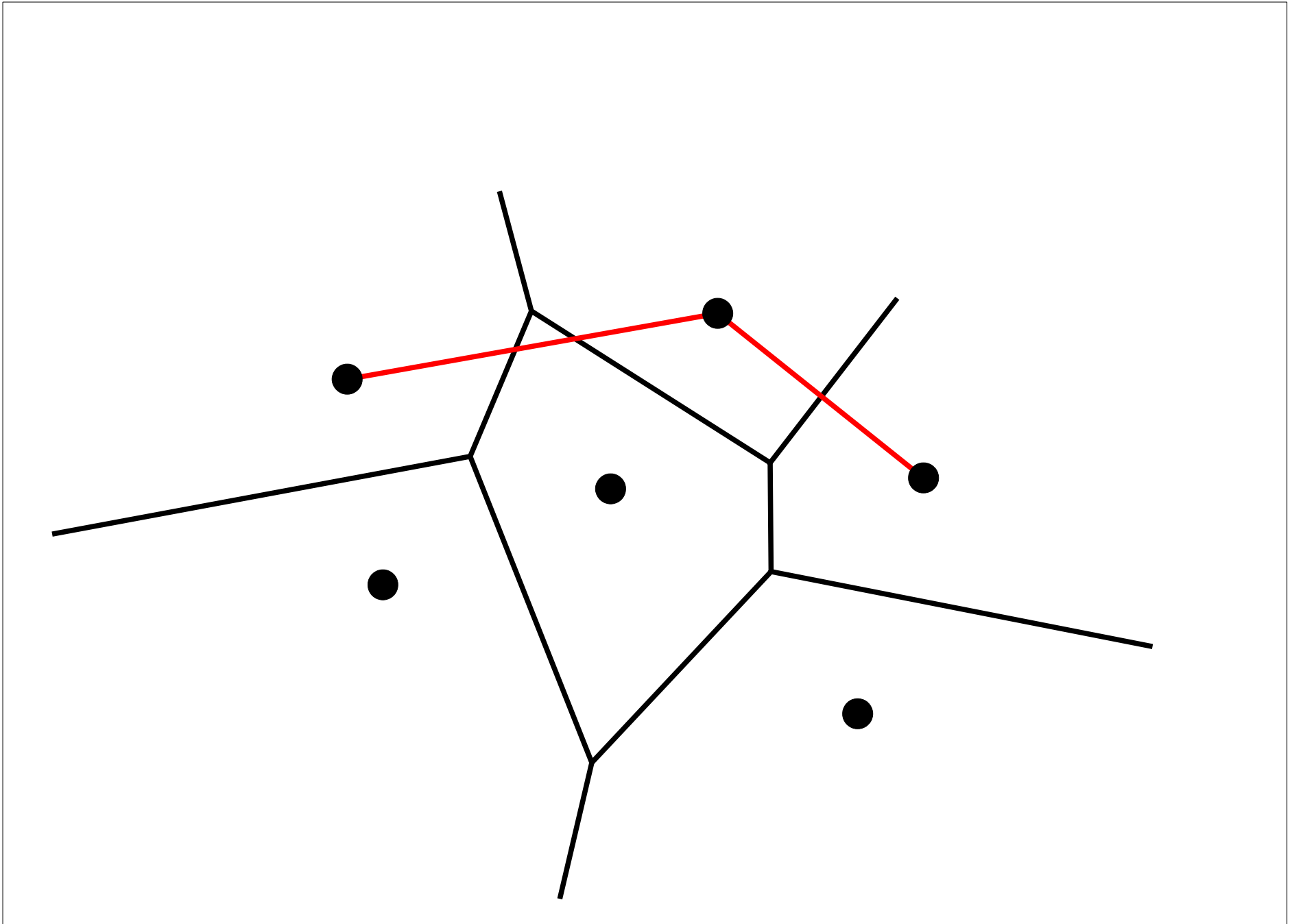
3. Effiziente Berechnung des Voronoidiagramms

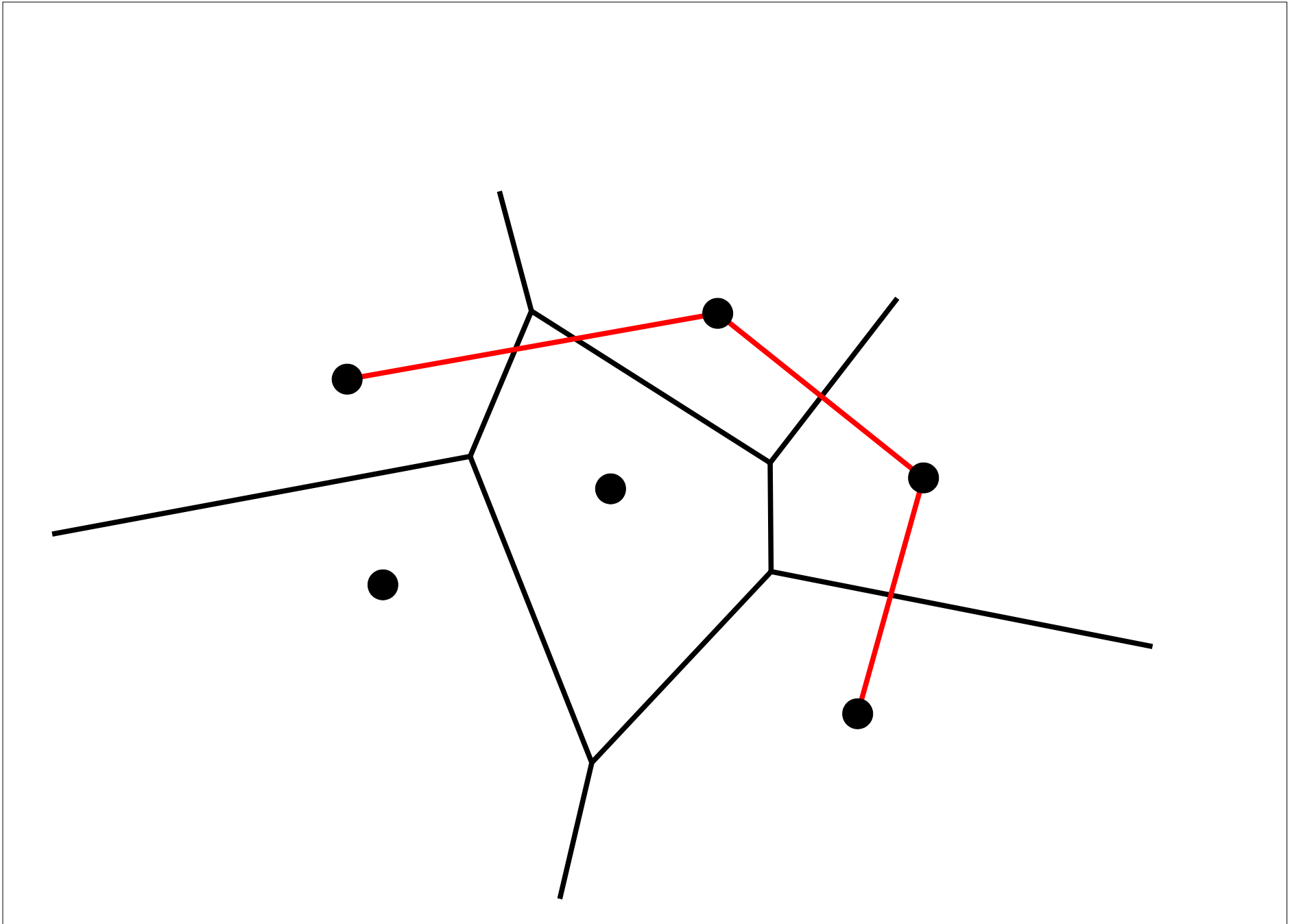
Zunächst halten wir fest, dass man leicht aus dem Voronoidiagramm die **konvexe Hülle** von P erhalten kann.

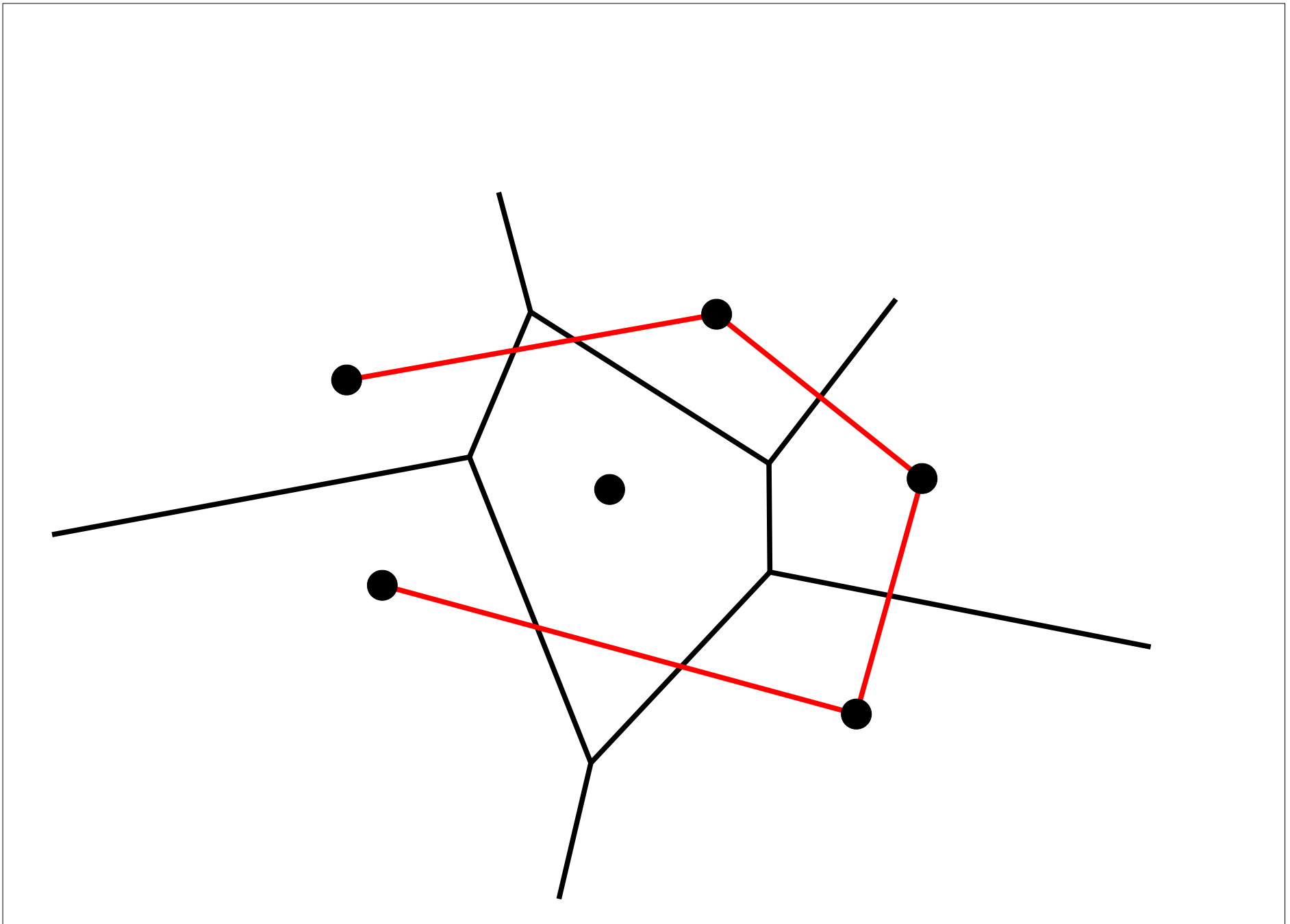


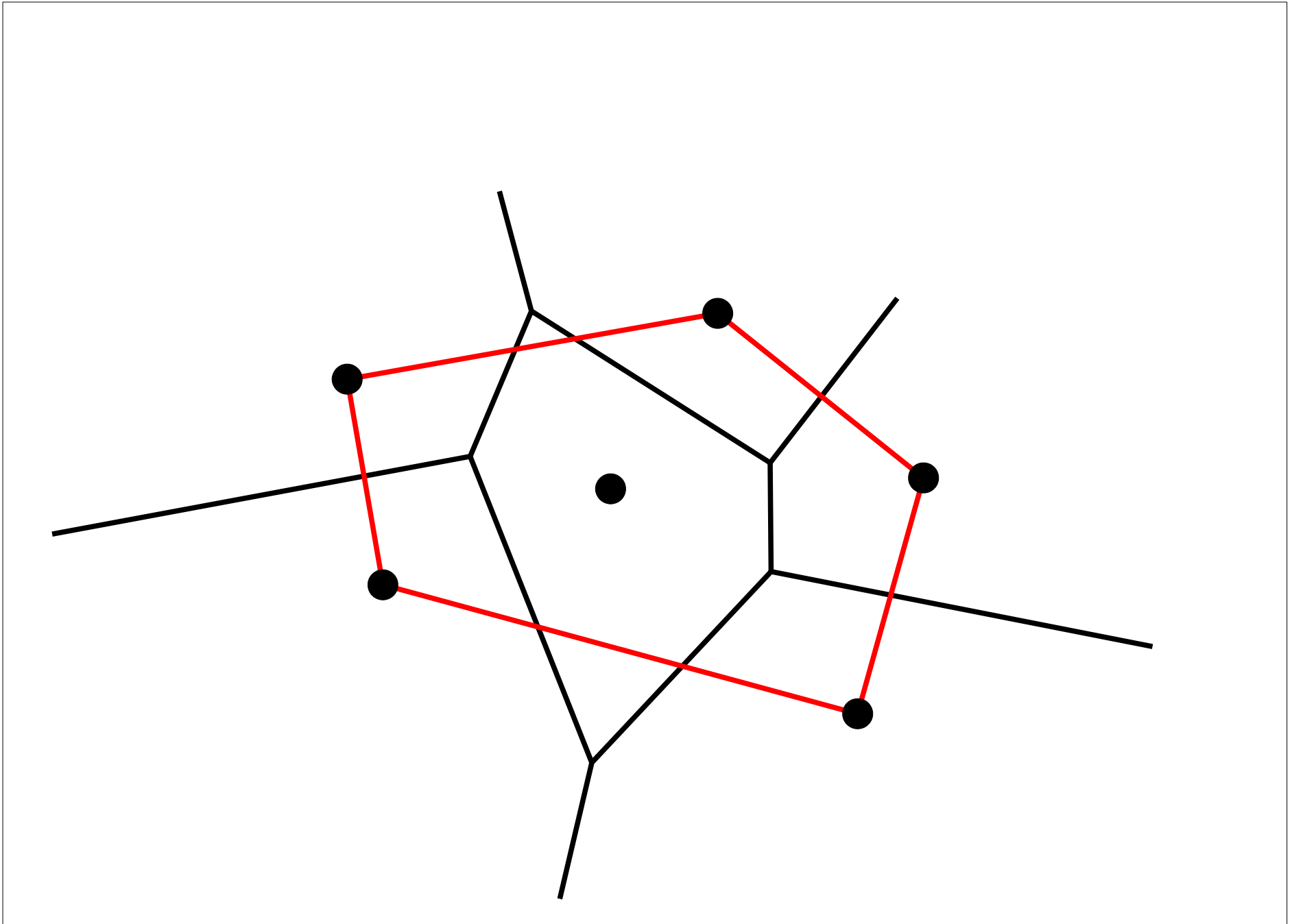
Die Eckpunkte der konvexen Hülle sind die Punkte mit einer **unbeschränkten Voronoiregion**.











Die Berechnung des Voronoidiagramms für eine Menge von Punkten in der Ebene unter Verwendung des euklidischen Abstands ist ein Problem, bei dem man zur Lösung **sehr viele Entwurfsprinzipien für Algorithmen** einsetzen kann:

Gleitgeradenverfahren

(randomisierte) inkrementelle Konstruktion

Teile und Herrsche

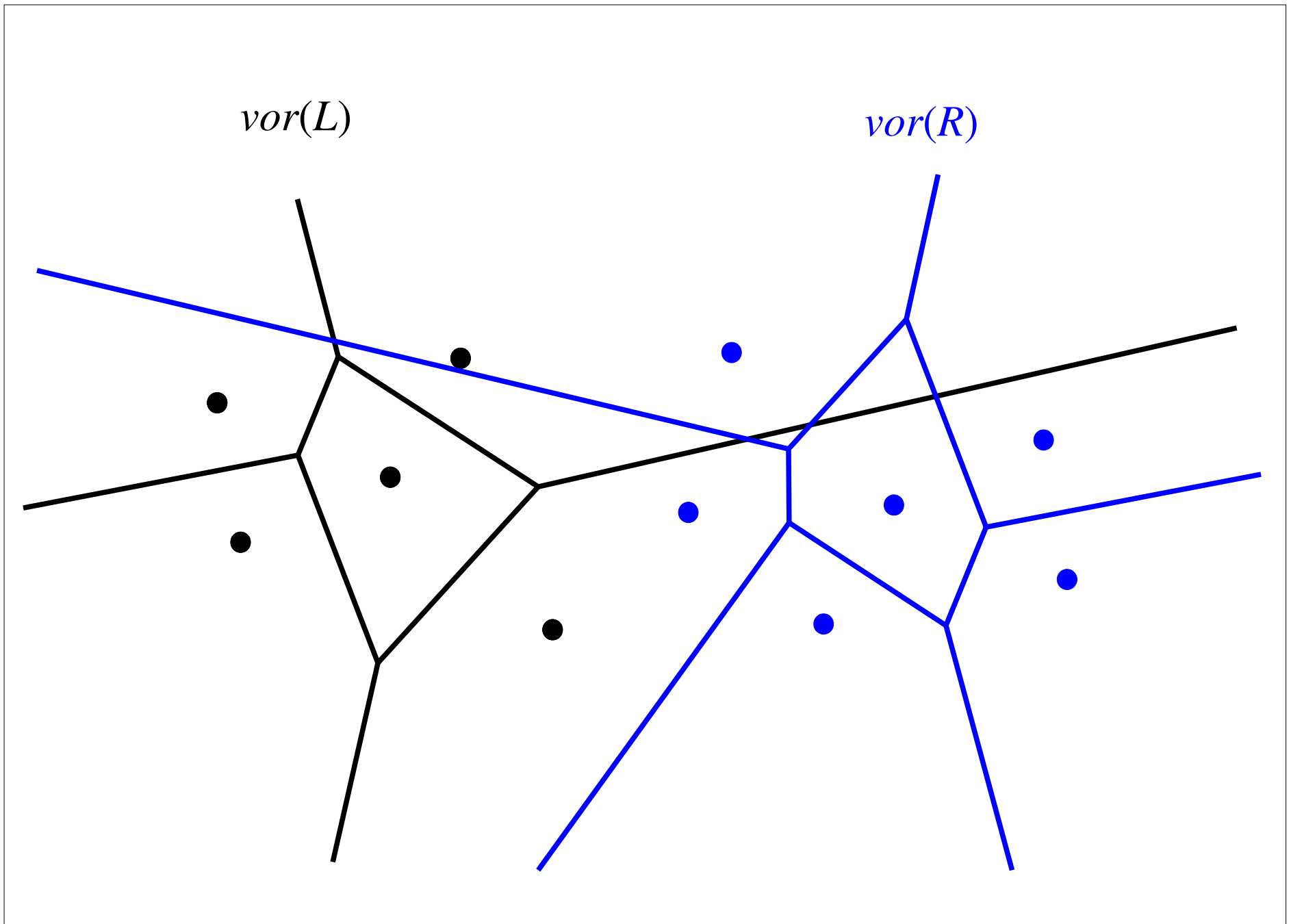
Grundidee eines Teile-und-Herrsche-Algorithmus:

Zuerst wird die gegebene Menge P von Punkten wieder nach aufsteigenden x -Koordinaten **sortiert**.

