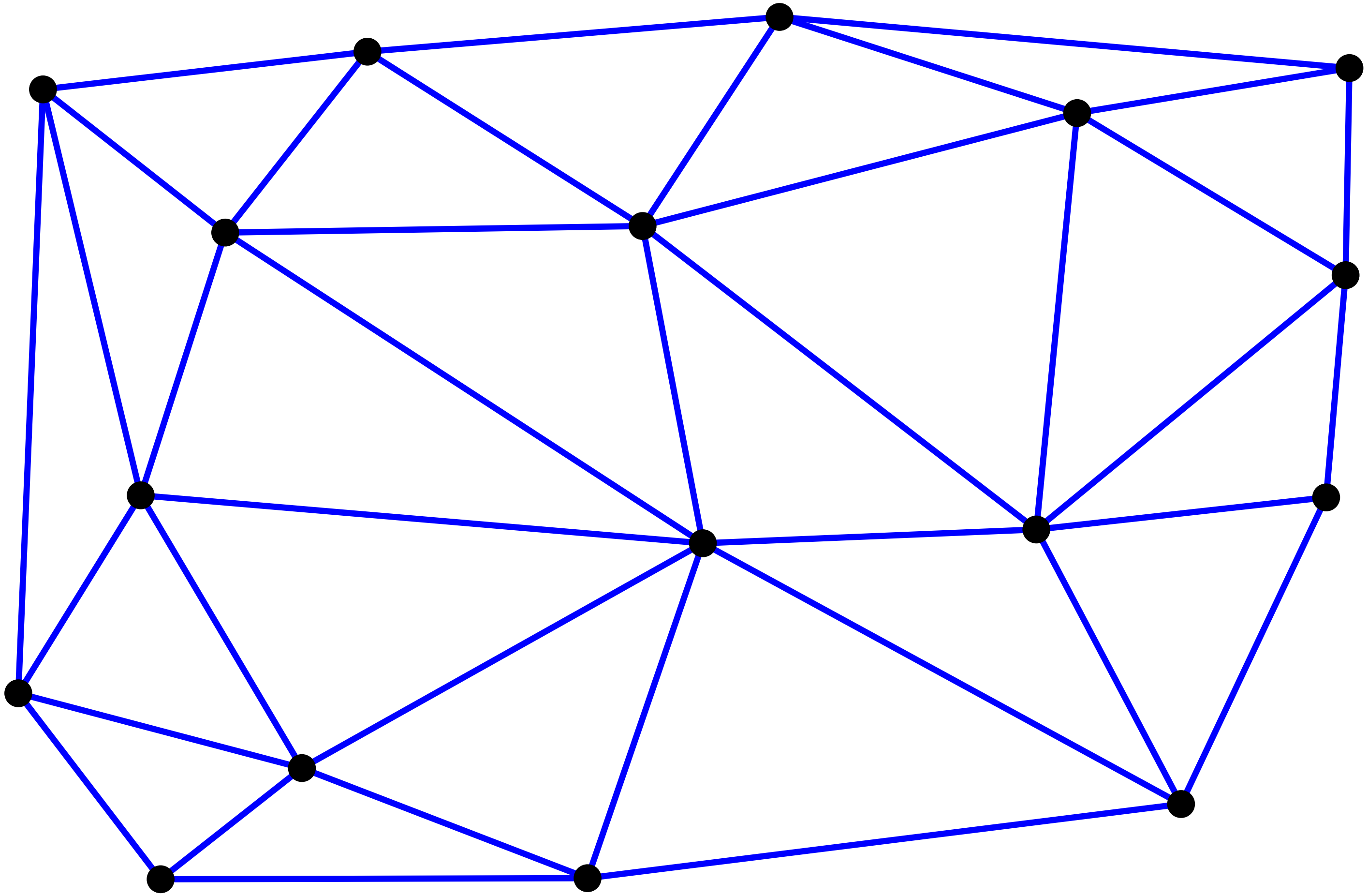
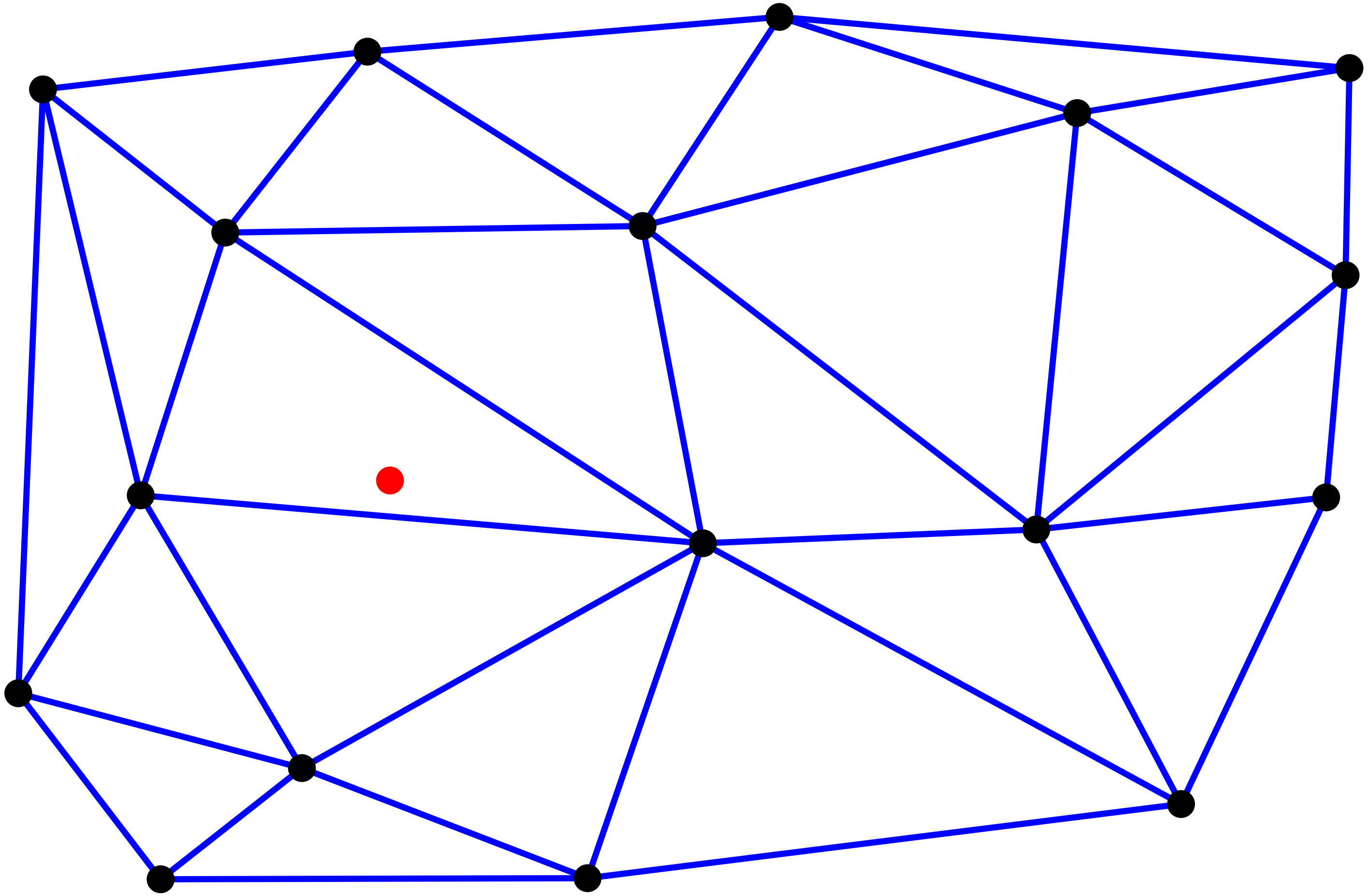
Delaunay Triangulation