Dann wählen wir den Median und bilden **zwei Teilprobleme** mit nur noch halb so vielen Punkten:

- Die Menge L der Punkte aus P , die **links vom Median** liegen.
- Die Menge R der Punkte aus P , die nicht in L sind.

Für L und R berechnen wir **rekursiv** die Voronoidiagramme $vor(L)$ und $vor(R)$.



Die Voronoidiagramme $vor(L)$ und $vor(R)$ müssen nun noch zusammengesetzt (**gemischt**) werden.

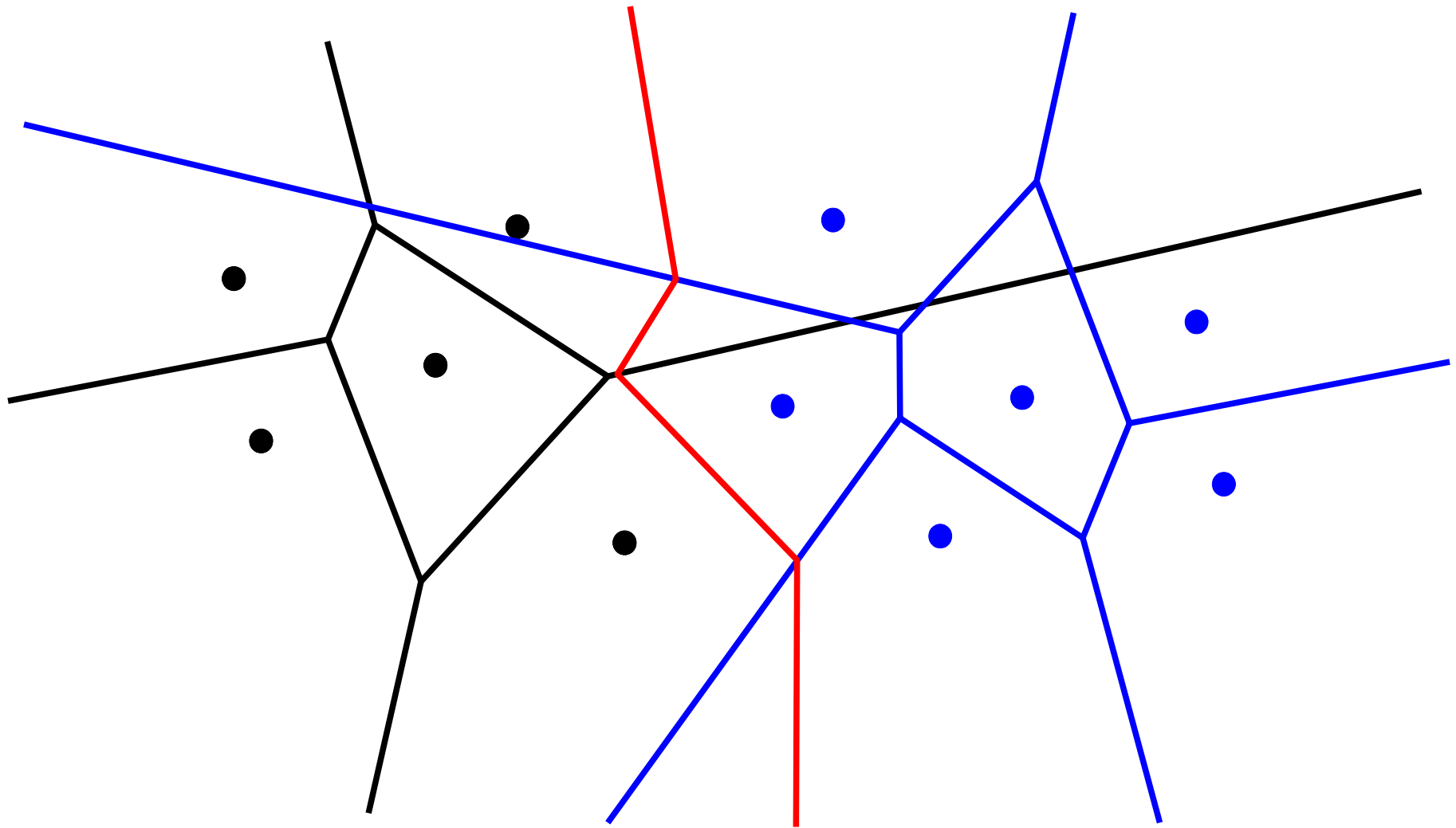
Dazu überlegt man sich:

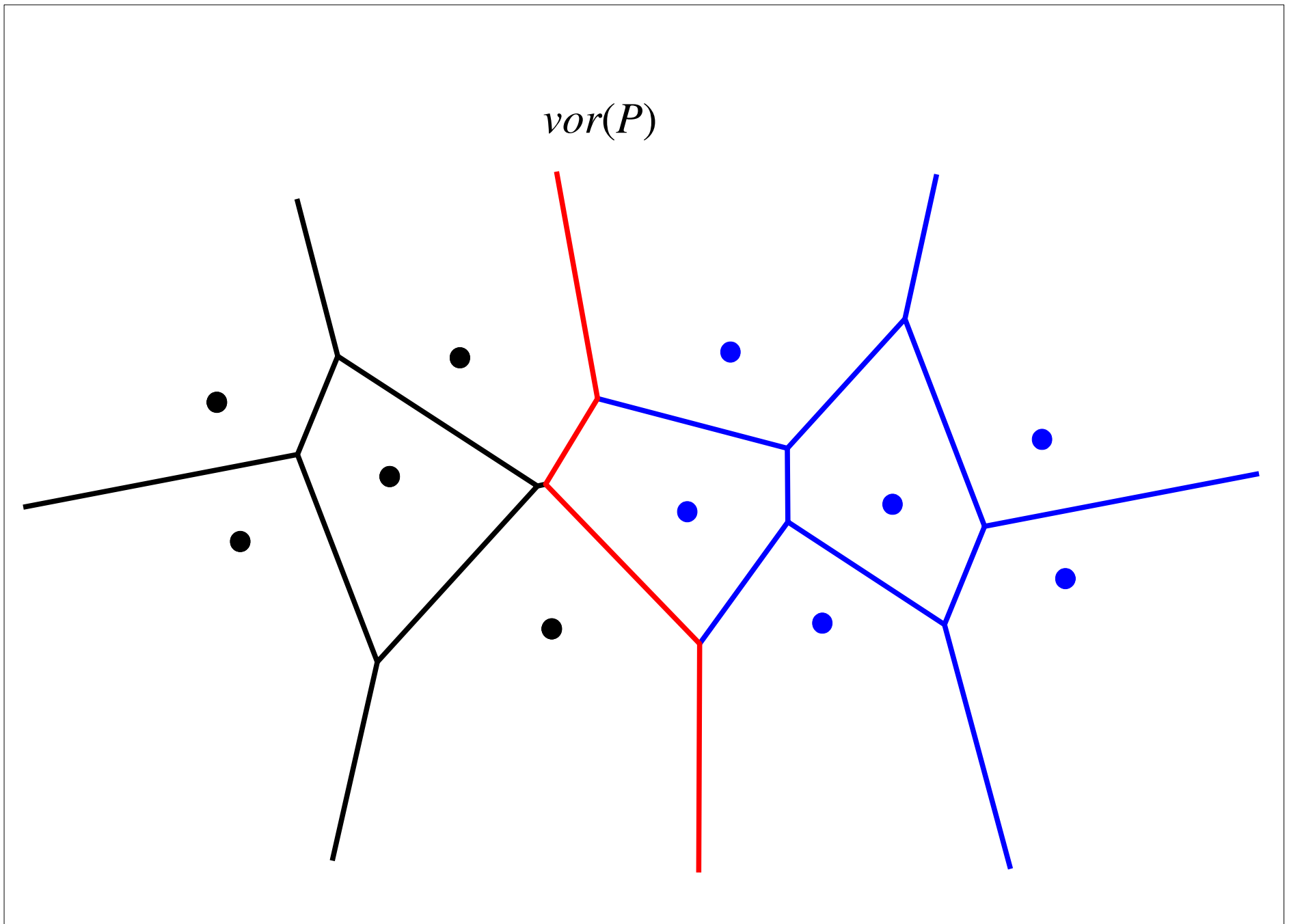
Kanten im zu berechnenden Voronoidiagramm $vor(P)$ sind

entweder **Teil einer Voronoikante** aus $vor(L)$ bzw. $vor(R)$

oder verlaufen **zwischen zwei Voronoiregionen**, von denen je eine zu einem Punkt aus L und einem Punkt aus R gehört.

Der **Bisektor** zwischen L und R .





Das **Mischen** von $vor(L)$ und $vor(R)$ läuft also darauf hinaus, den Bisektor zwischen L und R zu berechnen.

Deshalb überlegen wir uns zunächst einige **Eigenschaften dieses Bisektors**.

Lemma

Der Bisektor ist ein y -monotoner Polygonzug.

Beweis:

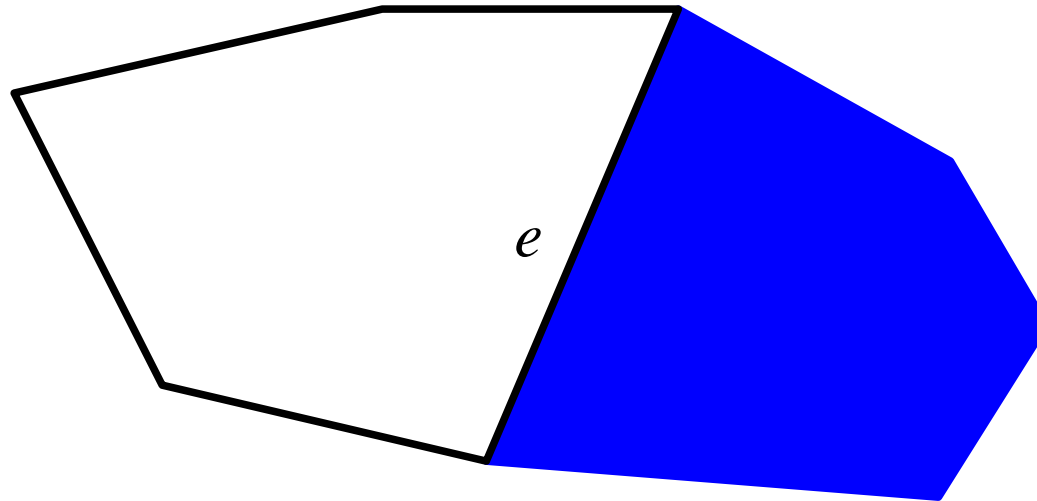
Wir betrachten $vor(P)$ und färben jede Voronoiregion entweder weiß oder blau.

Eine Voronoiregion $vor(q,P)$ wird weiß gefärbt, wenn q in L liegt, und blau, wenn q in R liegt.

Der Bisektor ist nun gerade die Grenze zwischen weißen und blauen Regionen.

Damit ist erst einmal klar, dass der Bisektor aus Strecken und Strahlen besteht.

Betrachten wir eine beliebige Kante e des Bisektors.



e liegt zwischen einer weißen und einer blauen Region.

Dabei kann nach unserer anfänglichen Aufteilung der Punkte in L und R die Kante e nicht waagerecht verlaufen und links muss sich die weiße Region befinden.

Wenn man nun von einem beliebigen Punkt des Bisektors losläuft, sodass die weißen Regionen immer rechts liegen, dann bewegt man sich immer weiter Richtung fallender y -Koordinaten.

Also besteht der Bisektor aus einem oder mehreren sich **nicht kreuzenden** unbeschränkten y -monotonen Polygonzügen.

Nehmen wir nun an, der Bisektor besteht aus mindestens zwei Polygonzügen.

Dann sind diese Polygonzüge in natürlicher Weise von links nach rechts angeordnet.

Links vom ersten Polygonzug befinden sich nur weiße Regionen, rechts davon nur blaue Regionen.

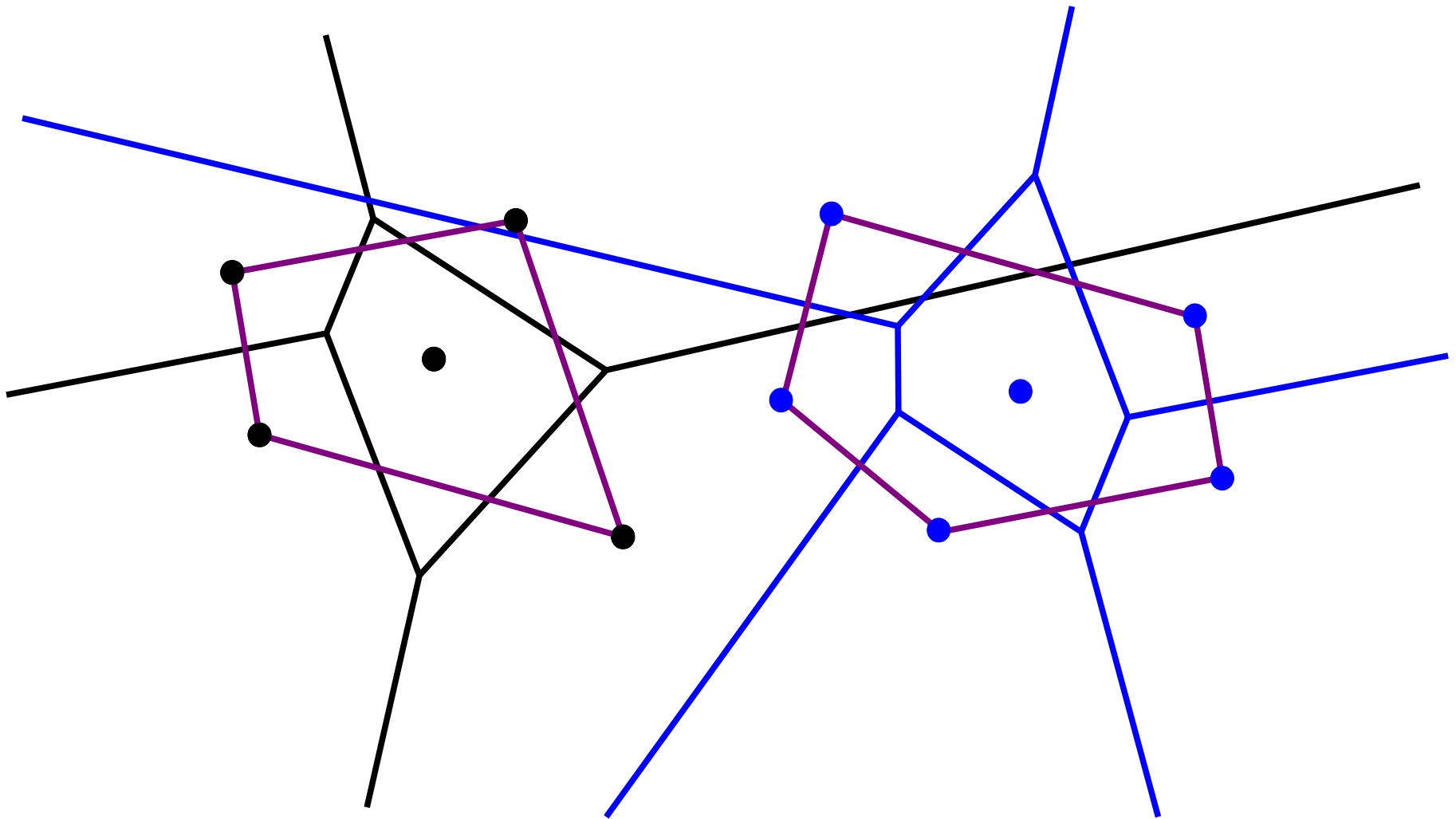
Links vom zweiten Polygonzug müssen sich aber auch wieder nur weiße Regionen befinden. Das geht natürlich nicht.

Also besteht der Bisektor zwischen L und R aus nur einem Polygonzug. ■

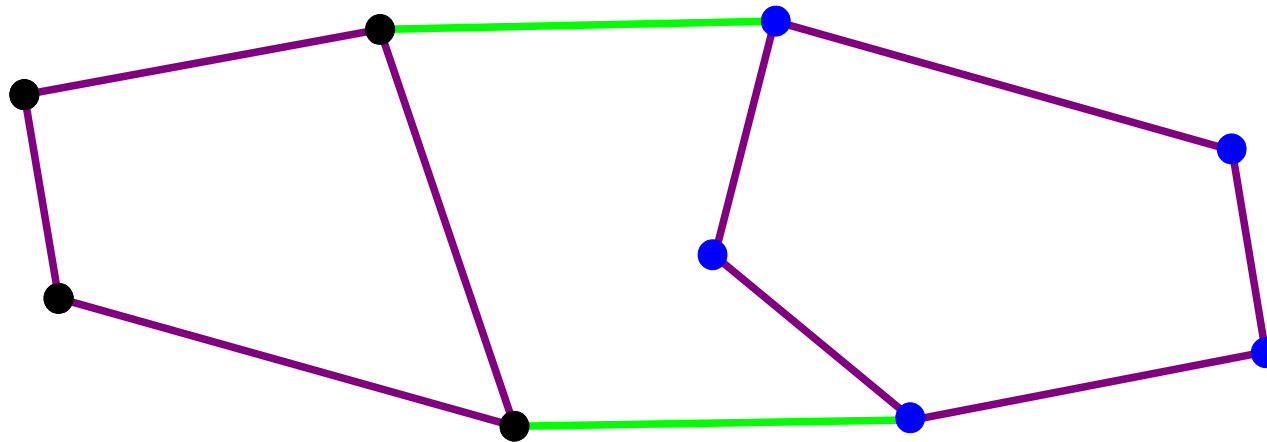
Unsere Strategie wird nun darin bestehen, eines der **Enden des Bisektors**, also einen der beiden Strahlen, zu konstruieren und davon ausgehend den Bisektor simultan durch $vor(L)$ und $vor(R)$ zu verfolgen.

Deshalb machen wir uns zuerst Gedanken, wie wir an eines der Enden herankommen.

Mit $vor(L)$ und $vor(R)$ haben wir auch leichten Zugang zu den **konvexen Hüllen** von L und R .

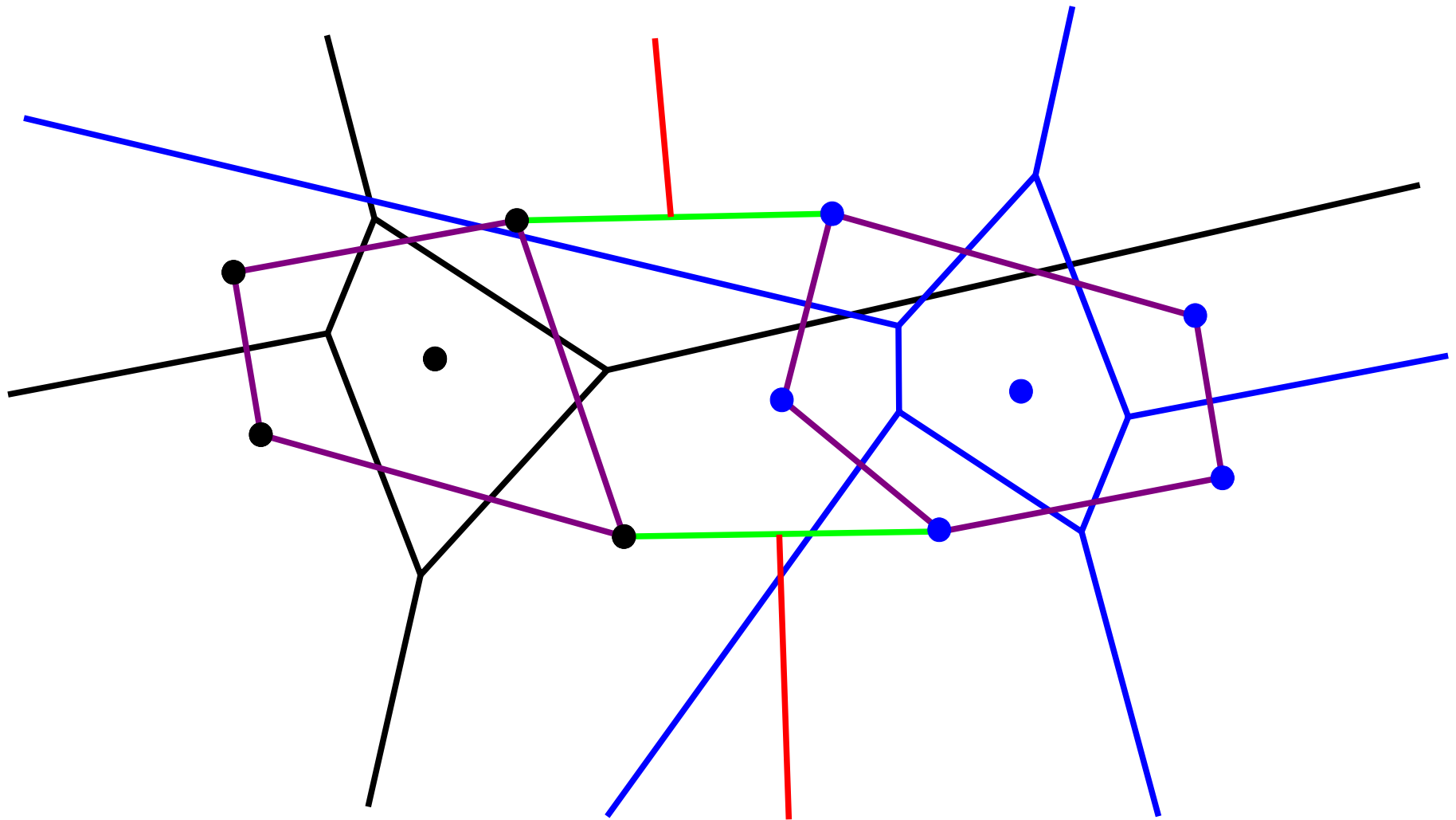


Aus diesen beiden konvexen Hüllen erhält man nun die konvexe Hülle von P .



Dazu muss man die **obere und untere Tangente** an beide konvexen Hüllen finden.

Die **Mittelsenkrechten** auf den Tangenten liefern uns aber die Enden des Bisektors zwischen L und R .



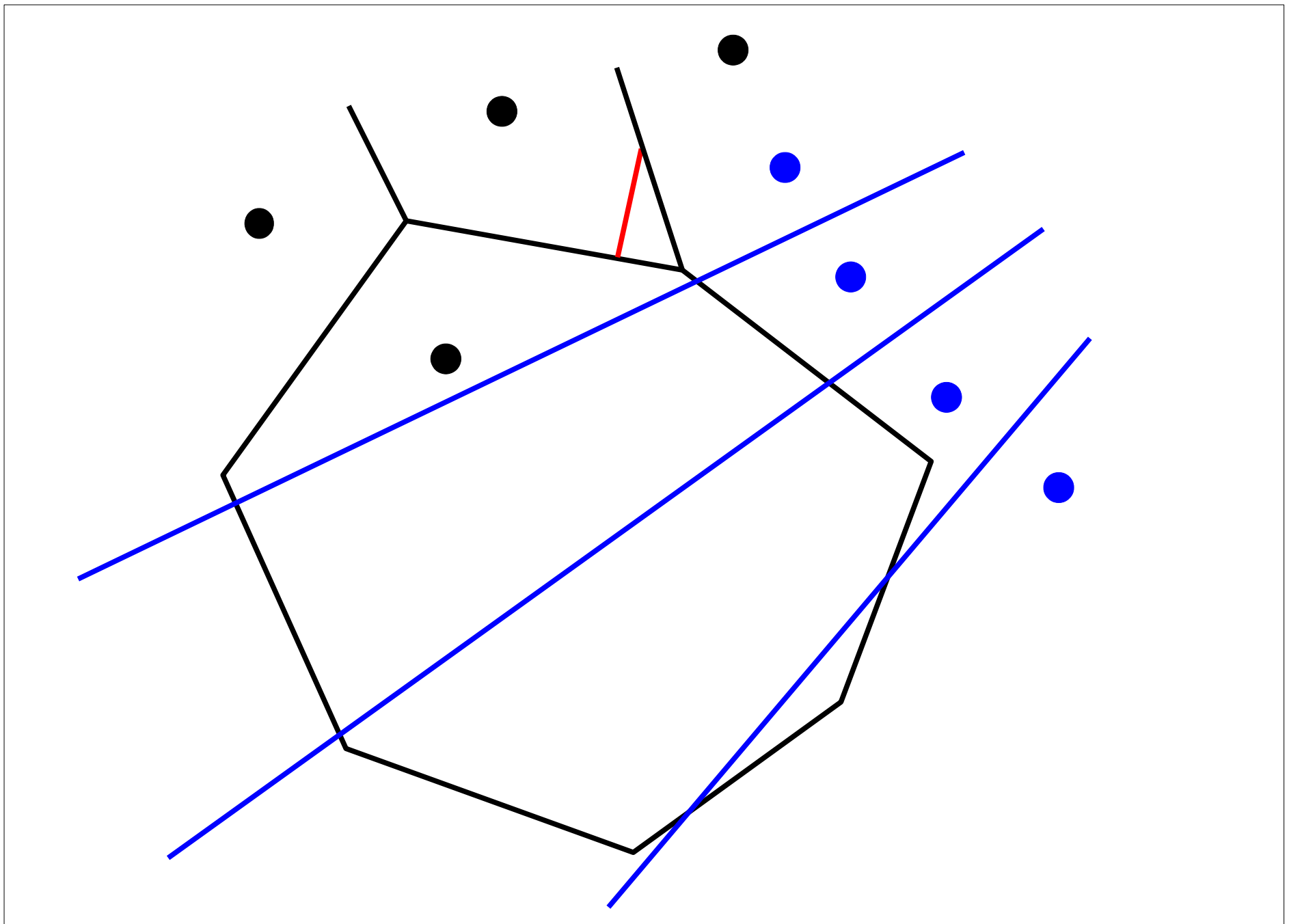
Jetzt geht es darum, wie wir den **Bisektor** zwischen L und R simultan durch $vor(L)$ und $vor(R)$ **verfolgen**.

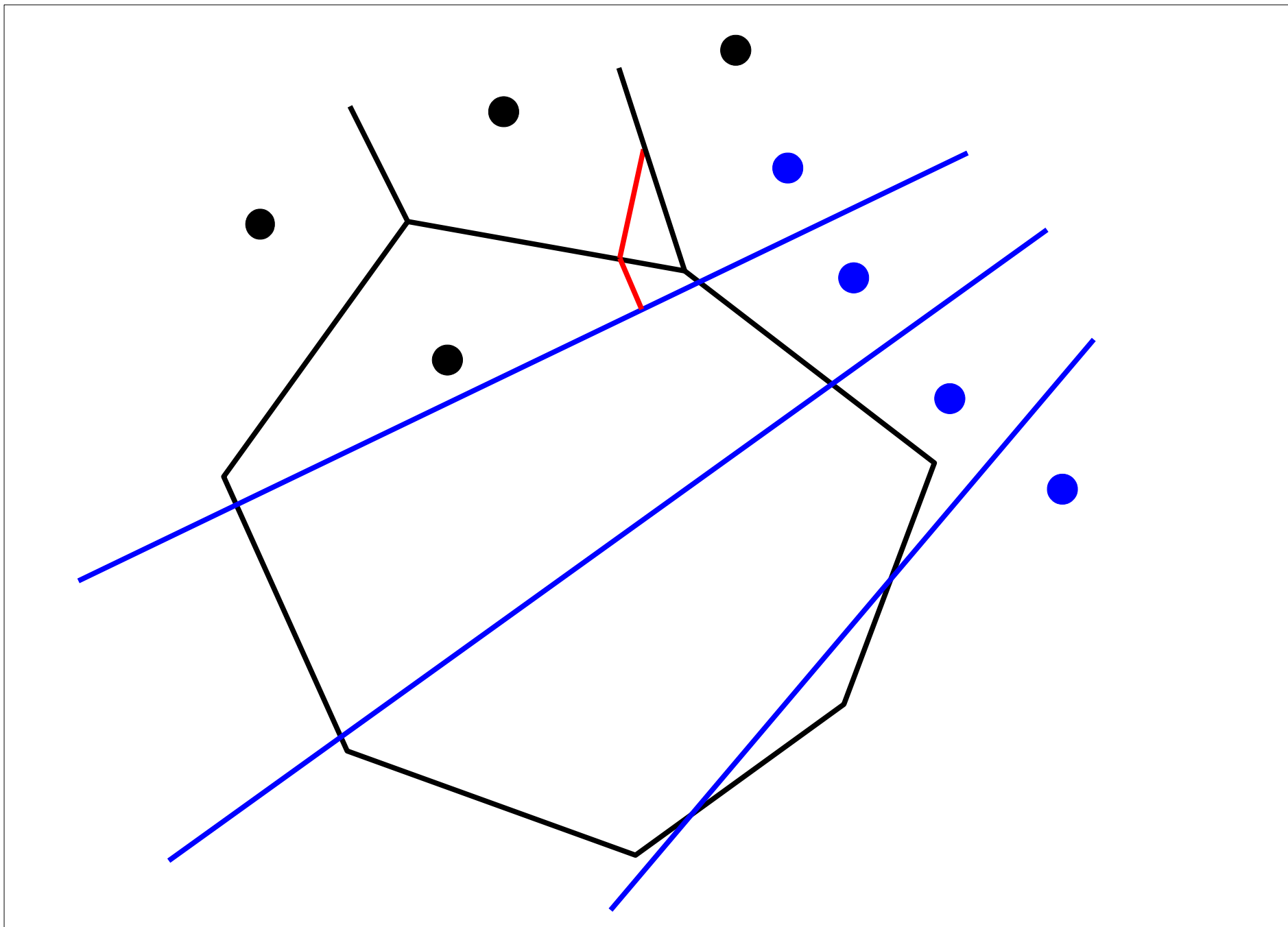
Wir gehen im Folgenden davon aus, dass wir gerade eine Voronoiregion von $vor(L)$ **betreten**.

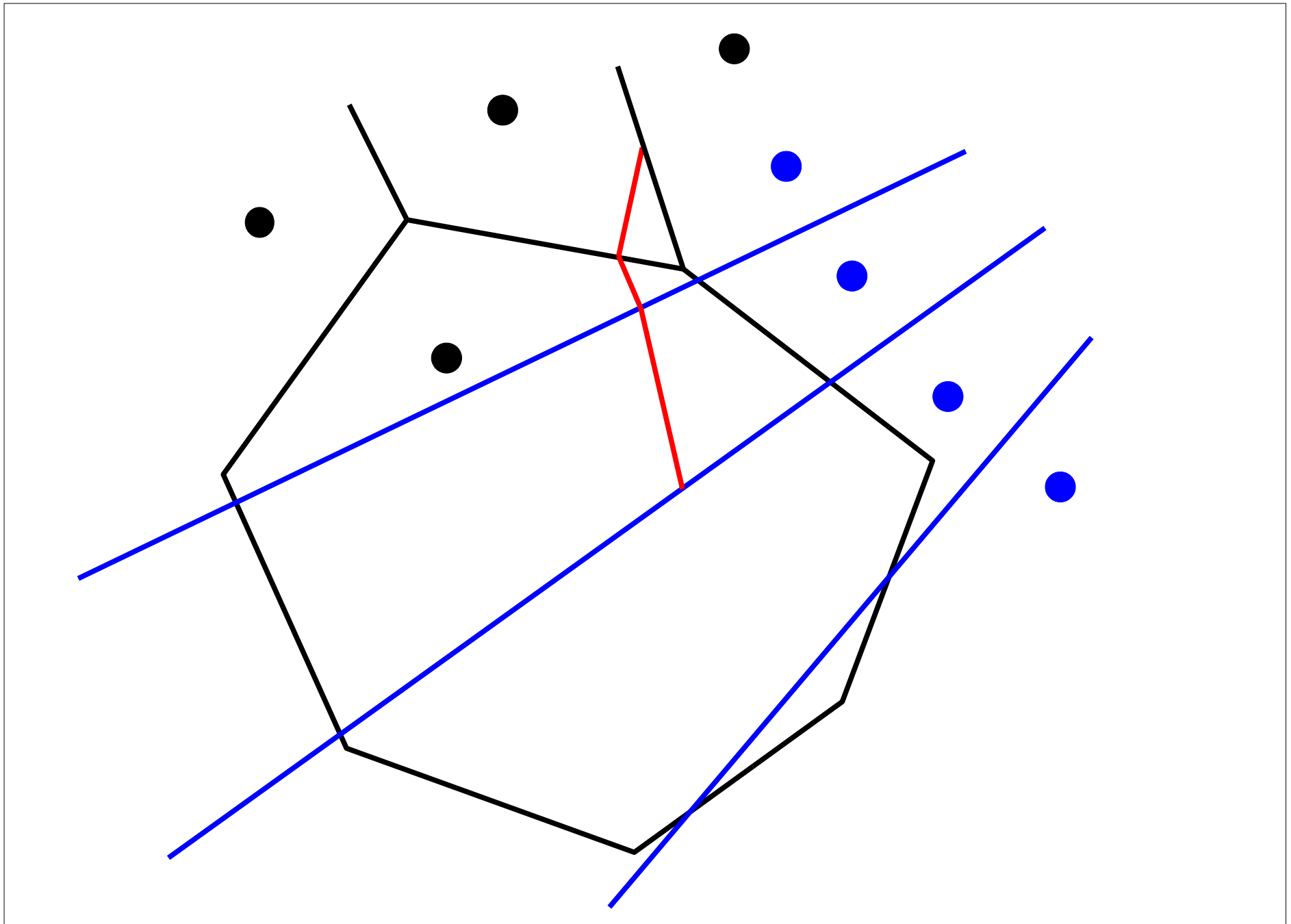
Diese ist ein konvexes Polygon, durch welches wir uns hindurchbewegen müssen.

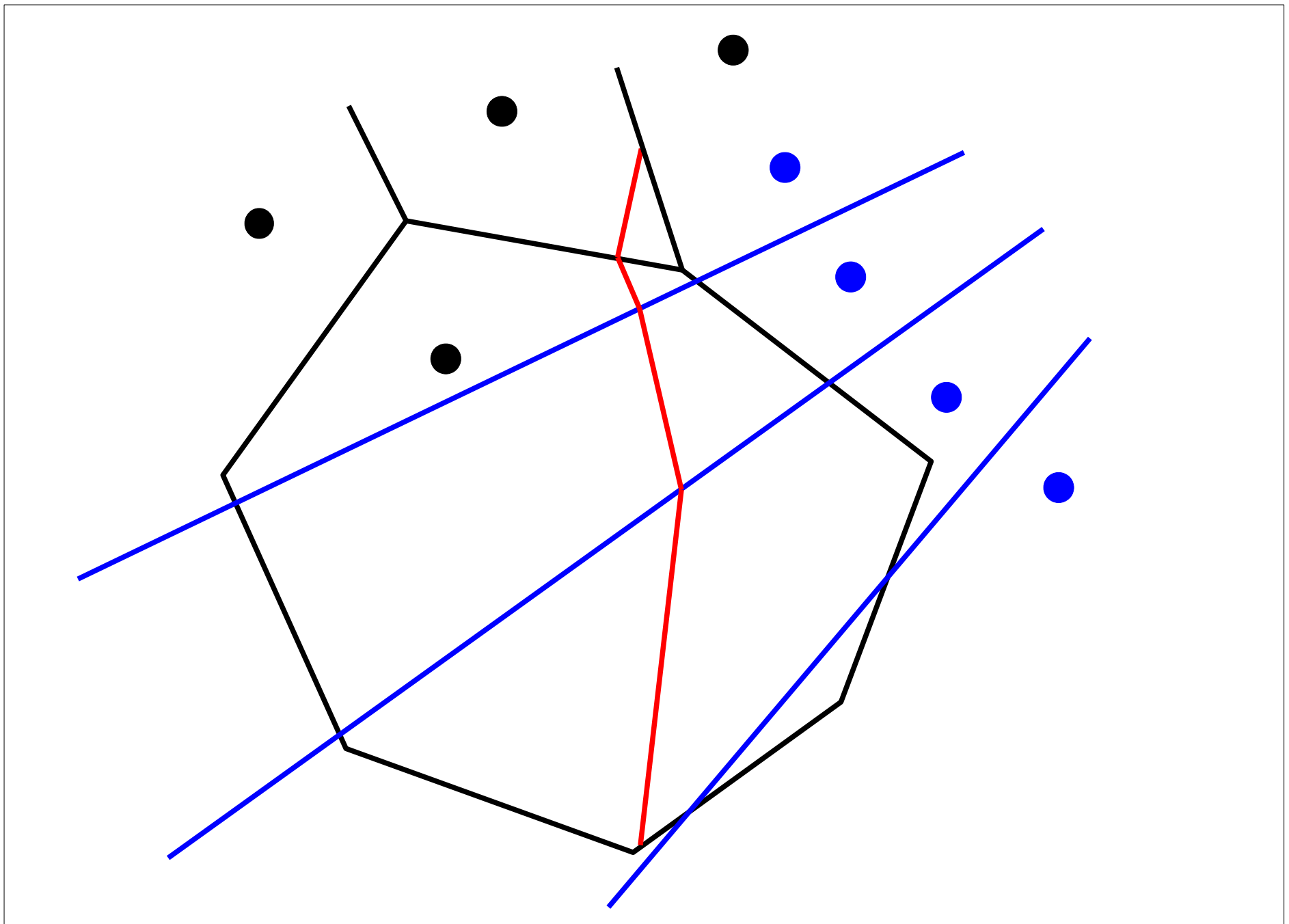
Dabei stoßen wir aber möglicherweise auf Kanten zwischen Voronoiregionen von $vor(R)$.

An diesen Stellen müssen wir immer die **Richtung**, in die wir uns bewegen, entsprechend **anpassen**.