Delaunay Triangulation: incremental algorithm



Delaunay Triangulation: incremental algorithm

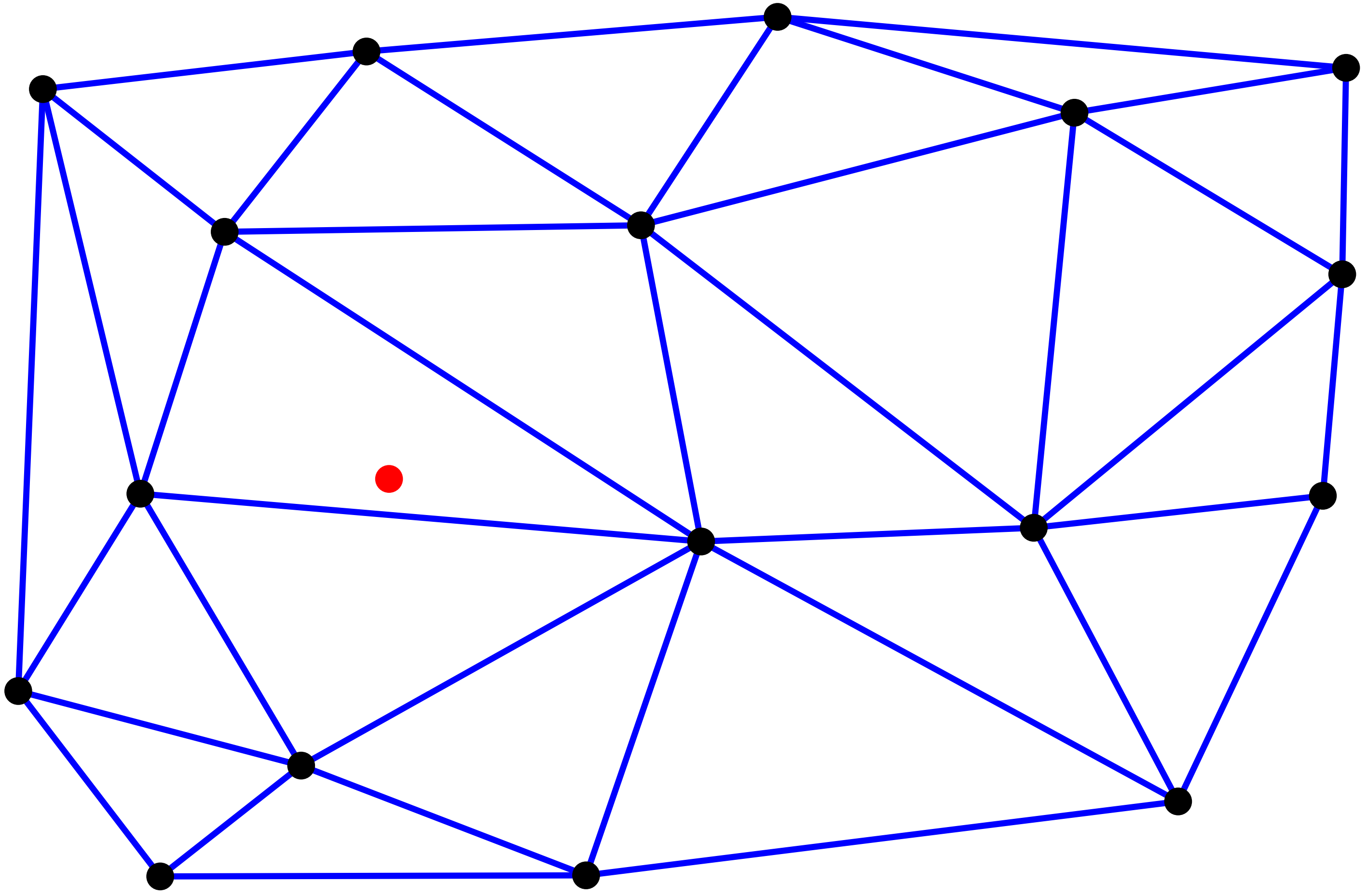
New point



Delaunay Triangulation: incremental algorithm

New point

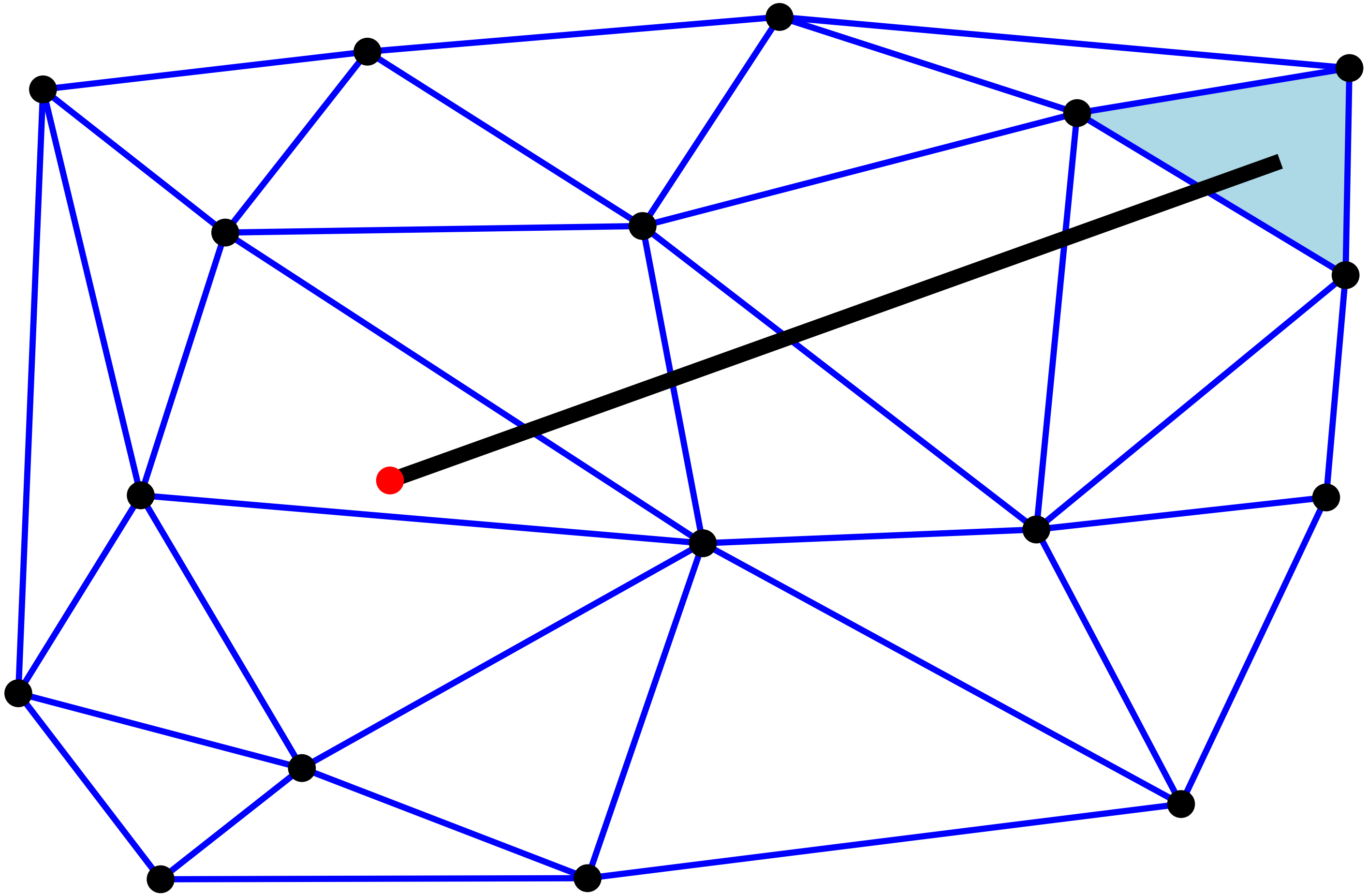
Locate



Delaunay Triangulation: incremental algorithm

New point

Locate

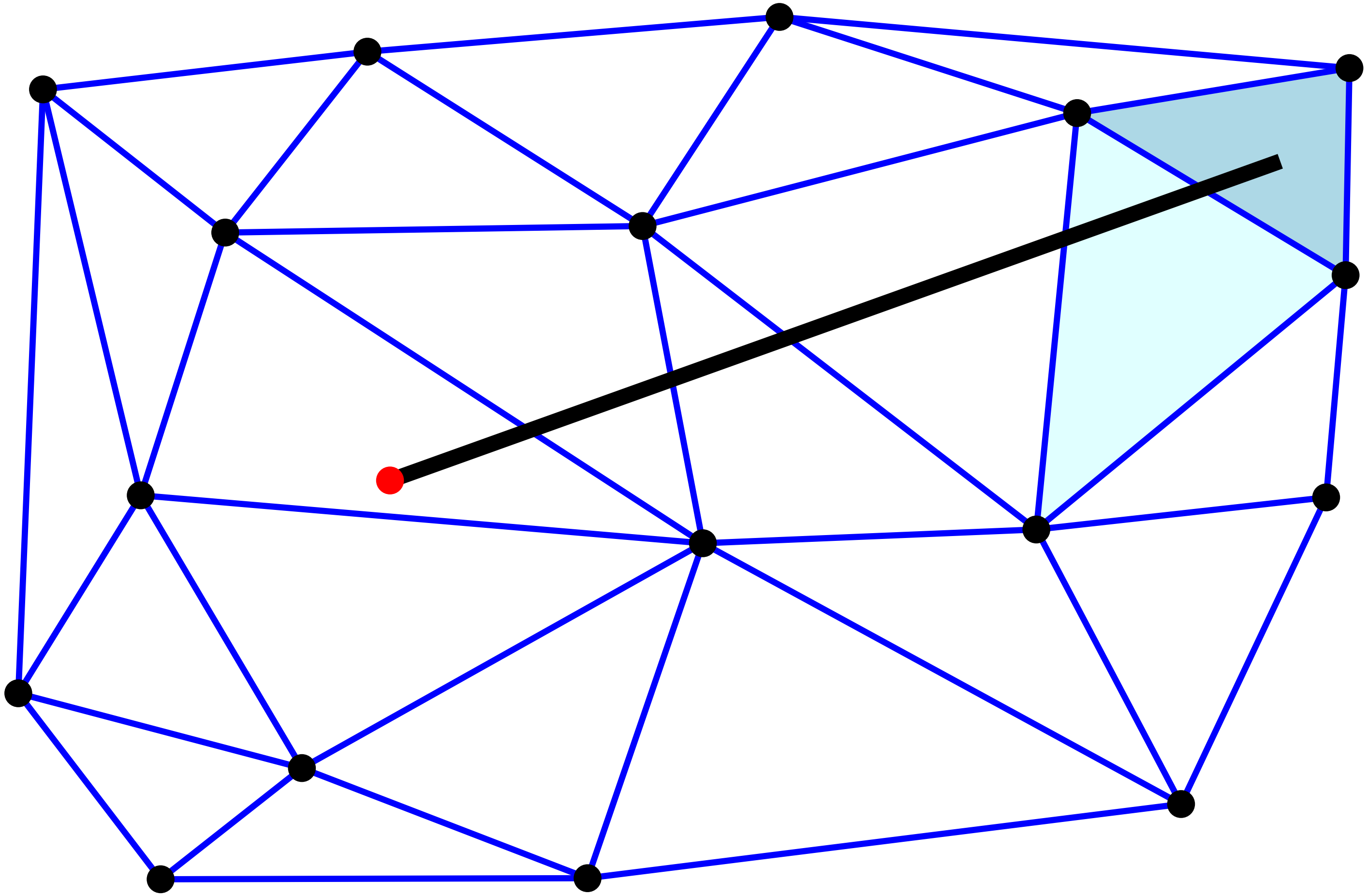


e.g.: straight walk

Delaunay Triangulation: incremental algorithm

New point

Locate

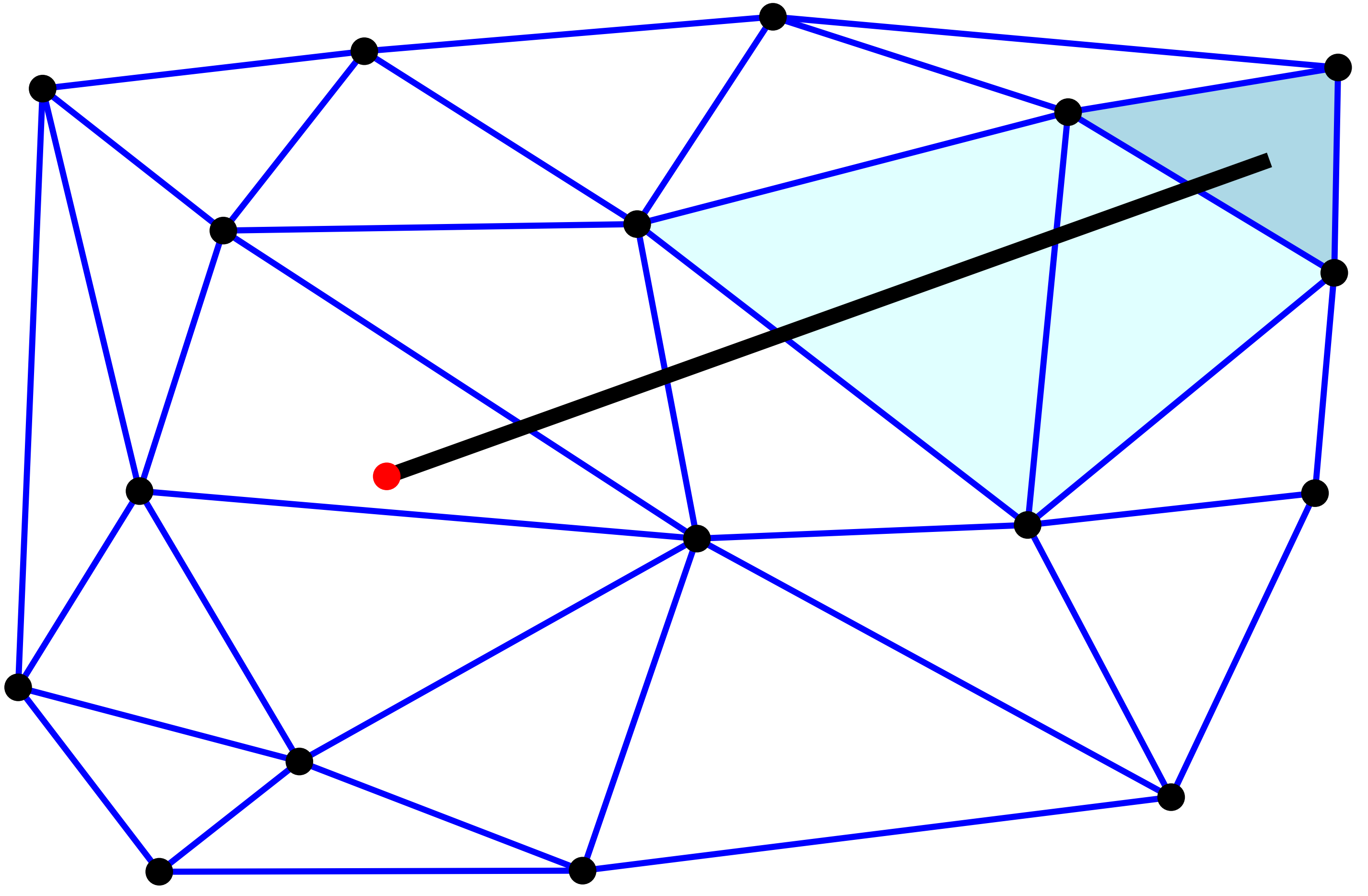


e.g.: straight walk

Delaunay Triangulation: incremental algorithm

New point

Locate

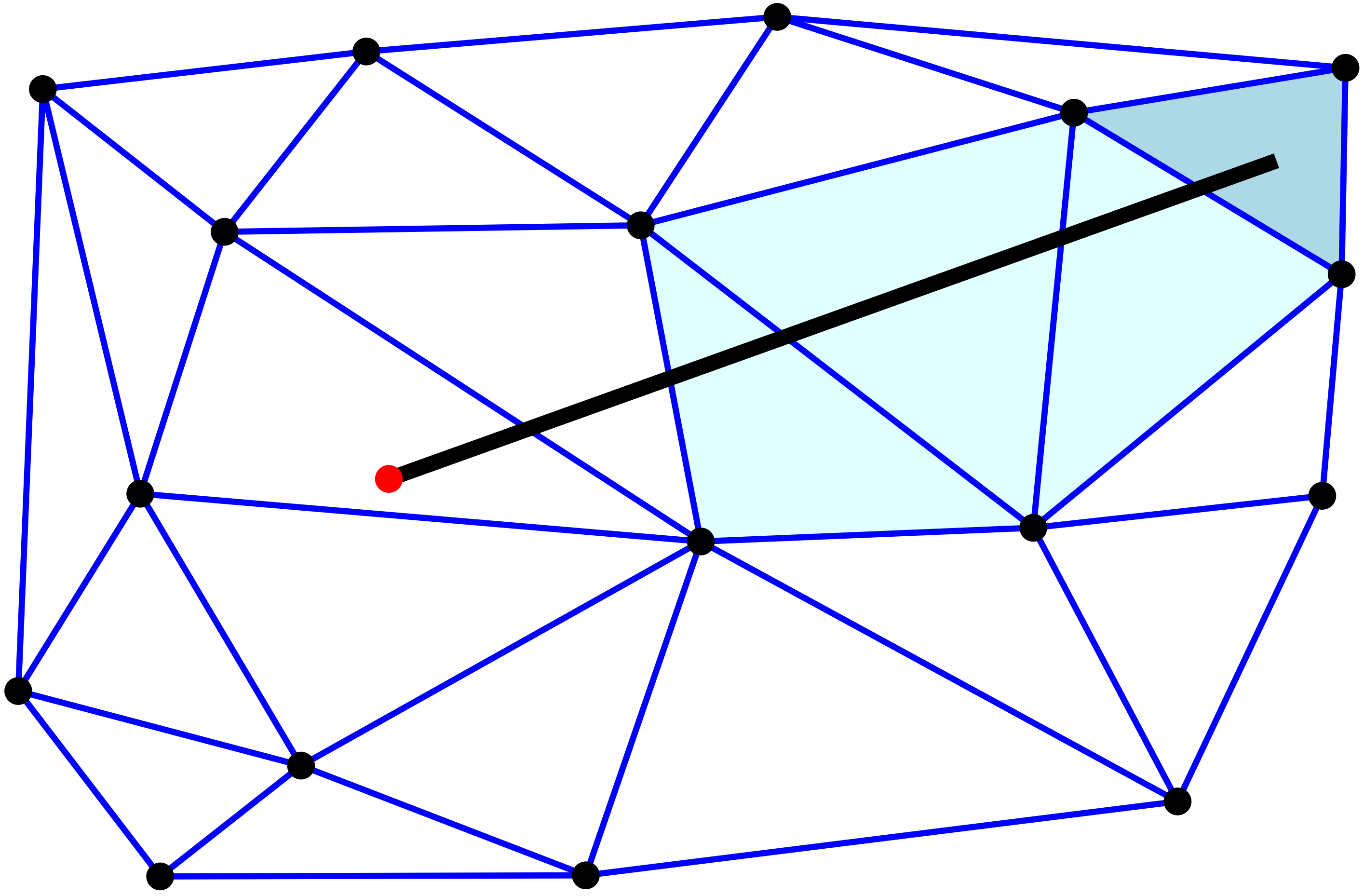


e.g.: straight walk

Delaunay Triangulation: incremental algorithm

New point

Locate

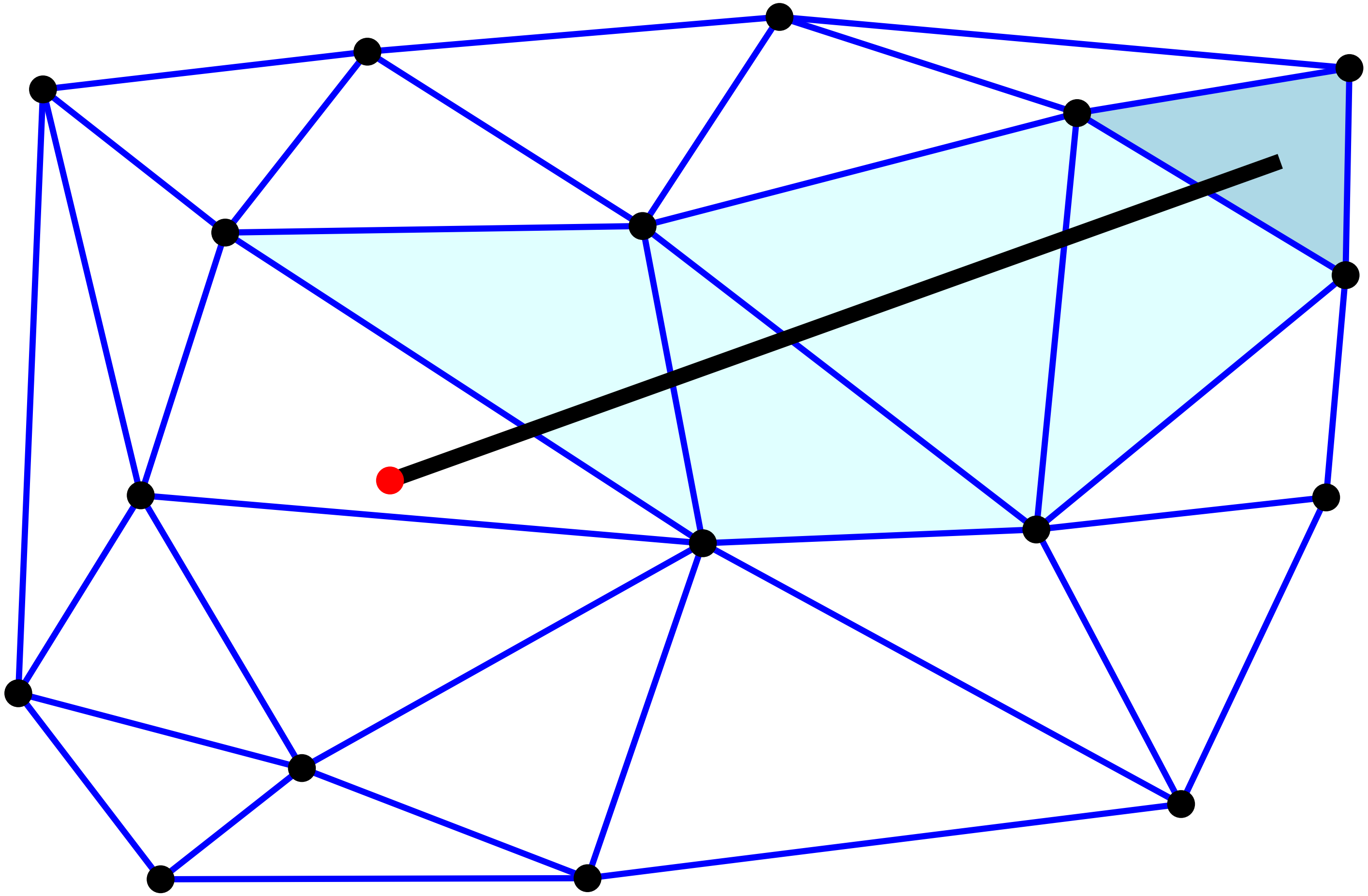


e.g.: straight walk

Delaunay Triangulation: incremental algorithm

New point

Locate

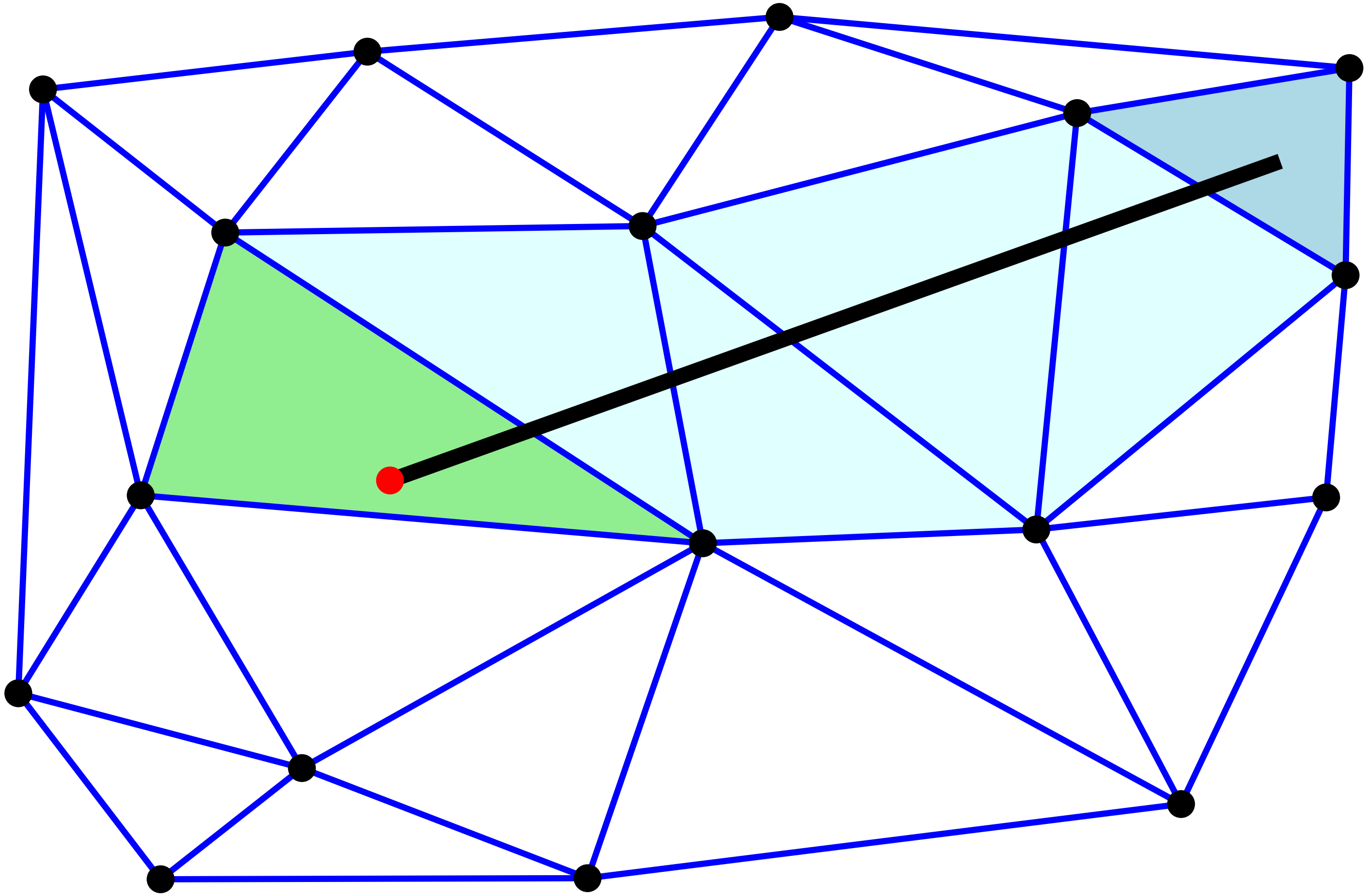


e.g.: straight walk

Delaunay Triangulation: incremental algorithm

New point

Locate

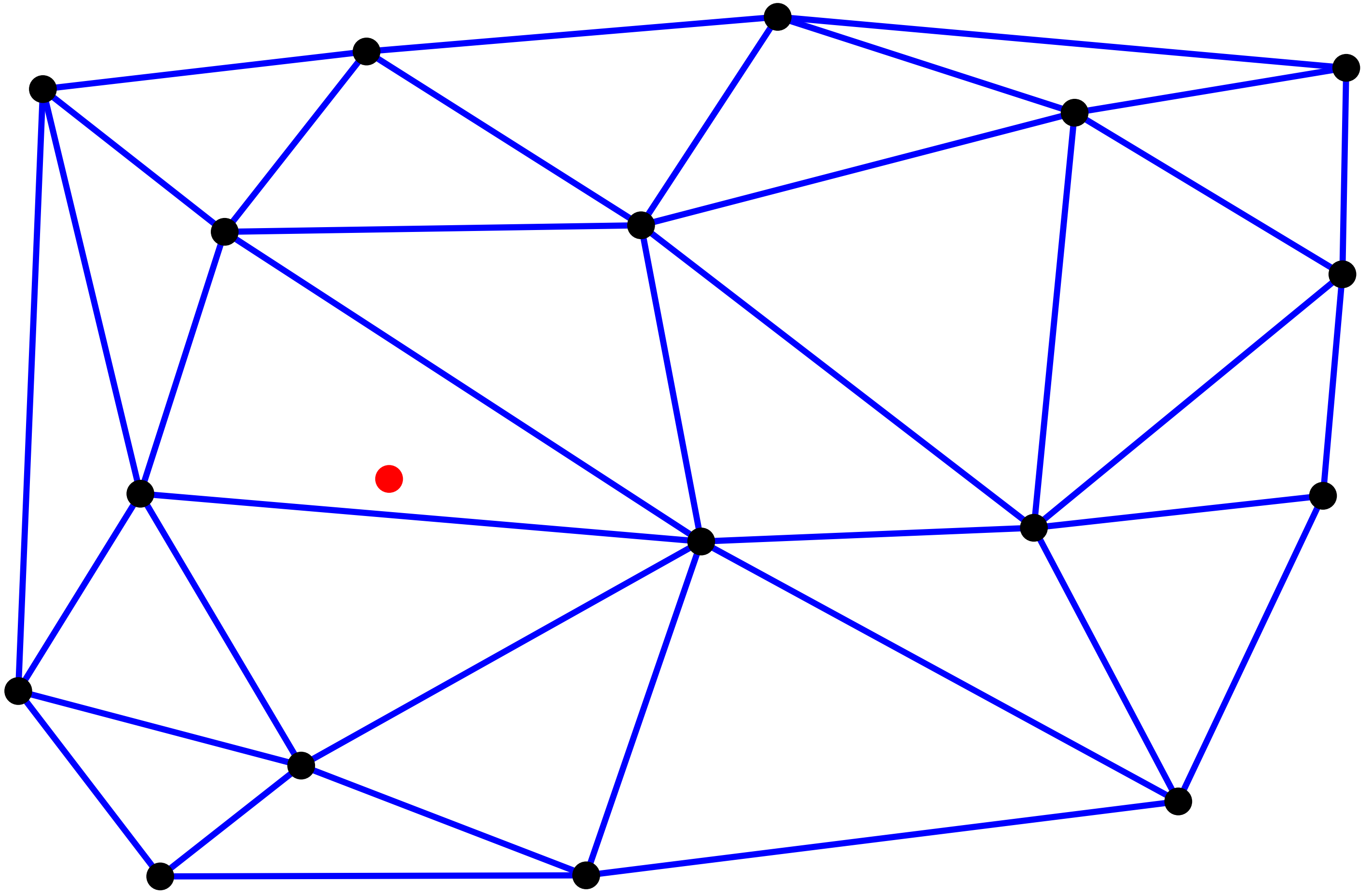


e.g.: straight walk

Delaunay Triangulation: incremental algorithm

New point

Locate

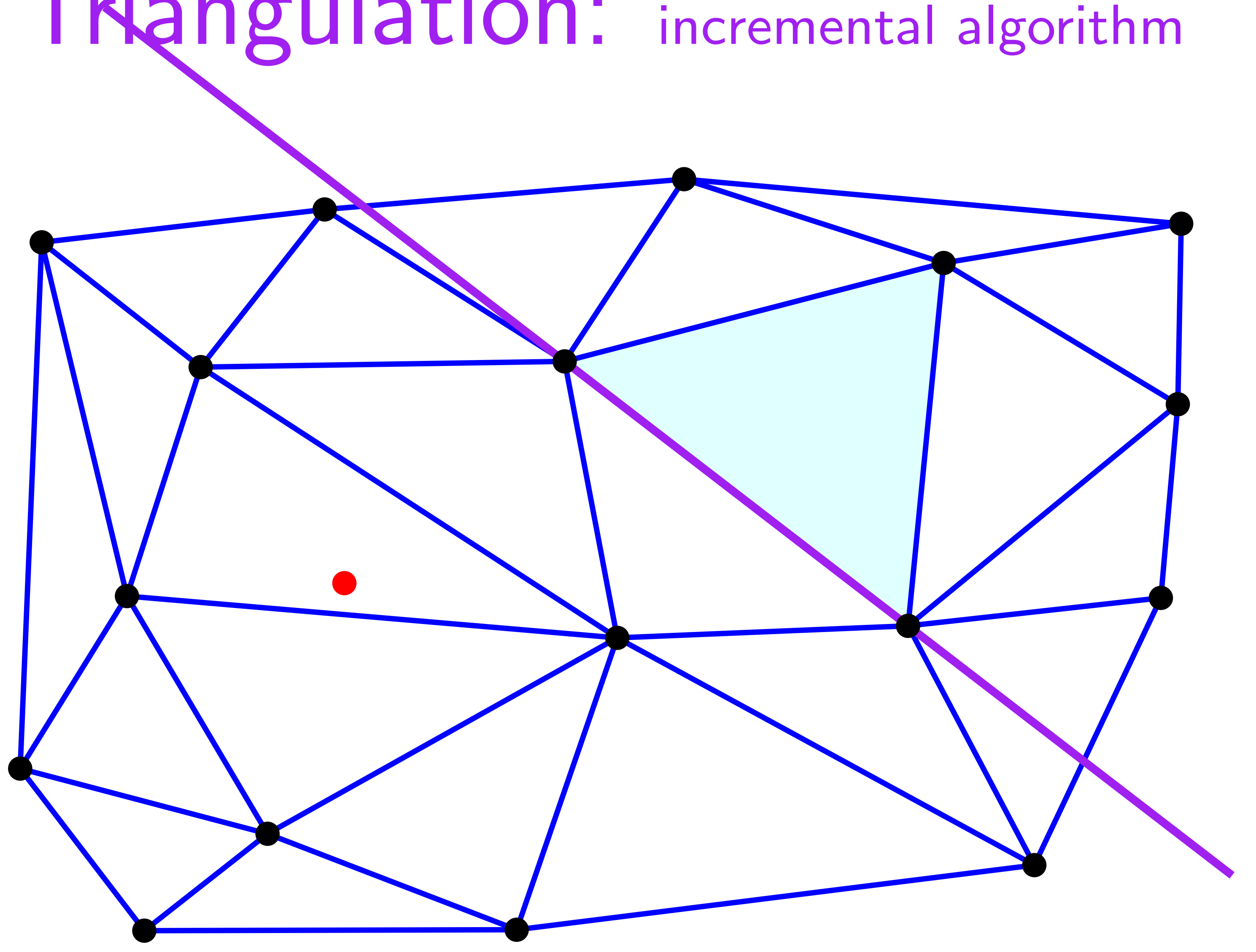


e.g.: visibility walk

Delaunay Triangulation: incremental algorithm

New point

Locate

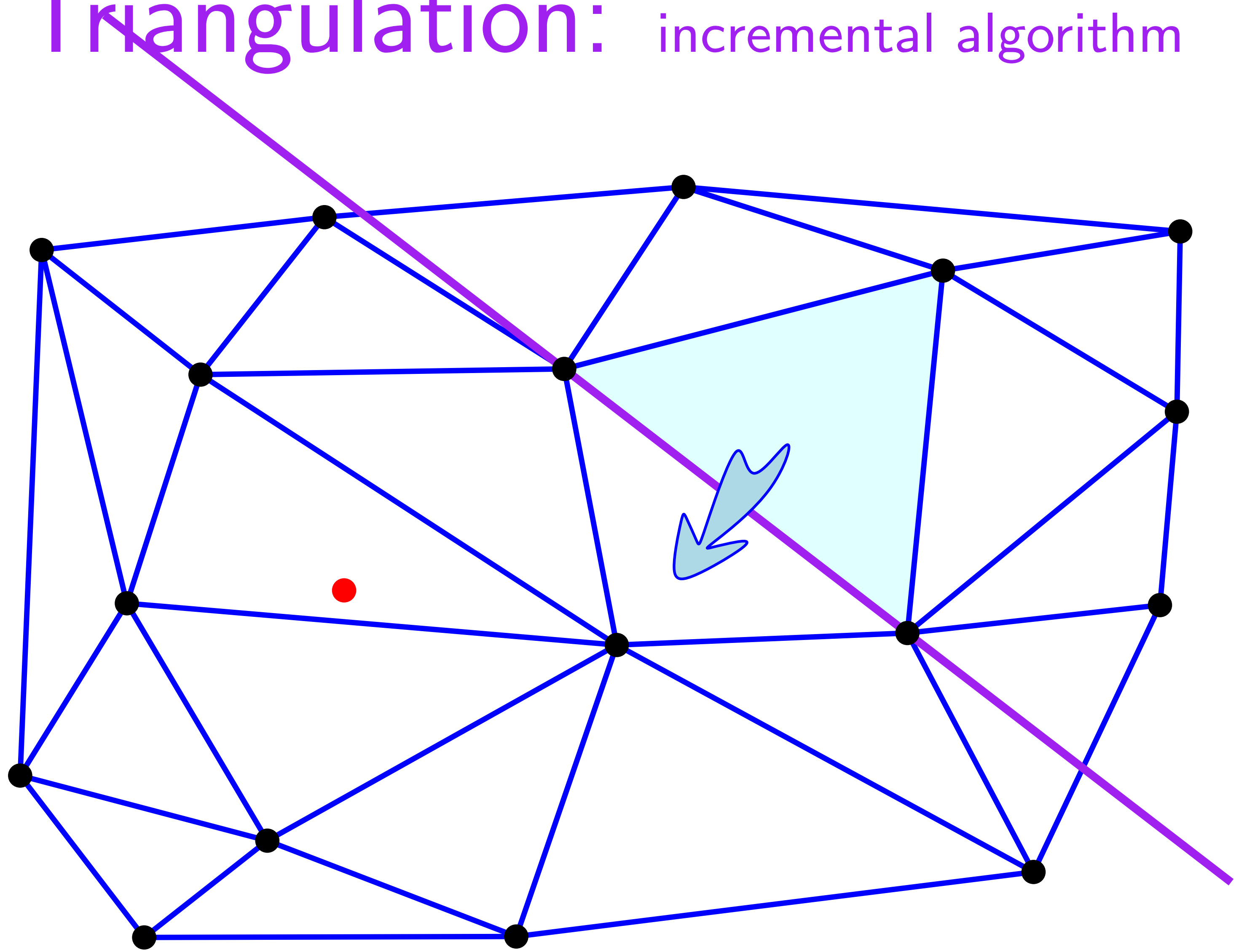


e.g.: visibility walk

Delaunay Triangulation: incremental algorithm

New point

Locate

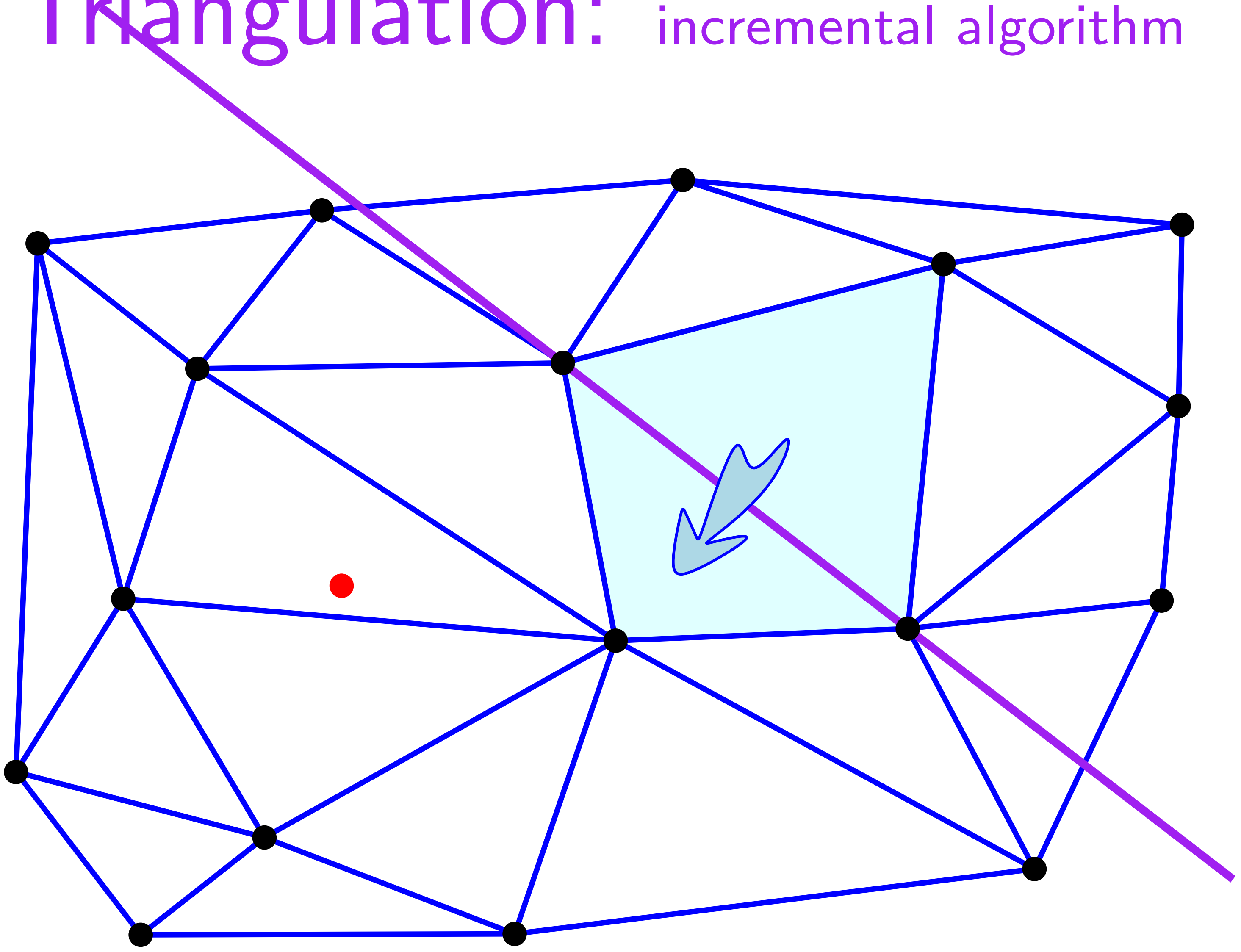


e.g.: visibility walk

Delaunay Triangulation: incremental algorithm

New point

Locate

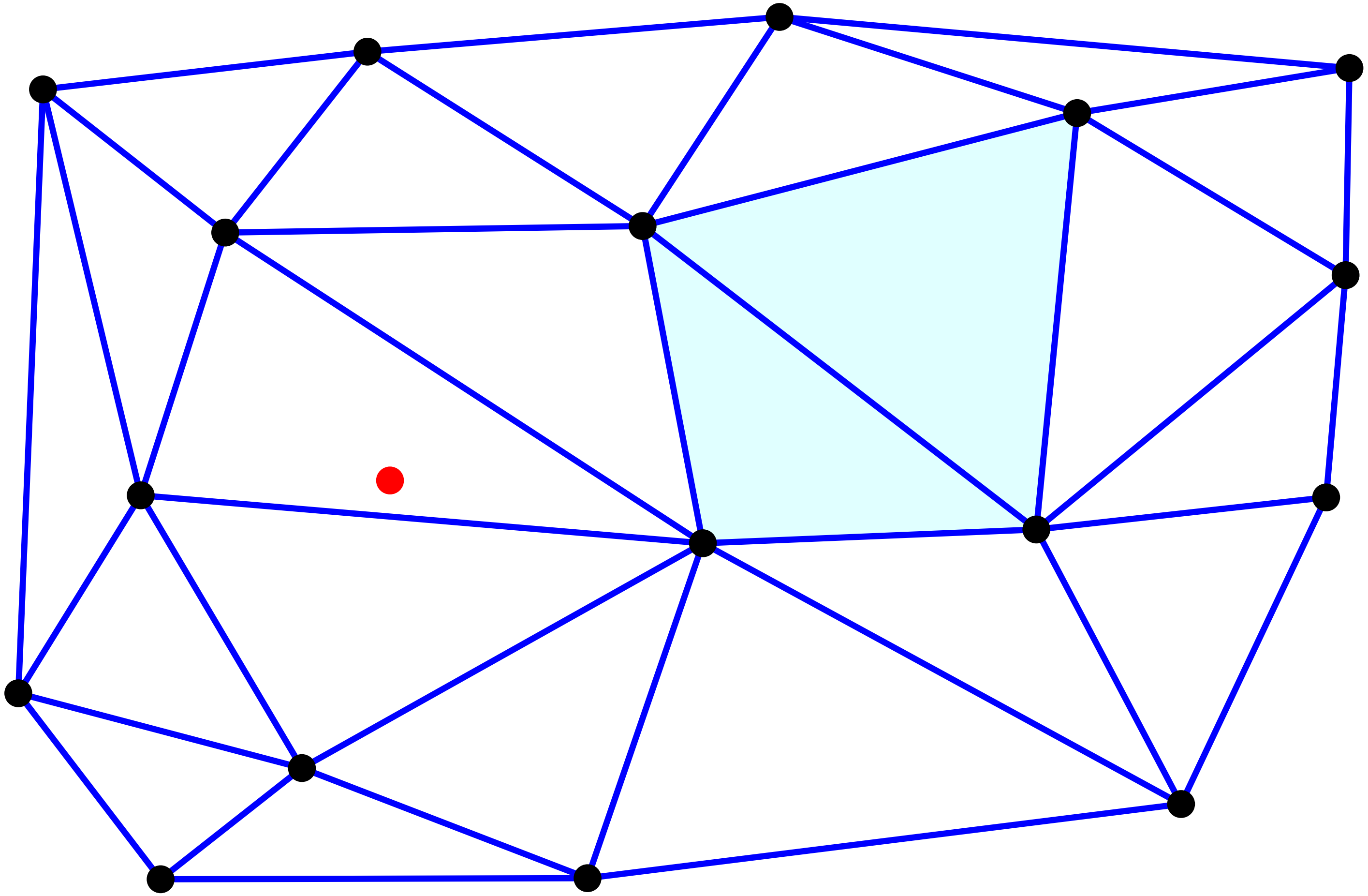


e.g.: visibility walk

Delaunay Triangulation: incremental algorithm

New point

Locate

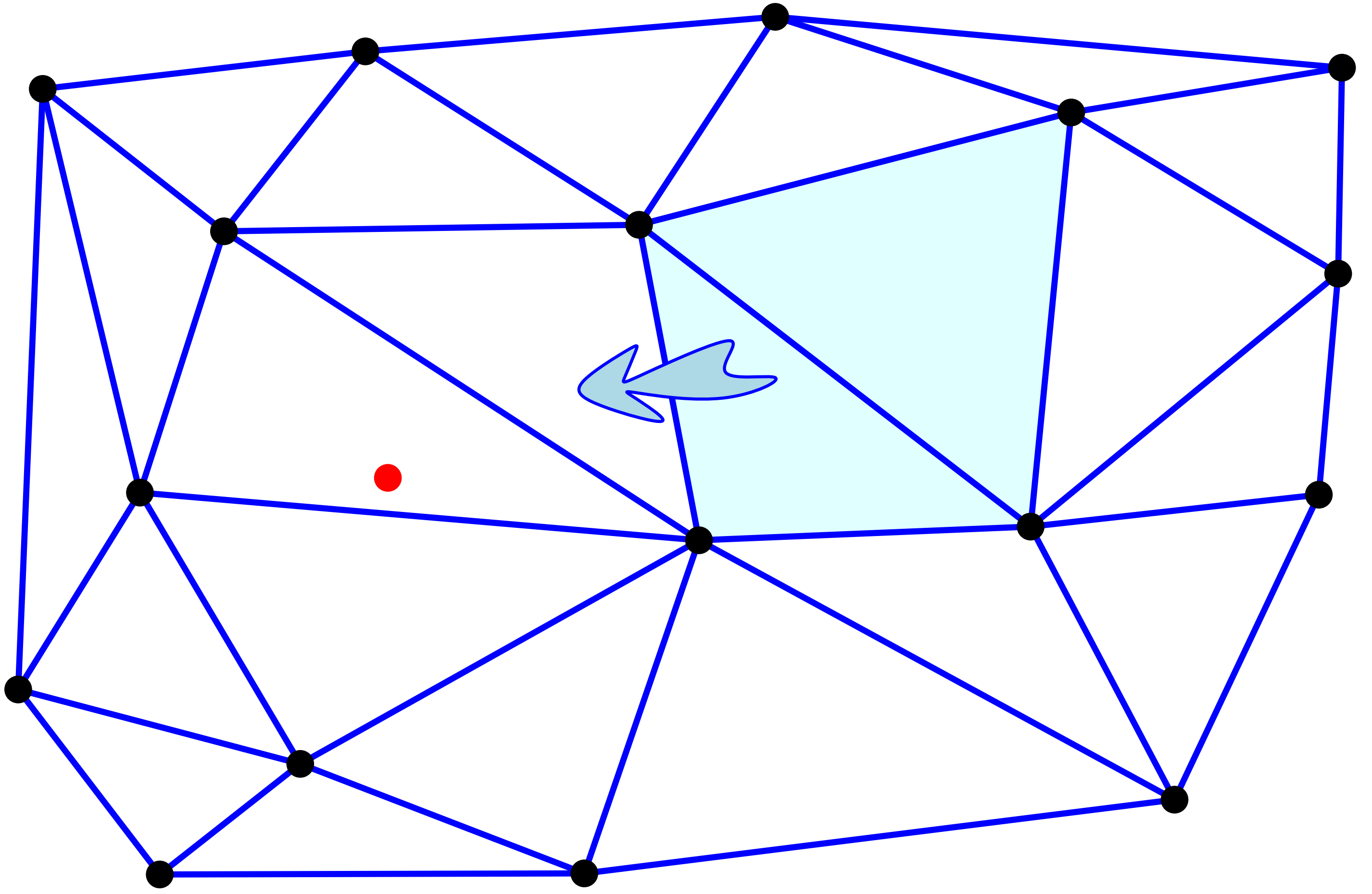


e.g.: visibility walk

Delaunay Triangulation: incremental algorithm

New point

Locate

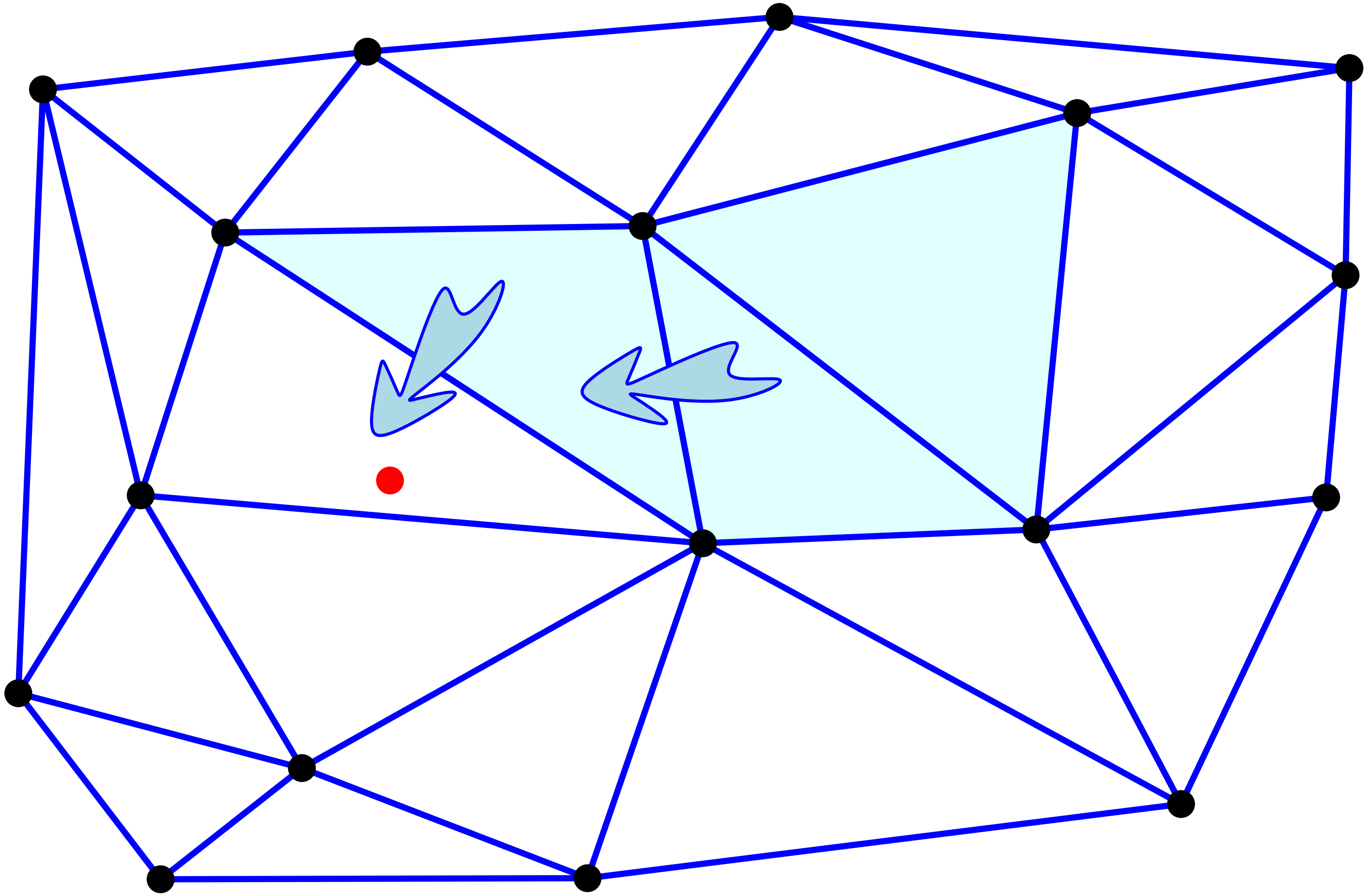


e.g.: visibility walk

Delaunay Triangulation: incremental algorithm

New point

Locate

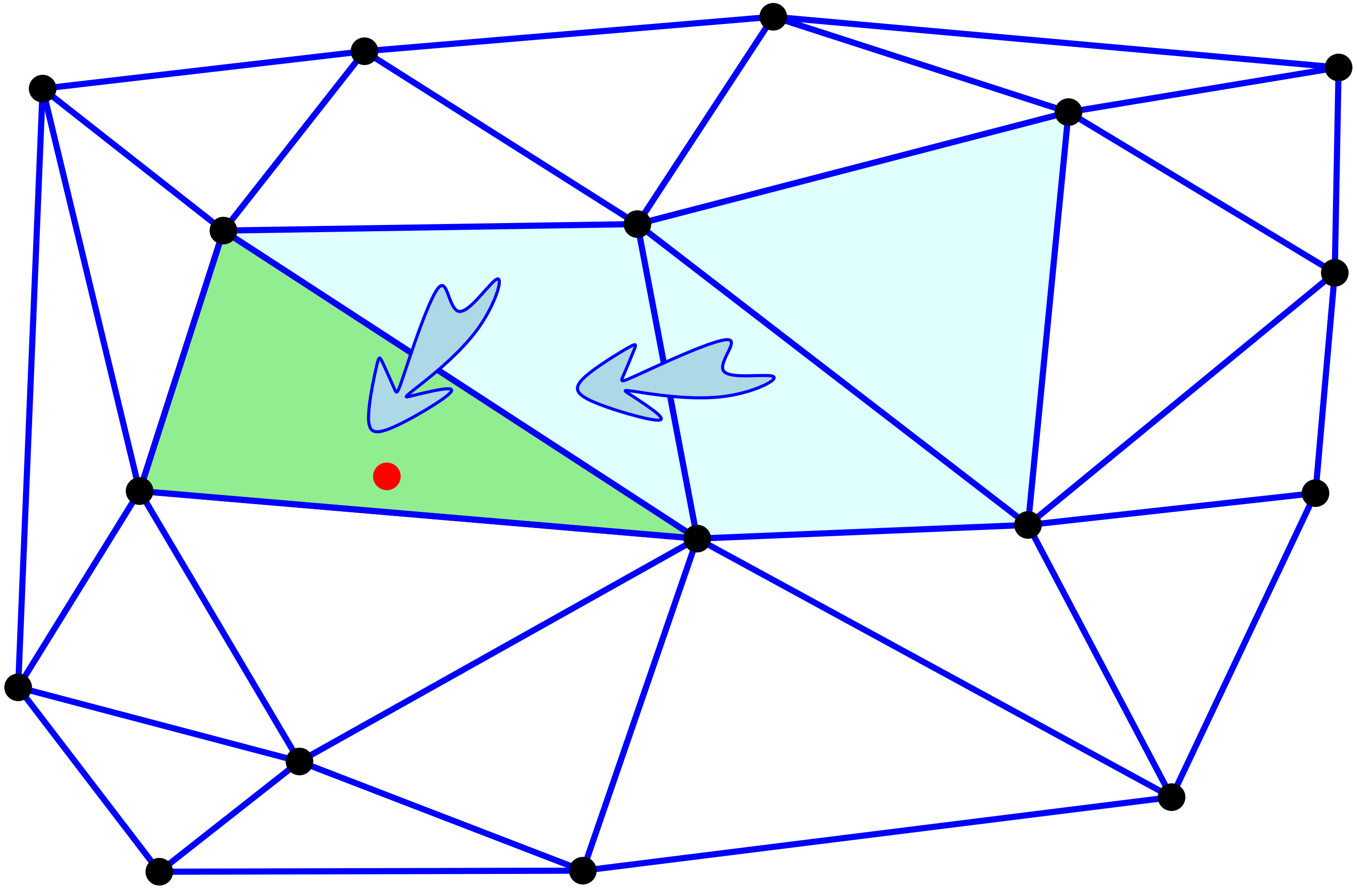


e.g.: visibility walk

Delaunay Triangulation: incremental algorithm

New point

Locate

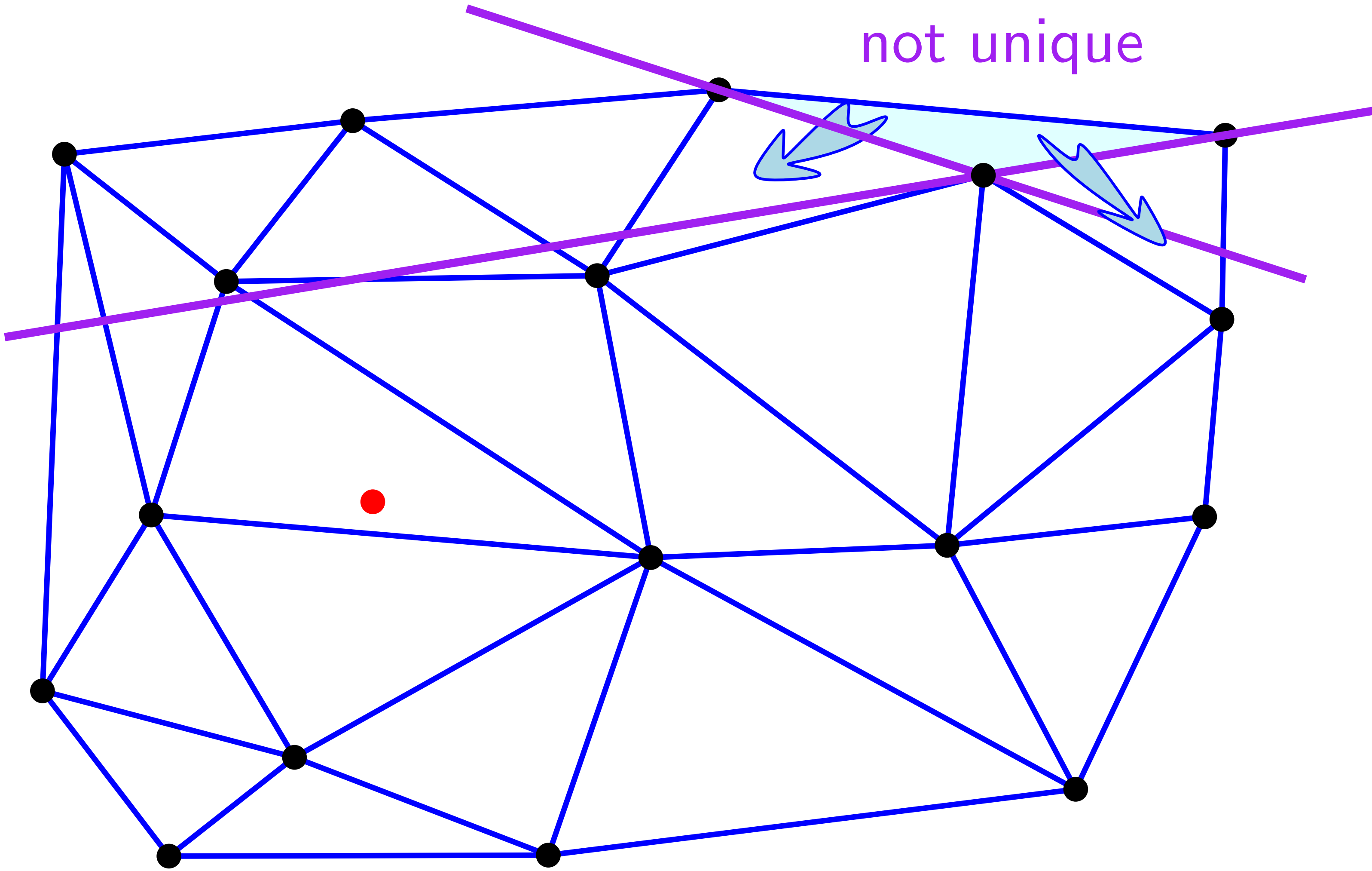


e.g.: visibility walk

Delaunay Triangulation: incremental algorithm

New point

Locate

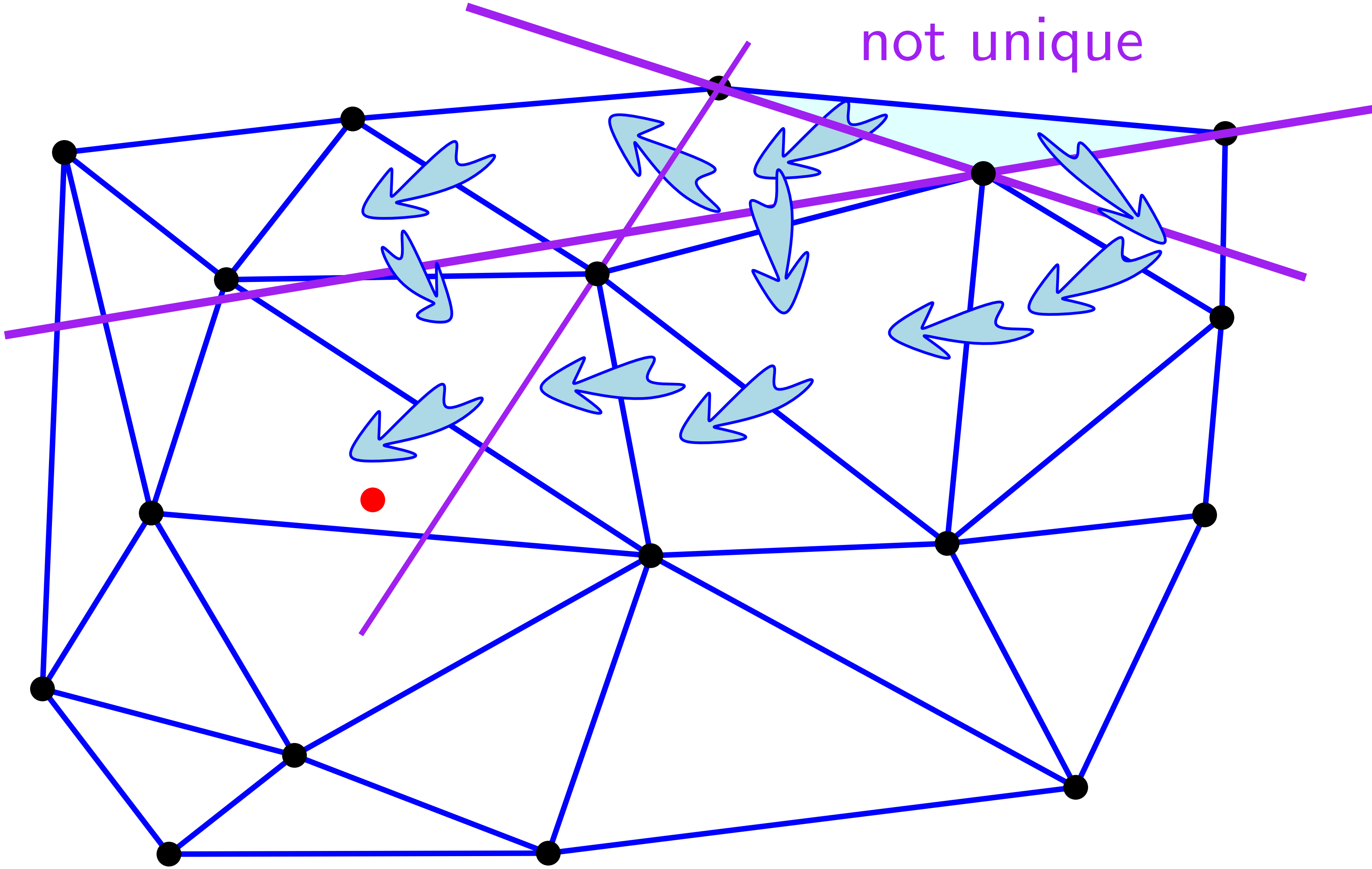


e.g.: visibility walk

Delaunay Triangulation: incremental algorithm

New point

Locate



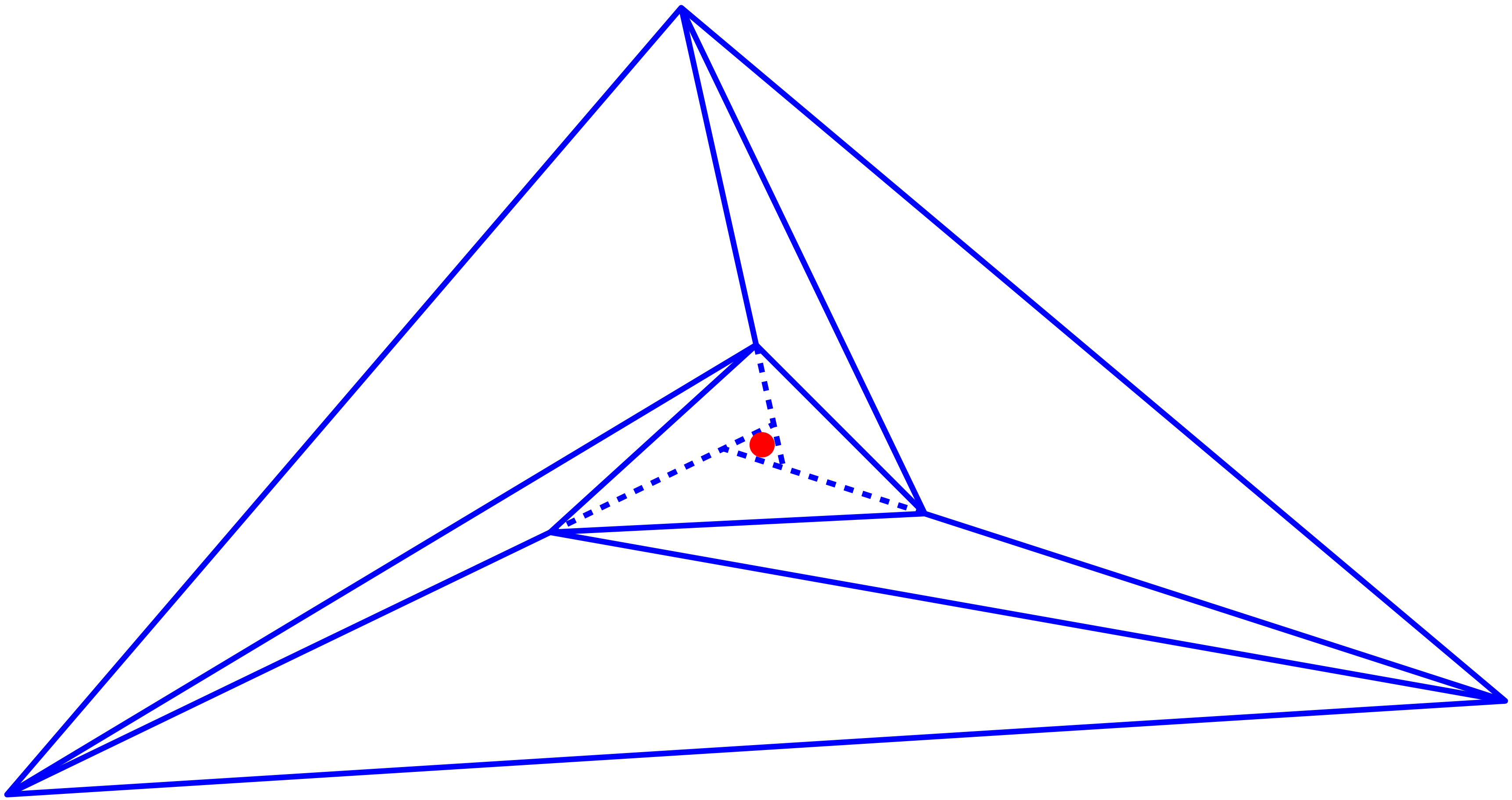
e.g.: visibility walk

Delaunay Triangulation: incremental algorithm

Visibility walk terminates ?