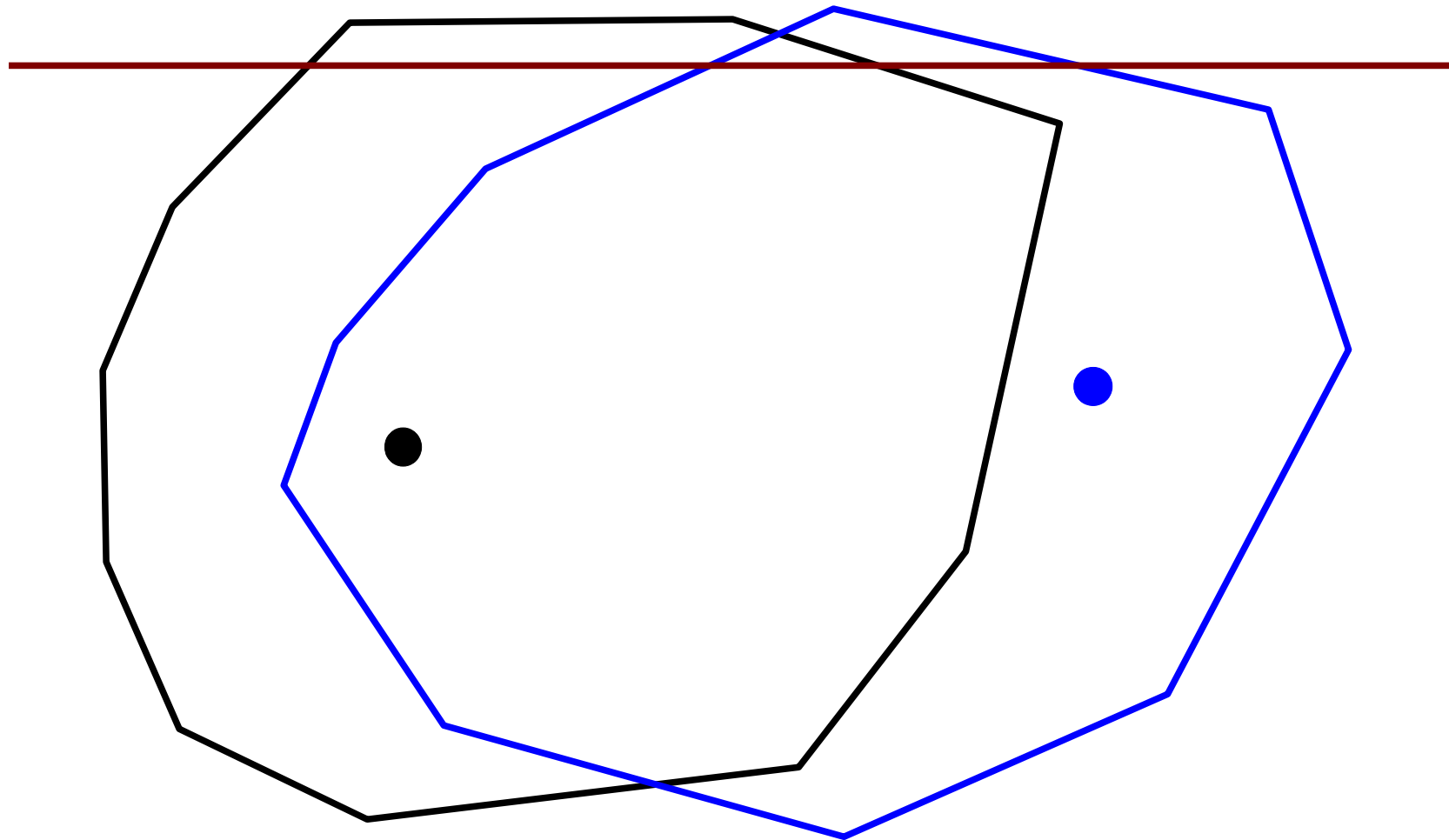
Es gibt zu einem bestimmten Zeitpunkt immer eine Voronoiregion von $vor(L)$ und eine Voronoiregion von $vor(R)$, in der wir uns bewegen.

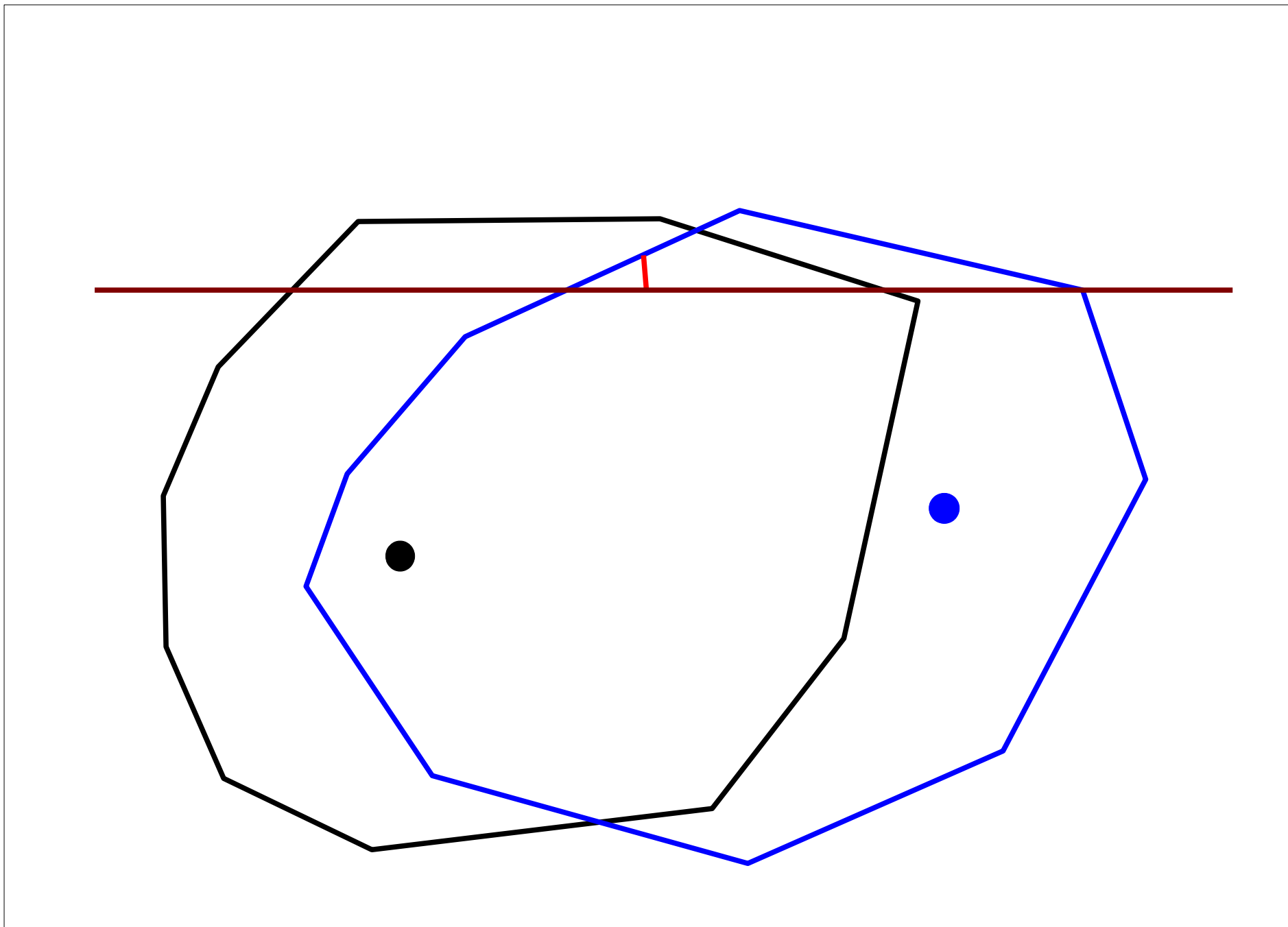
Wir müssen mit einem geeigneten Verfahren feststellen, welche der beiden Regionen wir zuerst verlassen und wo.

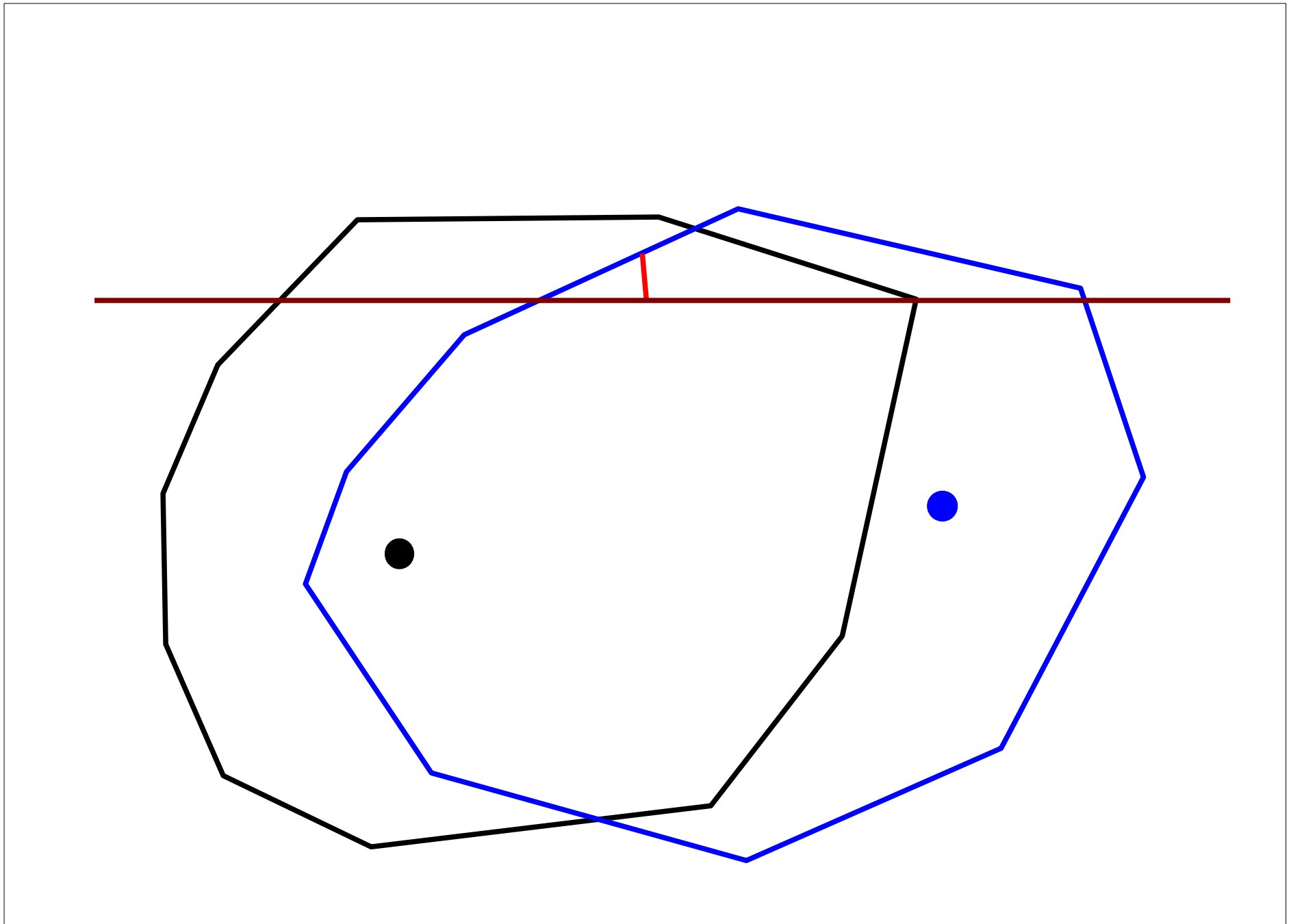
Dabei wollen wir **möglichst keine Voronoikanten** von $vor(L)$ oder $vor(R)$ **mehrmals durchlaufen**.

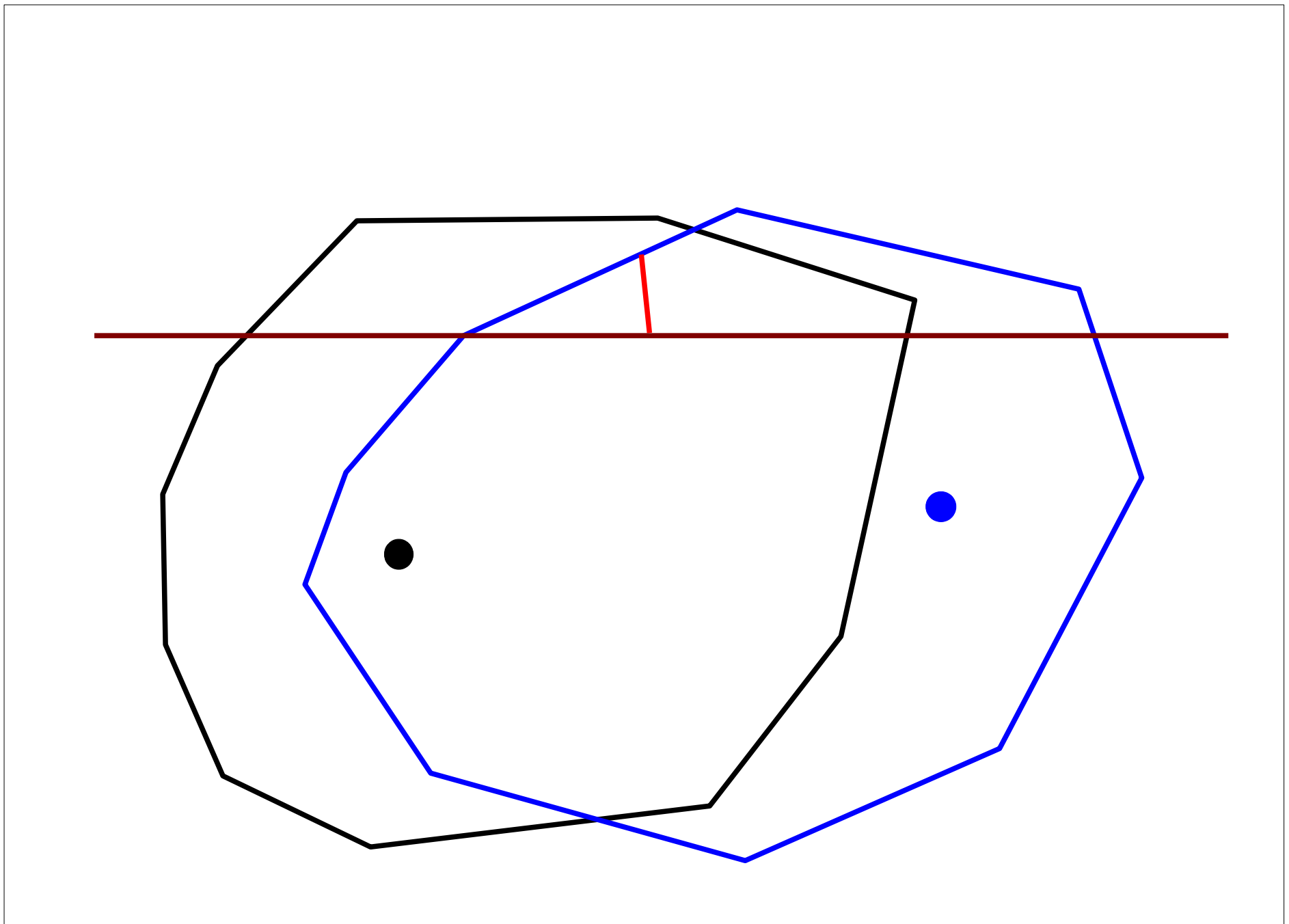
Dies ist insbesondere auch deshalb wichtig, weil eine Voronoiregion aus $vor(L)$ ($vor(R)$) durchaus **mehrmals vom Bisektor besucht** werden kann.

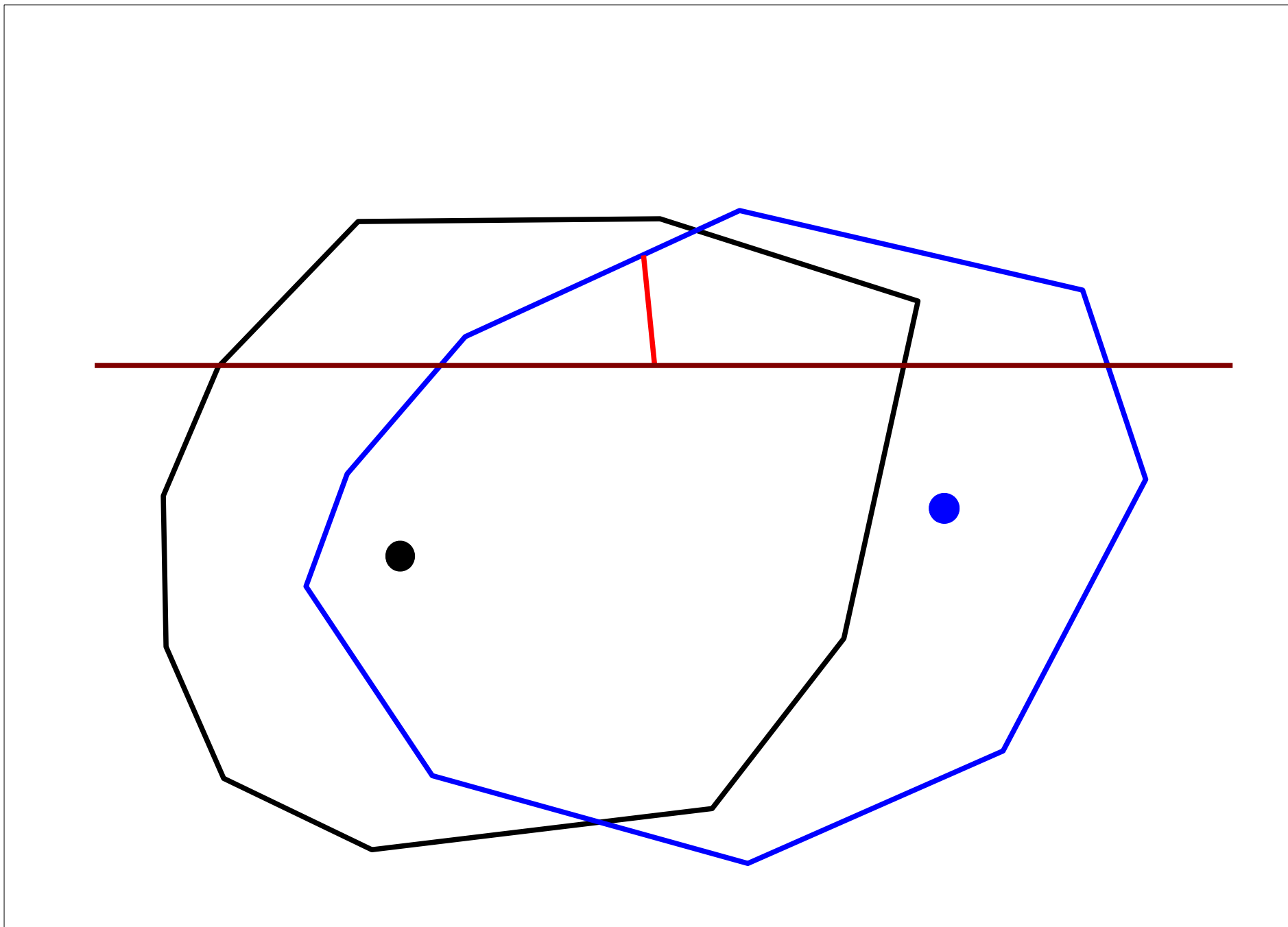
Wir verwenden eine waagerechte **Gleitgerade**, die simultan die Ecken der beiden Voronoiregionen als Ereignispunkte abarbeitet.

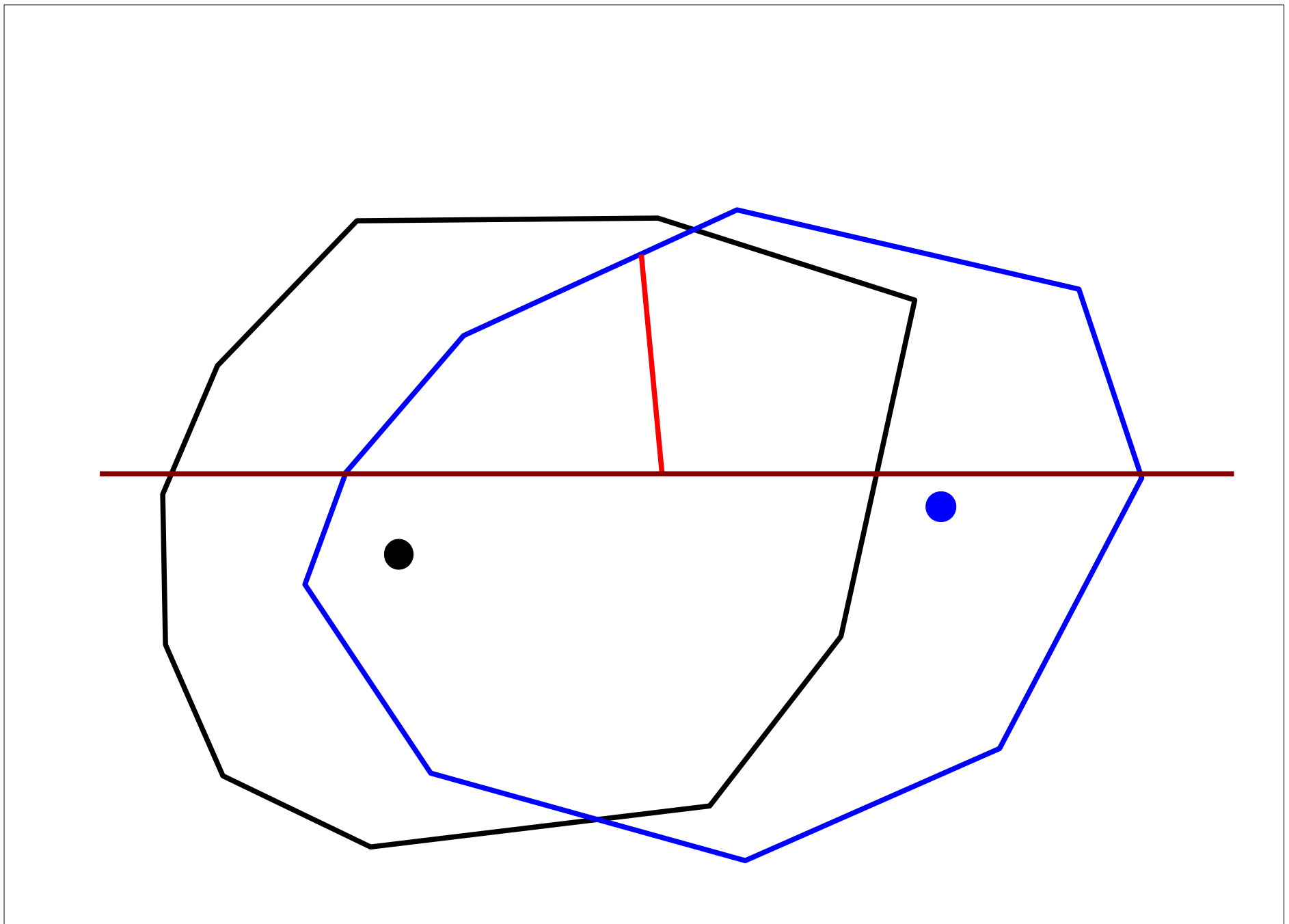


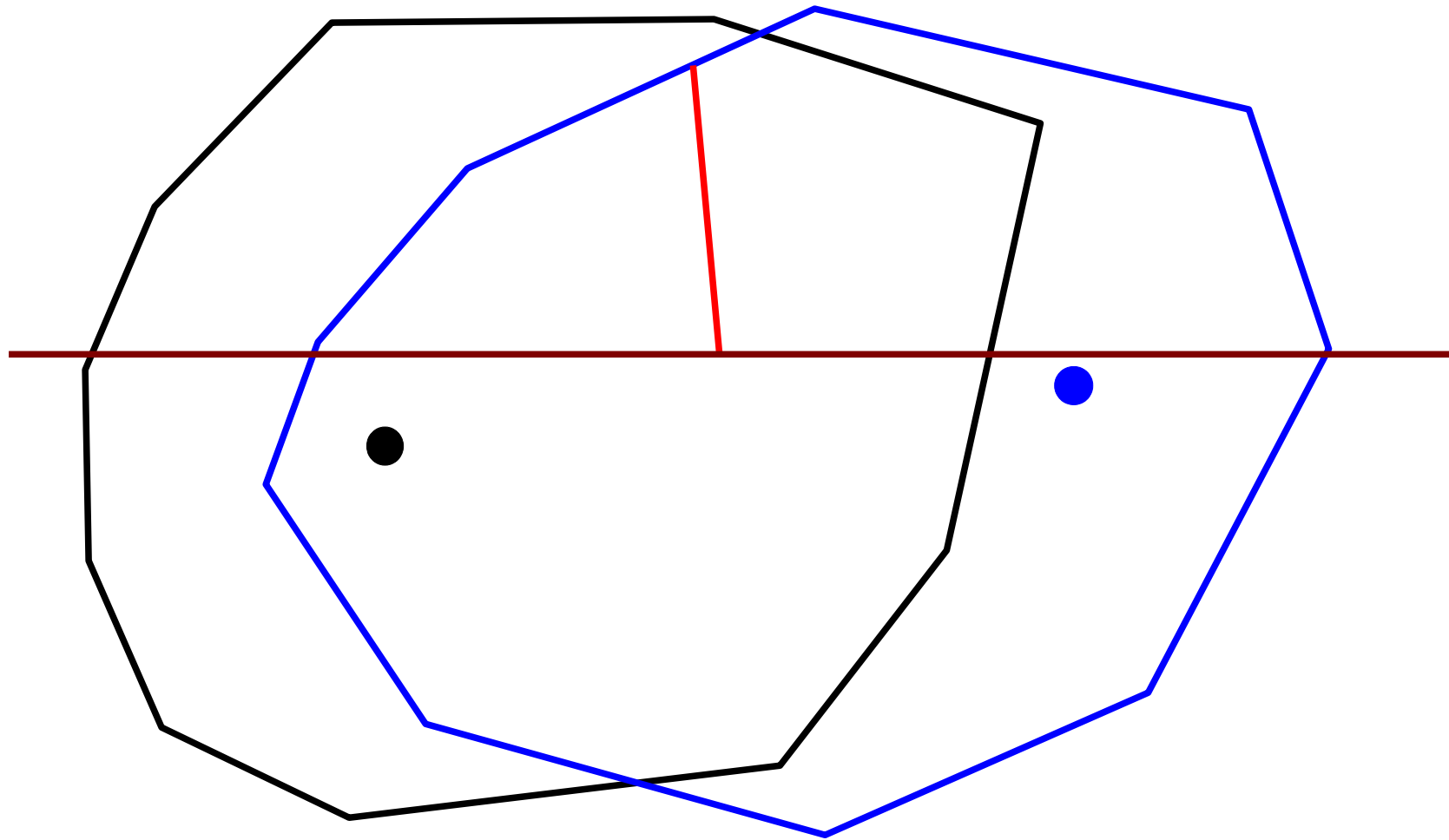


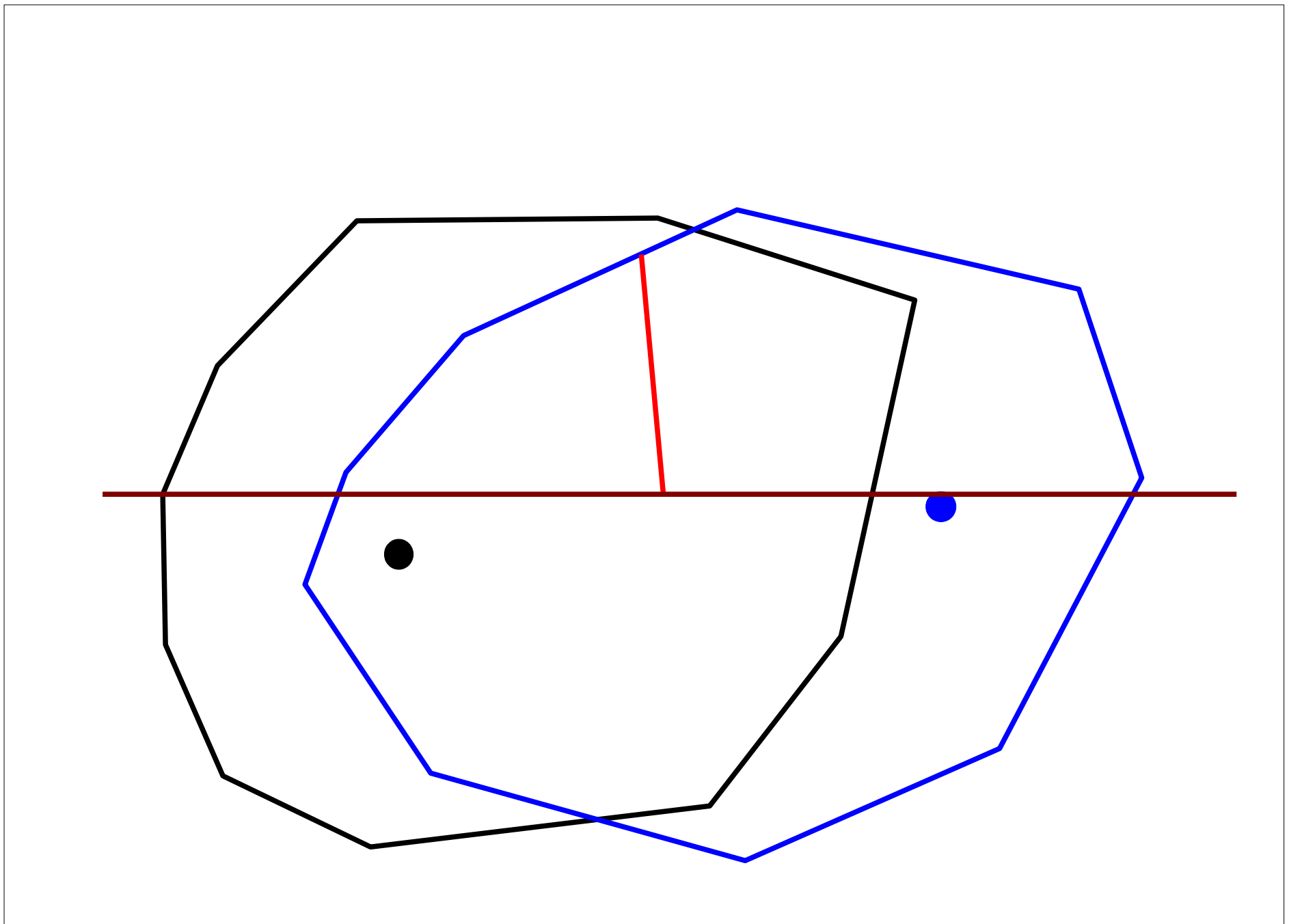


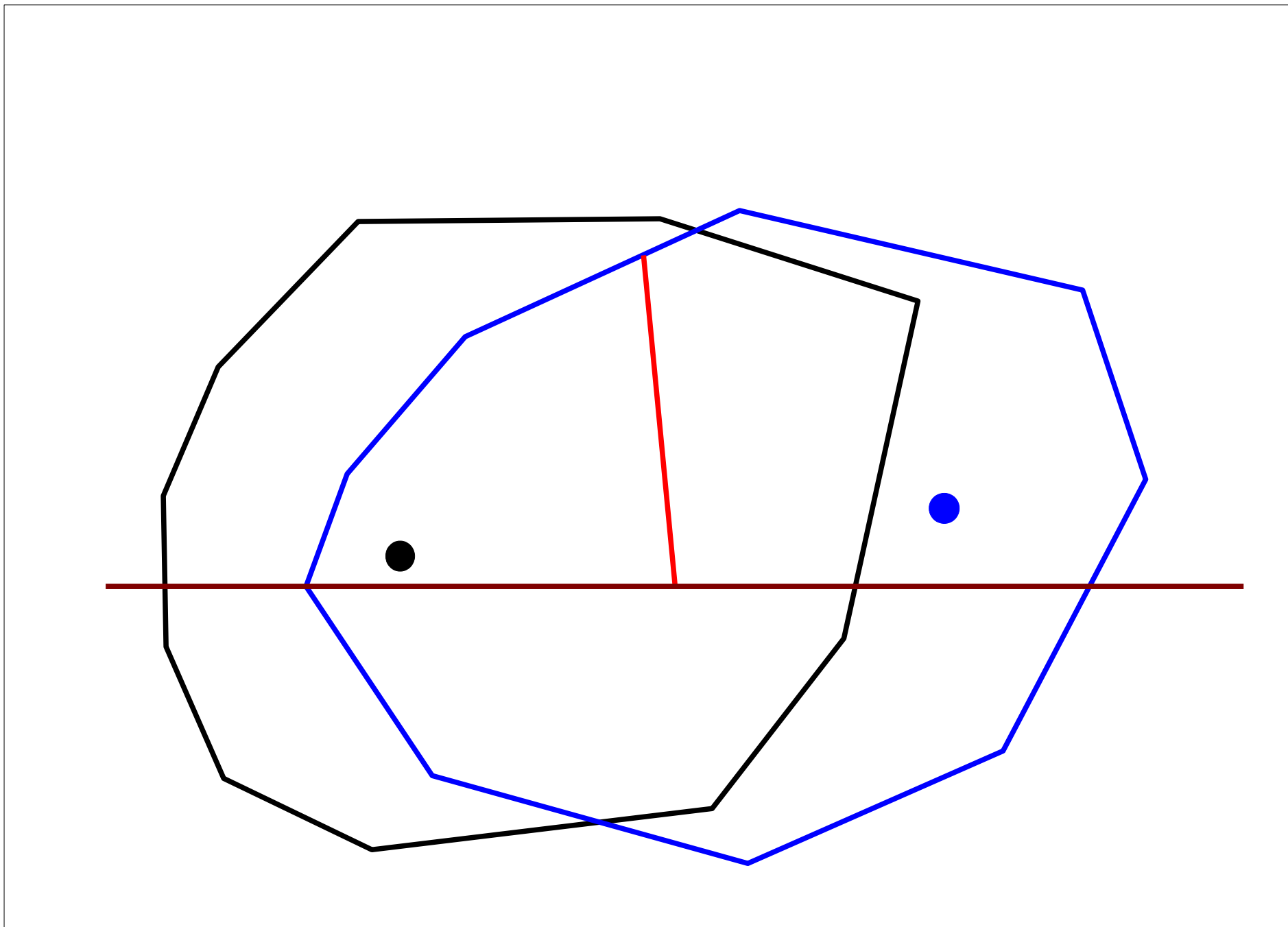


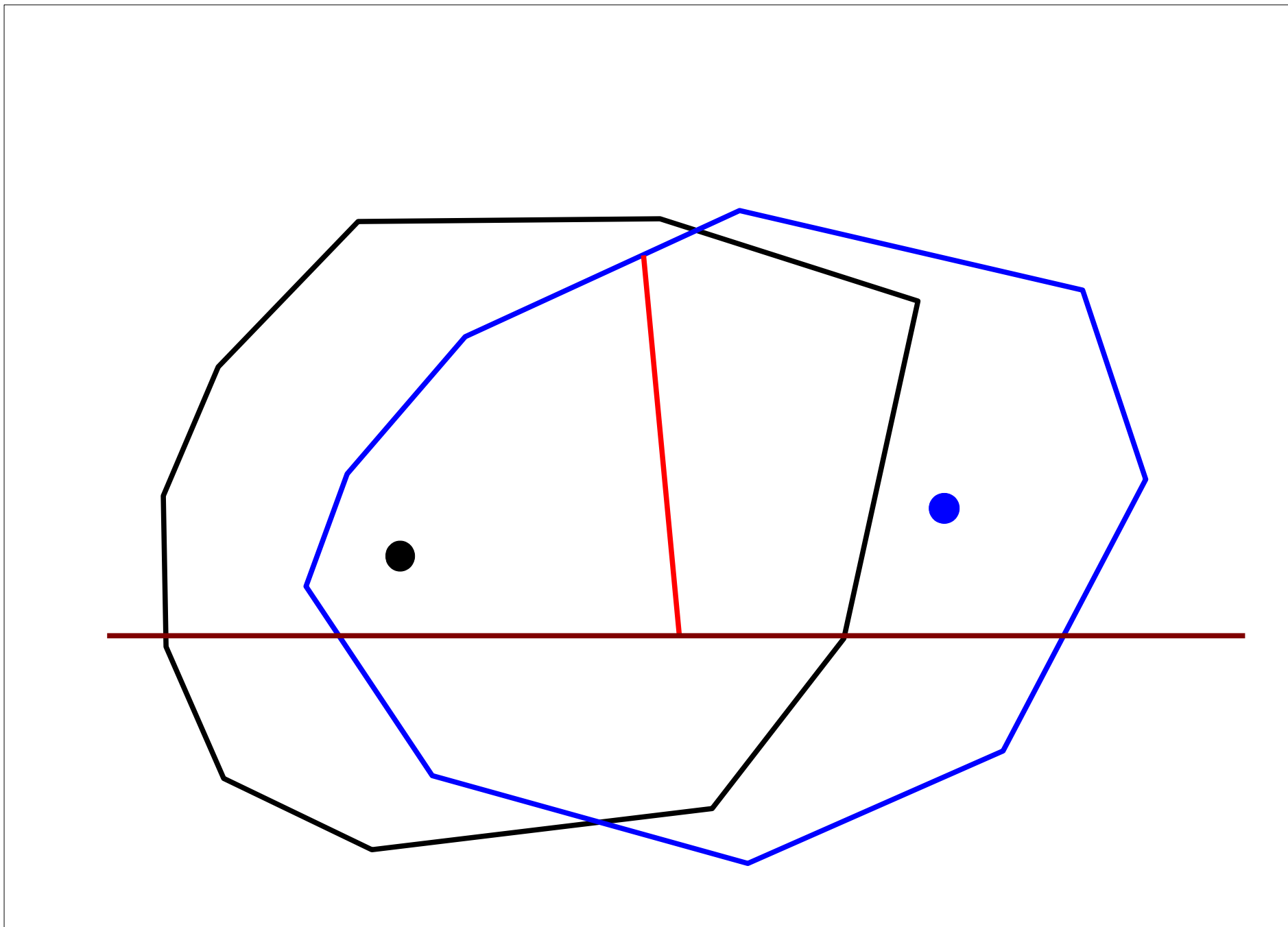


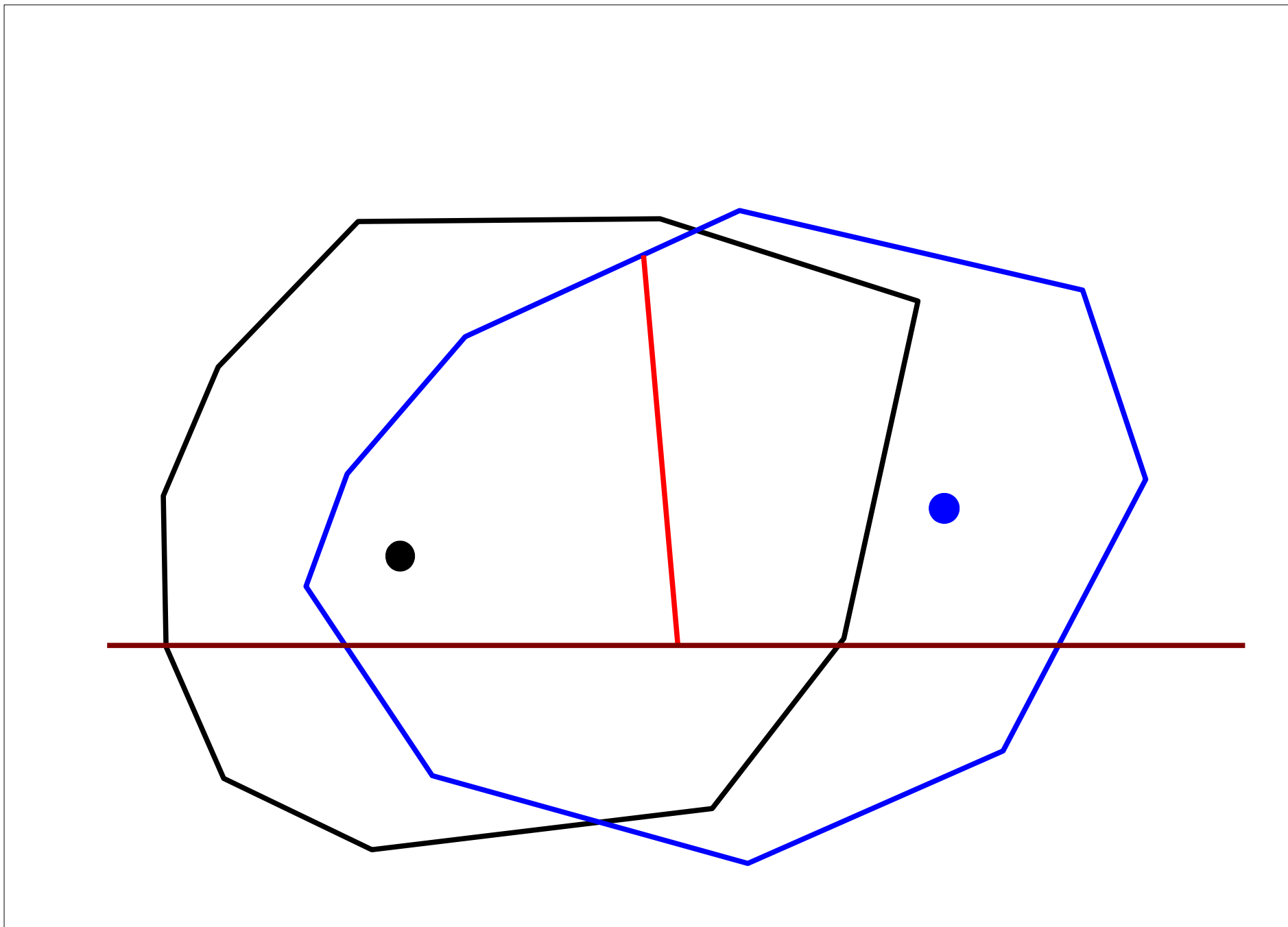


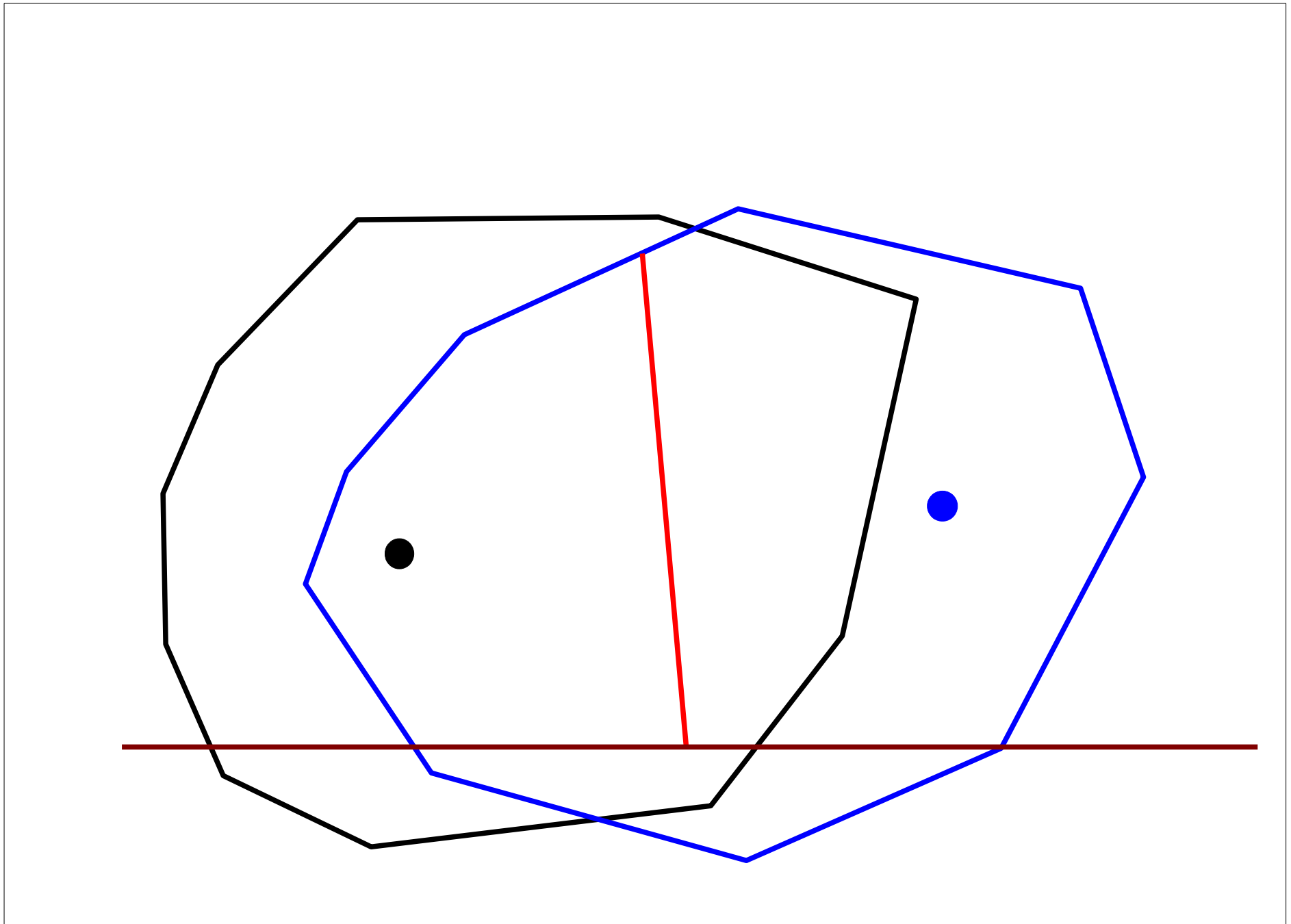


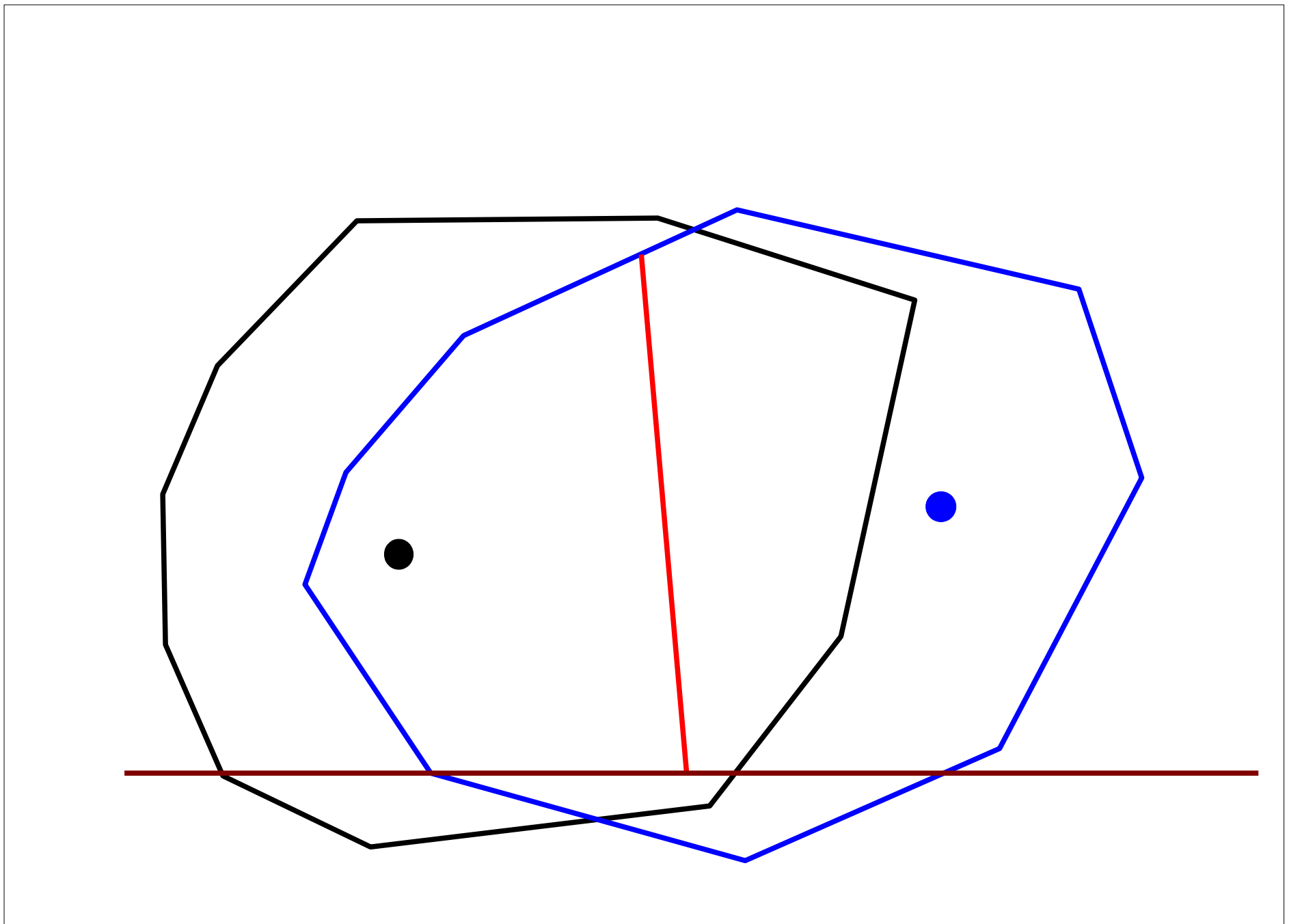


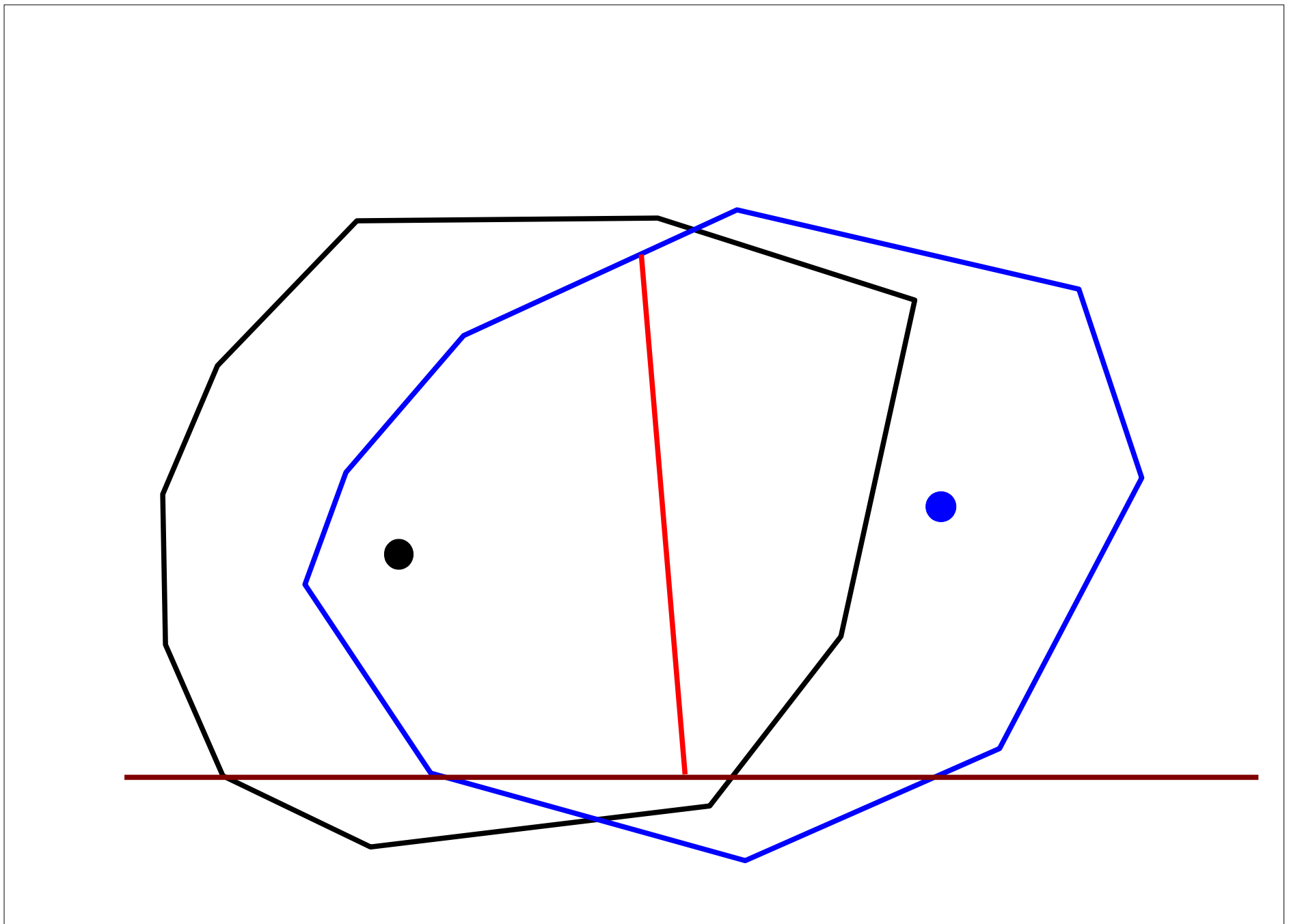


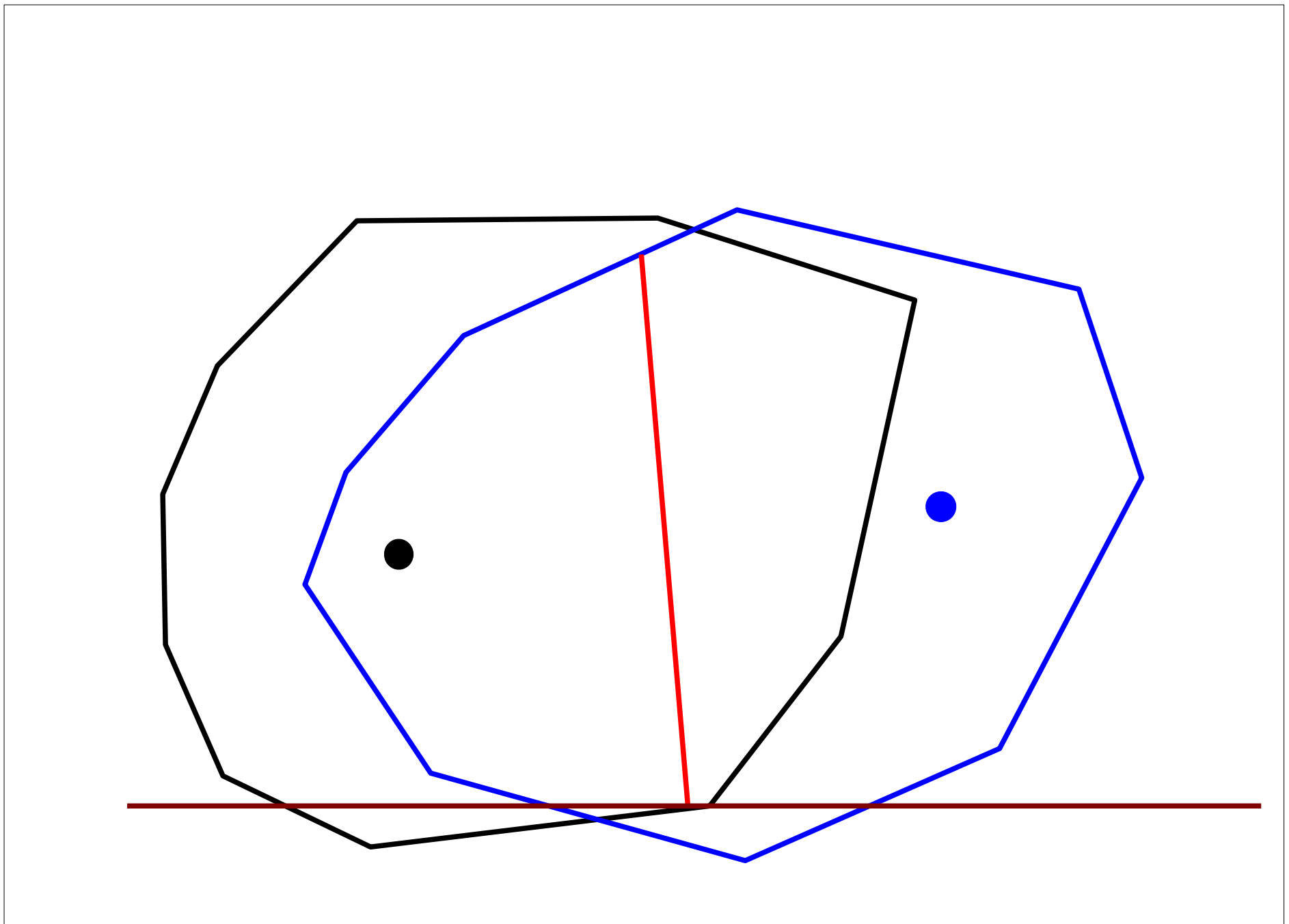