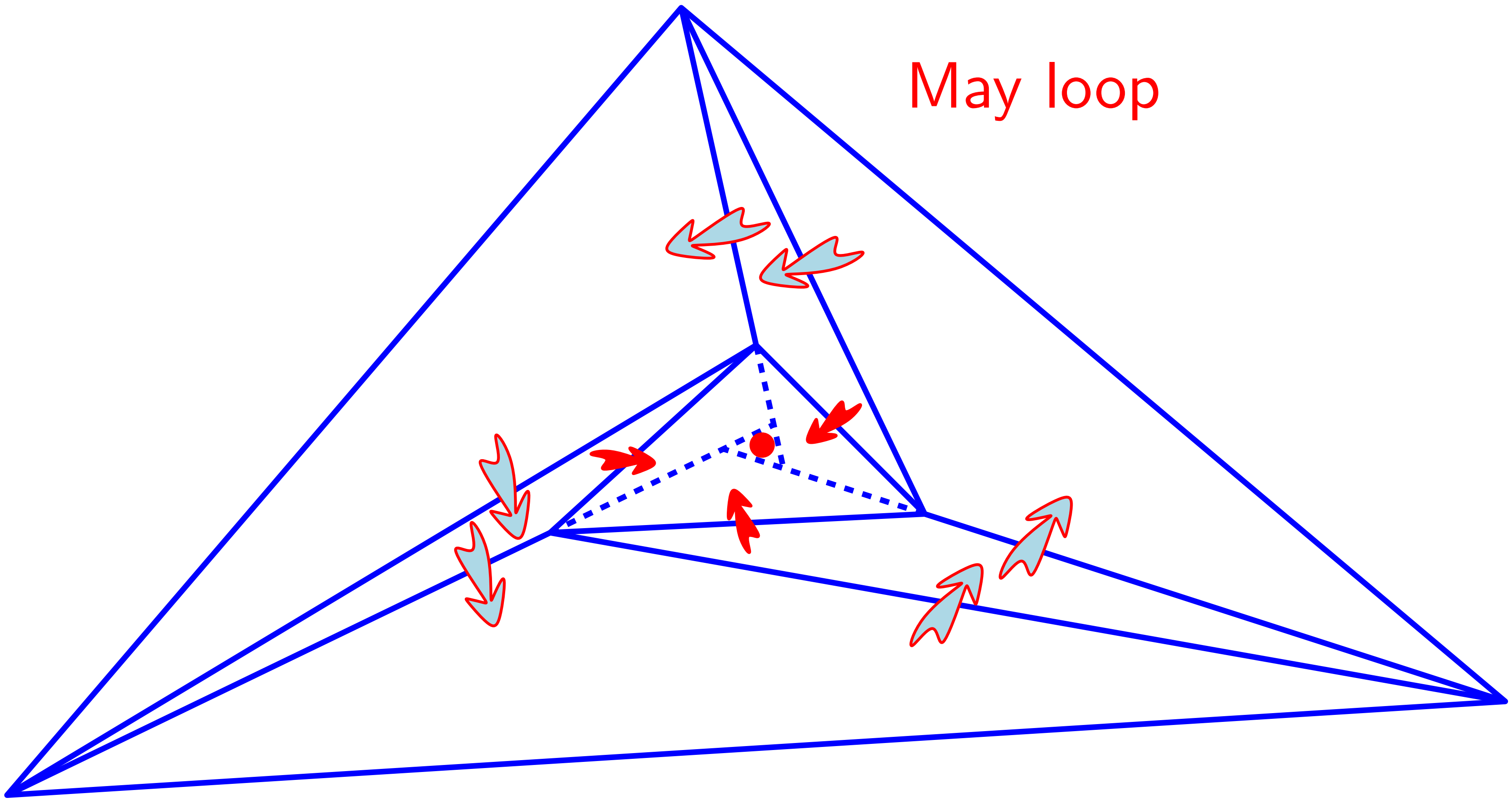
Delaunay Triangulation: incremental algorithm

Visibility walk terminates ?



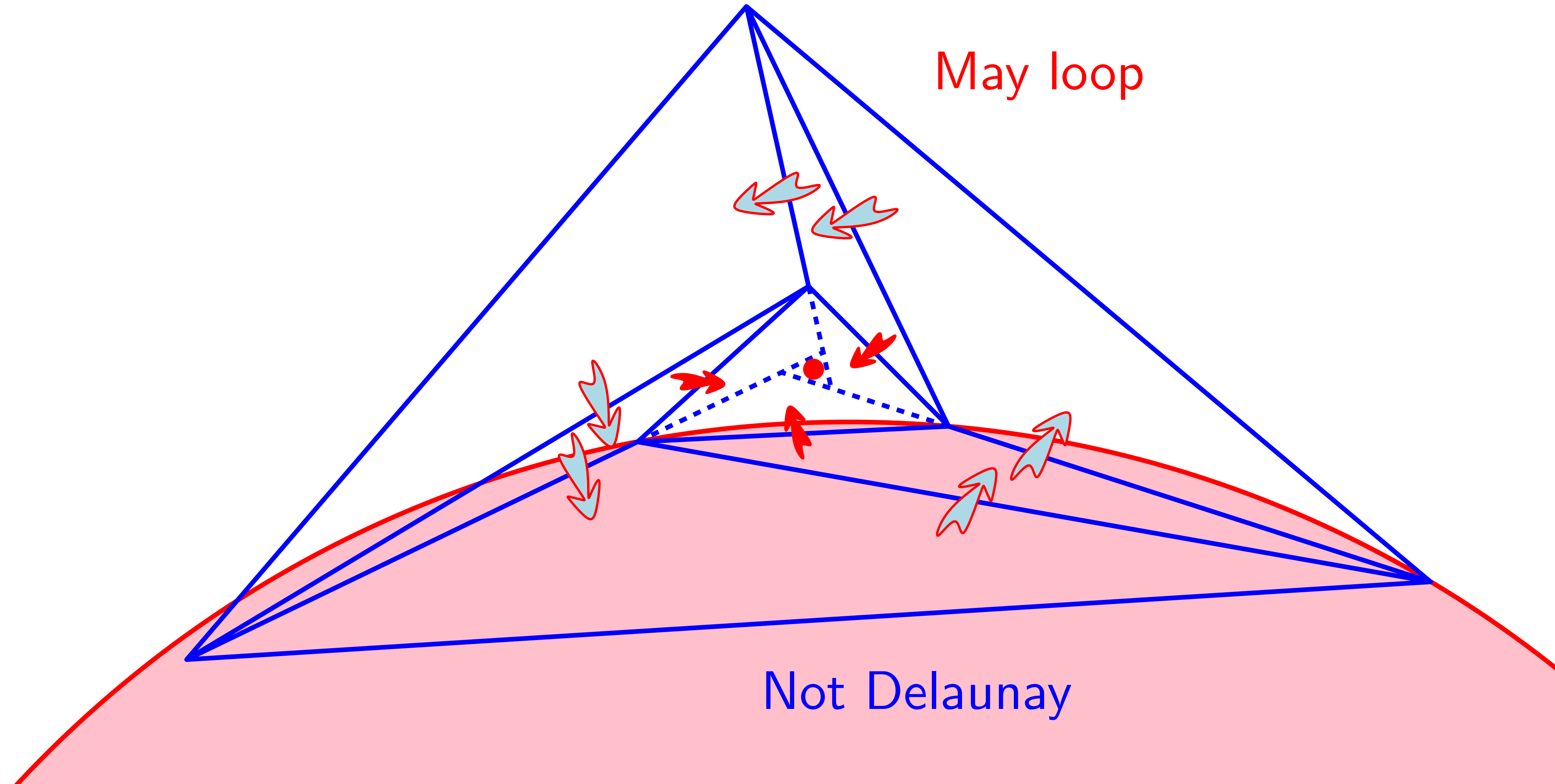
Delaunay Triangulation: incremental algorithm

Visibility walk terminates ?



Delaunay Triangulation: incremental algorithm

Visibility walk terminates ?



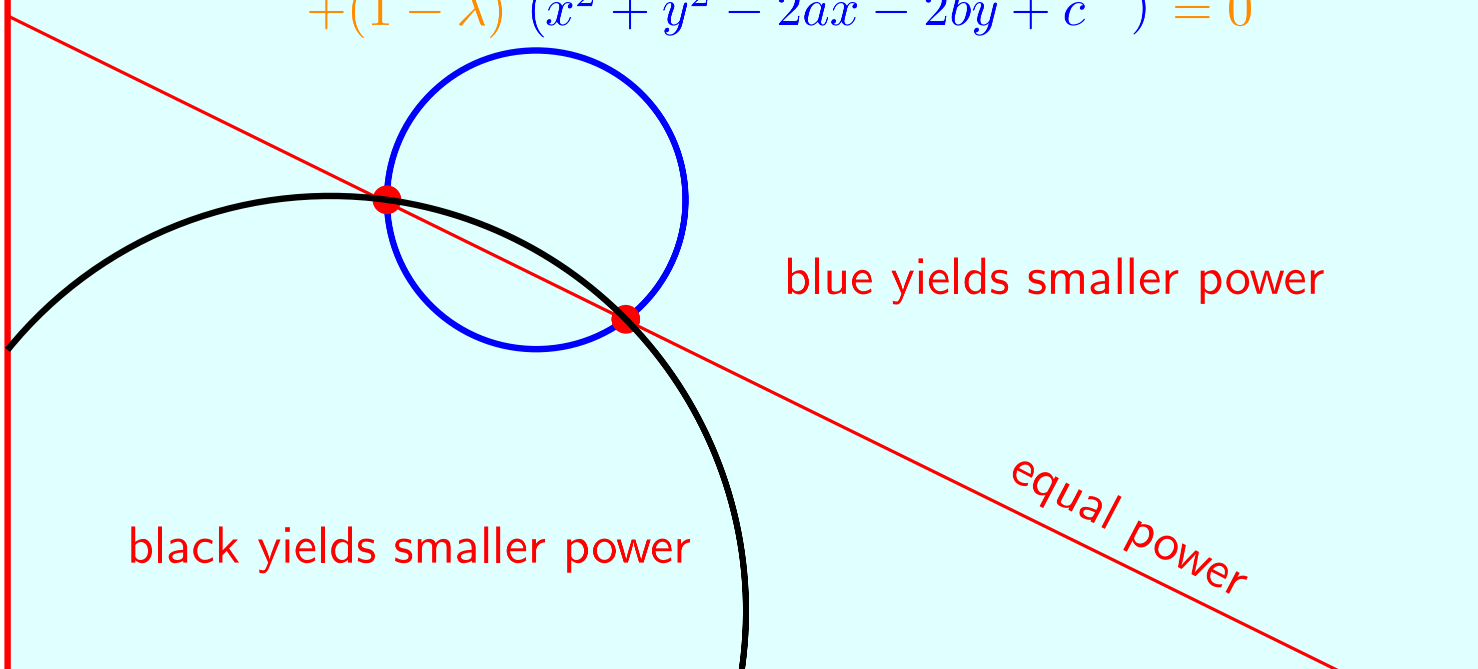
Delaunay Triangulation: incremental algorithm

Visibility walk terminates ?

Delaunay Triangulation: pencils of circles

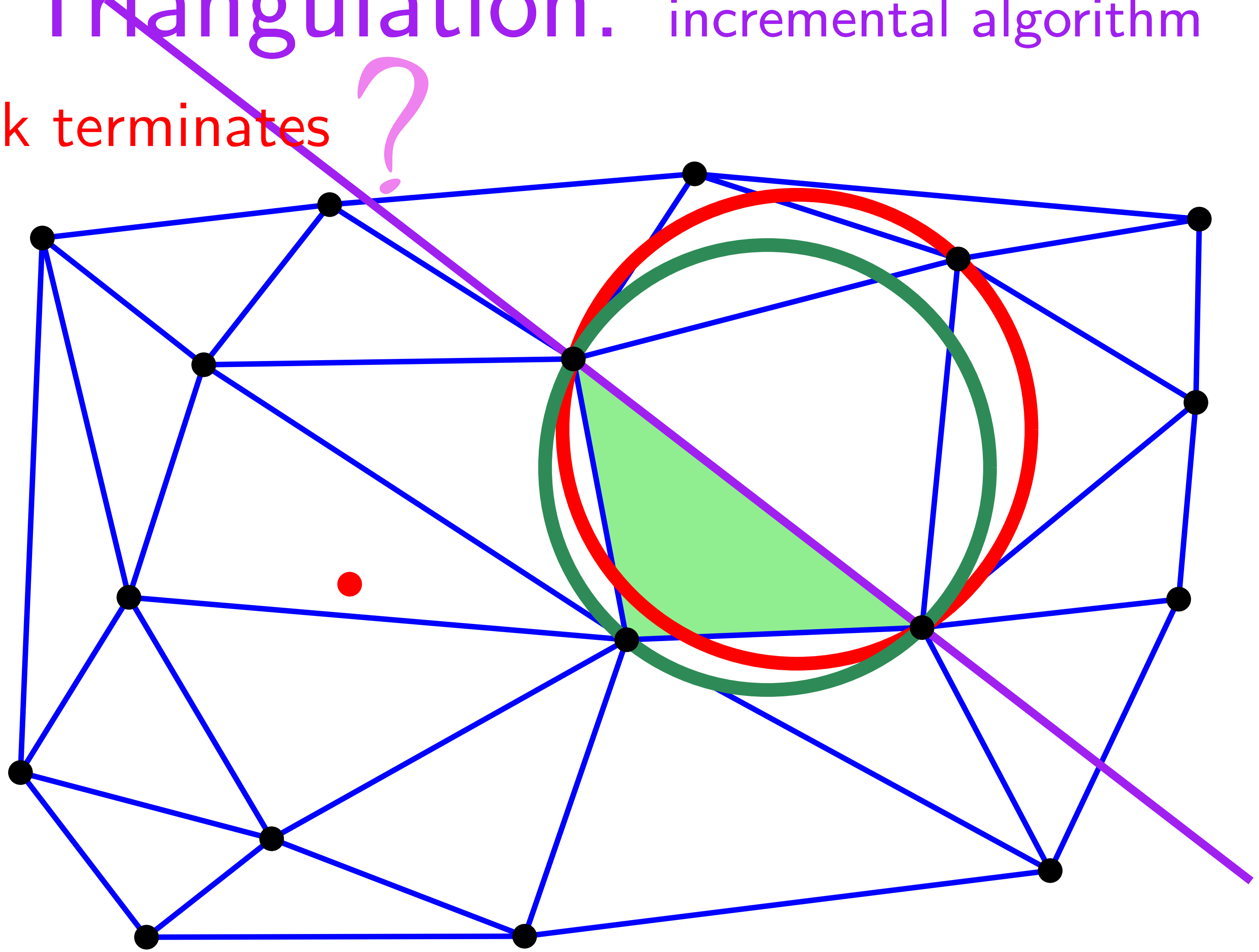
Power of a point w.r.t a circle

$$\lambda \quad (x^2 + y^2 - 2a'x - 2b'y + c')$$
$$+(1-\lambda) \; (x^2 + y^2 - 2ax - 2by + c \quad) = 0$$



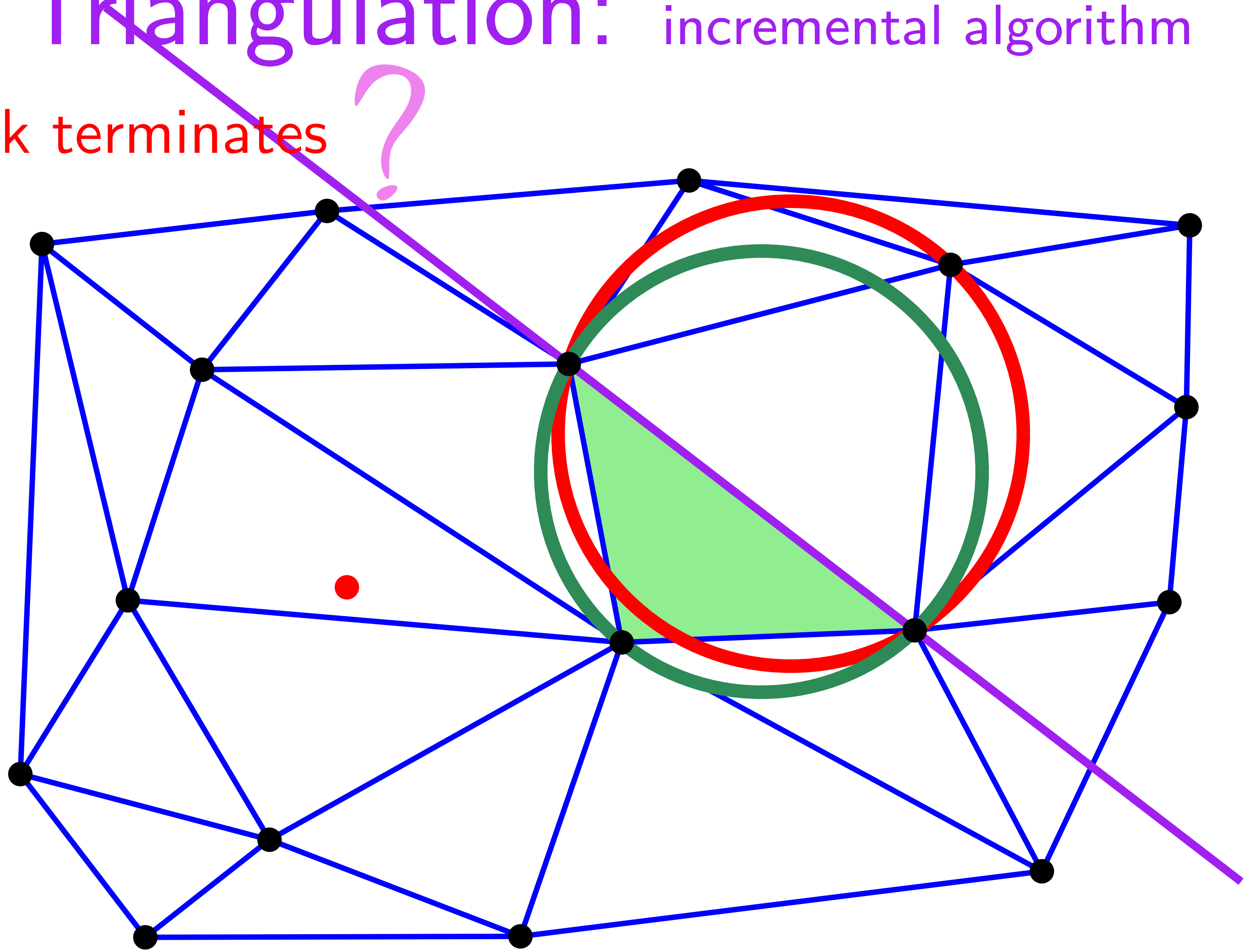
Delaunay Triangulation: incremental algorithm

Visibility walk terminates ?



Delaunay Triangulation: incremental algorithm

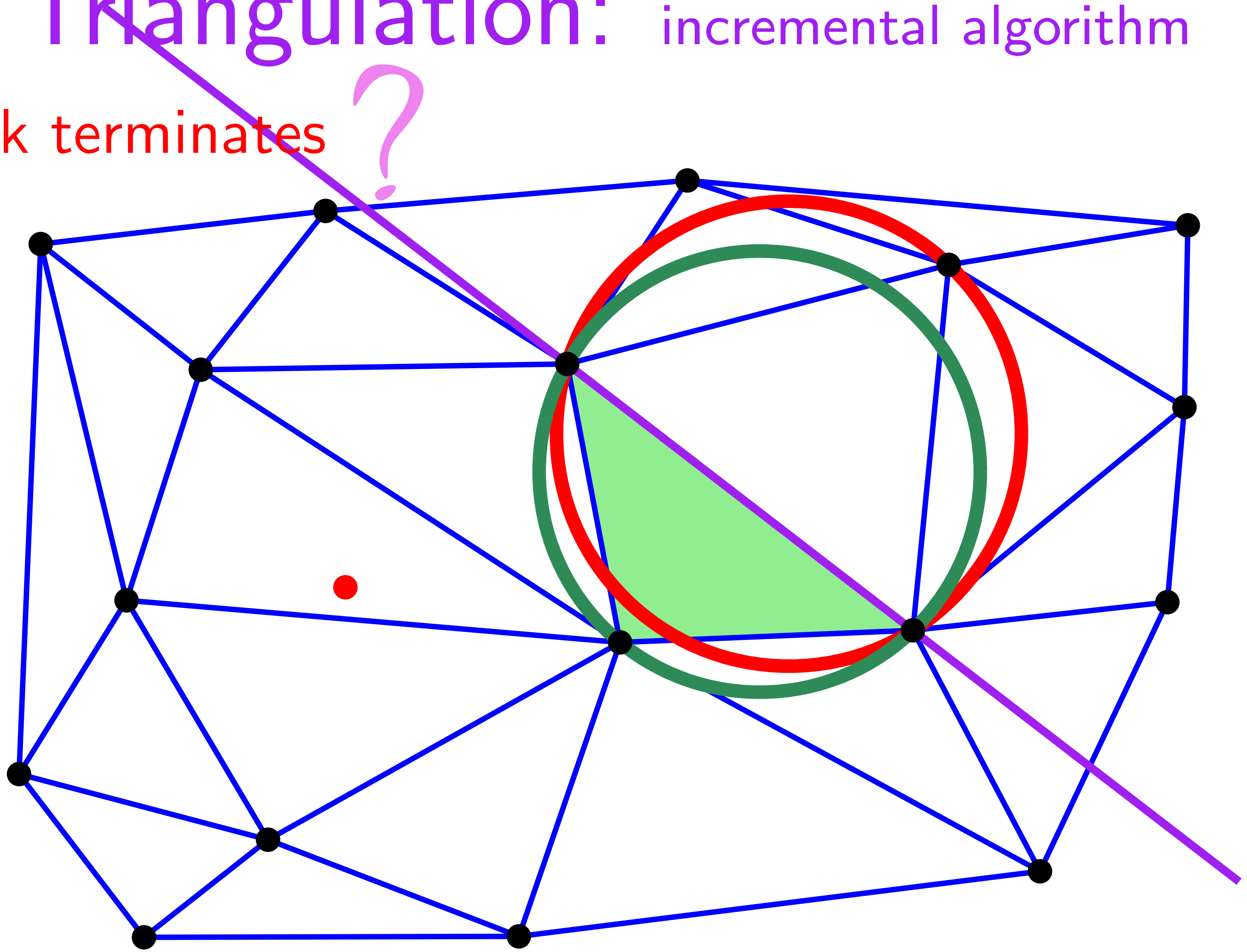
Visibility walk terminates ?



Green power $<$ Red power

Delaunay Triangulation: incremental algorithm

Visibility walk terminates ?