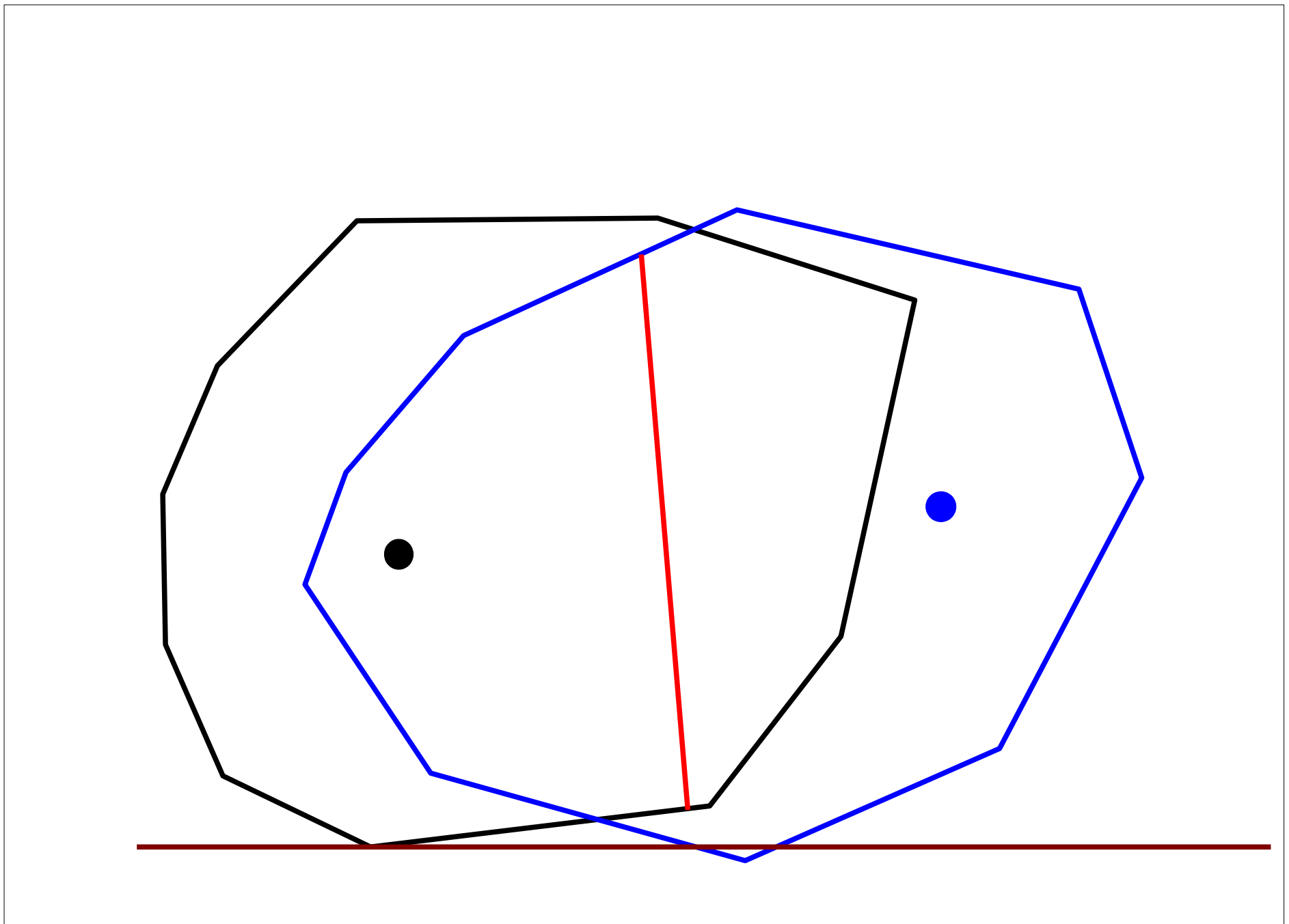












Analyse der Laufzeit

Der **Aufwand zum Mischen** von $vor(L)$ und $vor(R)$ setzt sich zusammen aus dem Aufwand für:

konvexe Hüllen für L und R : $O(n)$

Tangenten bestimmen: $O(n)$

Bisektor verfolgen: $O(n)$

Man beachte dabei, dass der Bisektor y -monoton verläuft, und man sich deshalb immer nur in Richtung fallender y -Koordinaten bewegt.