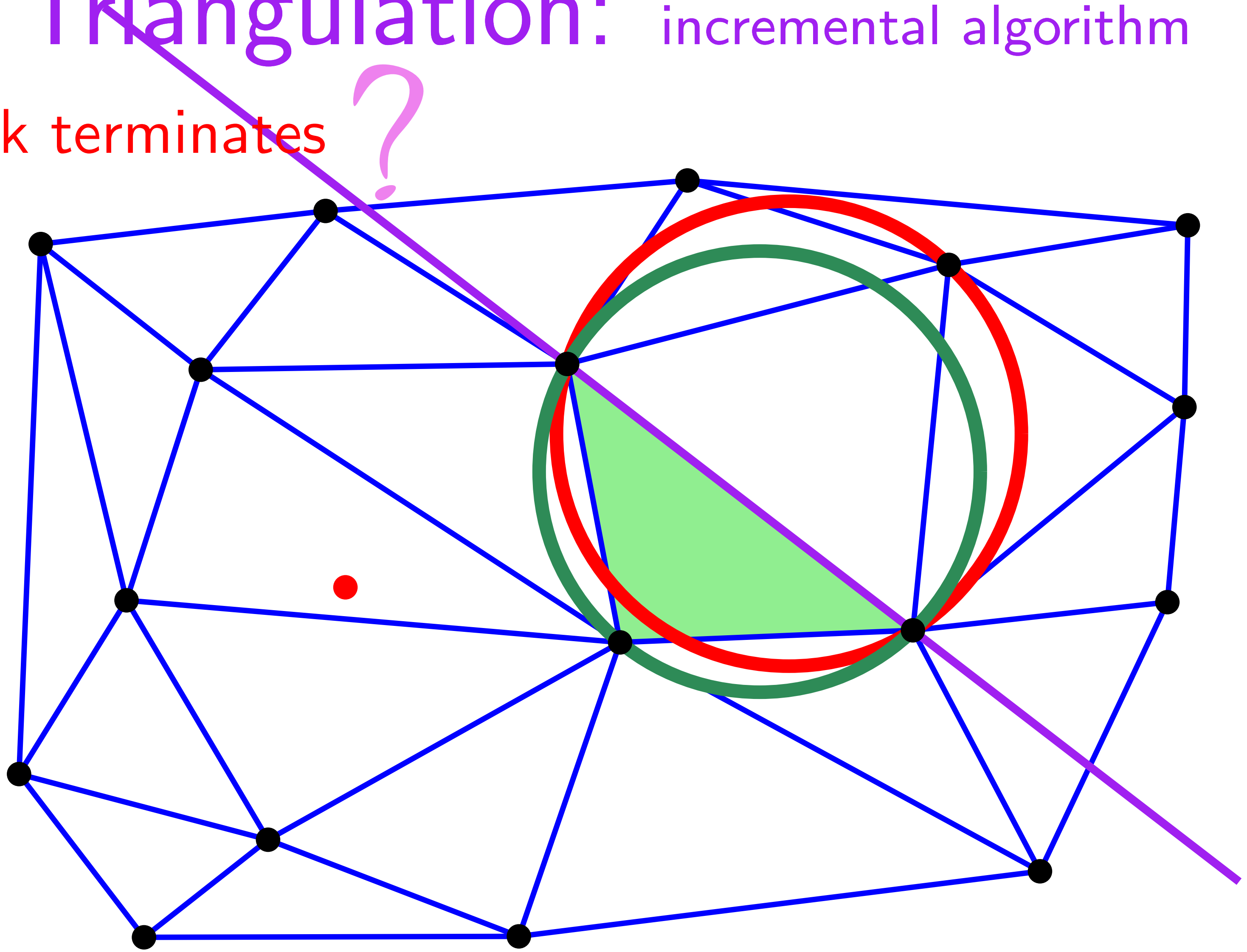


Green power < Red power

Power decreases

Delaunay Triangulation: incremental algorithm

Visibility walk terminates



Green power < Red power

Power decreases

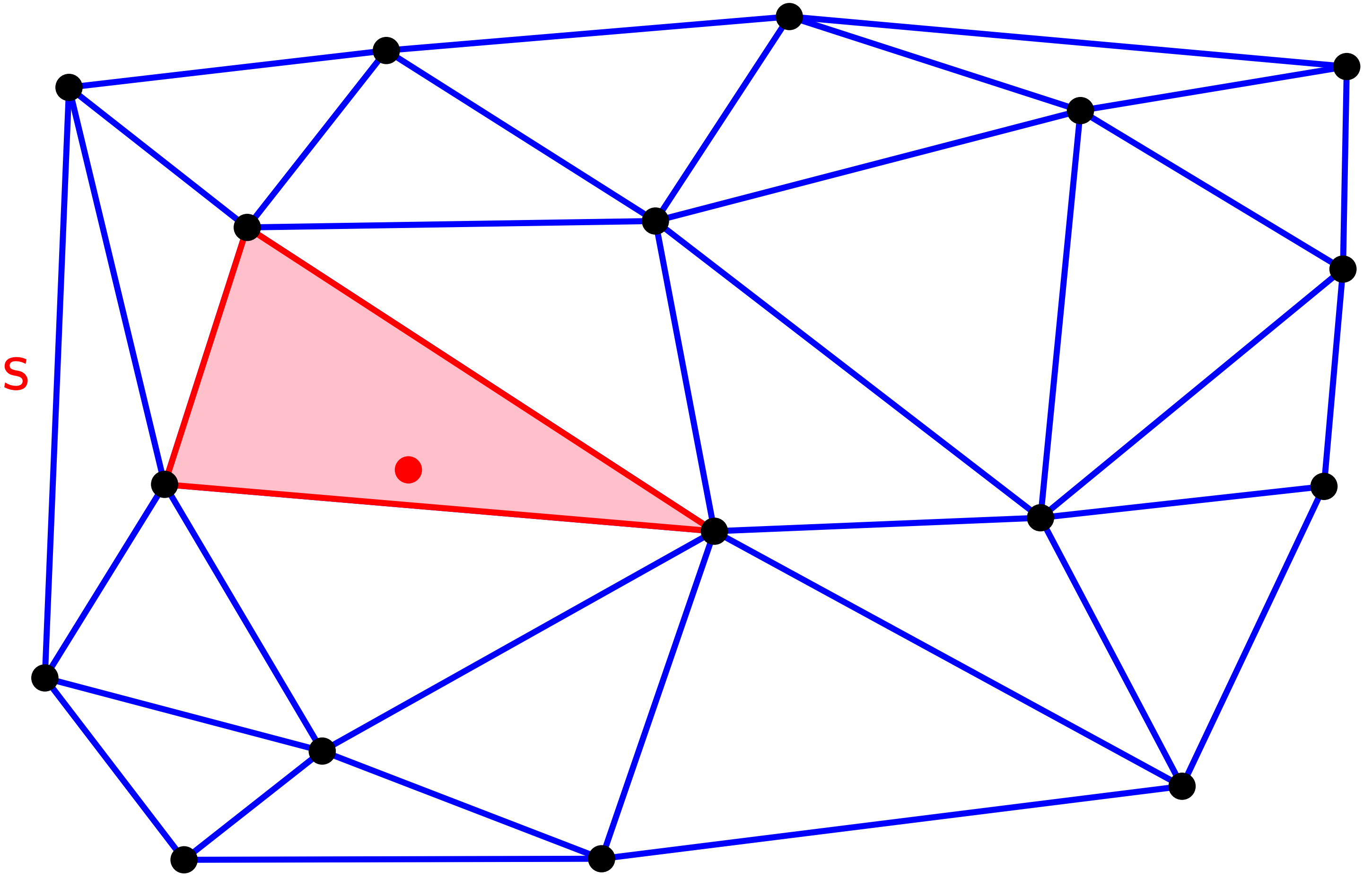
Visibility walk terminates

Delaunay Triangulation: incremental algorithm

New point

Locate

Search conflicts

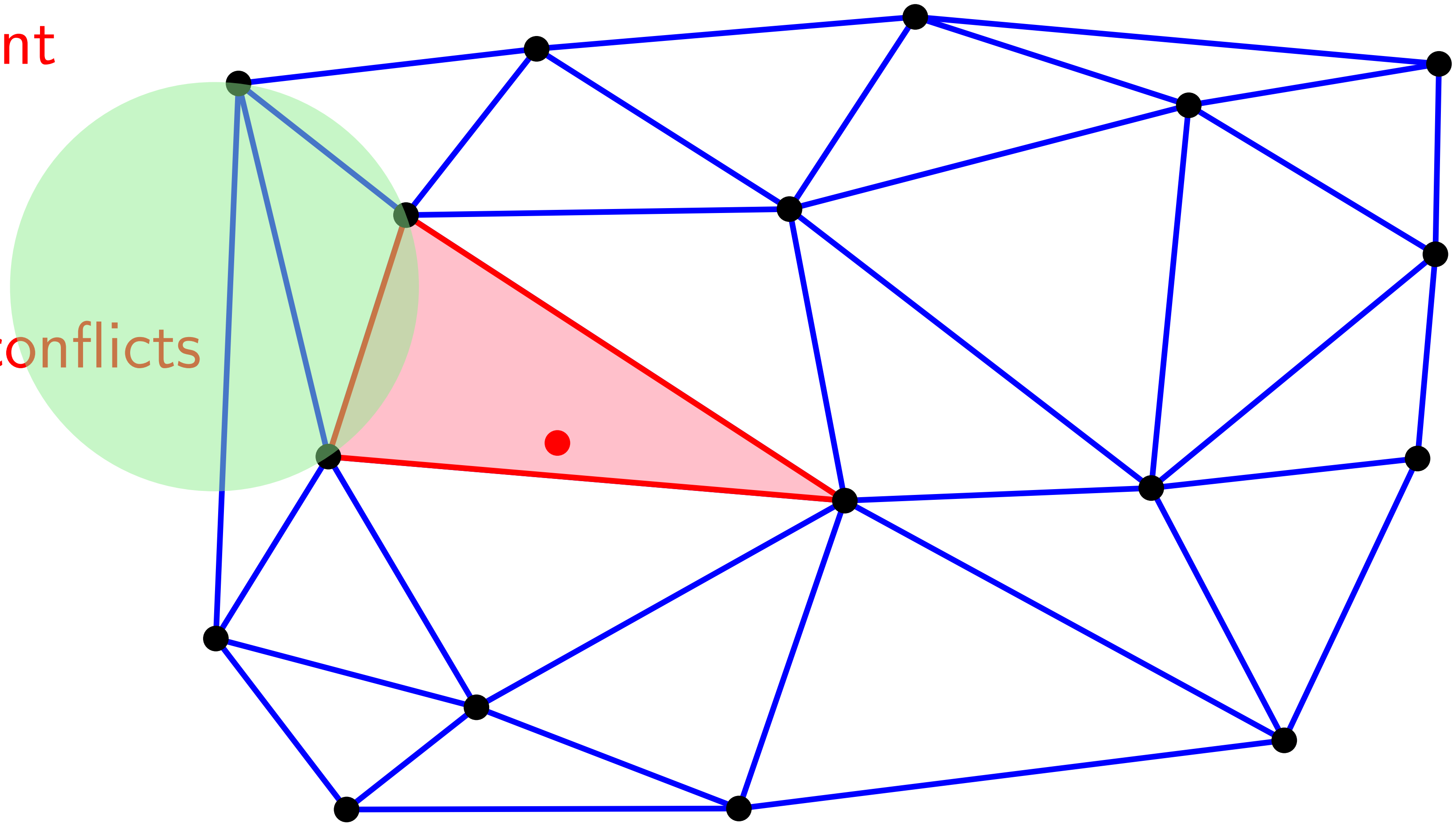


Delaunay Triangulation: incremental algorithm

New point

Locate

Search conflicts

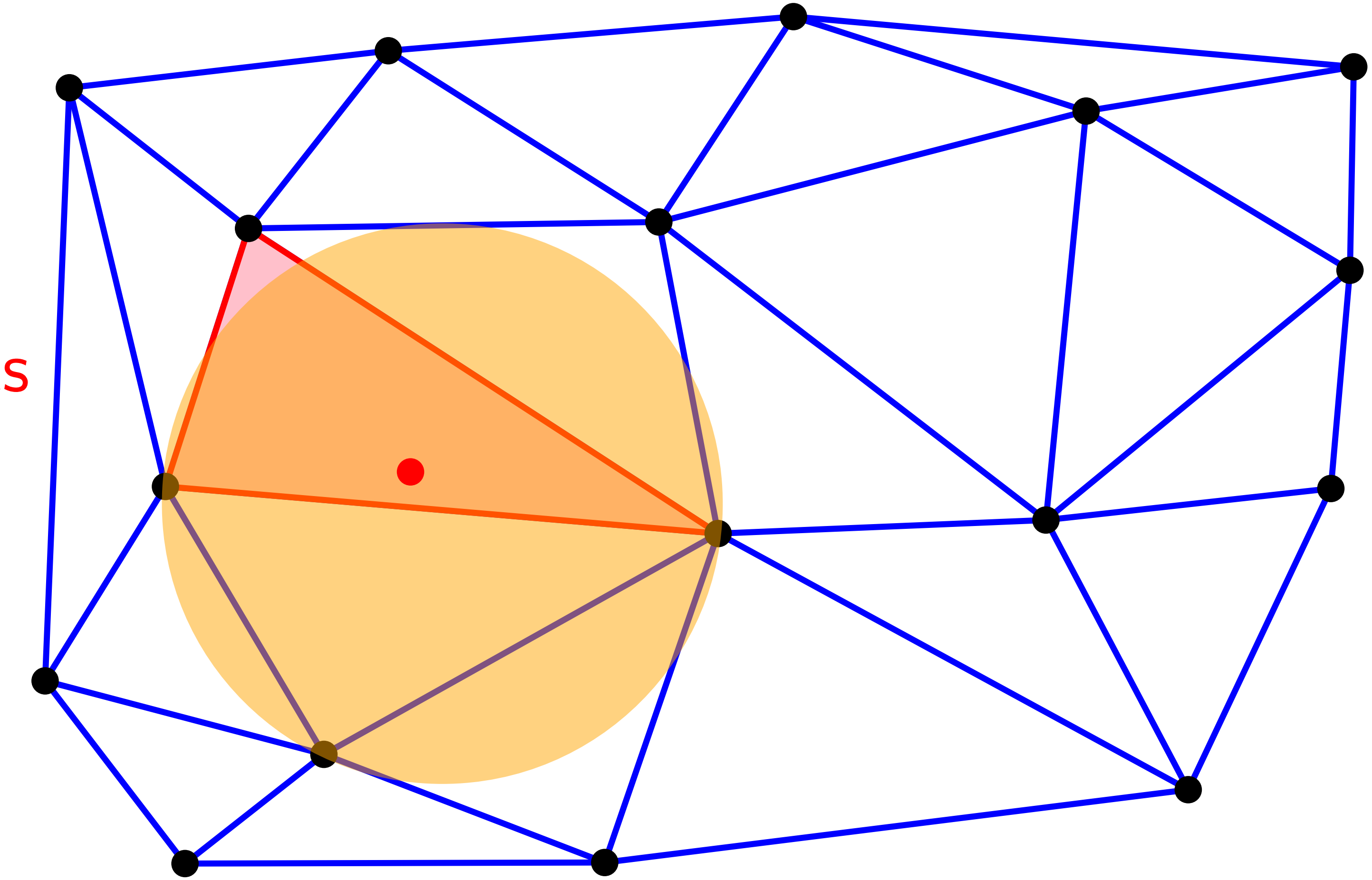


Delaunay Triangulation: incremental algorithm

New point

Locate

Search conflicts

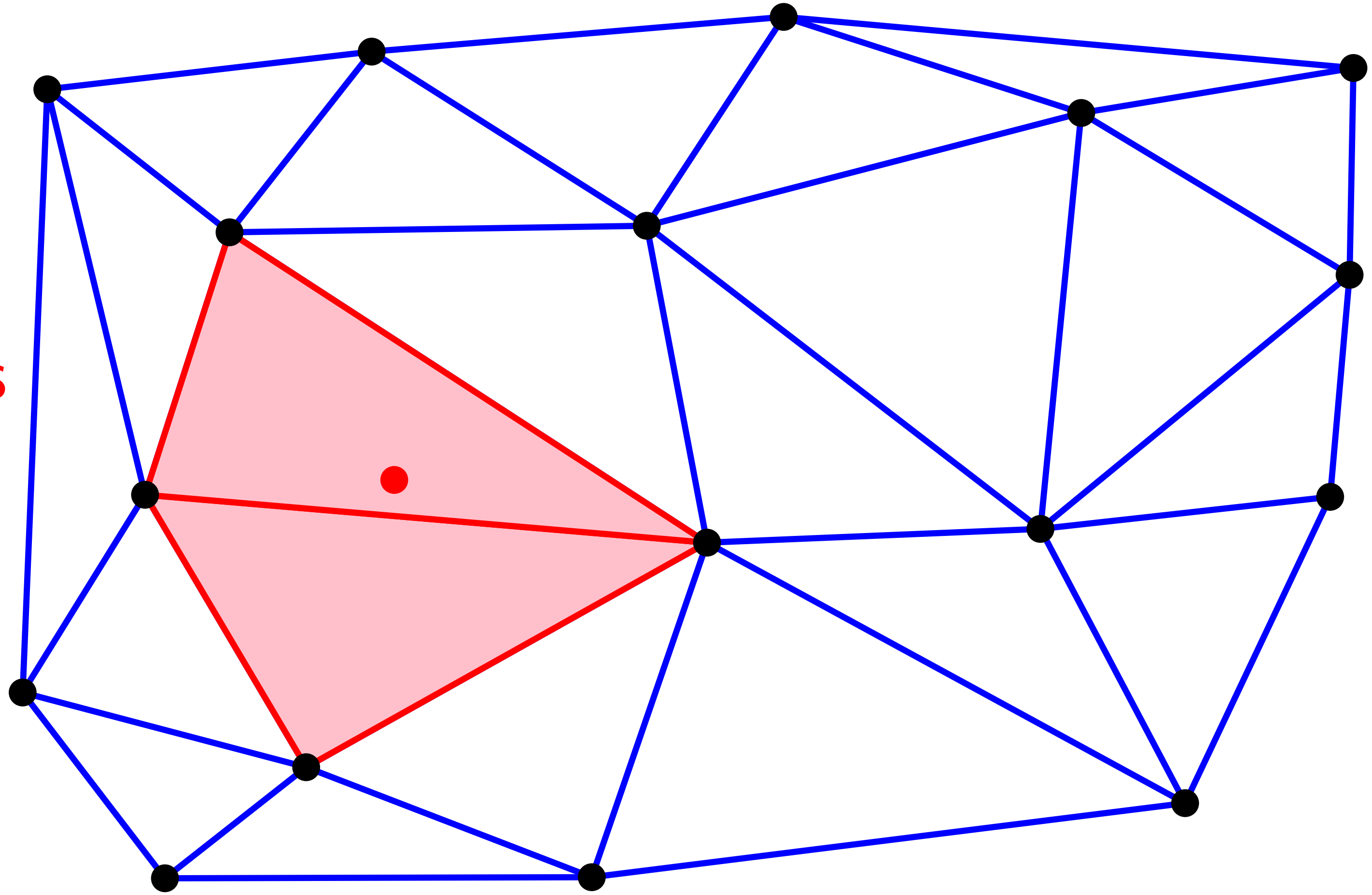


Delaunay Triangulation: incremental algorithm

New point

Locate

Search conflicts

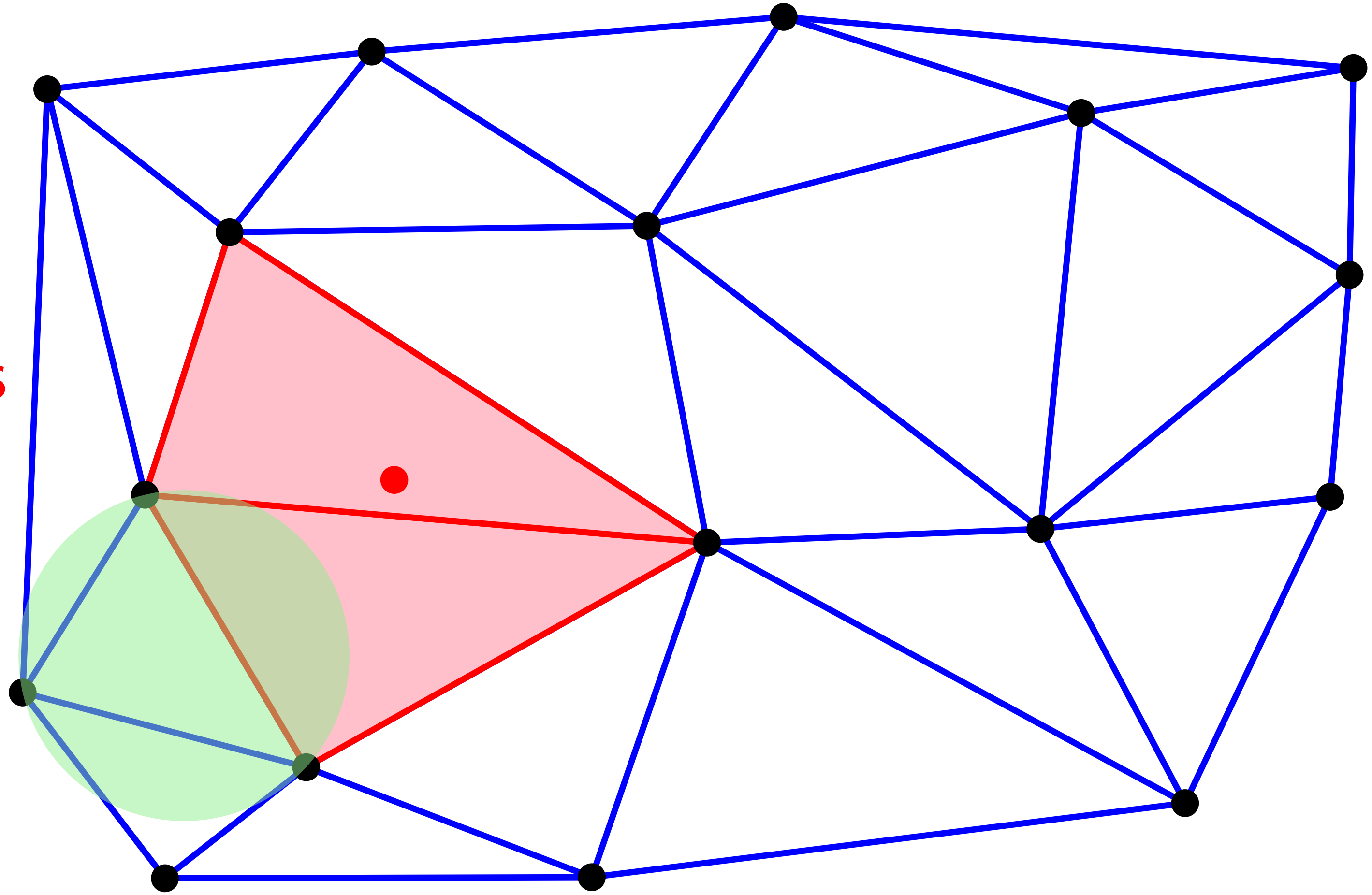


Delaunay Triangulation: incremental algorithm

New point

Locate

Search conflicts

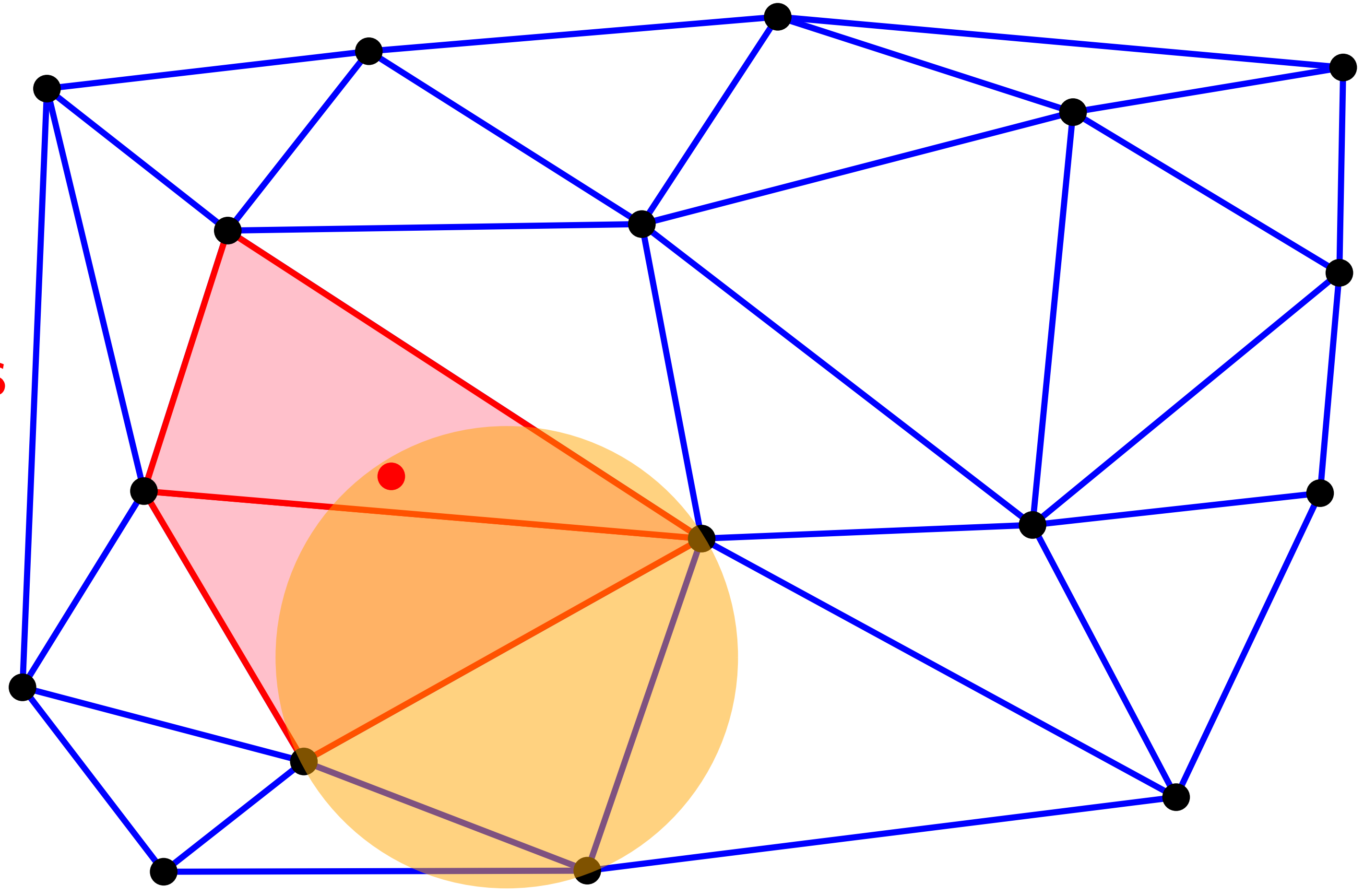


Delaunay Triangulation: incremental algorithm

New point

Locate

Search conflicts

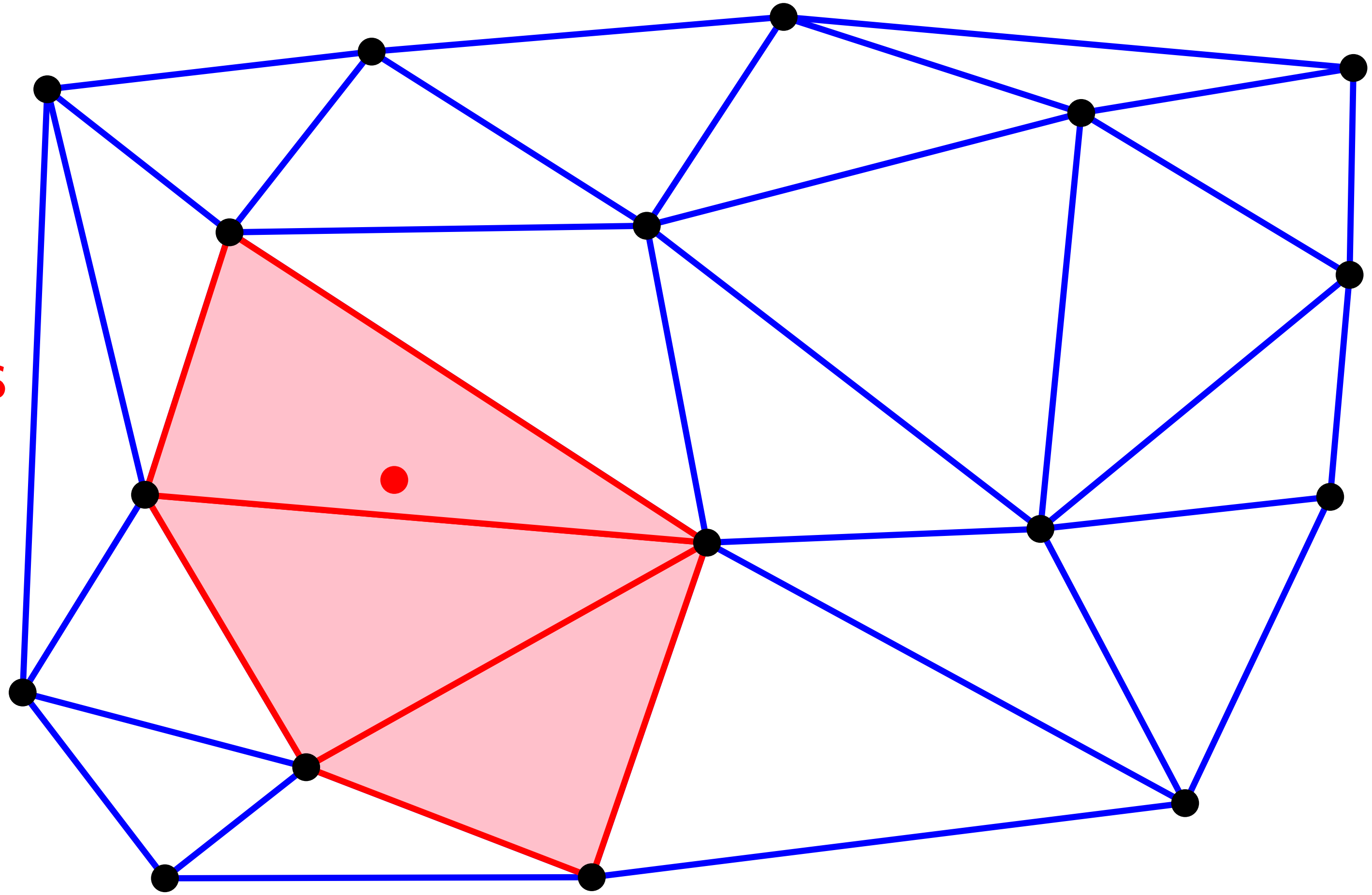


Delaunay Triangulation: incremental algorithm

New point

Locate

Search conflicts

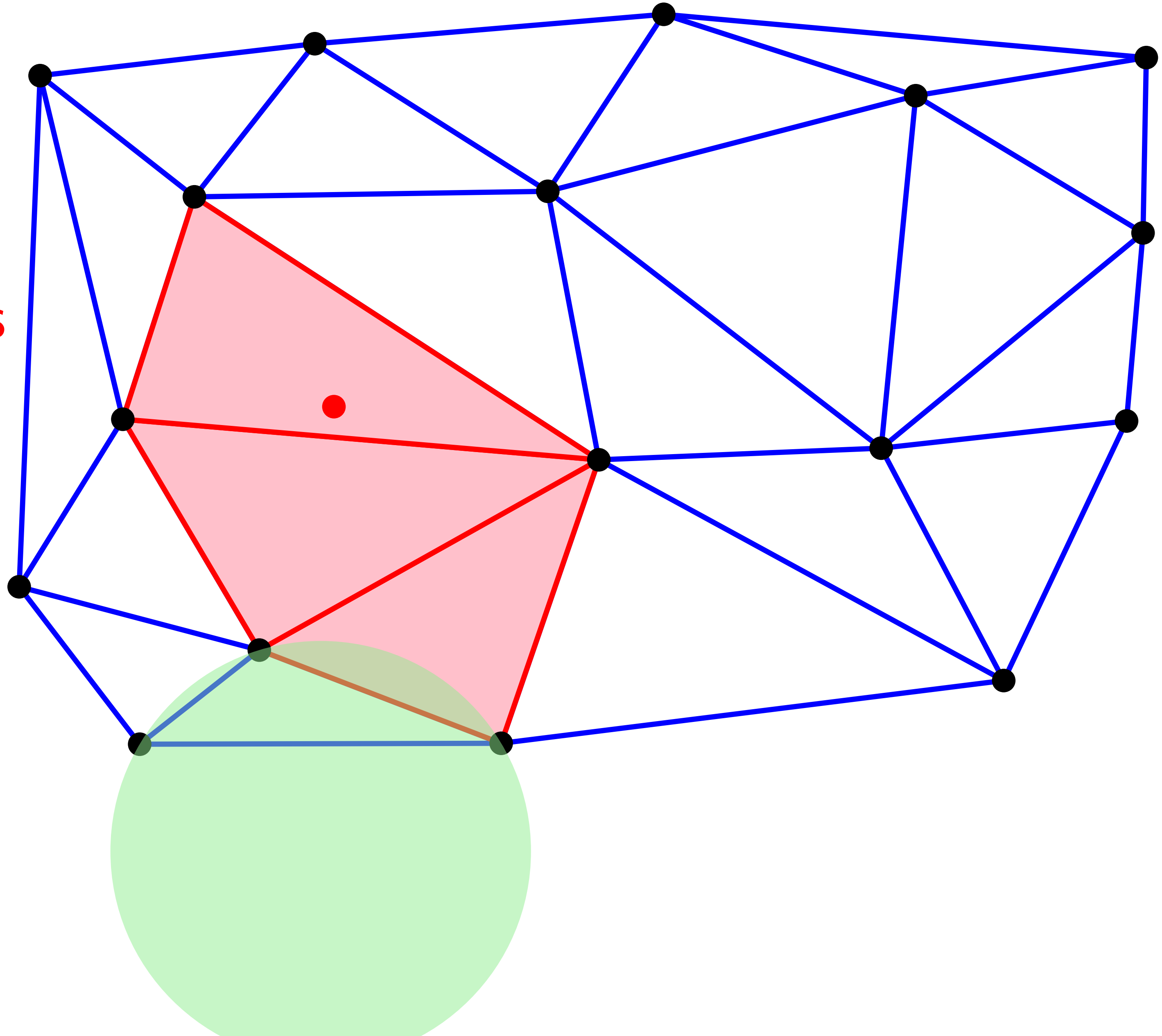


Delaunay Triangulation: incremental algorithm

New point

Locate

Search conflicts

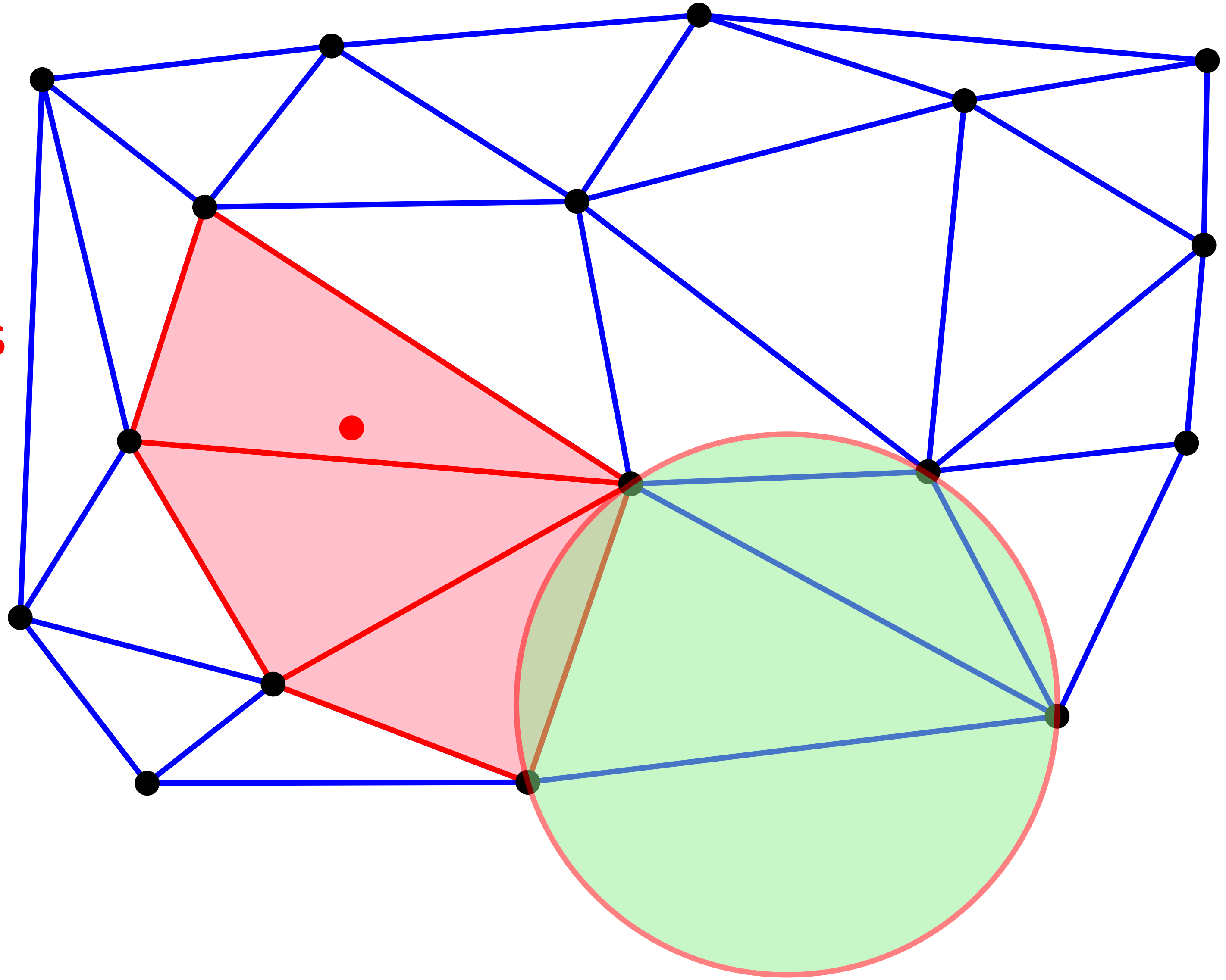


Delaunay Triangulation: incremental algorithm

New point

Locate

Search conflicts

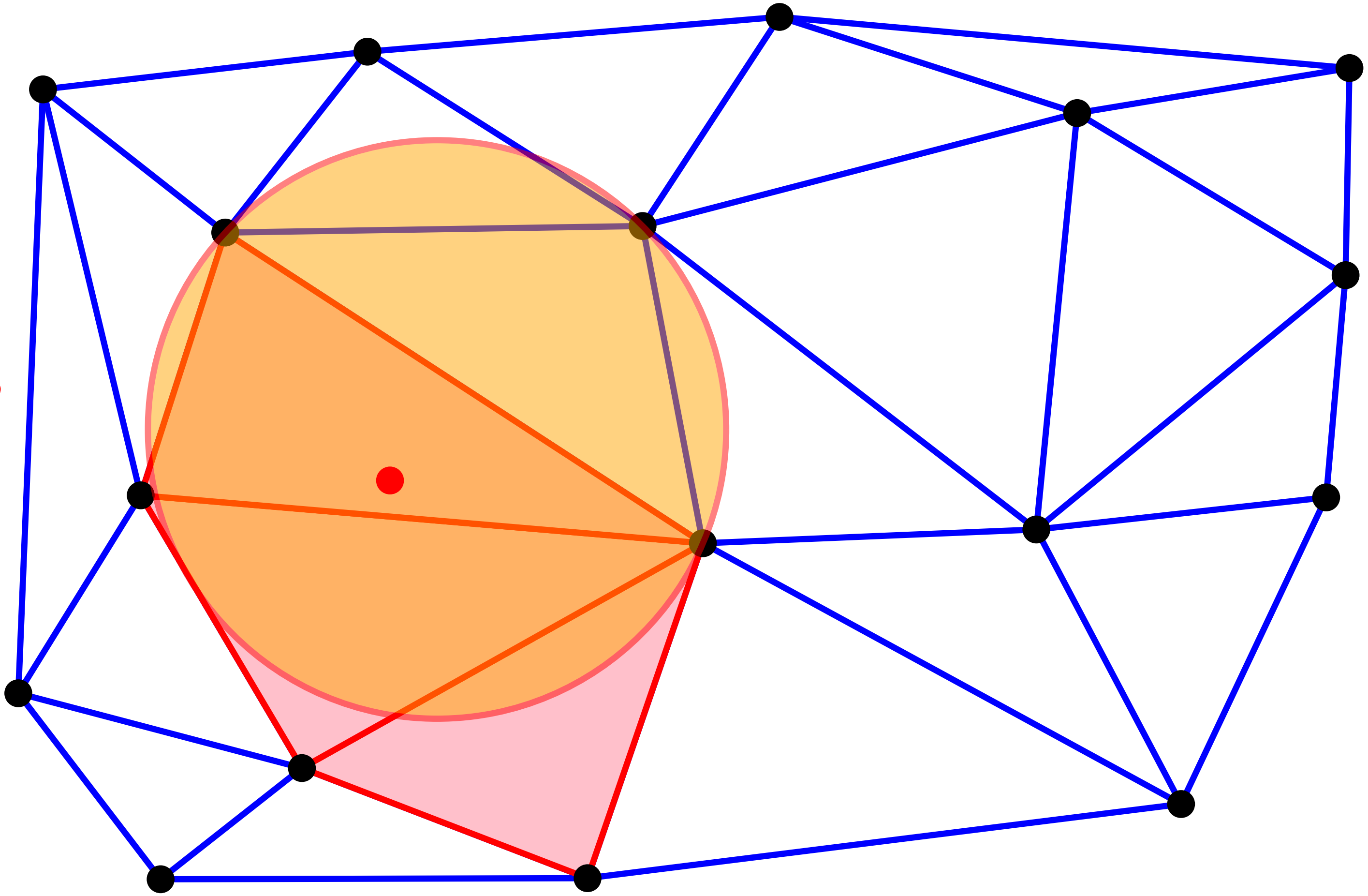


Delaunay Triangulation: incremental algorithm

New point

Locate

Search conflicts

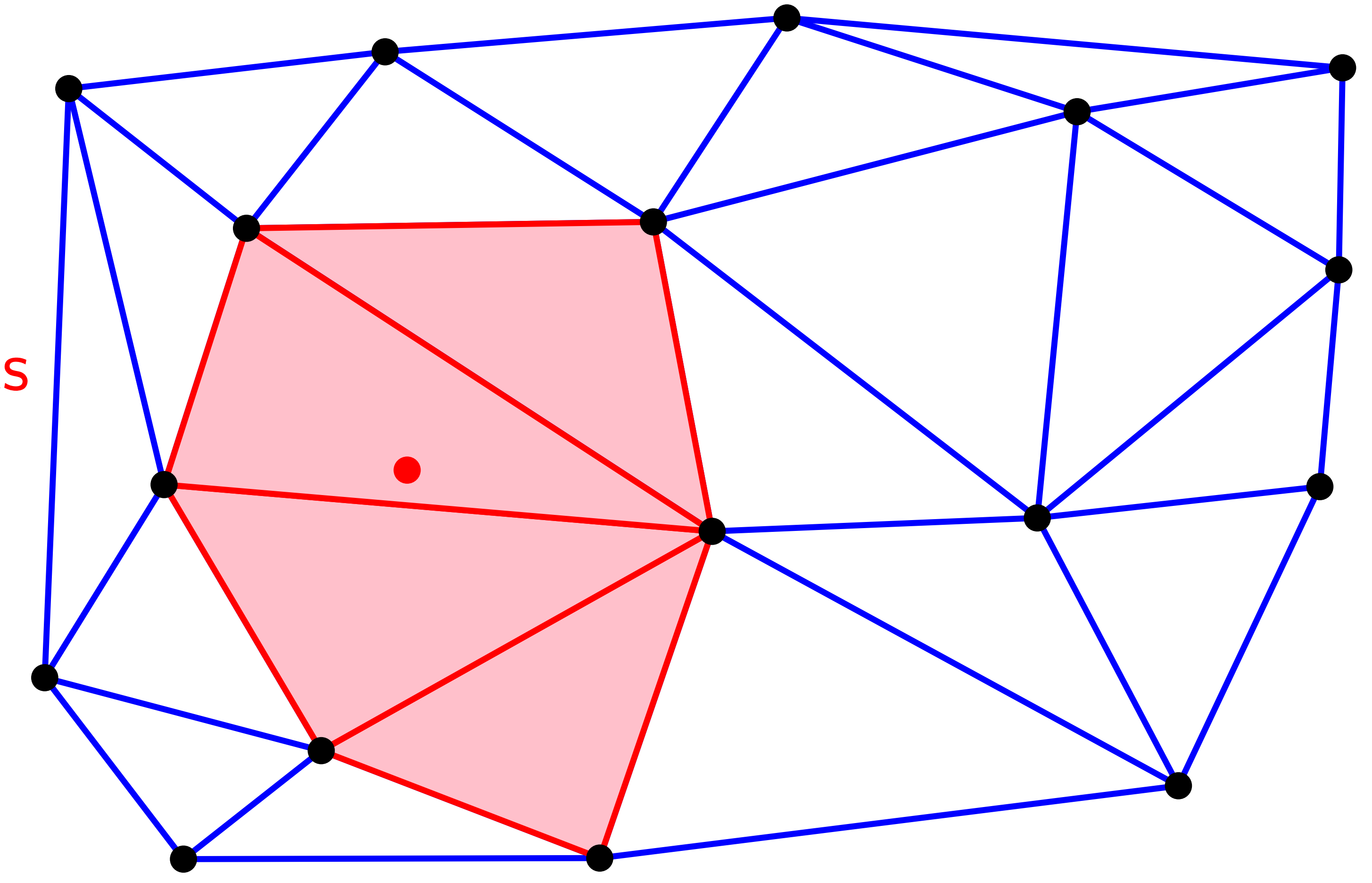


Delaunay Triangulation: incremental algorithm

New point

Locate

Search conflicts

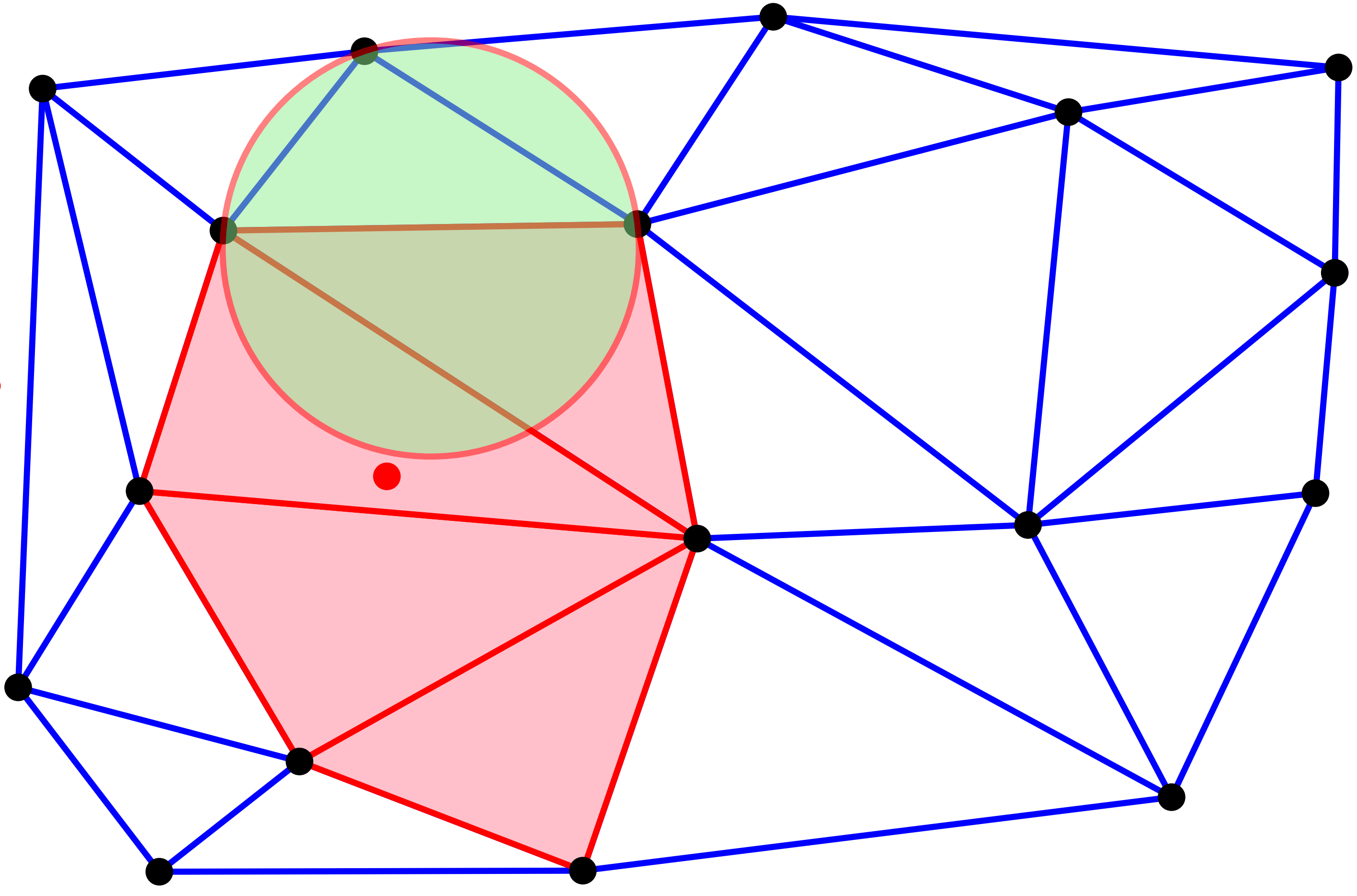


Delaunay Triangulation: incremental algorithm

New point

Locate

Search conflicts

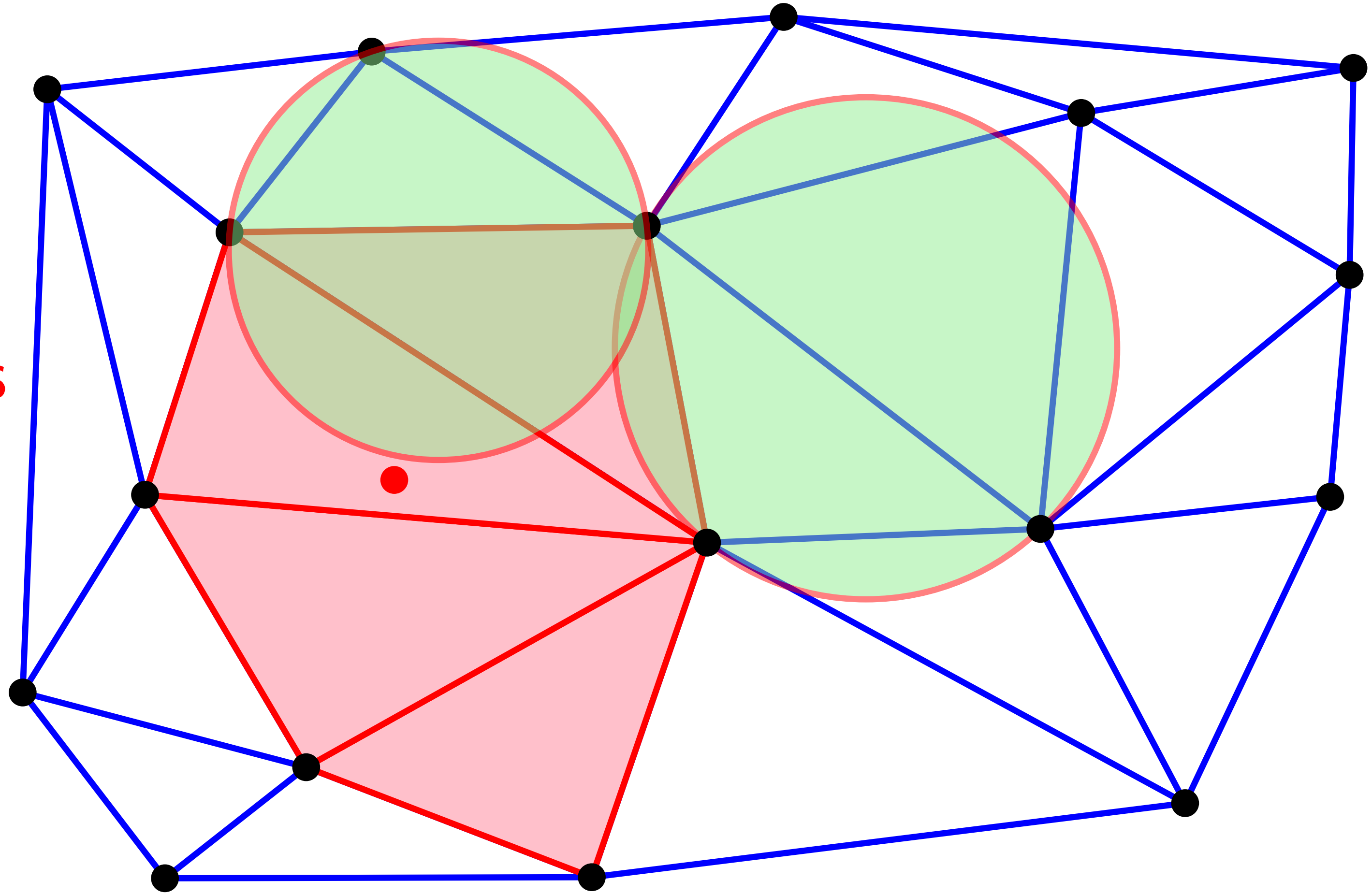


Delaunay Triangulation: incremental algorithm

New point

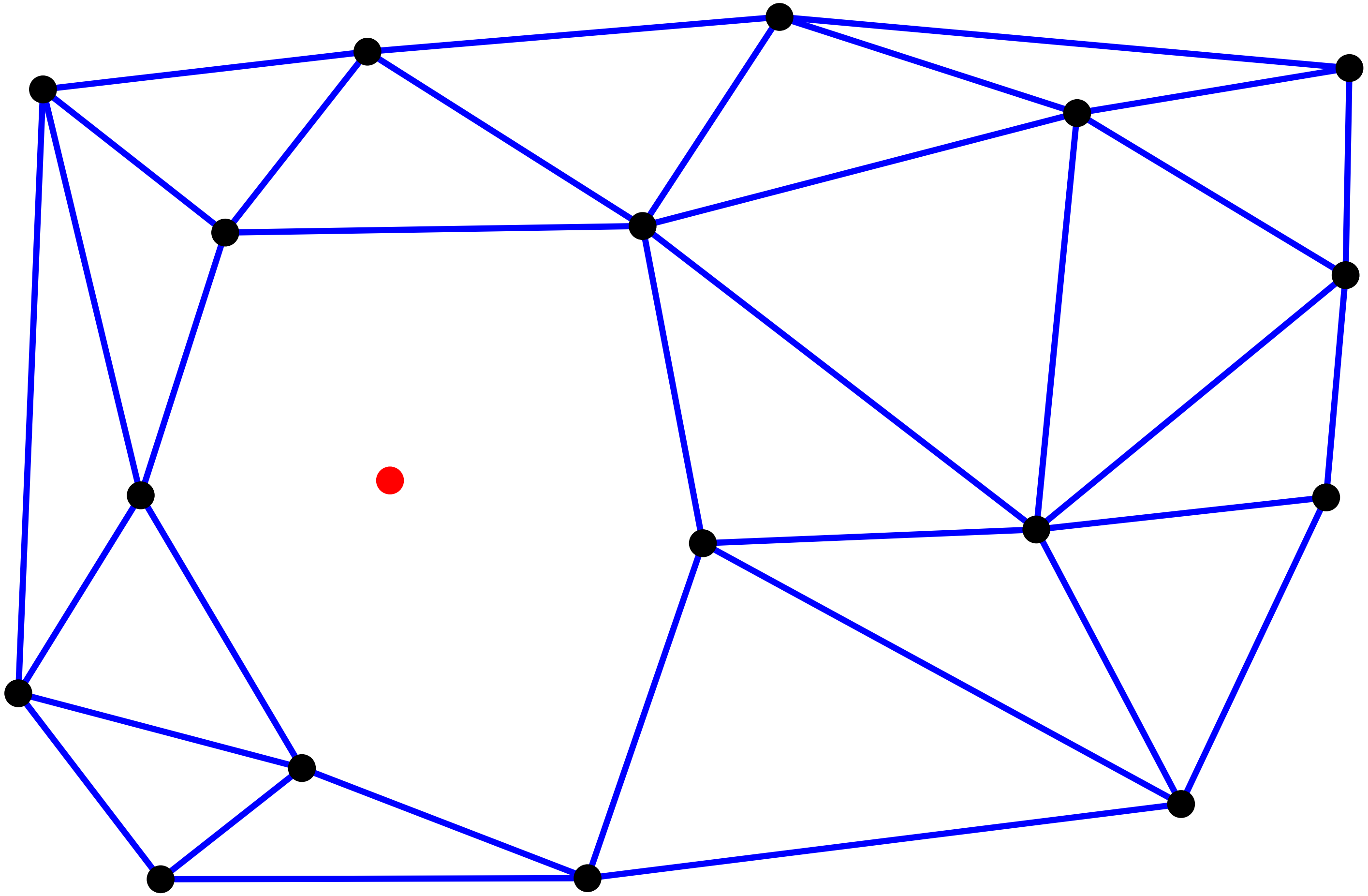
Locate

Search conflicts



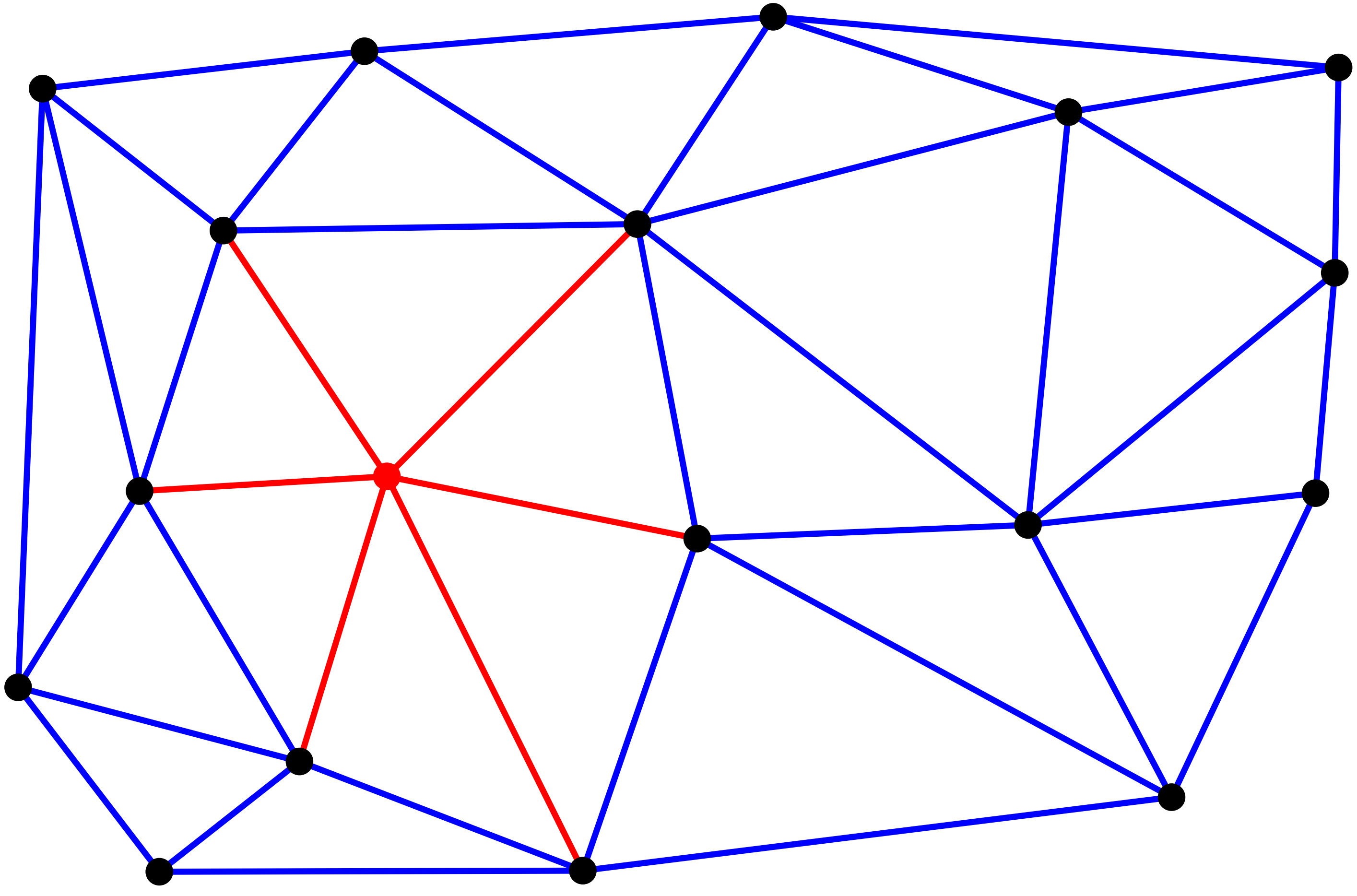
Delaunay Triangulation: incremental algorithm

New point



Delaunay Triangulation: incremental algorithm

New point



Delaunay Triangulation: incremental algorithm

Complexity

Locate

Search conflicts

Delaunay Triangulation: incremental algorithm

Complexity

Locate

Search conflicts $\sharp$ triangles in conflict

$\sharp$ triangles neighboring triangles in conflict

Delaunay Triangulation: incremental algorithm

Complexity

Locate