Deshalb durchläuft man keine Voronoikanten mehrmals.

Für die Laufzeit $T(n)$ unseres Teile-und-Herrsche Algorithmus **insgesamt** erhält man folgende Gleichung:

$$T(n) = 2T\left(\frac{n}{2}\right) + cn$$

Wie bei Mergesort erhält man als Lösung dieser Gleichung:

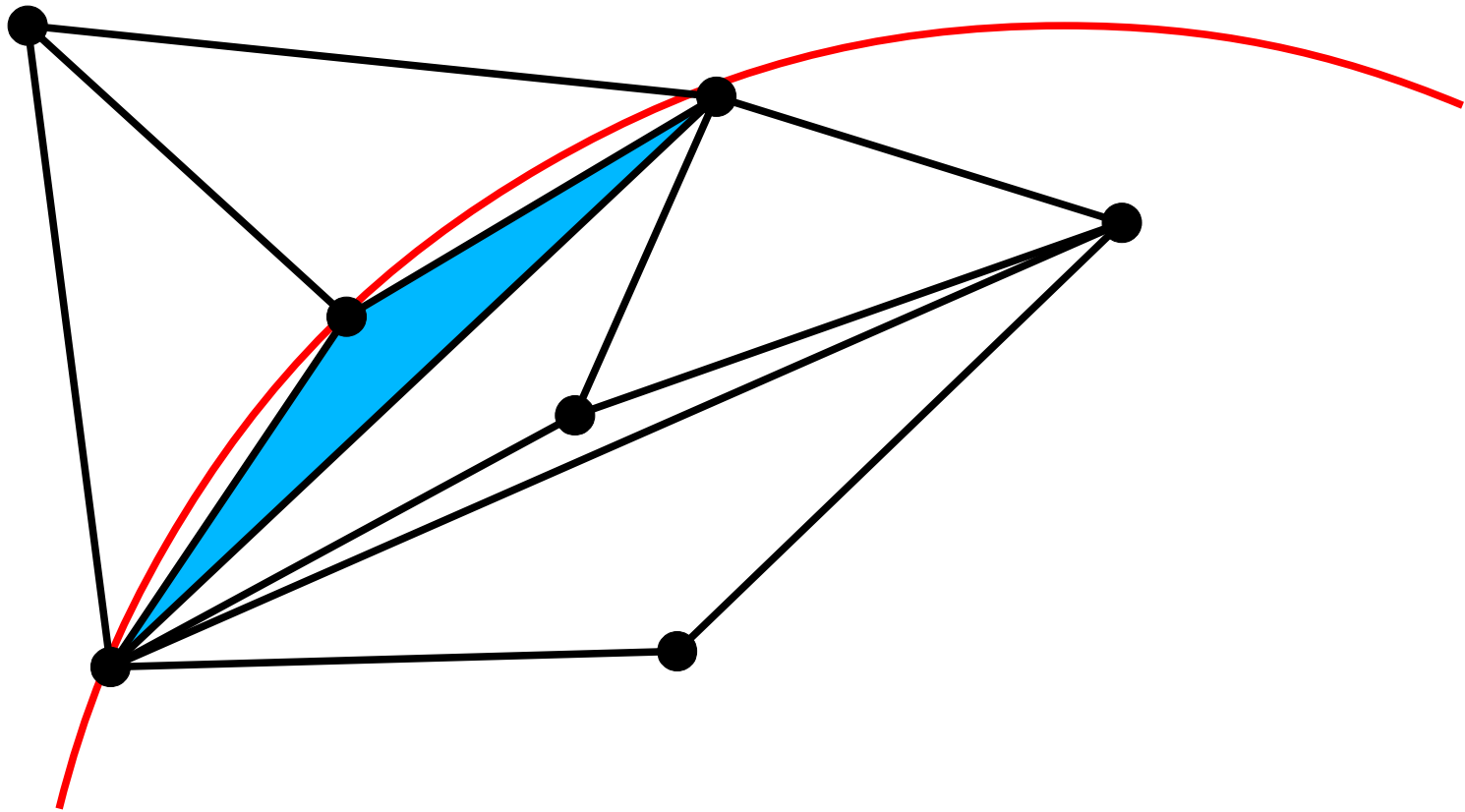
Laufzeit: $O(n \log n)$

4. Delaunay-Triangulationen

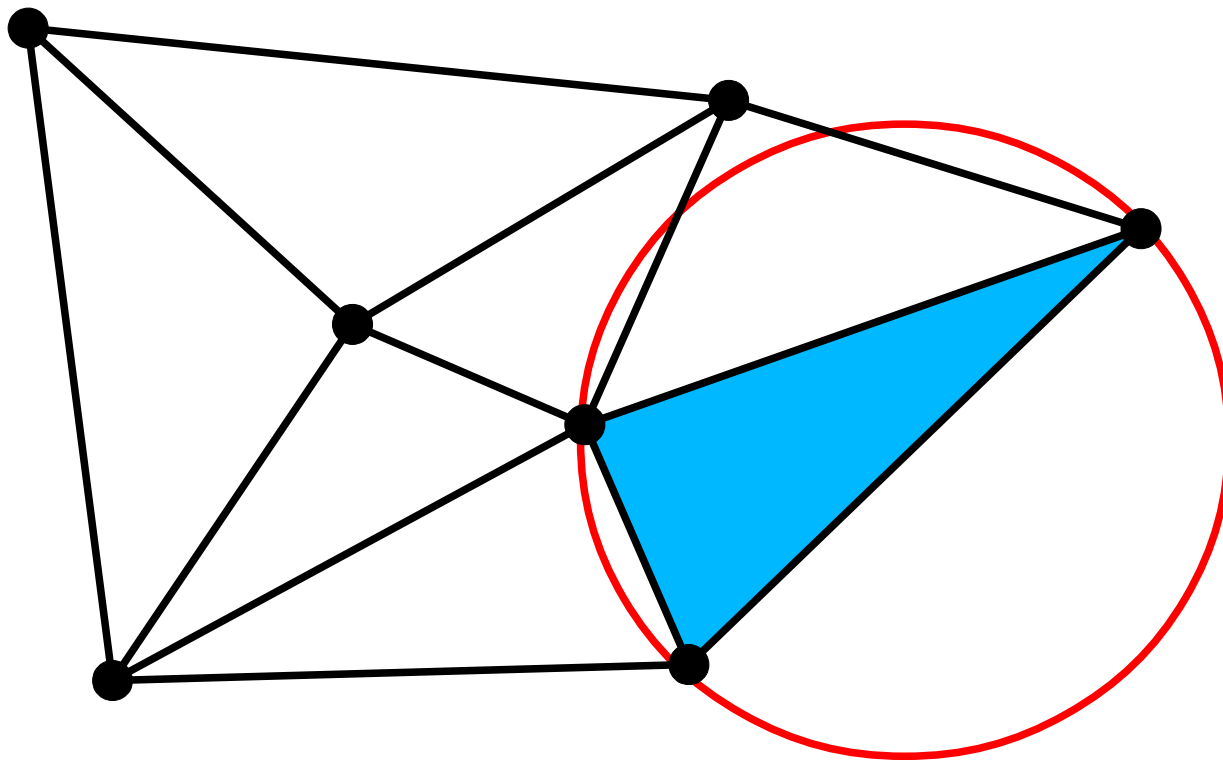
In einer **Delaunay-Triangulation** enthält der Umkreis eines jeden Dreiecks in seinem Inneren keine weiteren Punkte der Eingabe.

Diese Eigenschaft macht Delaunay-Triangulationen für viele Anwendungen sehr nützlich.

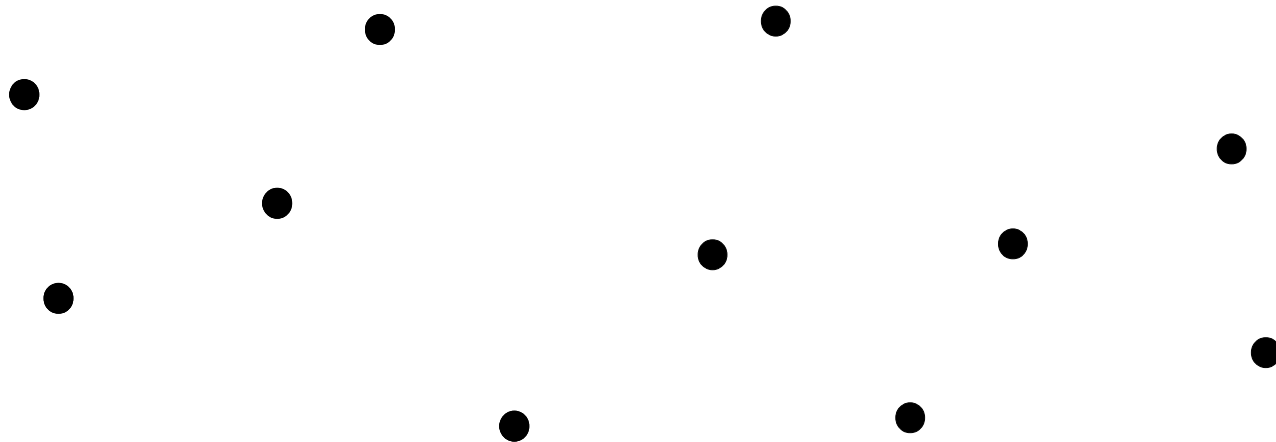
Dies ist zum Beispiel keine Delaunay-Triangulation.



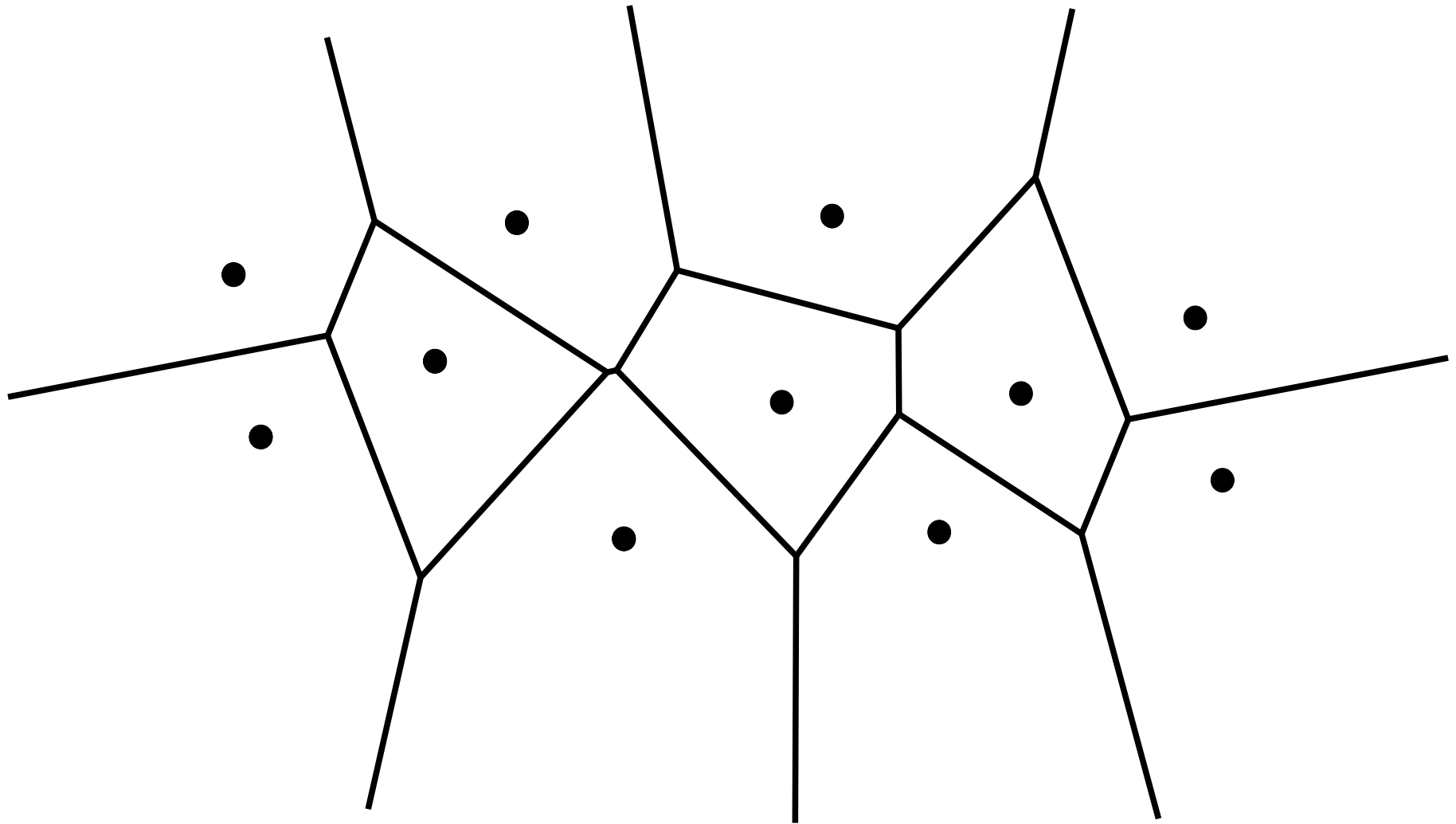
Aber dies ist eine Delaunay-Triangulation.



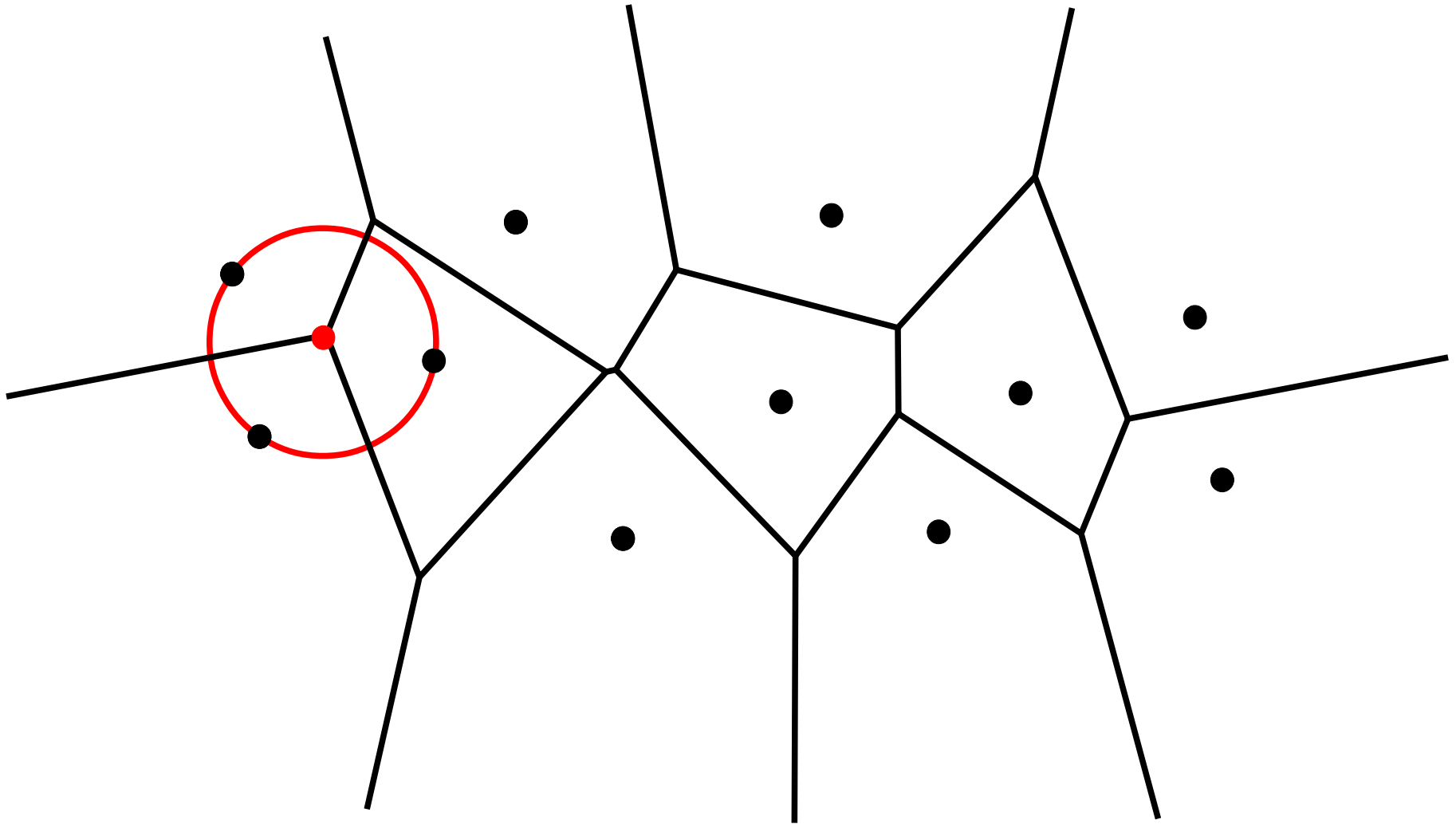
Betrachten wir allgemein eine Punktmenge.



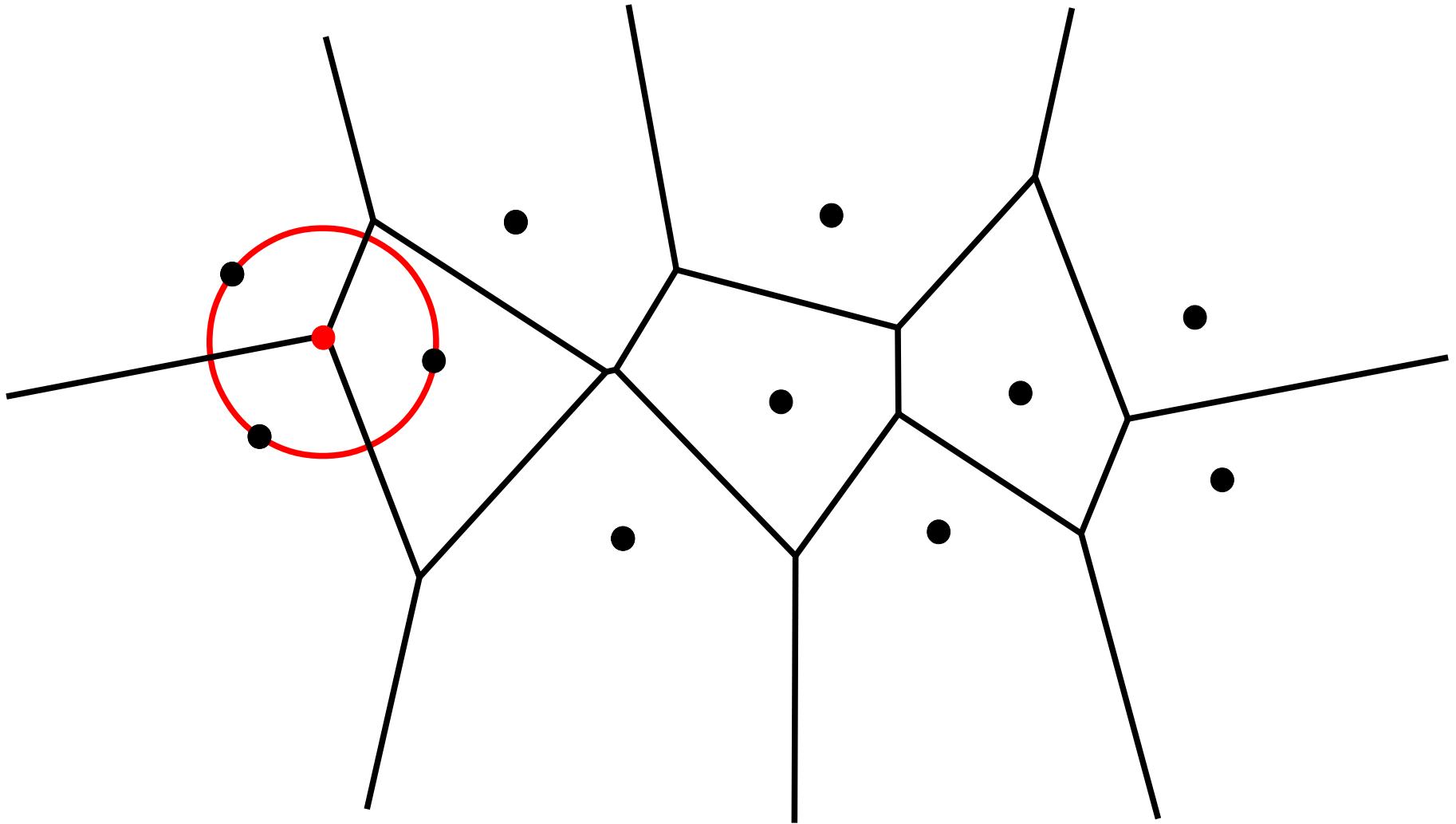
Wir berechnen das **Voronoidiagramm** zu dieser Punktmenge.



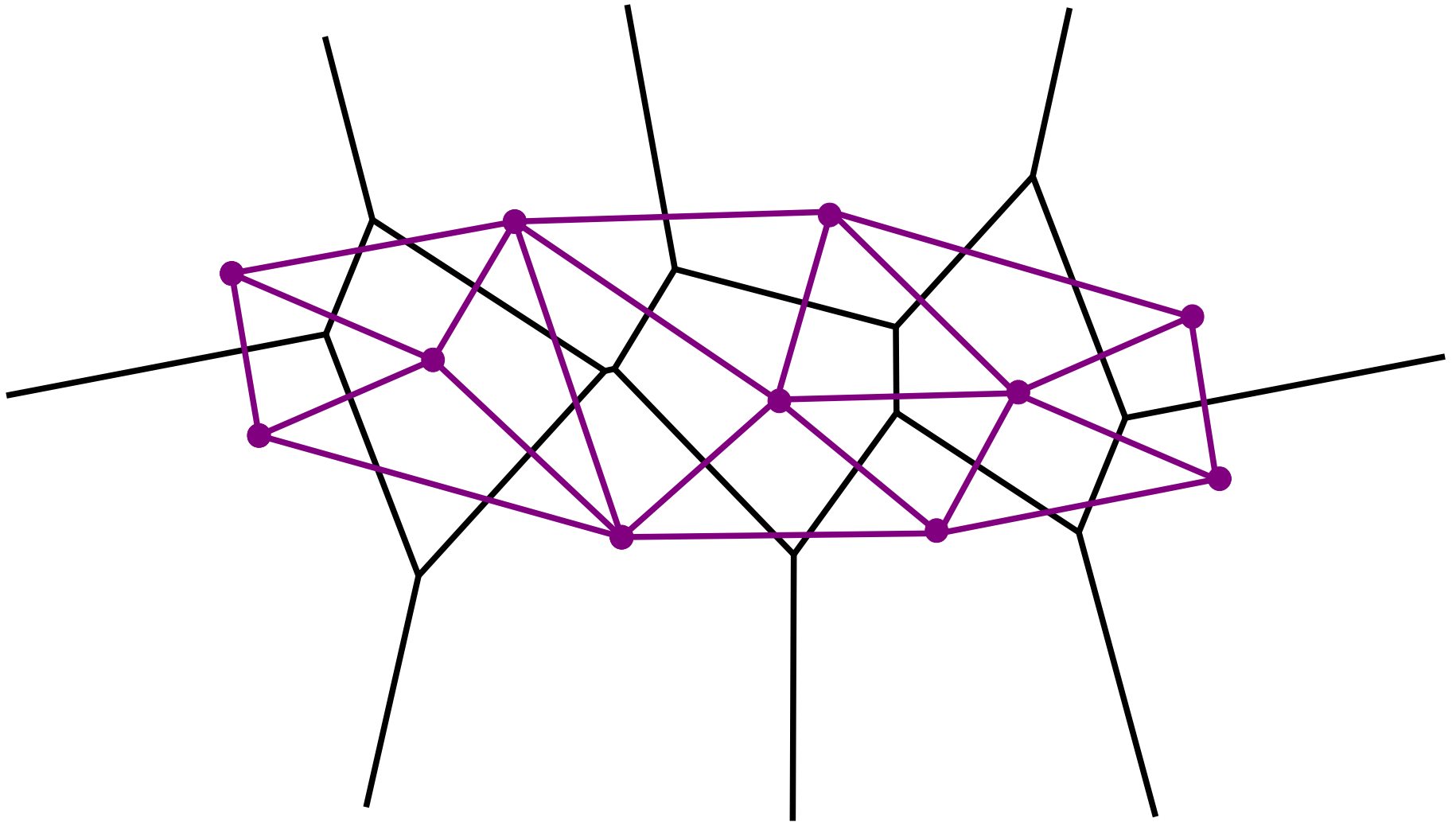
Betrachten wir nun die Kreise mit **Mittelpunkt in einer Voronoiecke** und Radius gleich dem Abstand zu den umgebenden drei Punkten.



Diese Kreise enthalten in ihrem Inneren **keine Punkte** der Eingabe.



Damit ist aber der **duale Graph** des Voronoidiagramms eine Delaunay-Triangulation.



5. Verallgemeinerungen des Voronoidiagramms

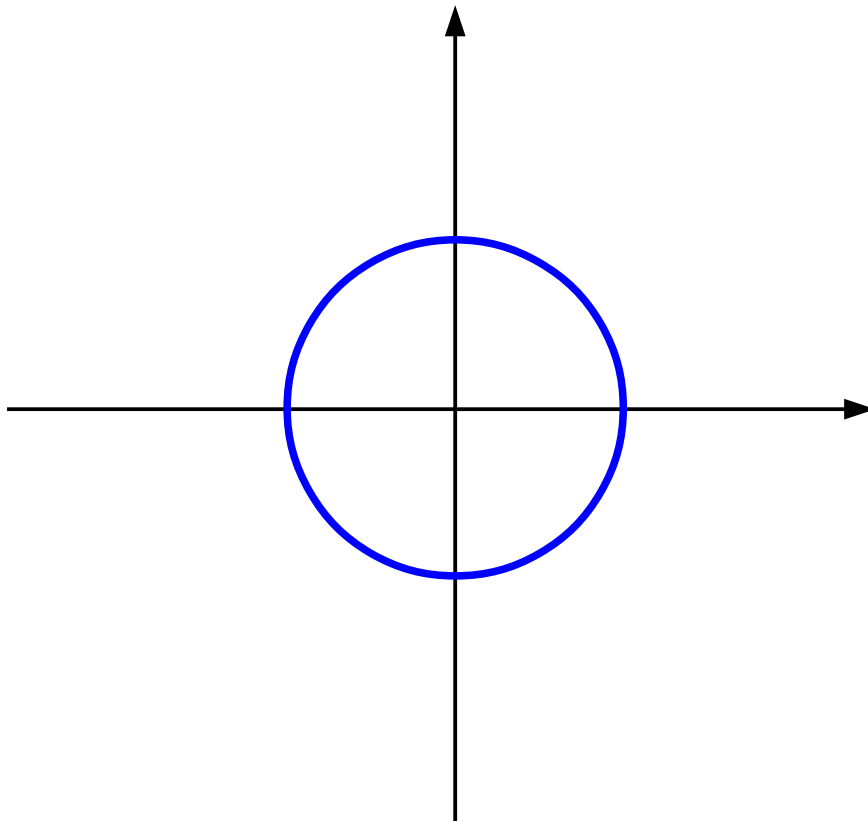
Man kann auch Voronoidiagramme für **andere geometrische Objekte** als Punkte betrachten.

Im Praktikum hatten wir zum Beispiel den kleinsten Abstand in einer **Menge von Strecken** gesucht

Diese Aufgabe kann man effizient mit einem Voronoidiagramm für die gegebene Streckenmenge lösen.

Im Folgenden schauen wir uns noch kurz an, welche **anderen Abstandsmaße** man verwenden kann.

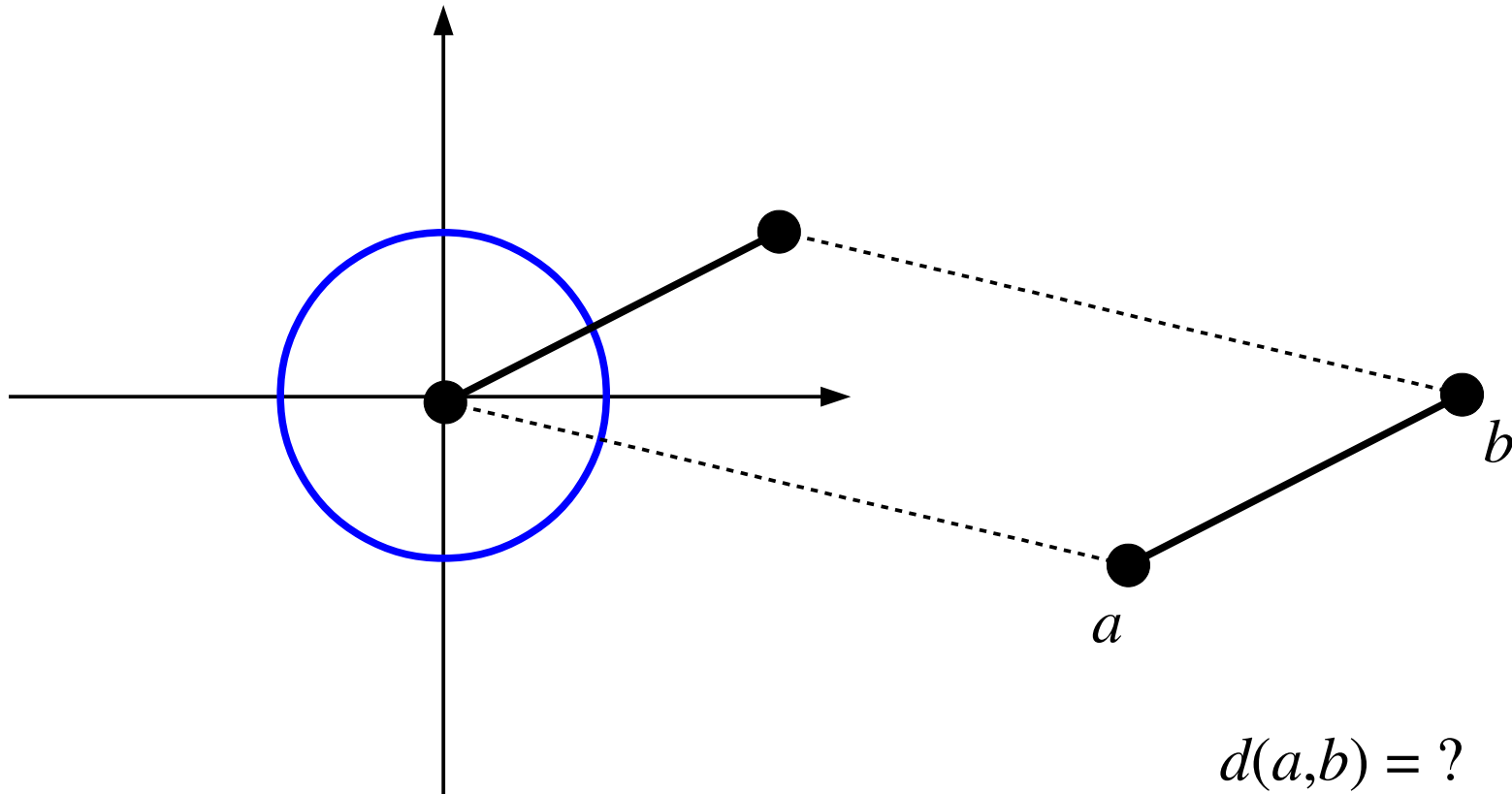
Einheitskreis des euklidischen Abstands



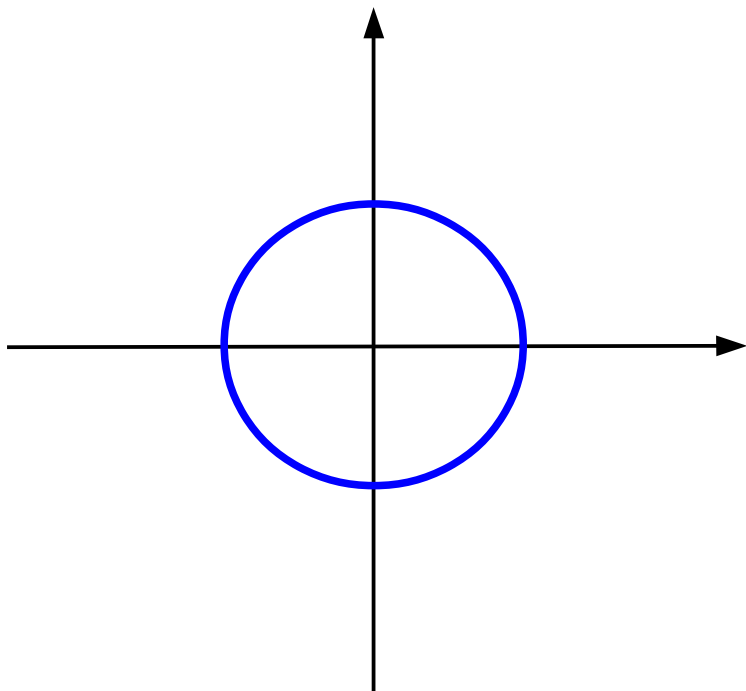
a

b

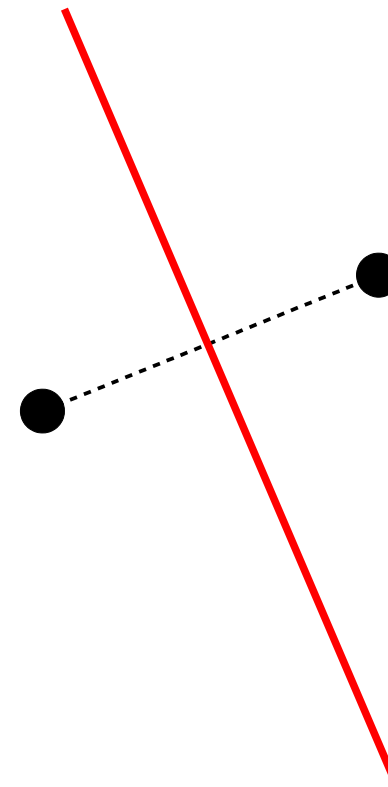
$$d(a,b) = ?$$

Messung des Abstands zwischen a und b 

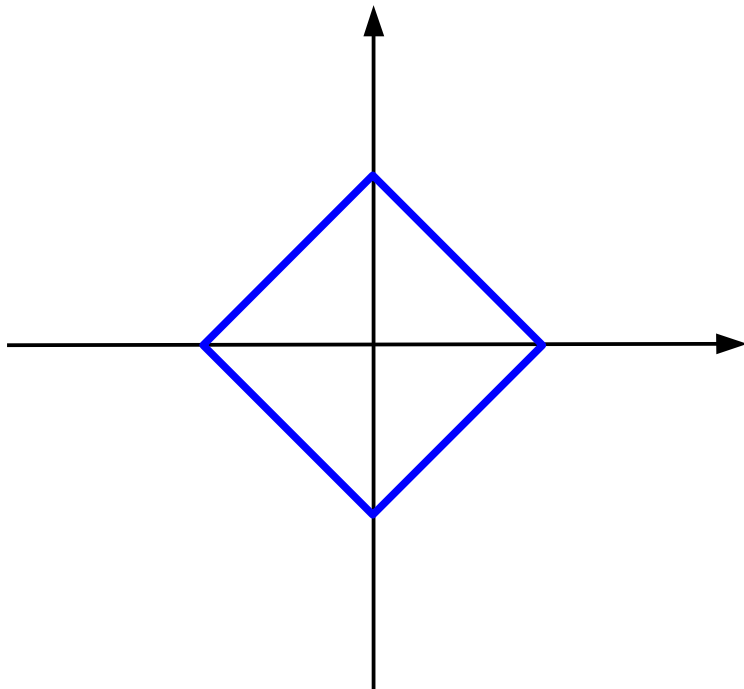
Einheitskreis



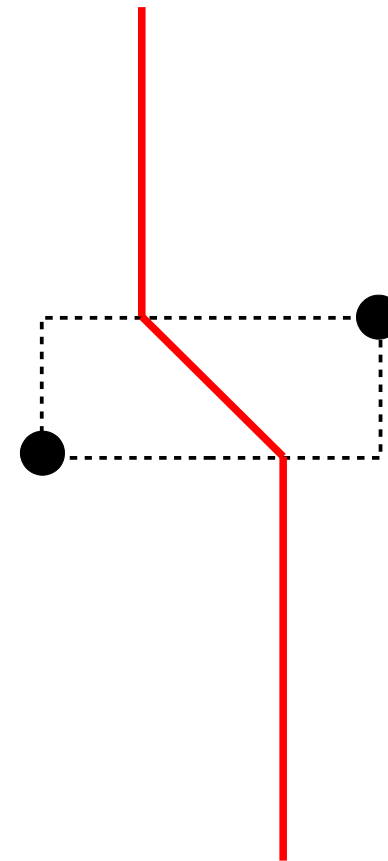
Bisektor



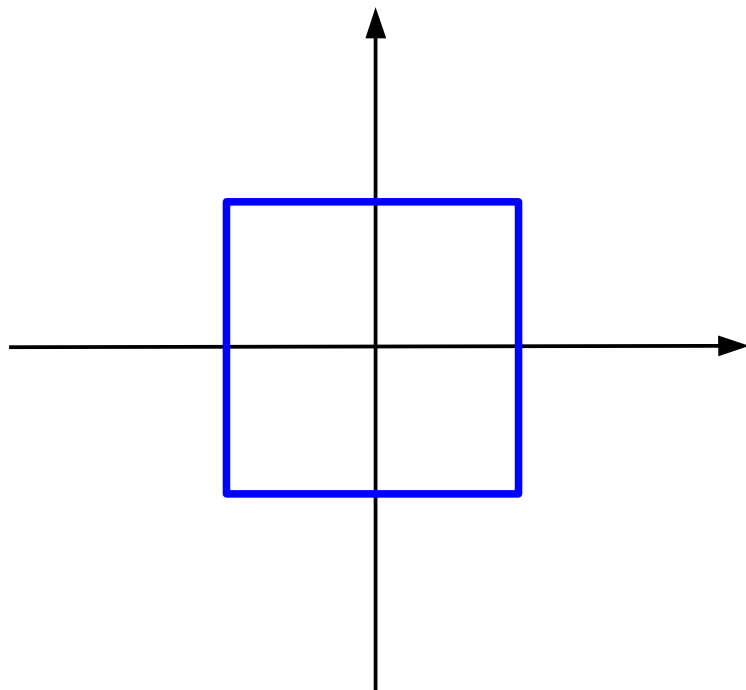
Einheitskreis



Bisektor



Einheitskreis



Bisektor