Search conflicts

$\sharp$ triangles in conflict

$\sharp$ triangles neighboring triangles in conflict

degree of new point in new triangulation

$< n$

Delaunay Triangulation: incremental algorithm

Complexity

Locate — Walk may visit all triangles

$< 2n$

Search conflicts

degree of new point in new triangulation

$< n$

Delaunay Triangulation: incremental algorithm

Complexity

Locate $O(n)$ per insertion

Search conflicts

Delaunay Triangulation: incremental algorithm

Complexity

Locate $O(n)$ per insertion

Search conflicts

$O(n^2)$ for the whole construction

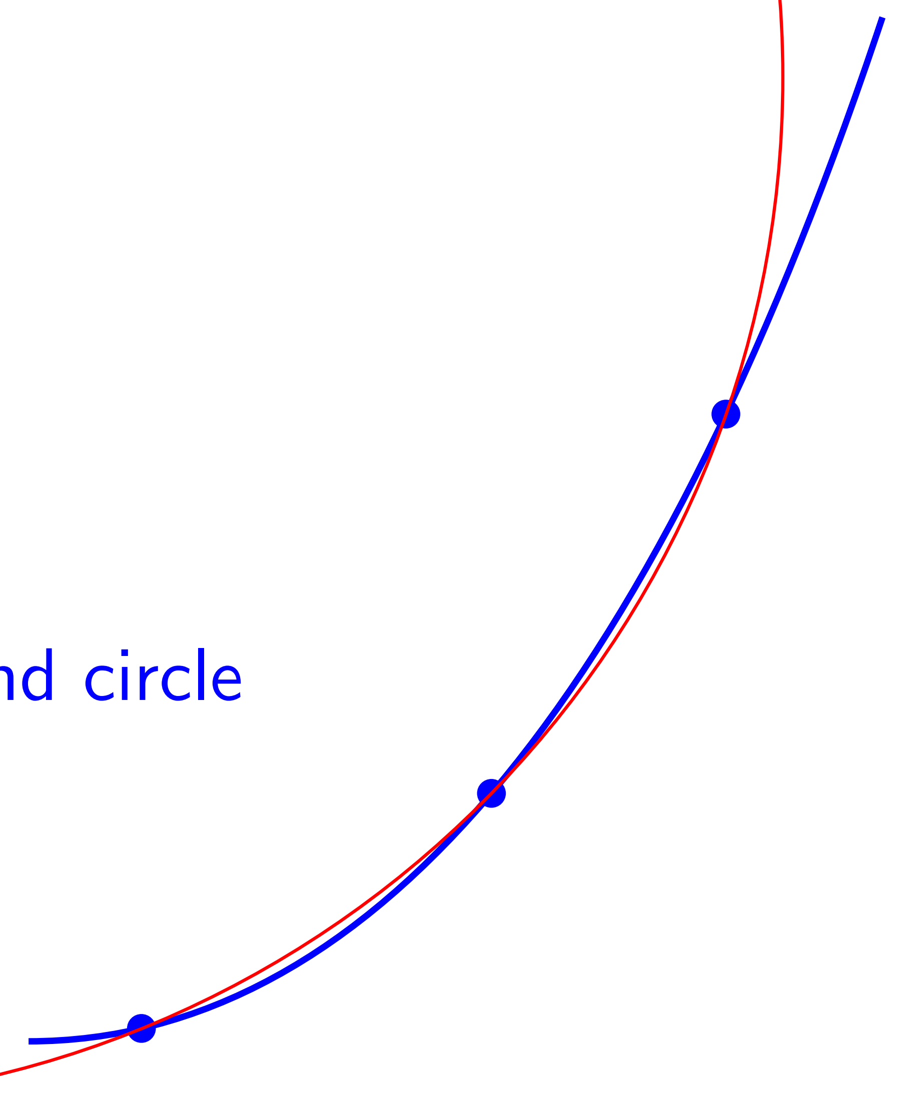
Delaunay Triangulation: incremental algorithm

Complexity

Locate

Search conflicts

half-parabola and circle

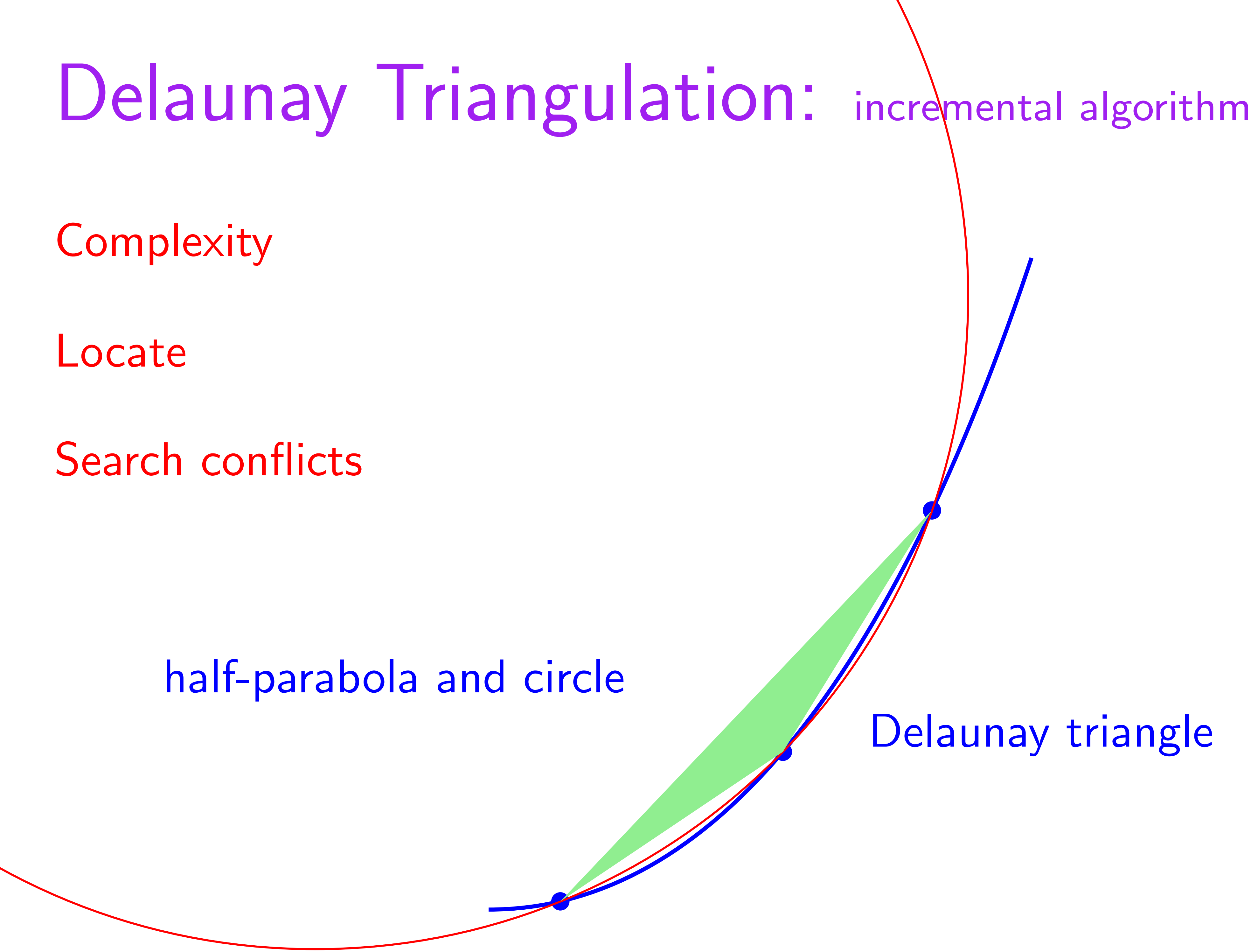


Delaunay Triangulation: incremental algorithm

Complexity

Locate

Search conflicts

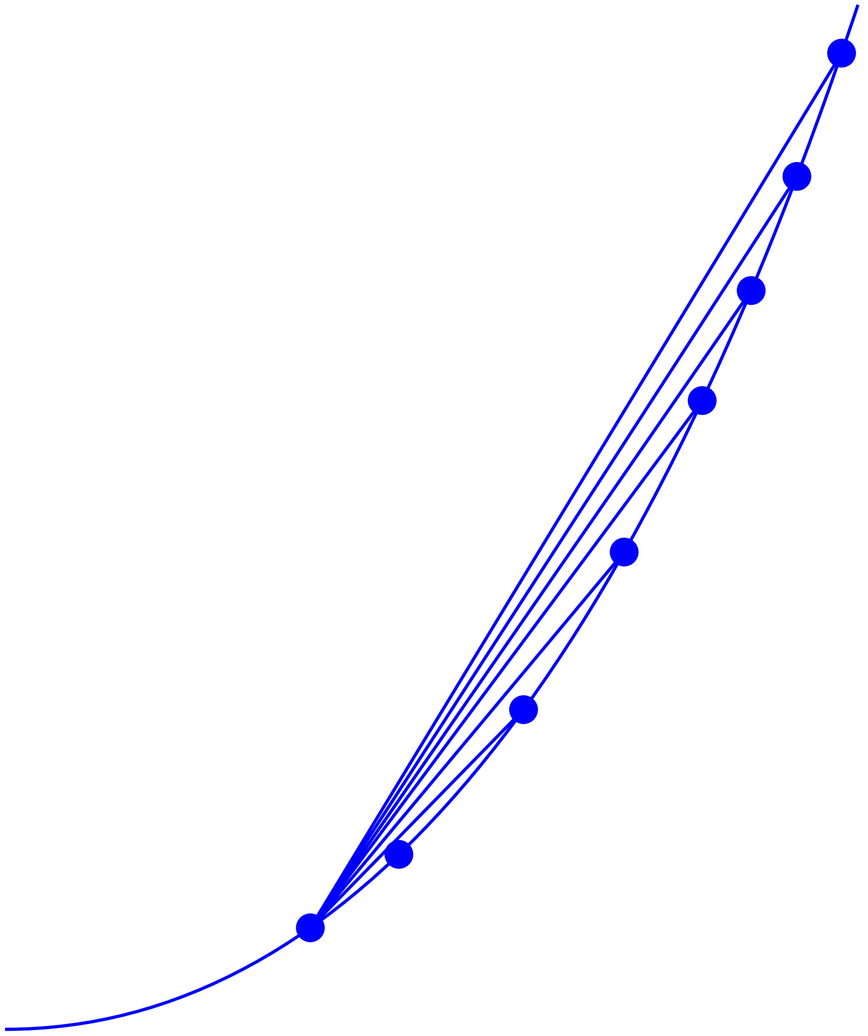


Delaunay Triangulation: incremental algorithm

Complexity

Locate

Search conflicts

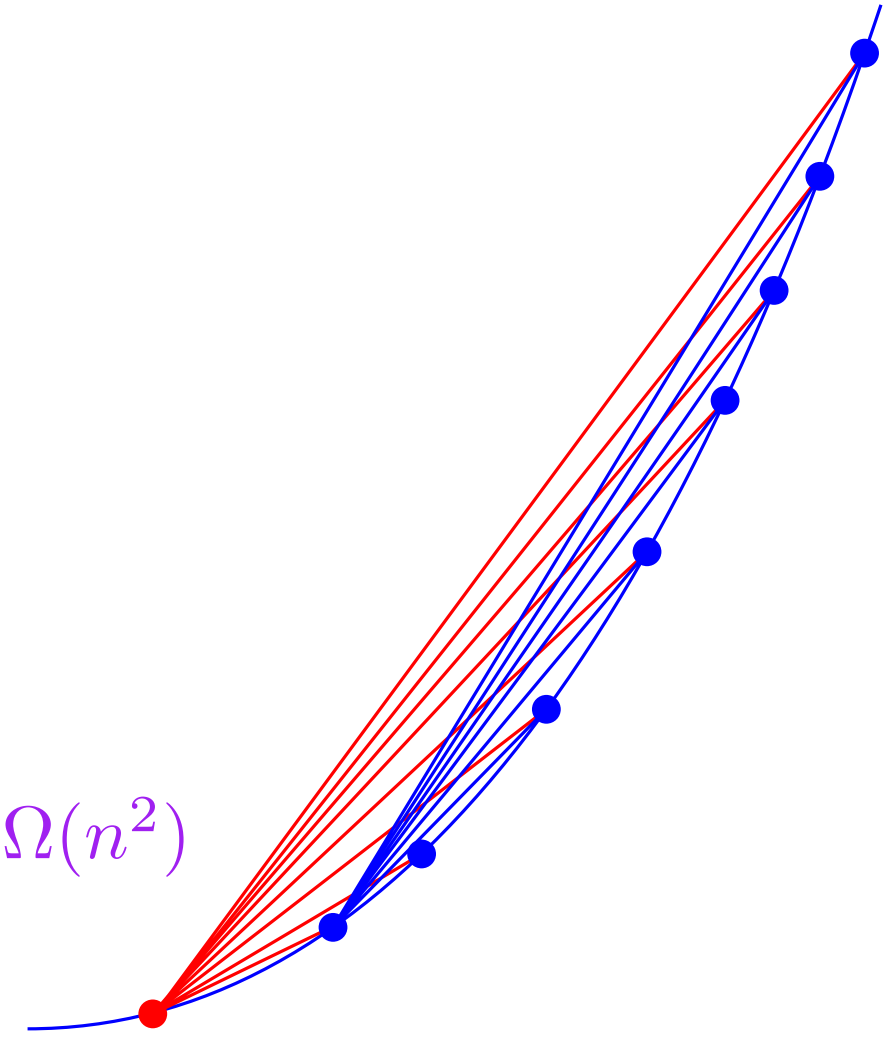


Delaunay Triangulation: incremental algorithm

Complexity

Locate

Search conflicts



Insertion: $\Omega(n)$

Whole construction: $\Omega(n^2)$

Delaunay Triangulation: incremental algorithm

In practice

Complexity

Locate — Many possibilities (walk, Delaunay hierarchy)

Search conflicts — Randomized

Teaser randomization lecture

Algorithmes optimaux pour la triangulation de Delaunay

Division-Fusion

Division-Fusion

Technique classique

exemple: tri

Problème de taille n

→ 2 sous-problèmes de taille $\frac{n}{2}$

résolution récursive

fusion

Division-Fusion

Problème de taille n $f(n)$

$O(n)$ → 2 sous-problèmes de taille $\frac{n}{2}$

résolution récursive $2 \cdot f(\frac{n}{2})$

fusion $O(n)$

Division-Fusion

$$f(n) = O(n) + 2f(\tfrac{n}{2})$$

Problème de taille n $f(n)$

$O(n)$ → 2 sous-problèmes de taille $\frac{n}{2}$

résolution récursive $2 \cdot f(\frac{n}{2})$

fusion $O(n)$

Division-Fusion

$$f(n) = O(n) + 2f\left(\tfrac{n}{2}\right)$$

$$f(n) = \underbrace{\begin{array}{l} n + 2f\left(\frac{n}{2}\right) \\ n + 2\left(\frac{n}{2} + 2f\left(\frac{n}{4}\right)\right) \\ n + 2\left(\frac{n}{2} + 2\left(\frac{n}{4} + 2f\left(\frac{n}{8}\right)\right)\right) \\ n + 2\frac{n}{2} + 2.2\frac{n}{4} + \ldots \end{array}}_{\log_2 n}$$

$$f(n) = O(n \log n)$$

Division

Fusion facile !

Partition équilibrée

$O(n)$

Division

Fusion facile !

Partition par une droite

Partition équilibrée

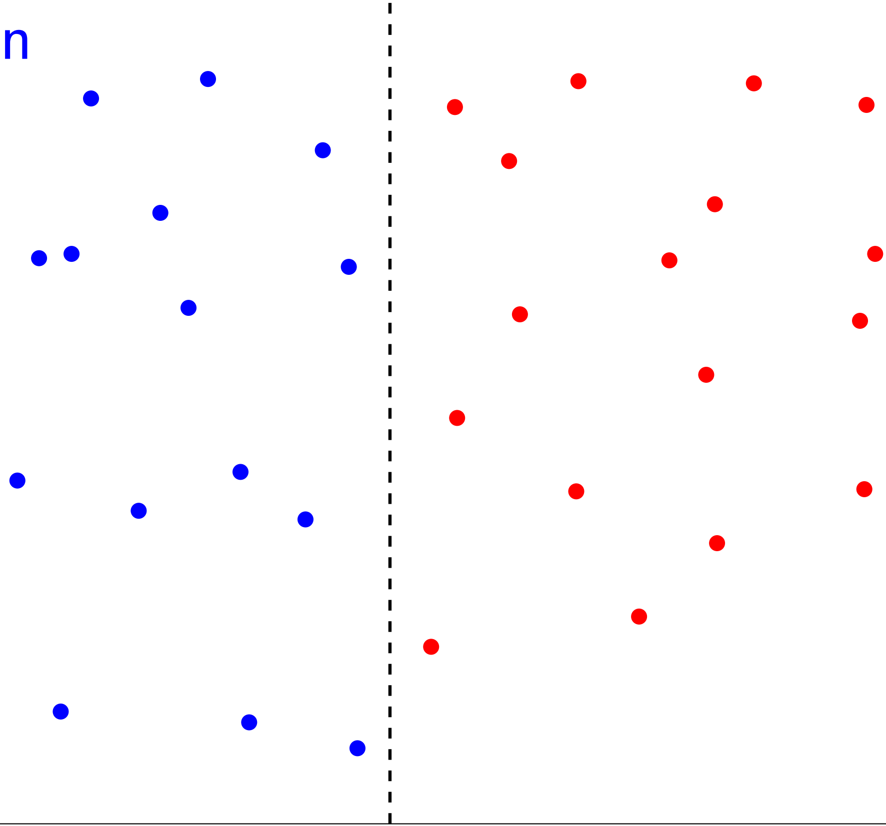
Droite médiane

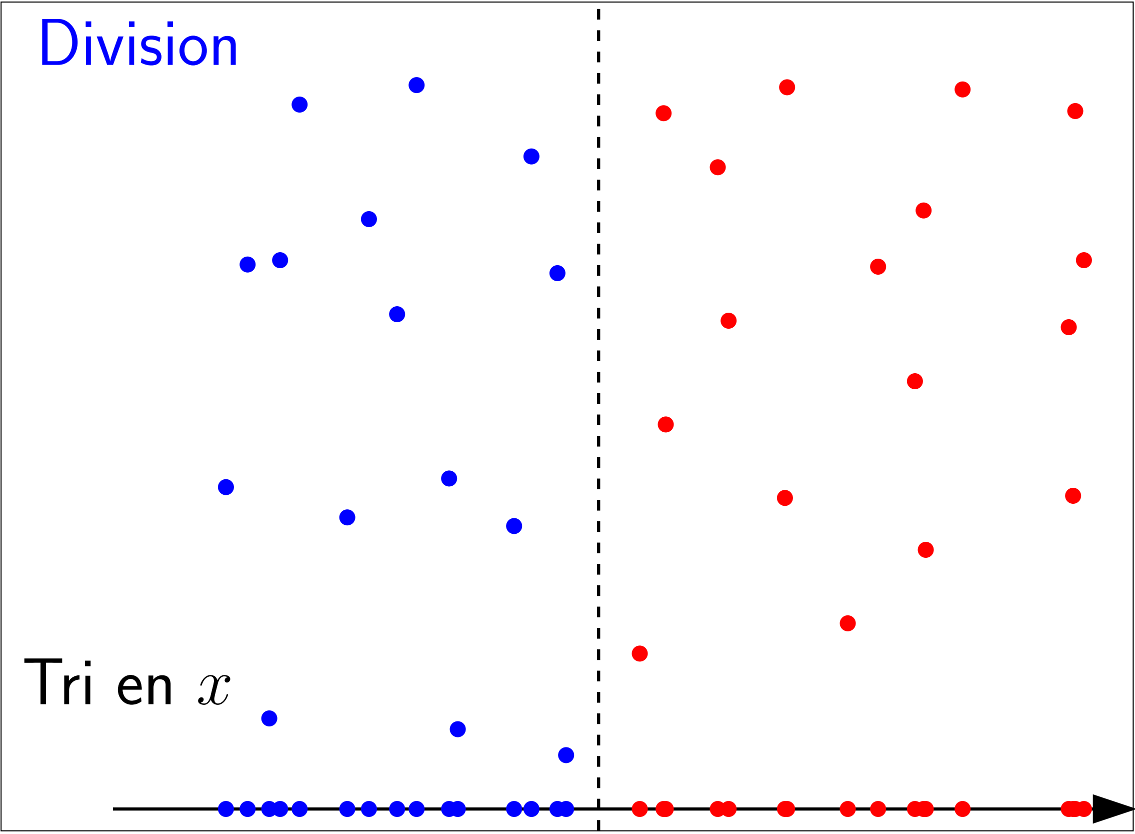
$O(n)$

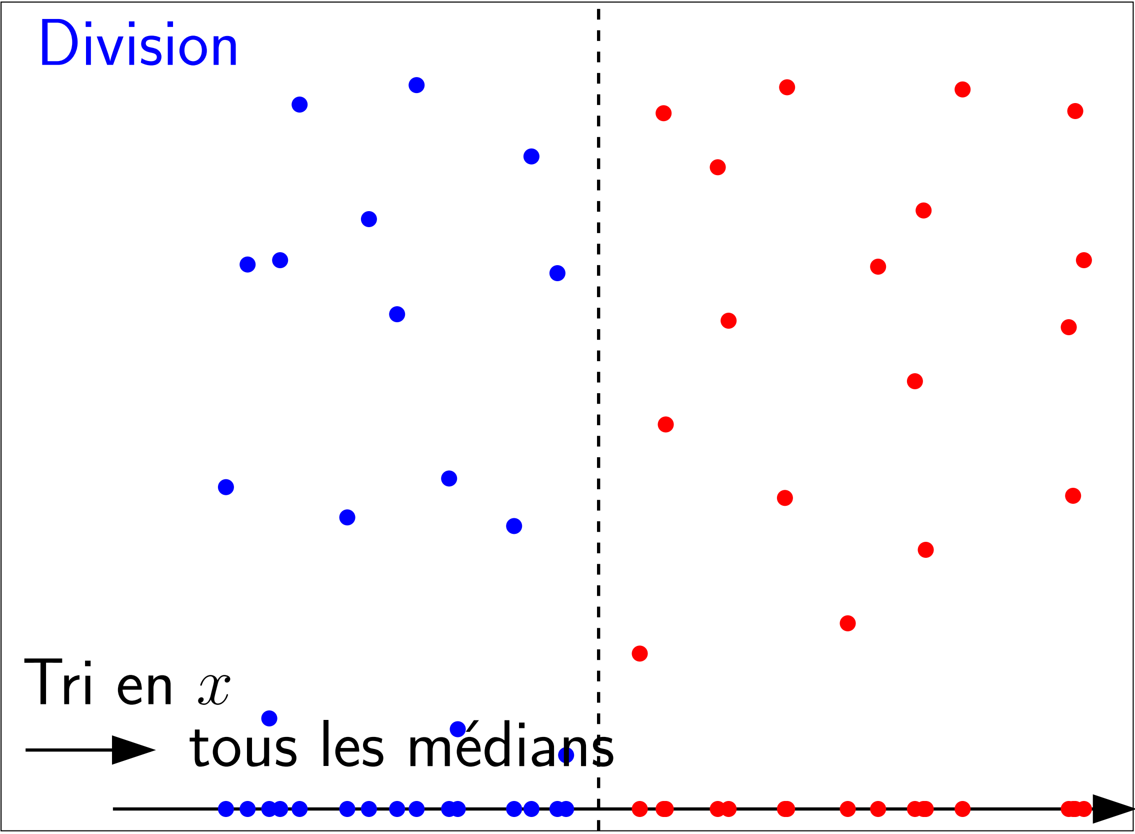
Médian linéaire ?

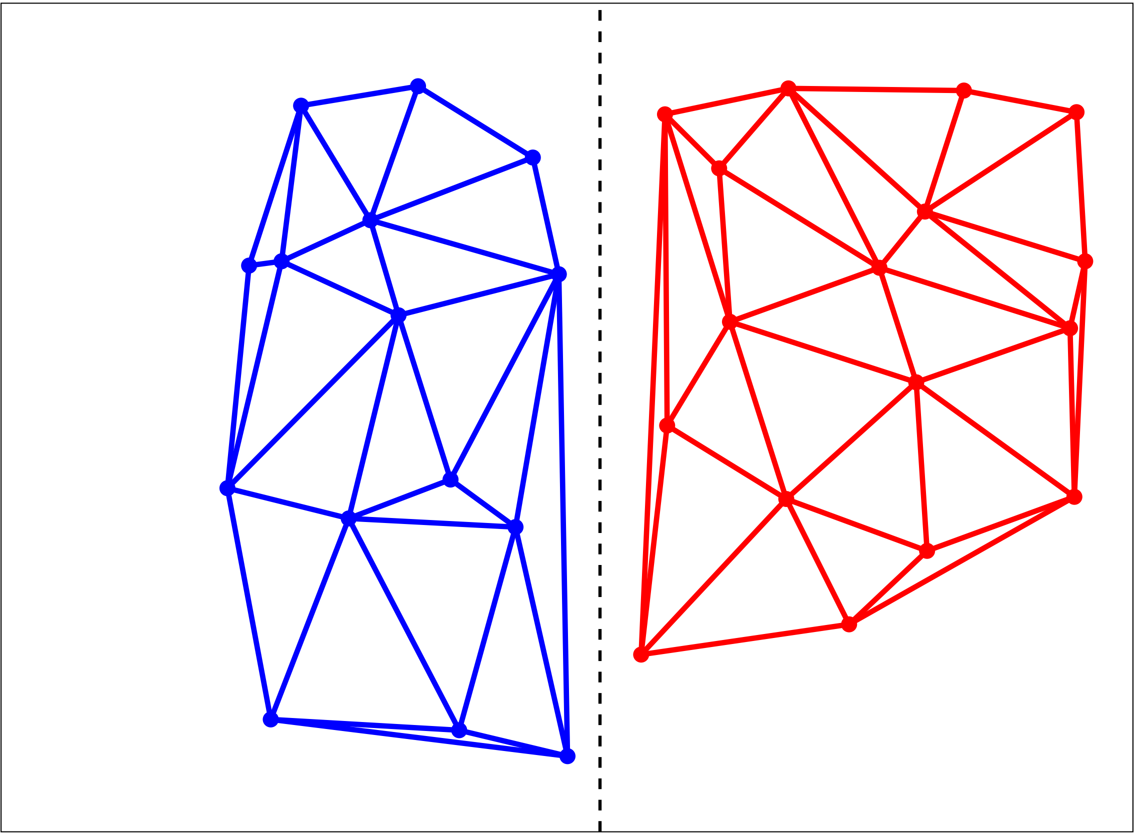
Prétraitement

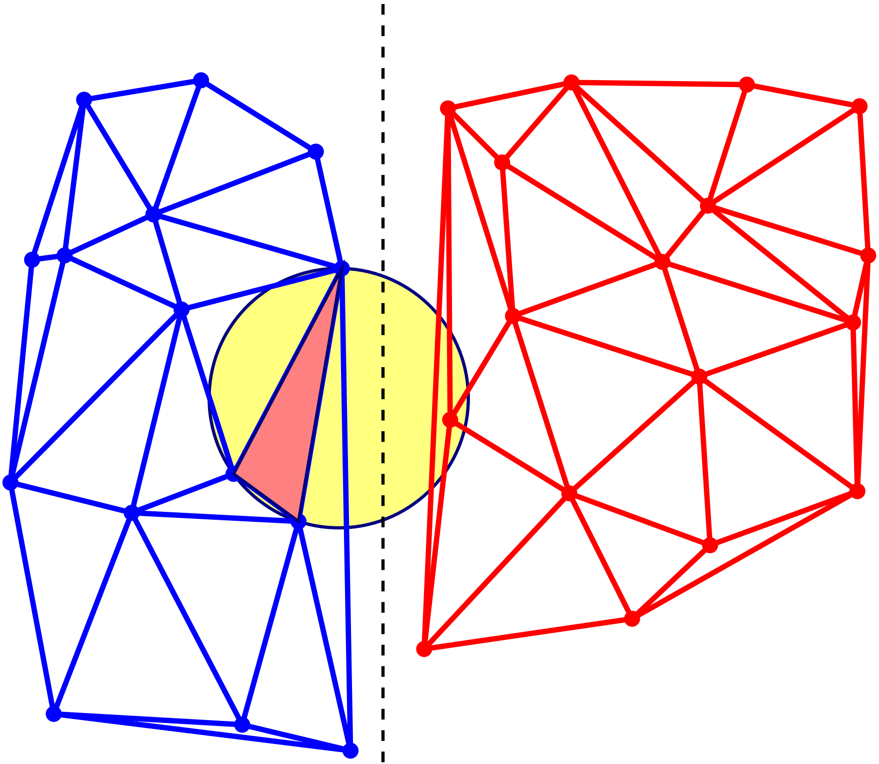
Division



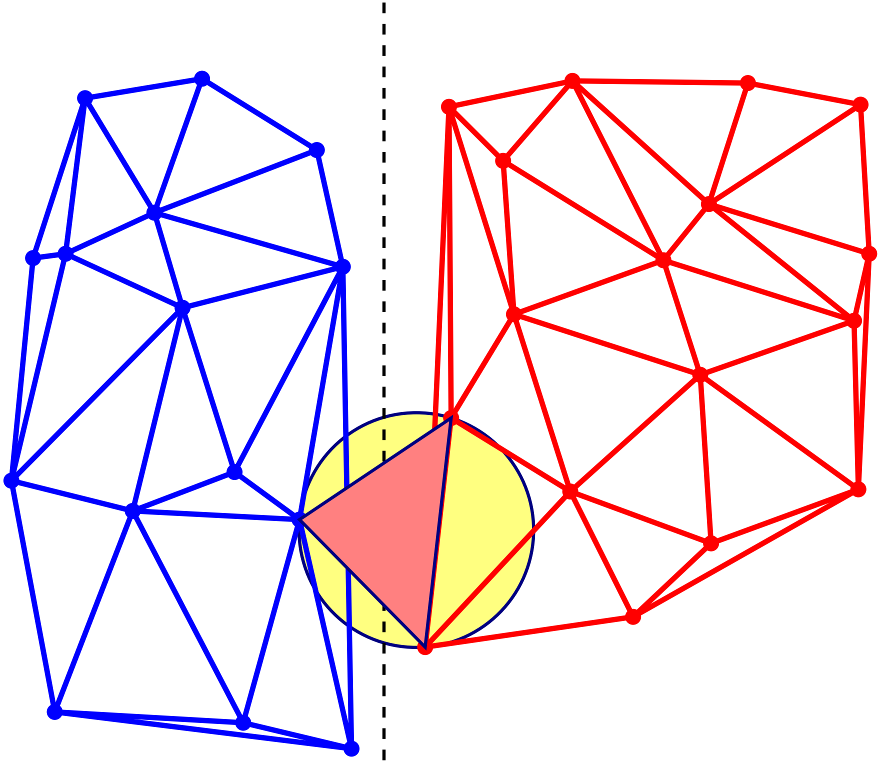

Division
Tri en x


Division
Tri en x
tous les médians

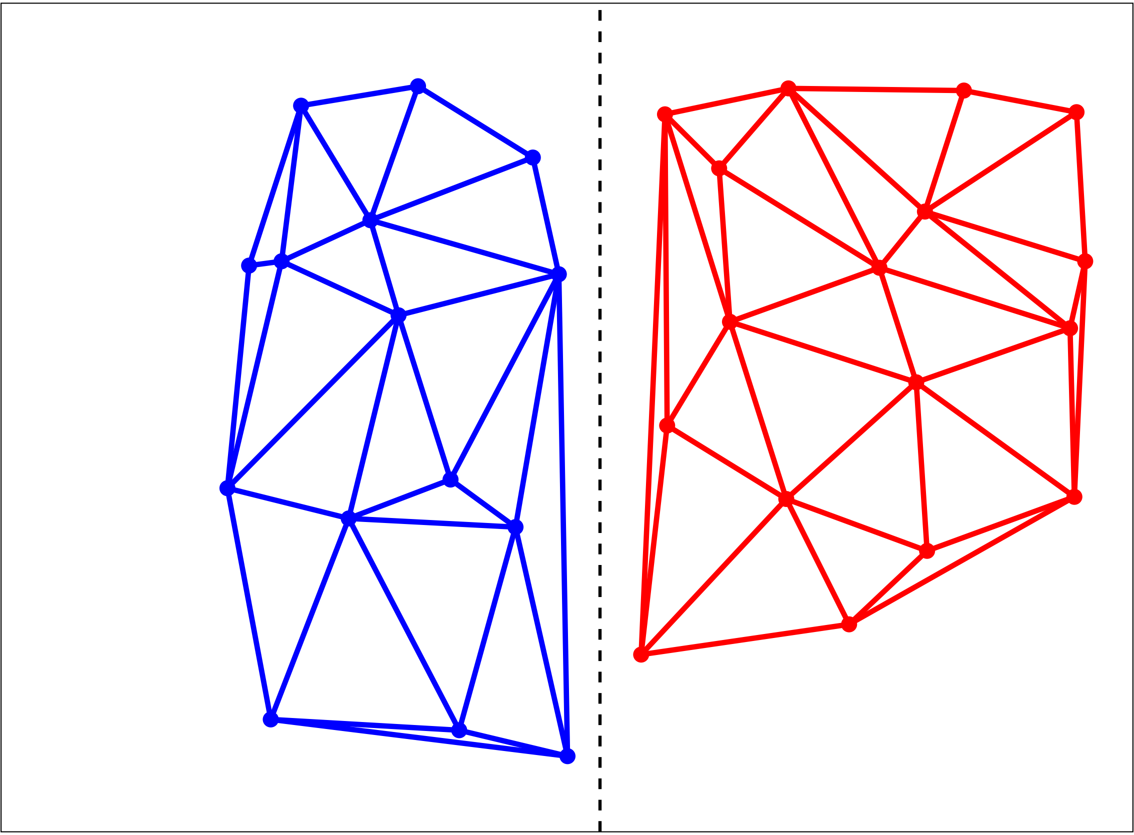




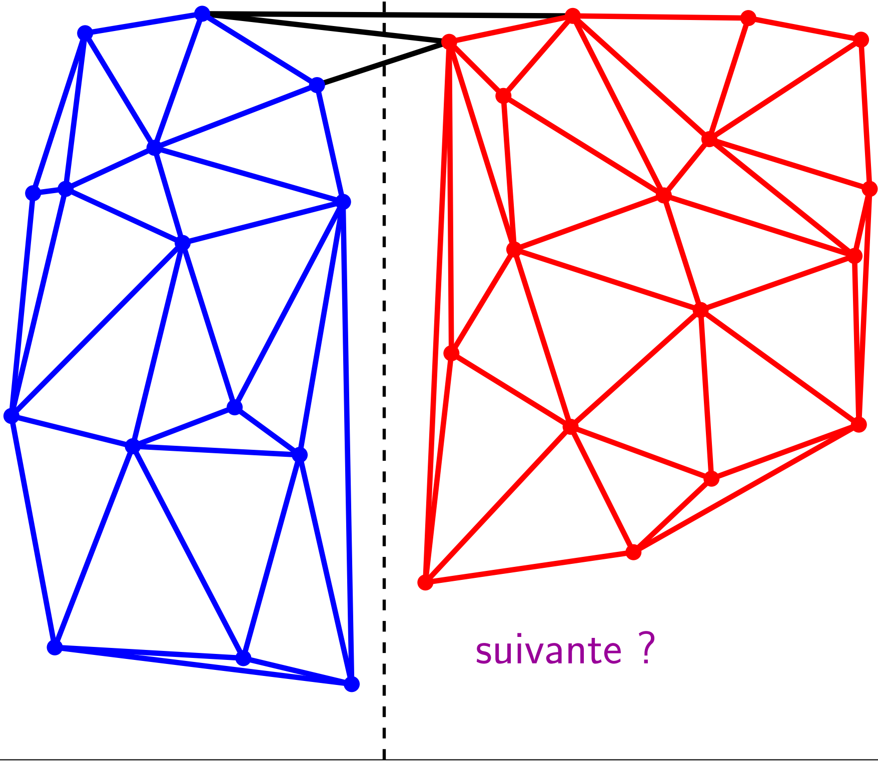
Triangles monocolores à éliminer



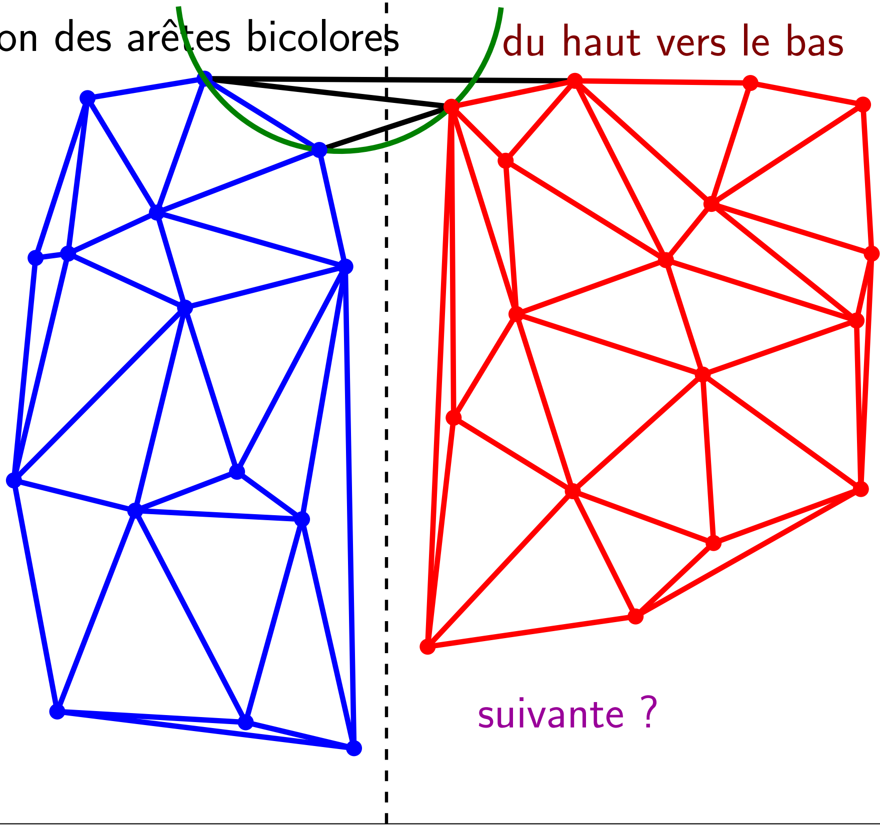
Triangles bicolores à construire



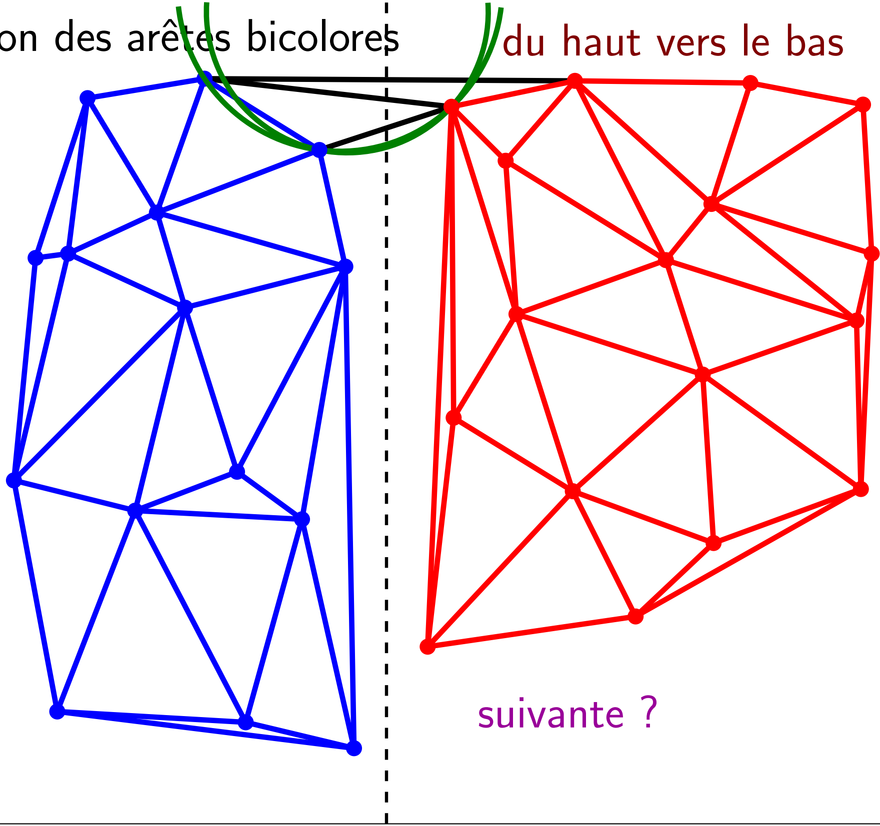
Construction des arêtes bicolores du haut vers le bas



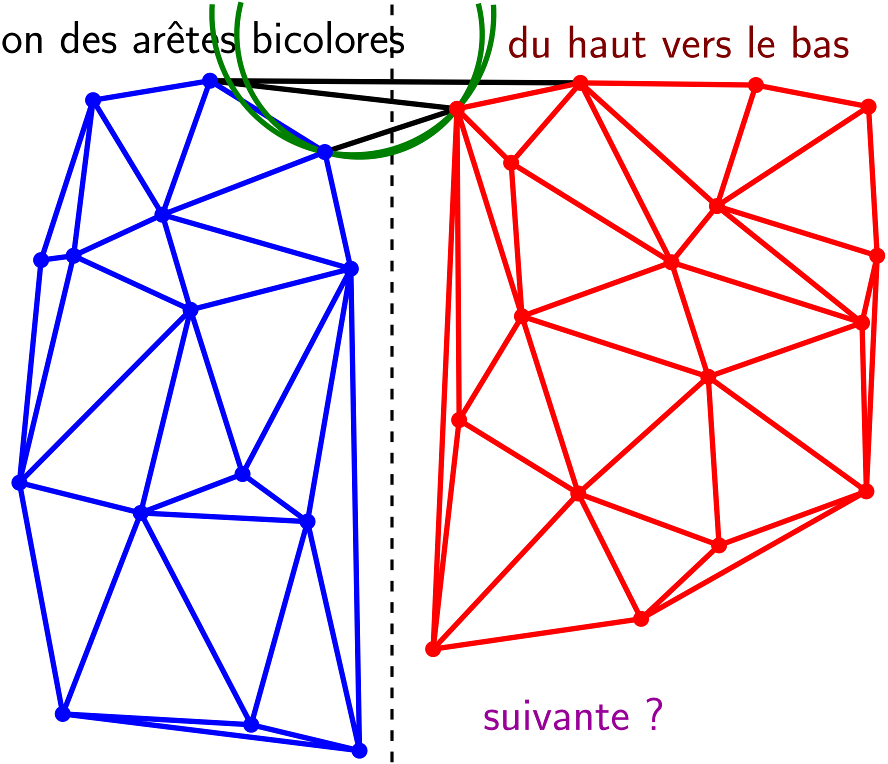
Construction des arêtes bicolores



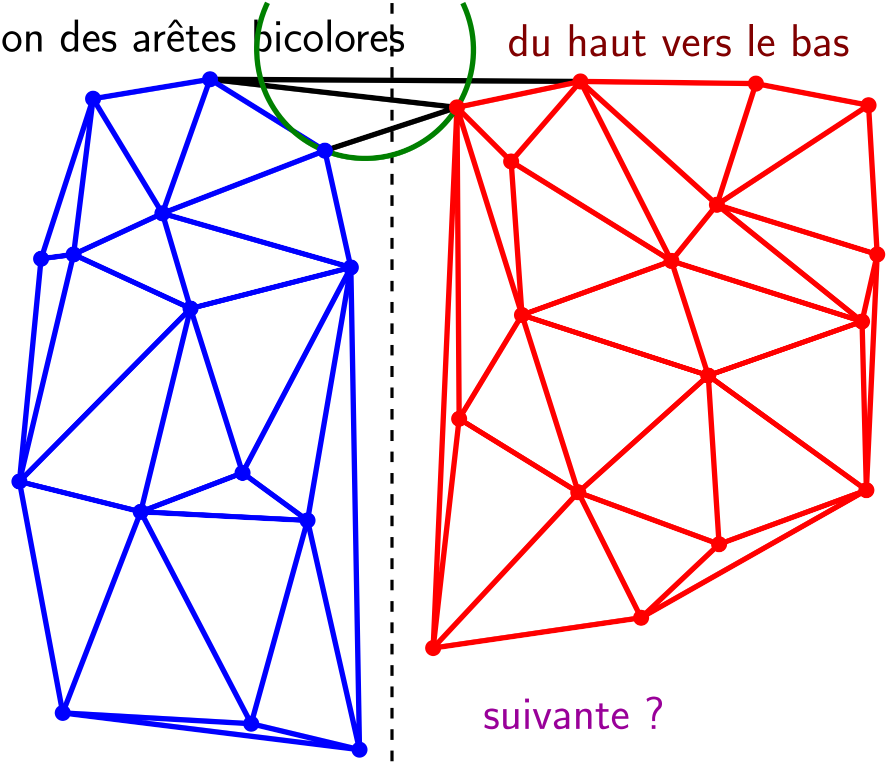
Construction des arêtes bicolores du haut vers le bas



Construction des arêtes bicolores

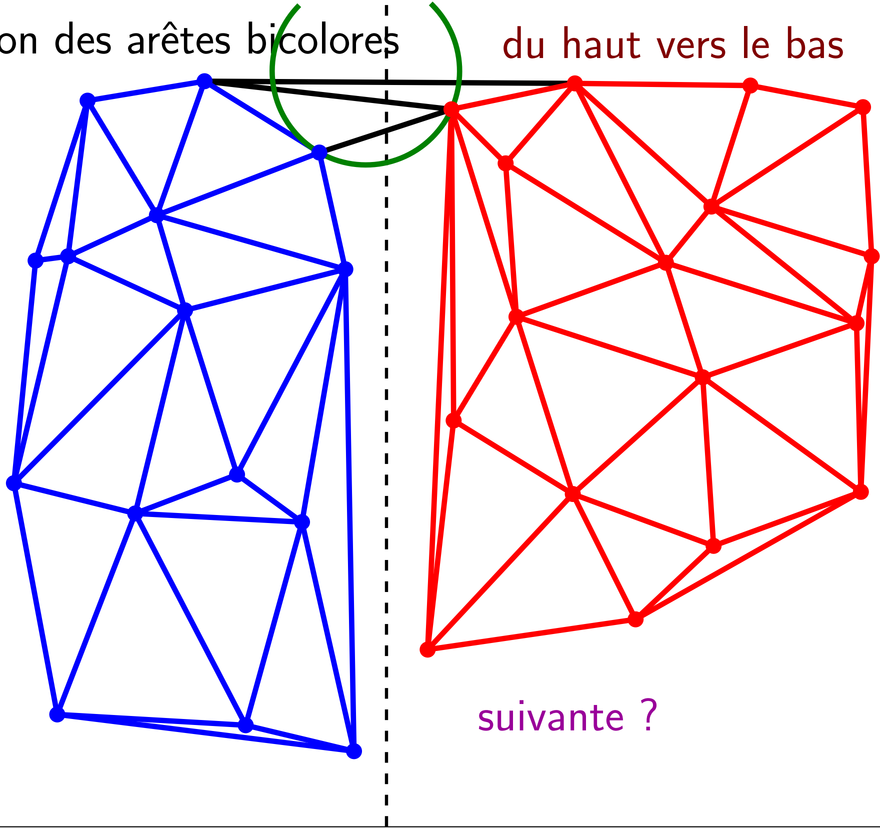


Construction des arêtes bicolores du haut vers le bas



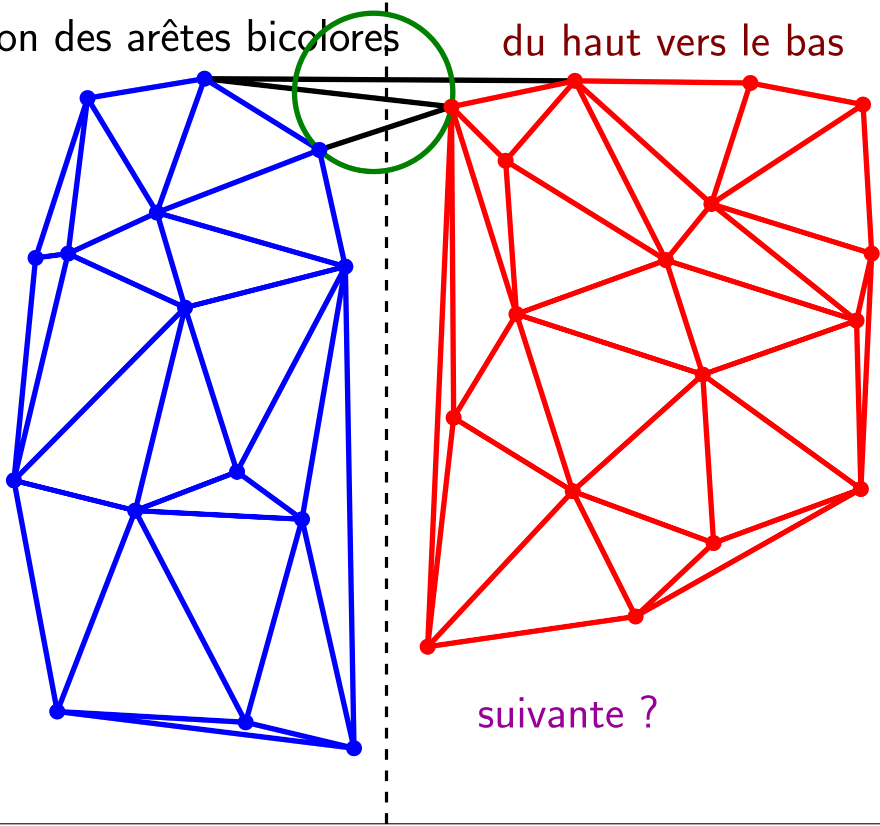
Construction des arêtes bicolores

du haut vers le bas

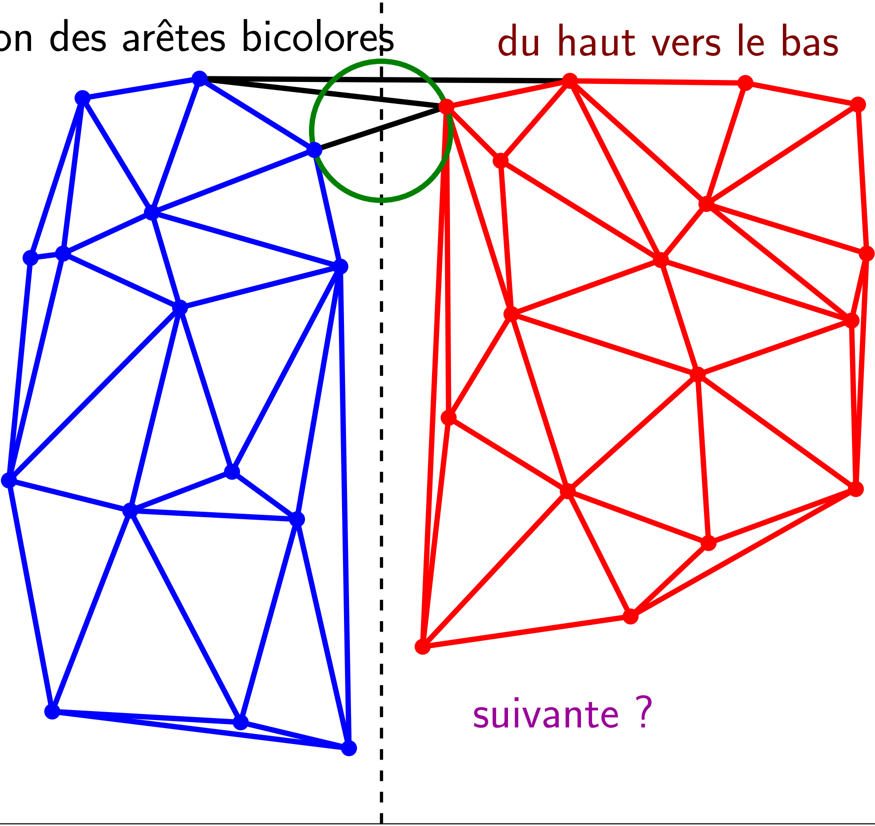


suivante ?

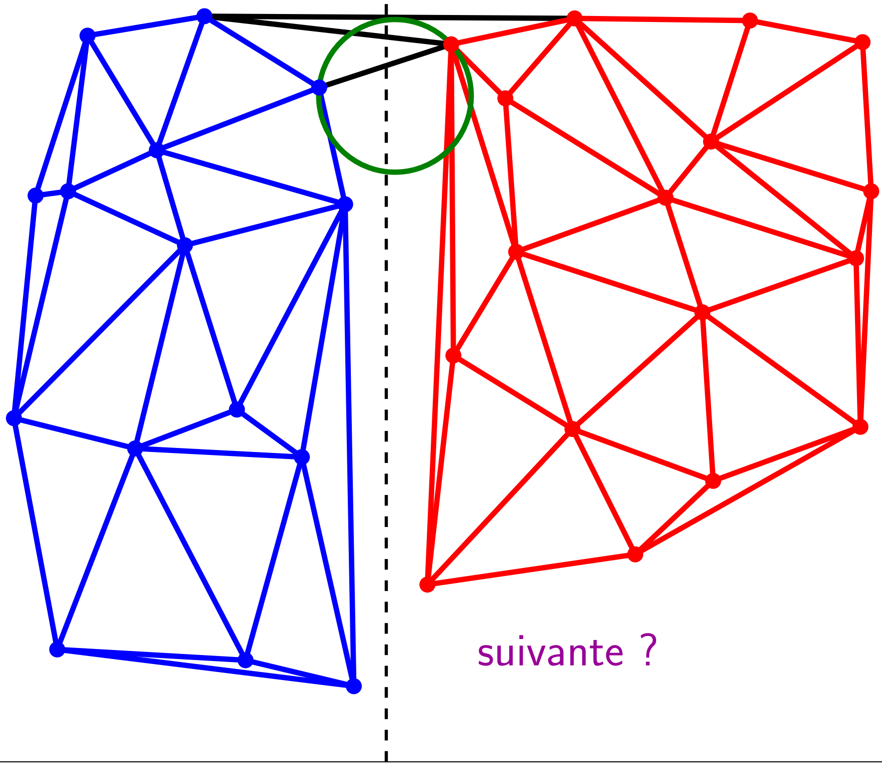
Construction des arêtes bicolores



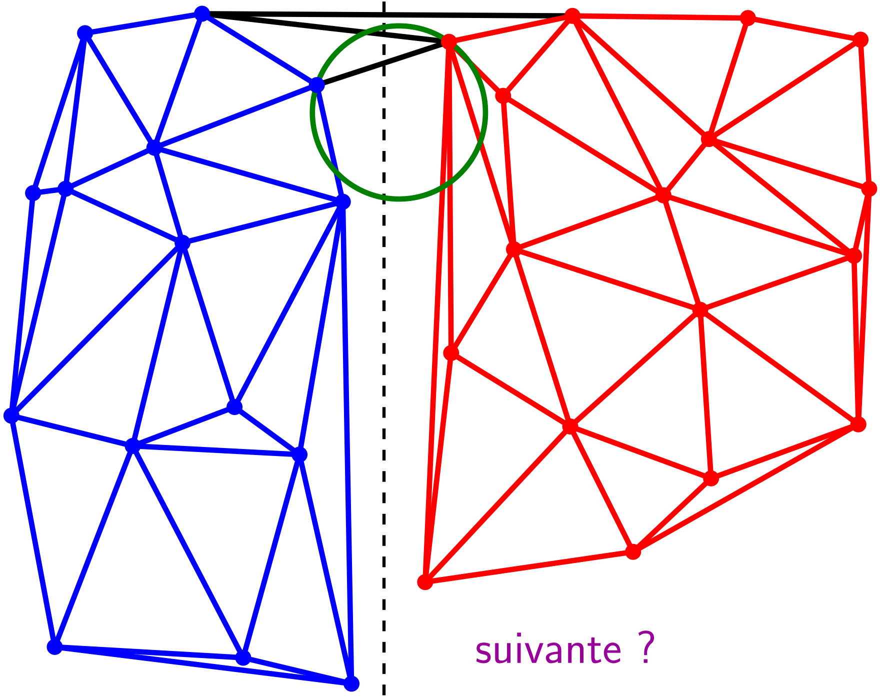
Construction des arêtes bicolores du haut vers le bas



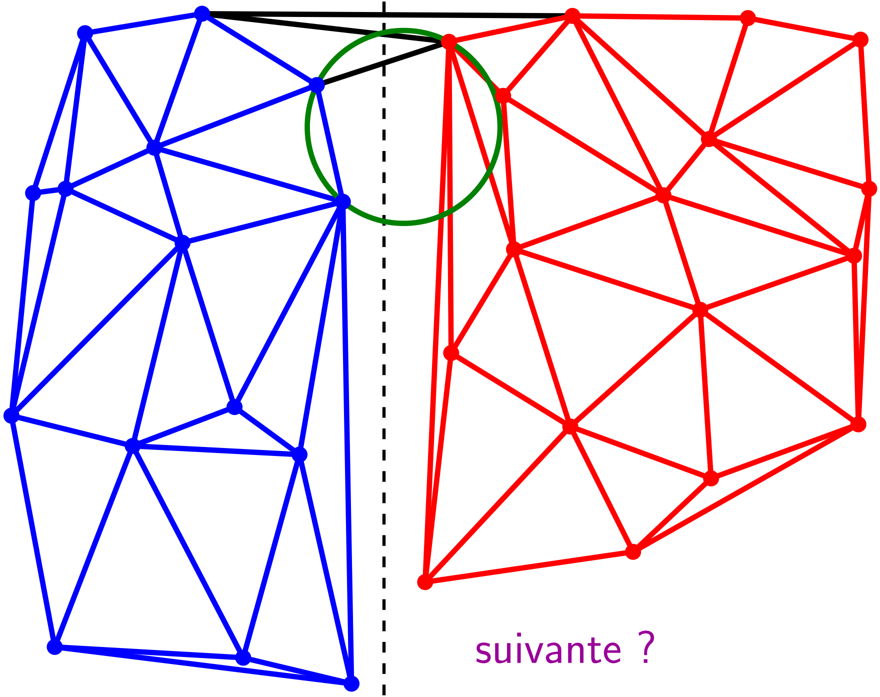
Construction des arêtes bicolores du haut vers le bas



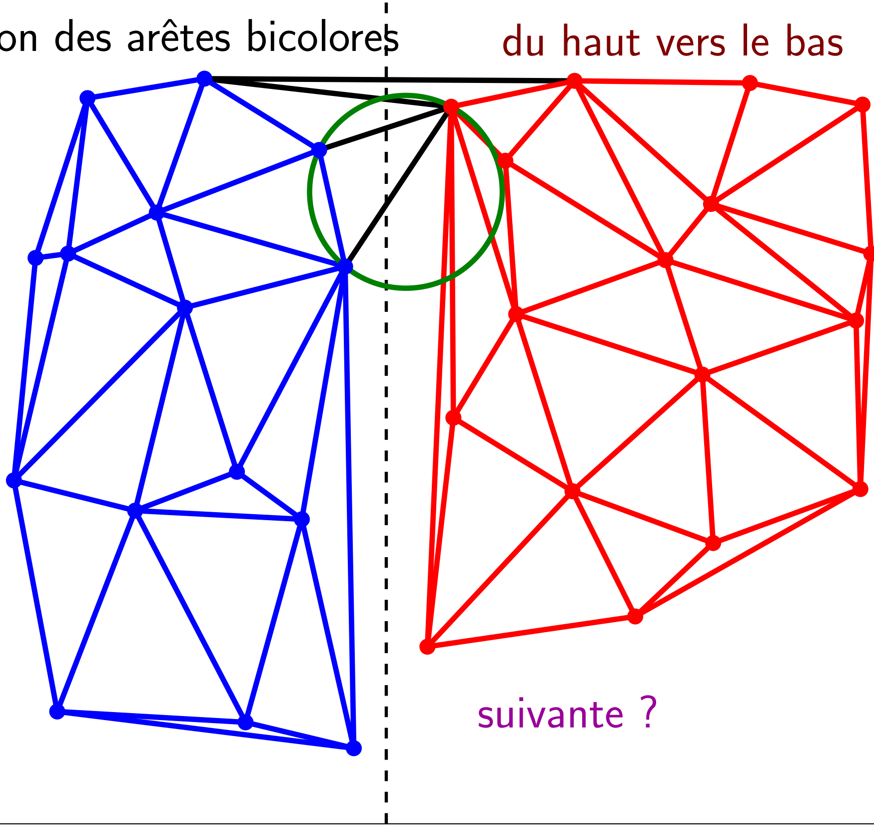
Construction des arêtes bicolores du haut vers le bas



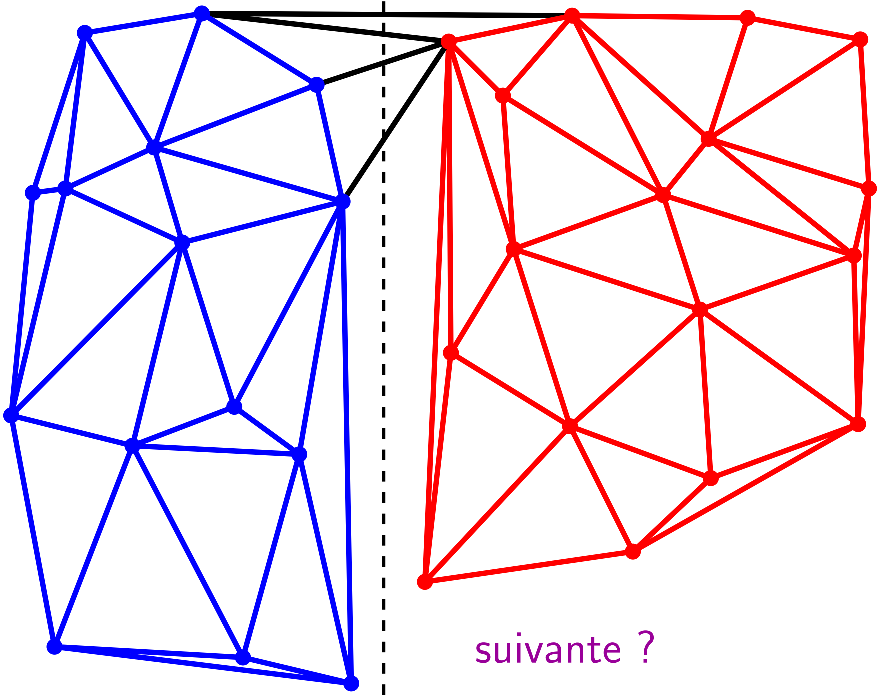
Construction des arêtes bicolores du haut vers le bas

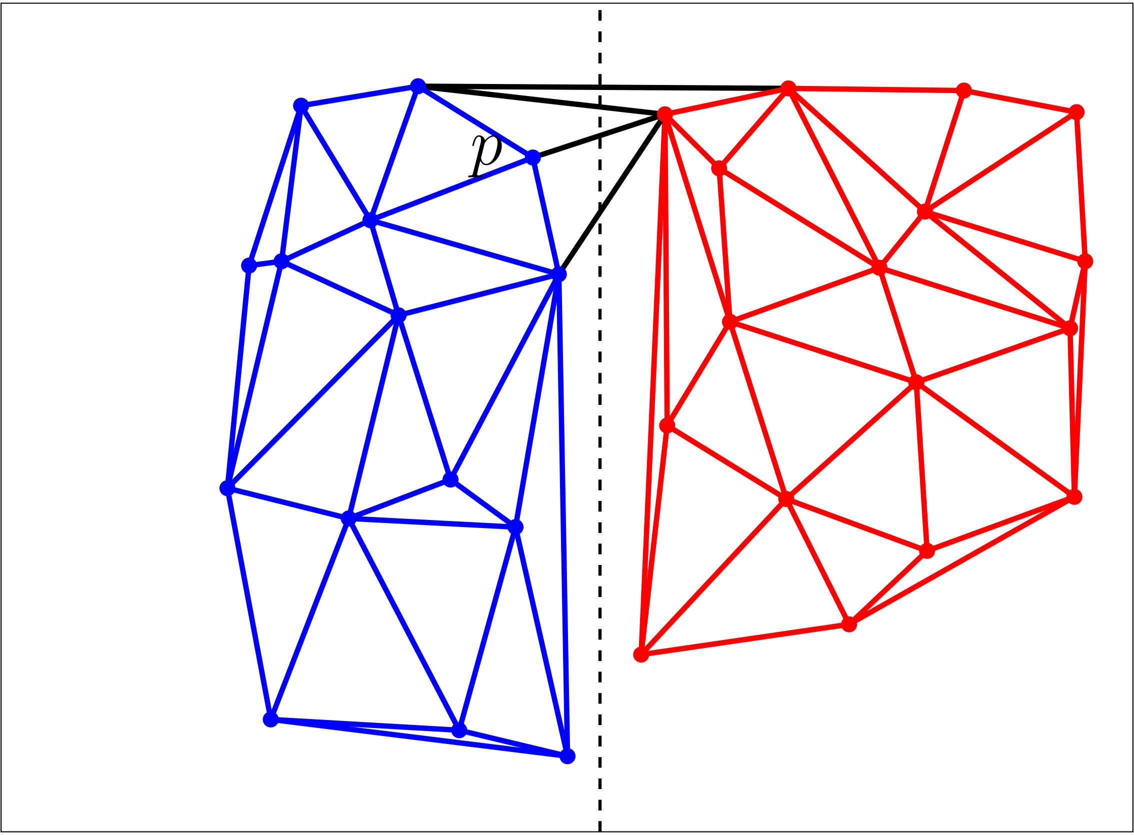


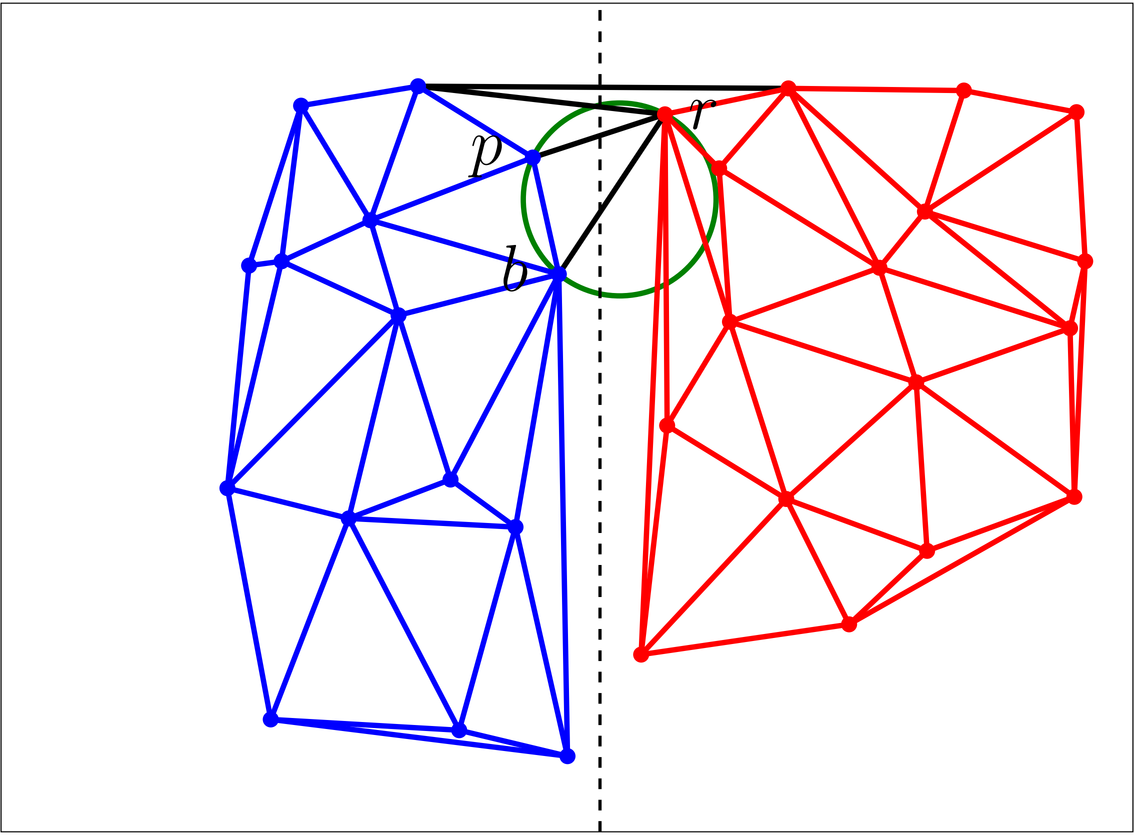
Construction des arêtes bicolores

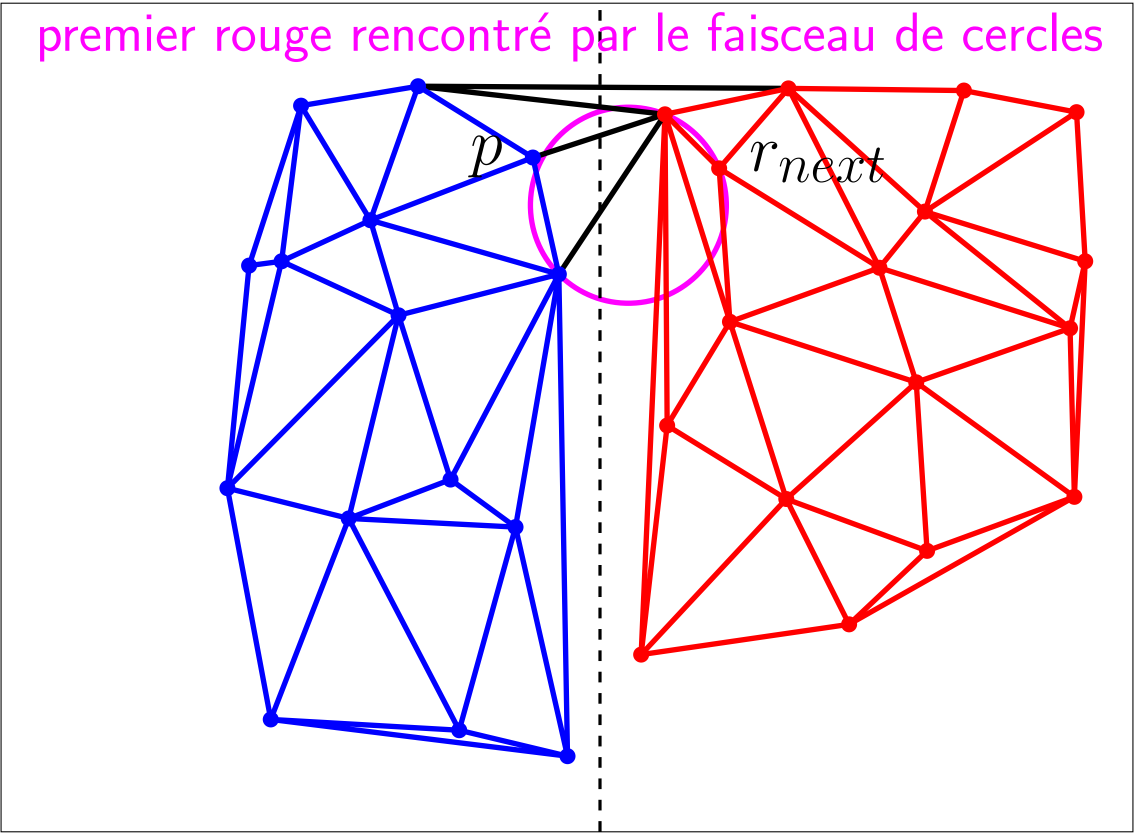


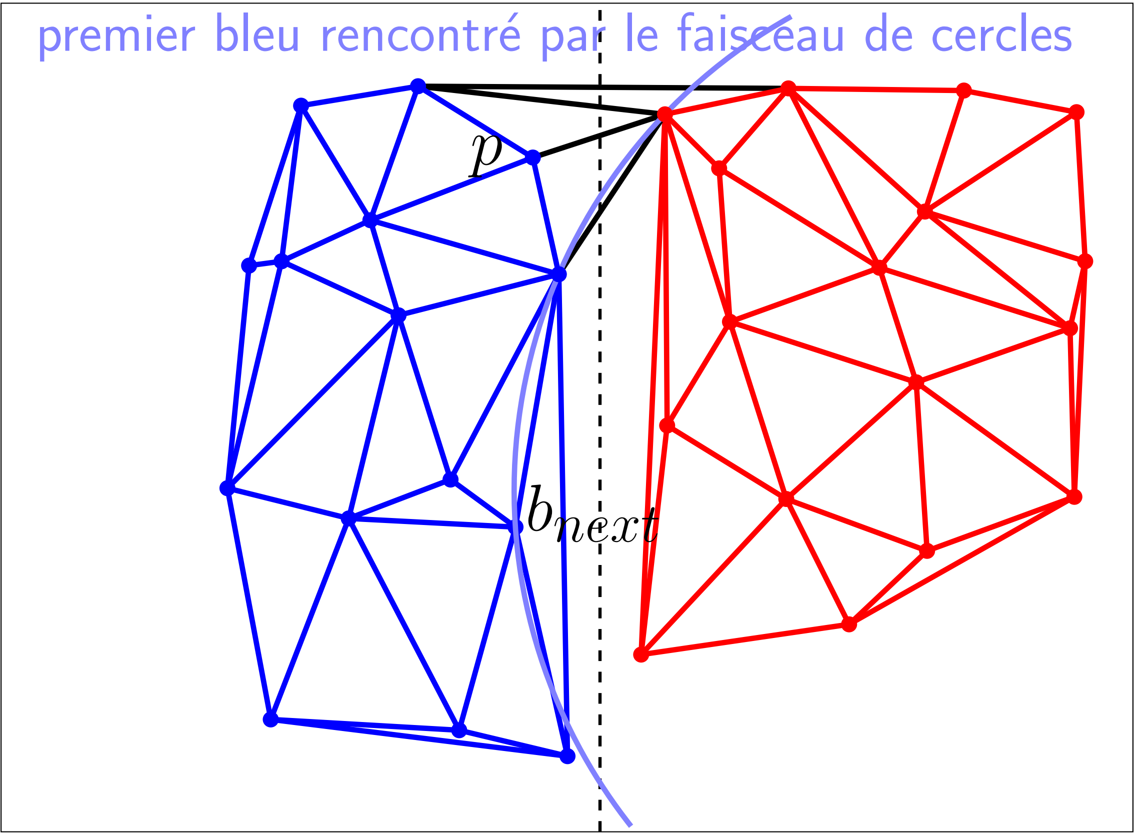
Construction des arêtes bicolores du haut vers le bas




p

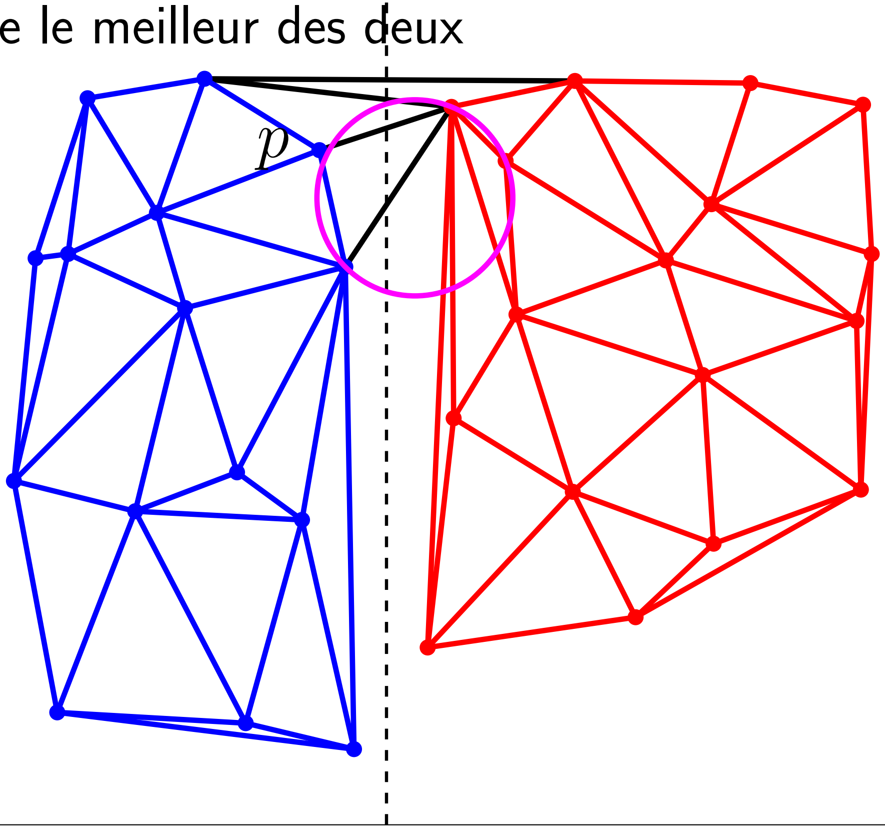

p
b
r

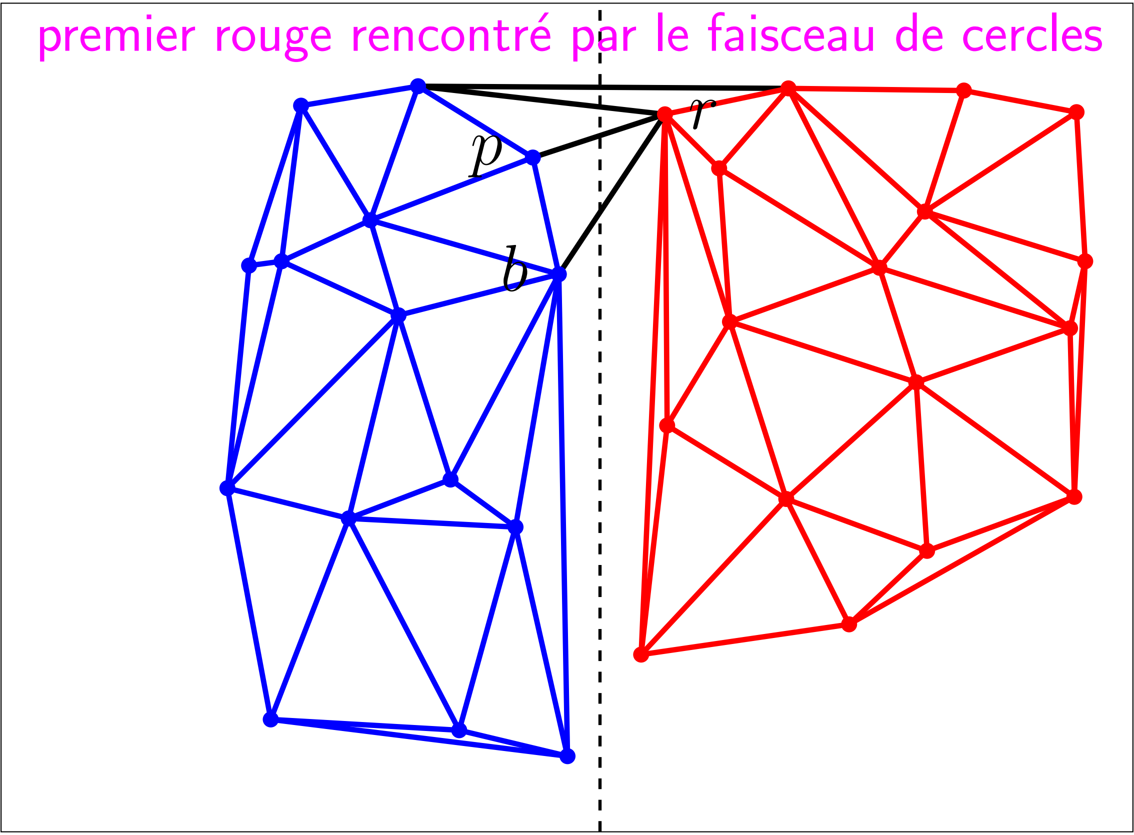

premier rouge rencontré par le faisceau de cercles
p
r_next

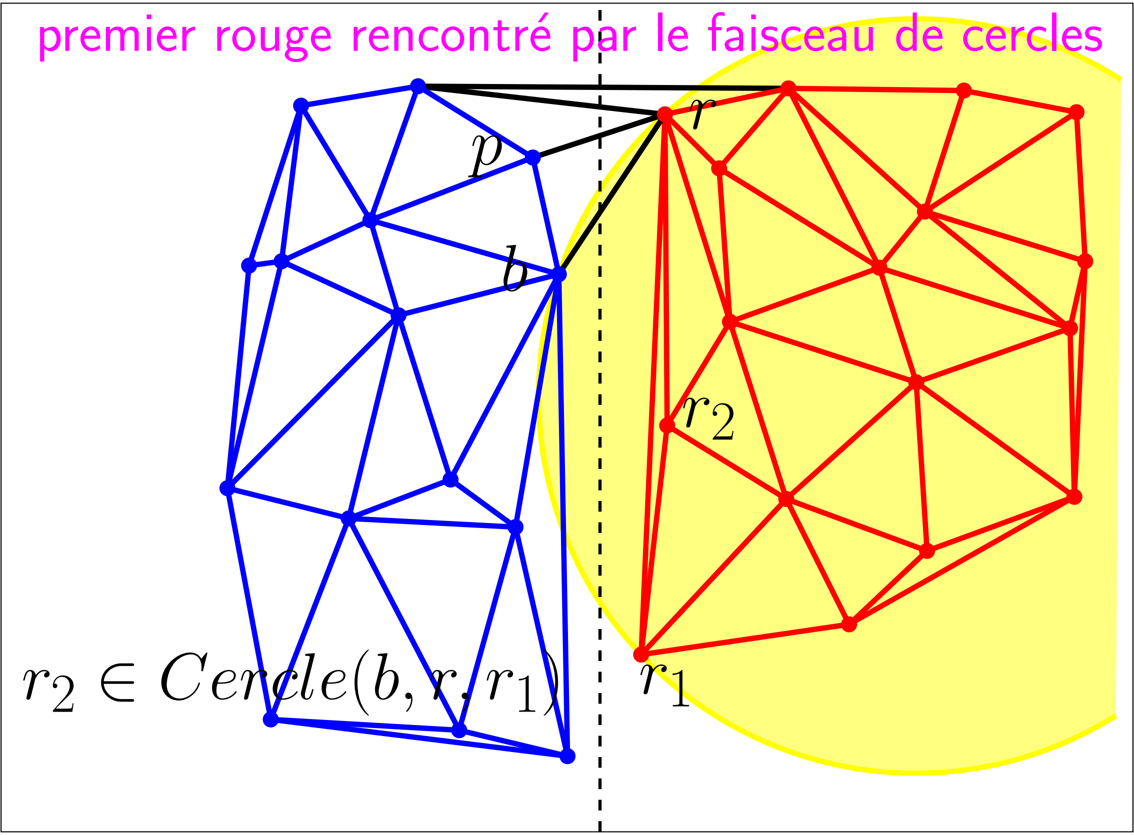


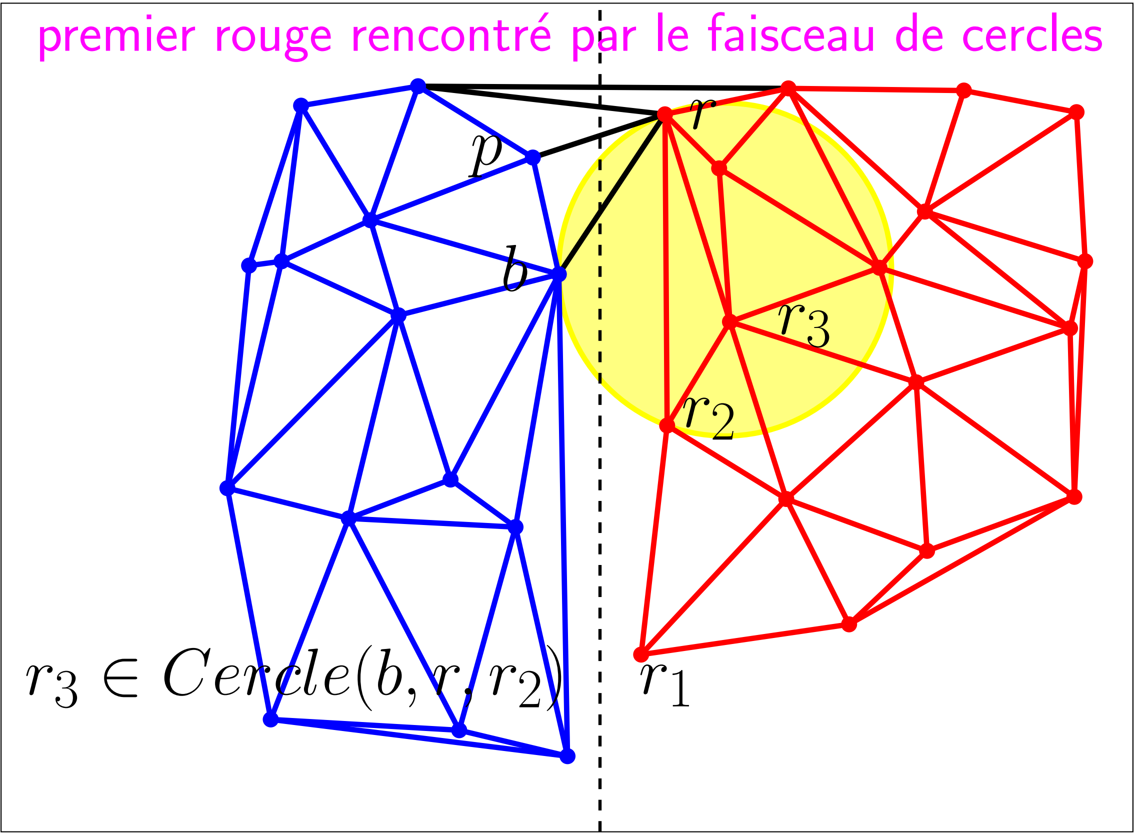
premier bleu rencontré par le faisceau de cercles
p
b_{next}

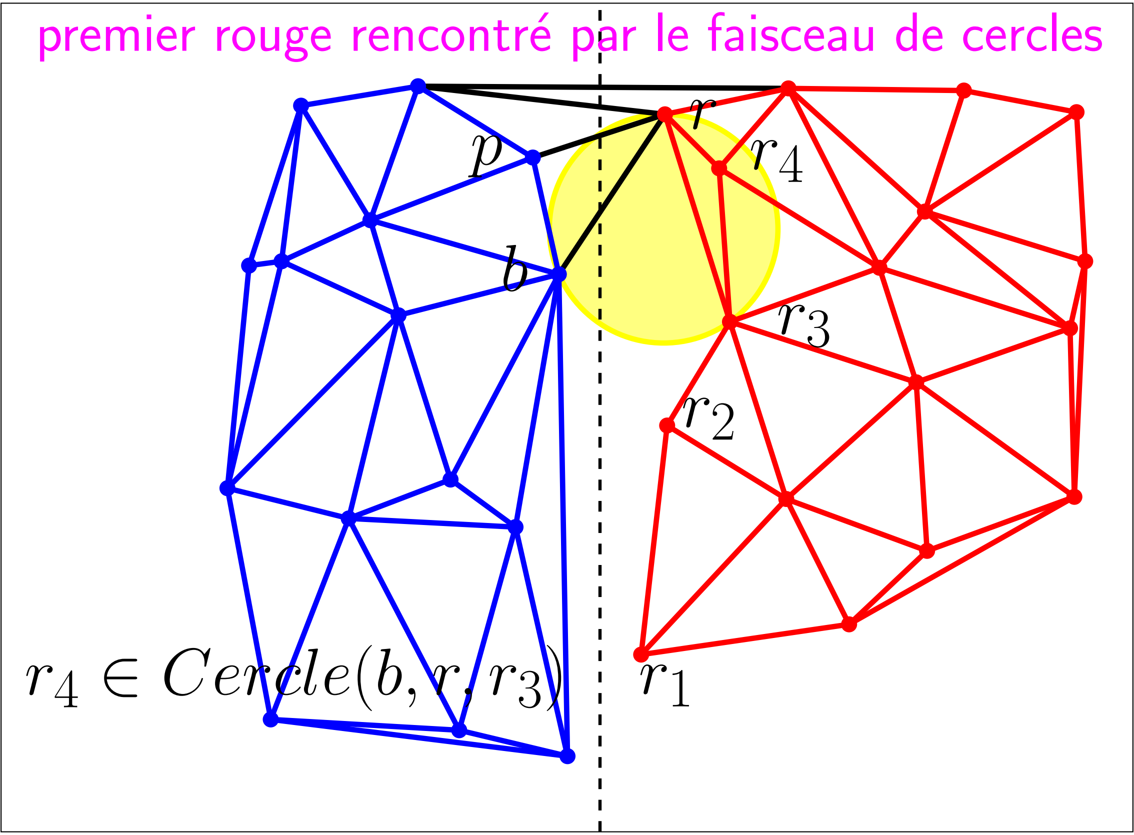
On garde le meilleur des deux

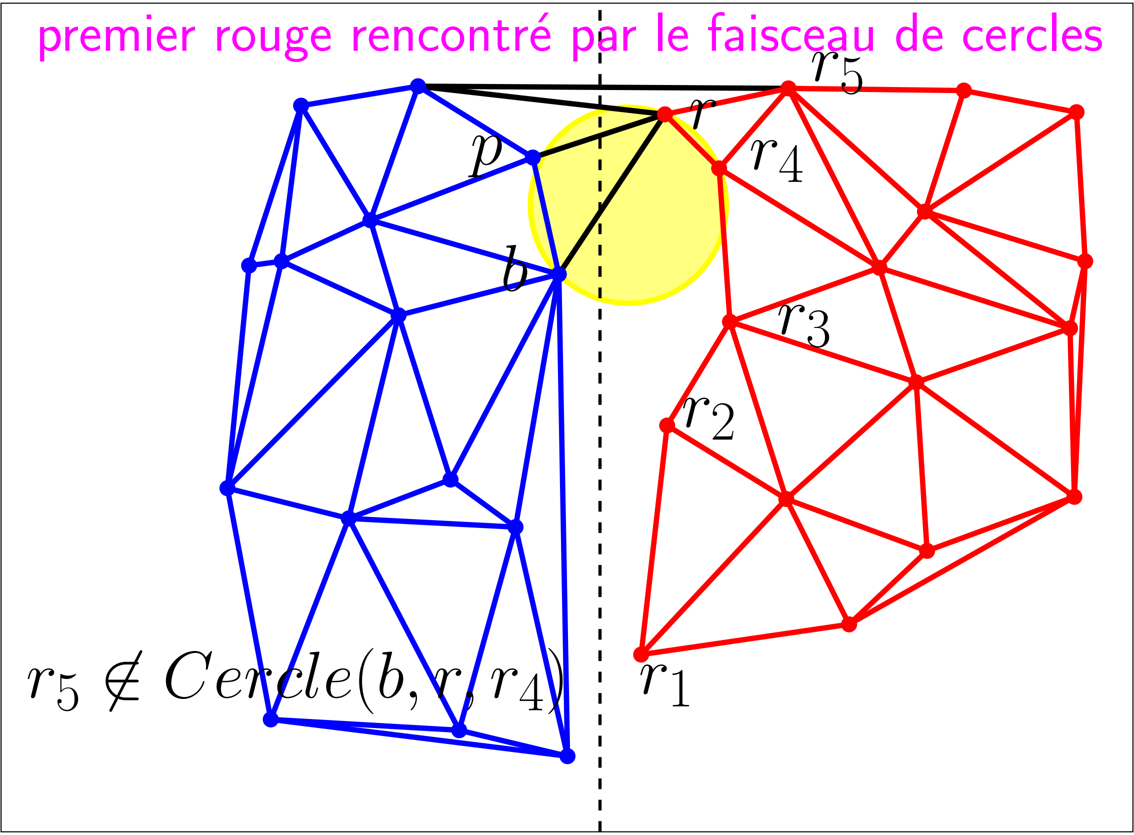



premier rouge rencontré par le faisceau de cercles
p
b
r

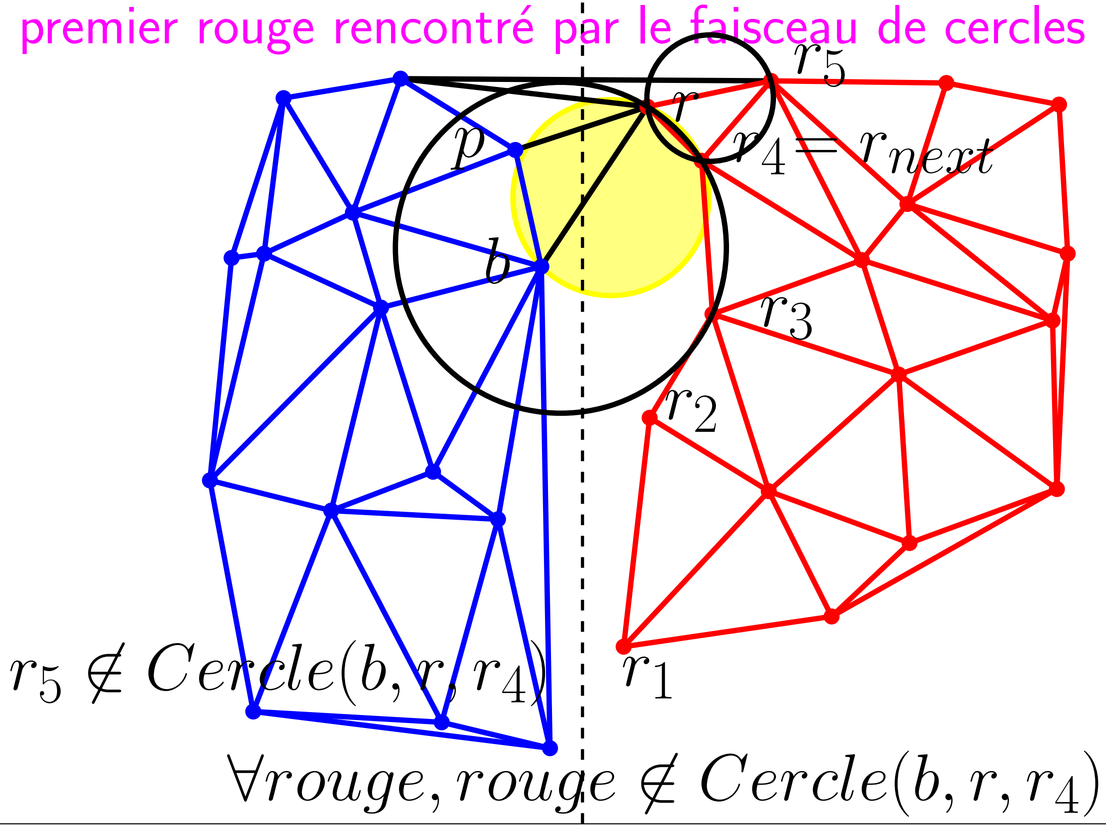

premier rouge rencontré par le faisceau de cercles
r
p
b
r_2
r_2 ∈ Cercle(b, r, r_1)
r_1


premier rouge rencontré par le faisceau de cercles
p
r
b
r3
r2
r1
r3 ∈ Cercle(b, r, r2)

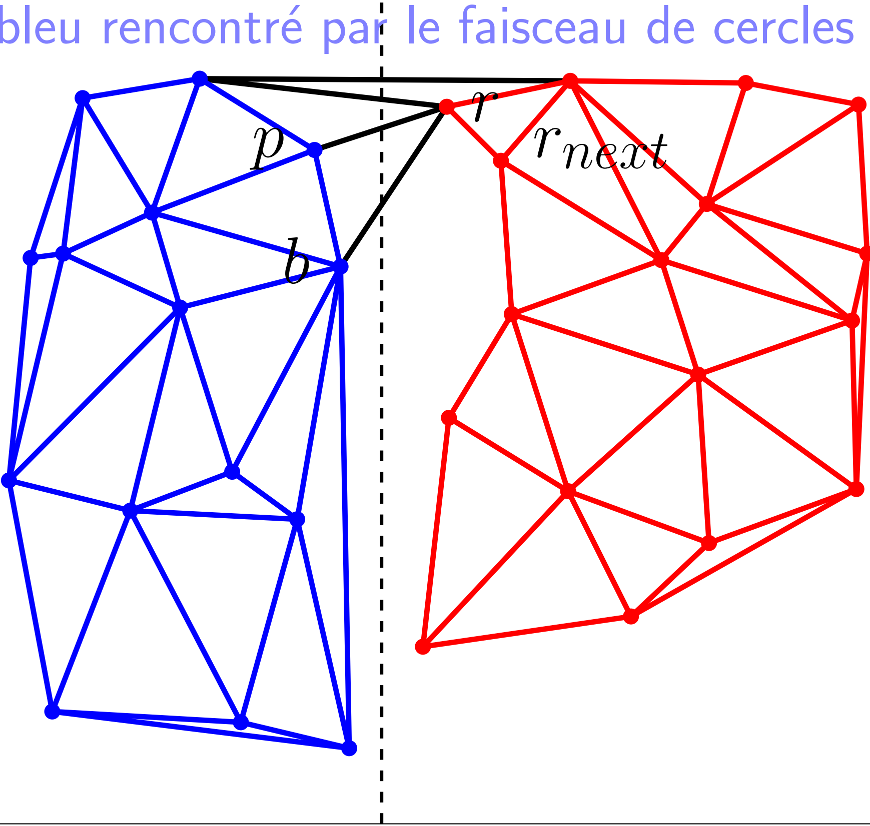

premier rouge rencontré par le faisceau de cercles
p
r
r_4
b
r_3
r_2
$r_4 \in Cercle(b, r, r_3)$
r_1

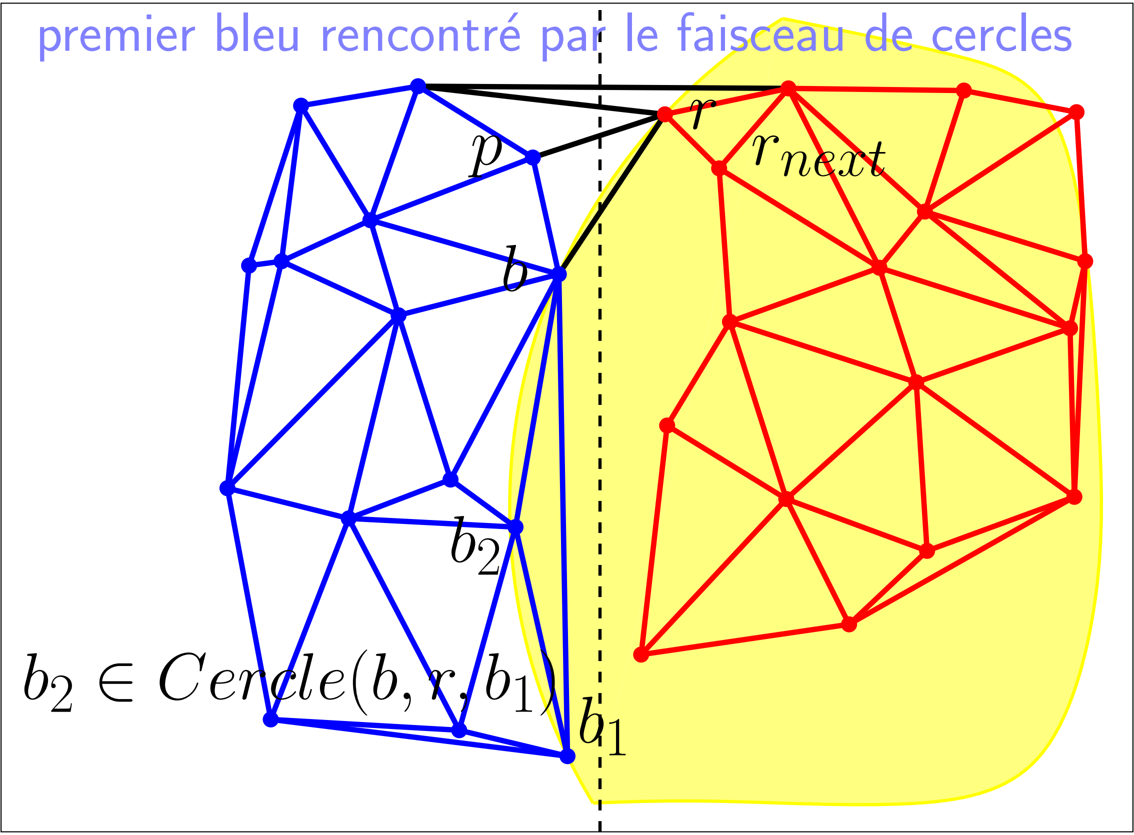

premier rouge rencontré par le faisceau de cercles
r_5
r
p
r_4
b
r_3
r_2
$r_5 \notin Cercle(b, r, r_4)$
r_1

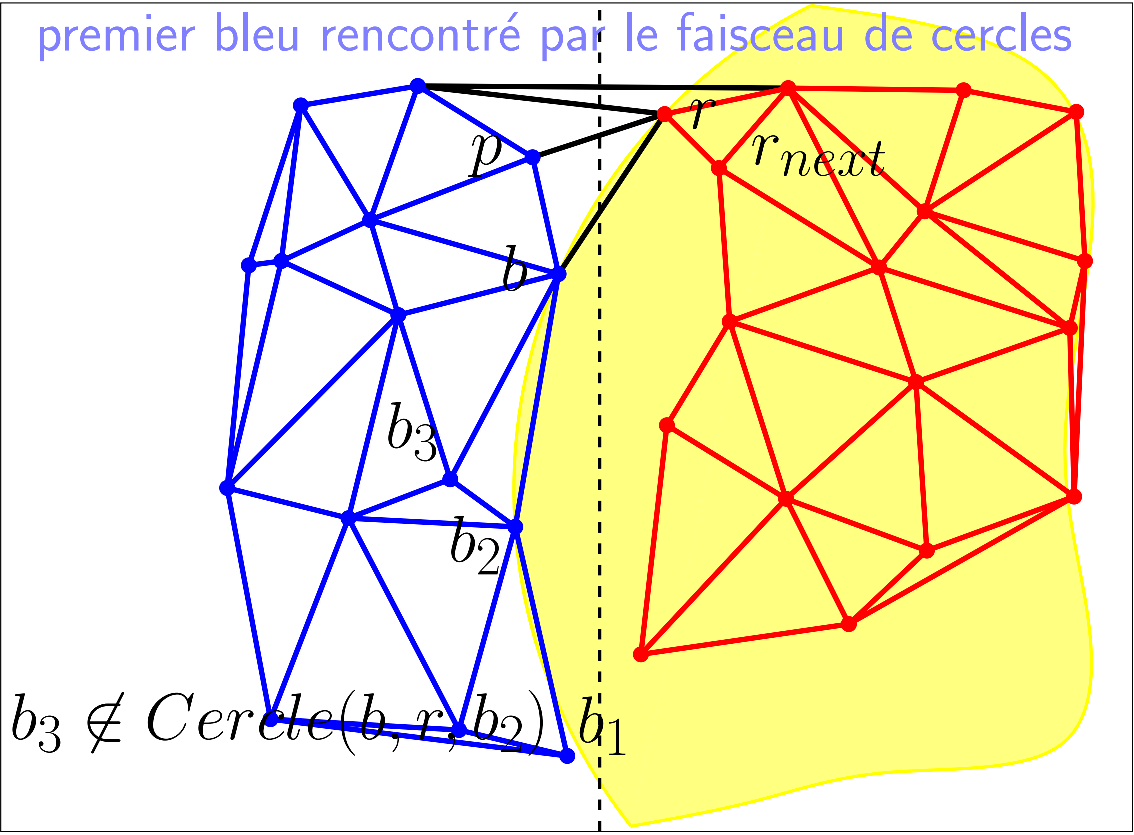
premier rouge rencontré par le faisceau de cercles

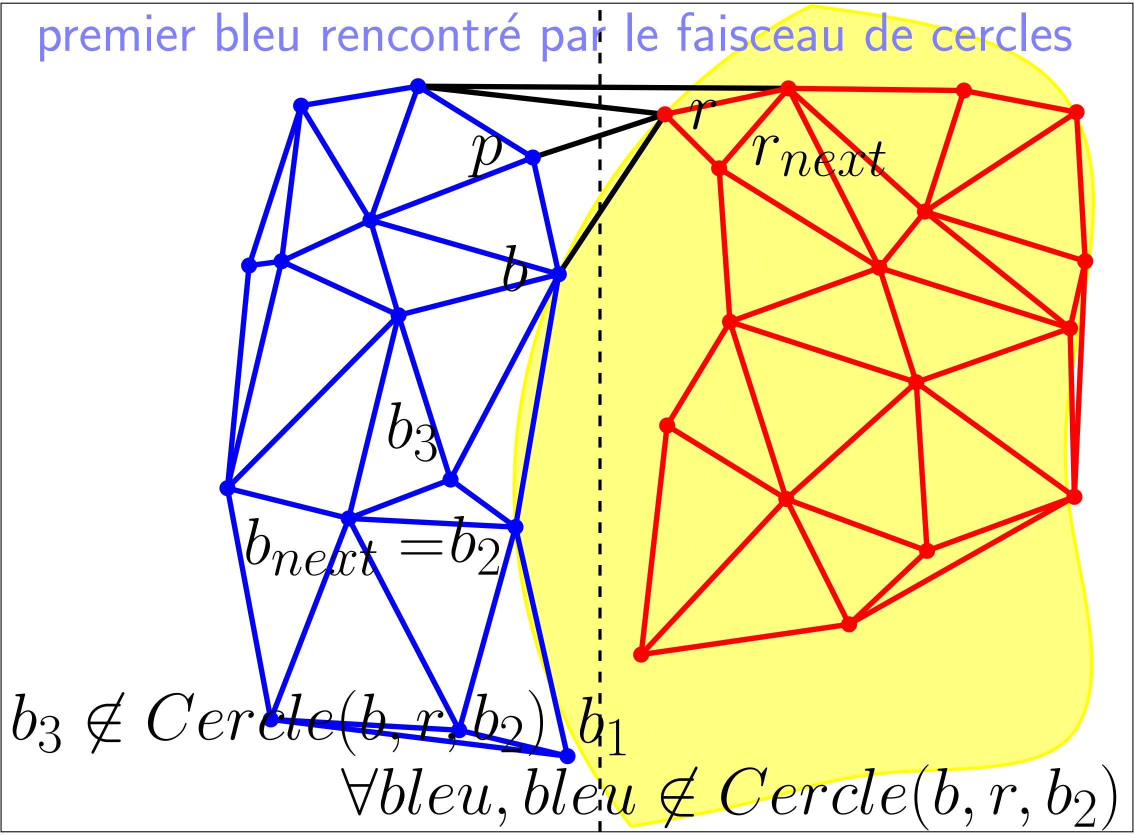


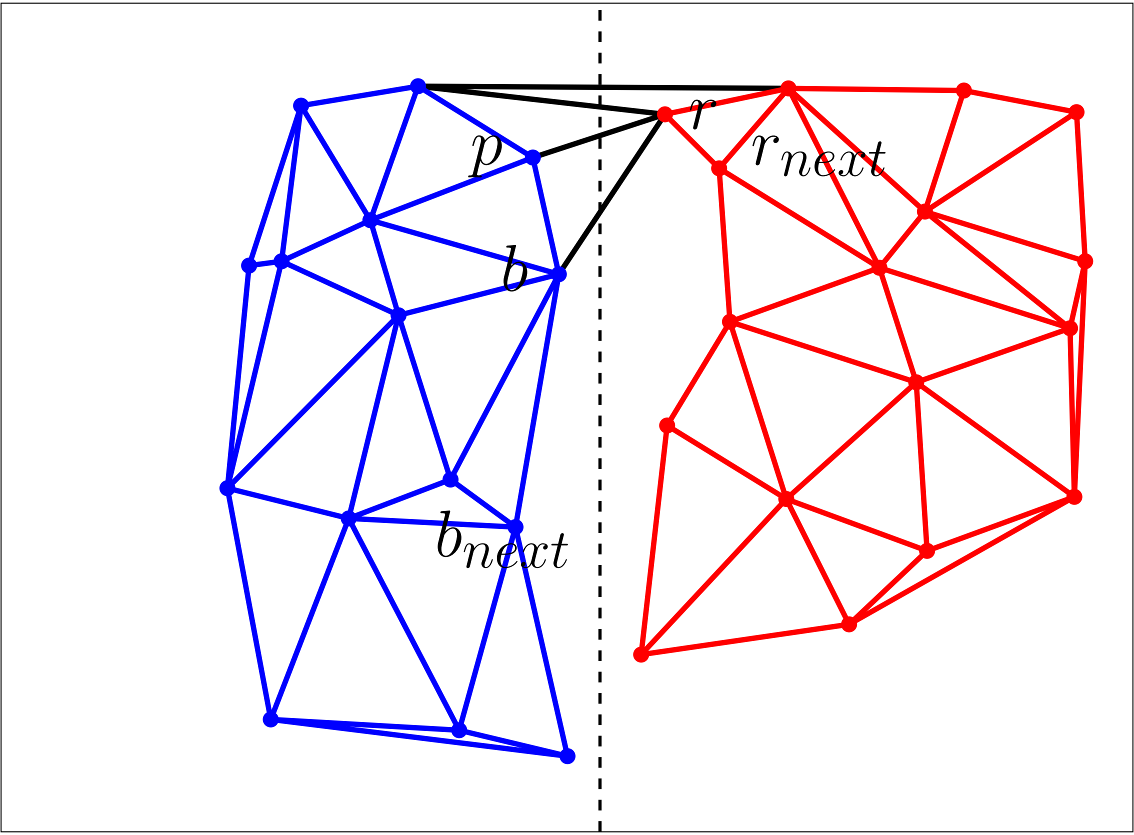
premier bleu rencontré par le faisceau de cercles

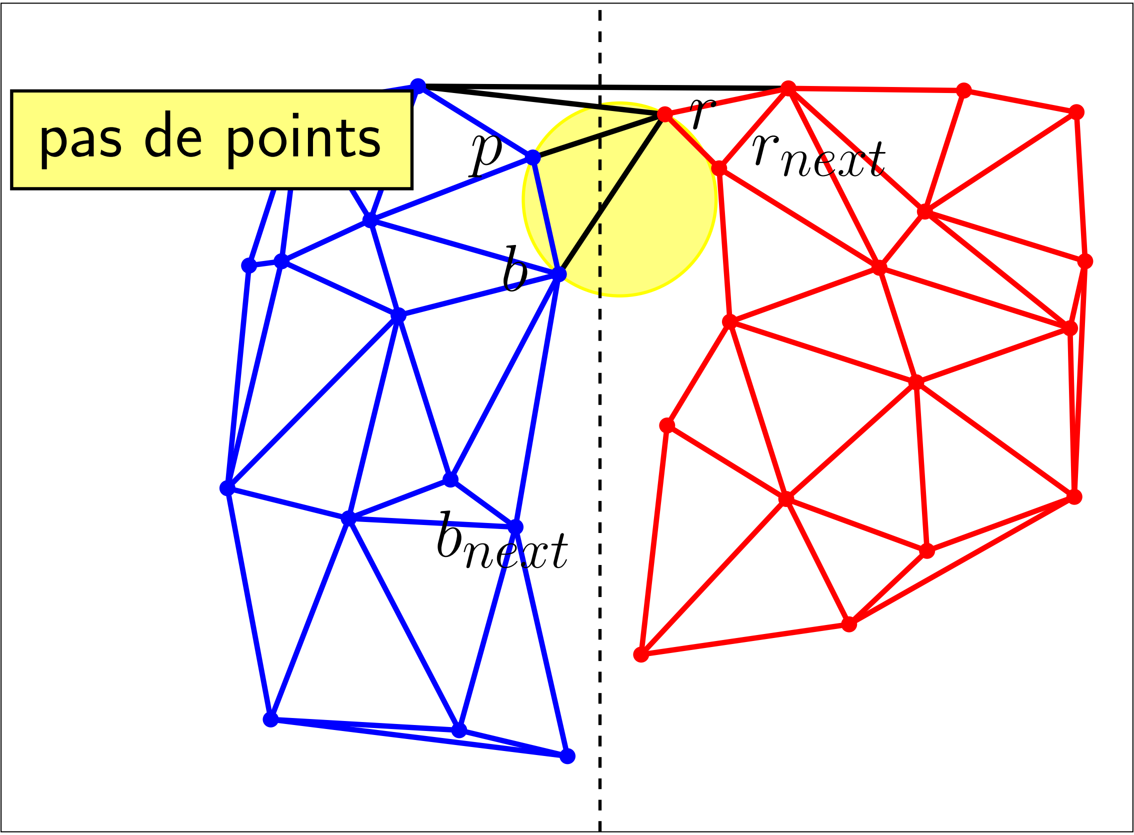



premier bleu rencontré par le faisceau de cercles
r
r_{next}
p
b
b_2
$b_2 \in Cercle(b, r, b_1)$
b_1

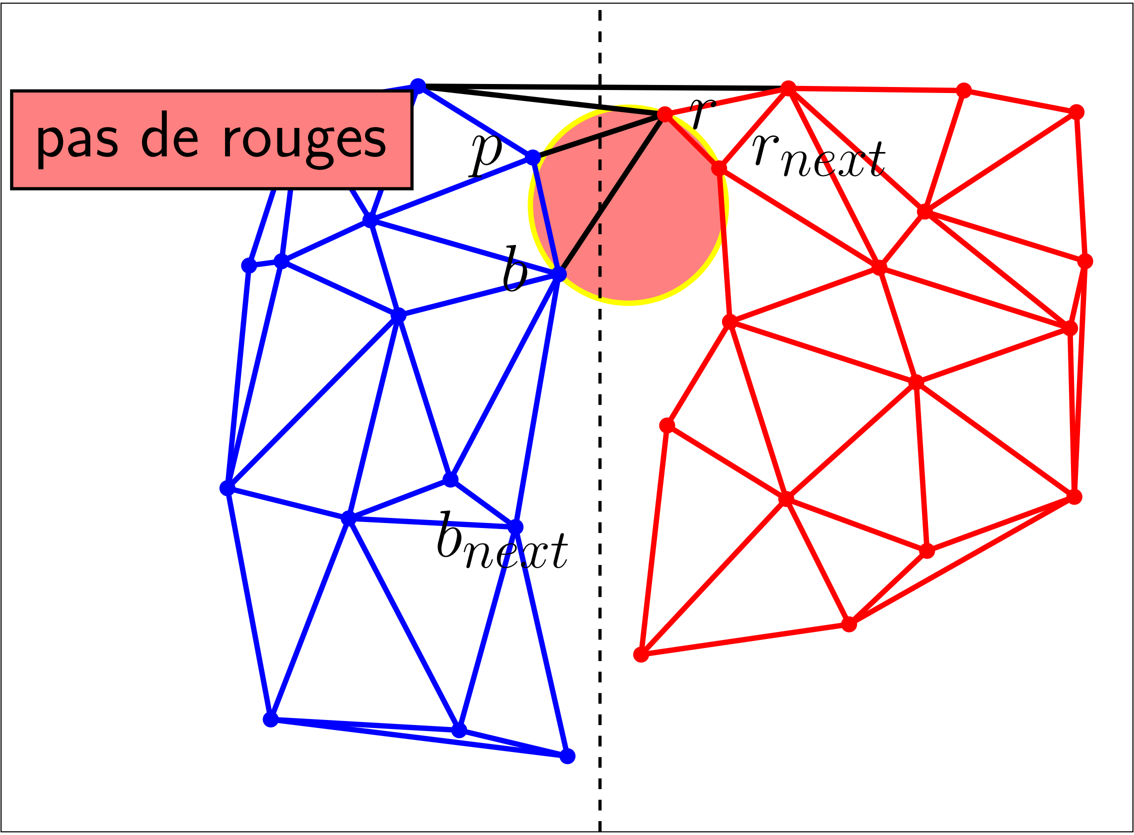

premier bleu rencontré par le faisceau de cercles
r
r_{next}
p
b
b_3
b_2
$b_3 \notin Cercle(b, r, b_2)$
b_1

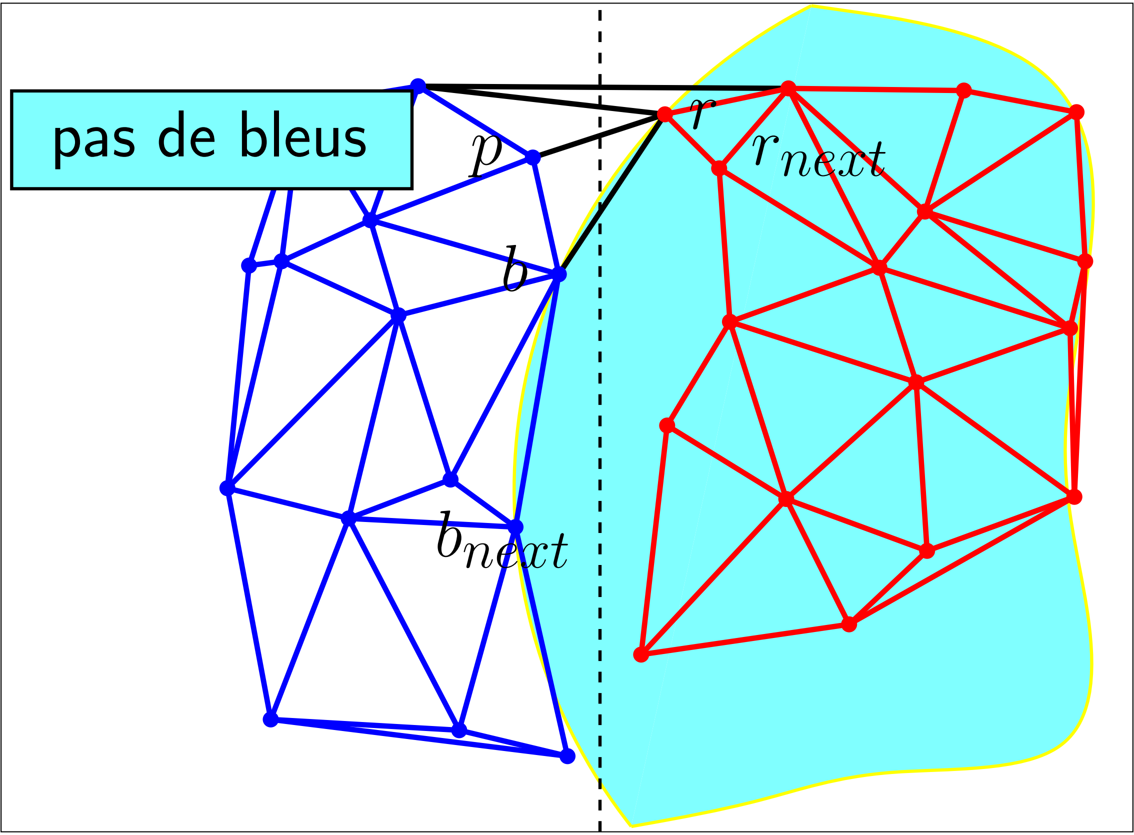

premier bleu rencontré par le faisceau de cercles
r
r_{next}
p
b
b_3
$b_{next} = b_2$
$b_3 \notin Cercle(b, r, b_2)$
b_1
$\forall bleu, bleu \notin Cercle(b, r, b_2)$

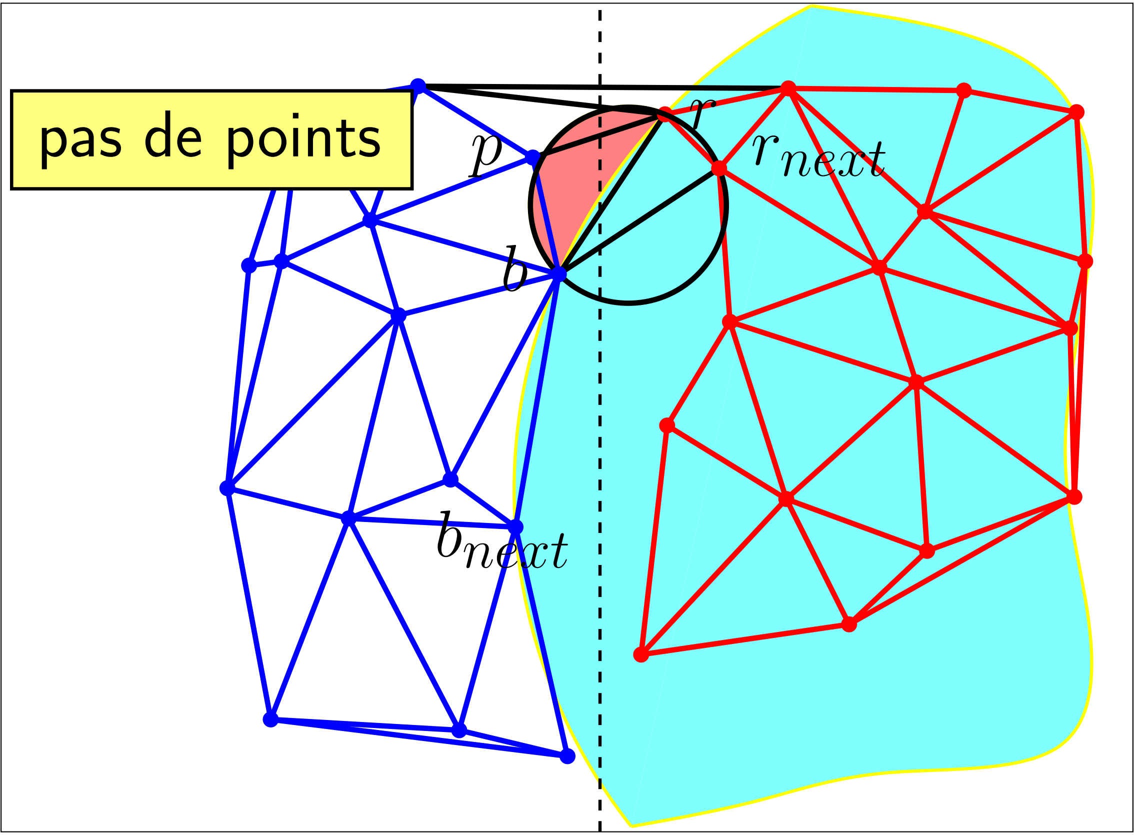

r
p
r_{next}
b
b_{next}

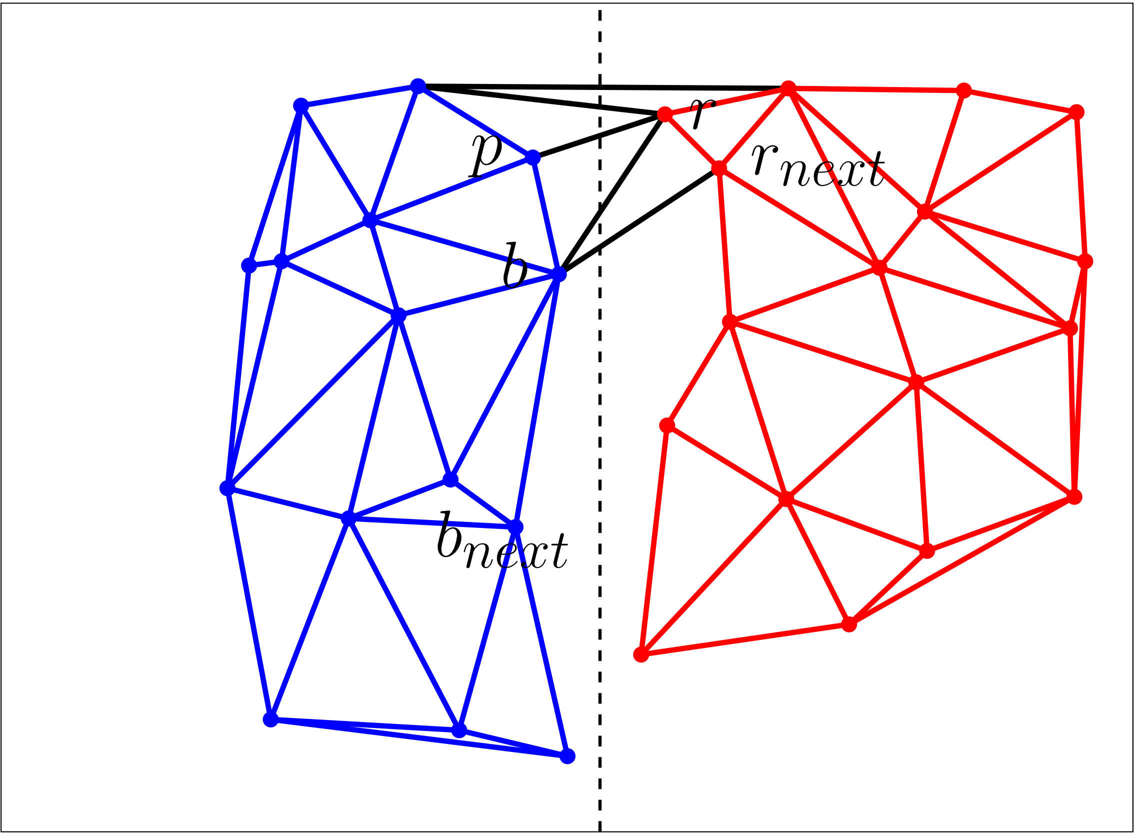


pas de points
p
r
r_{next}
b
b_{next}

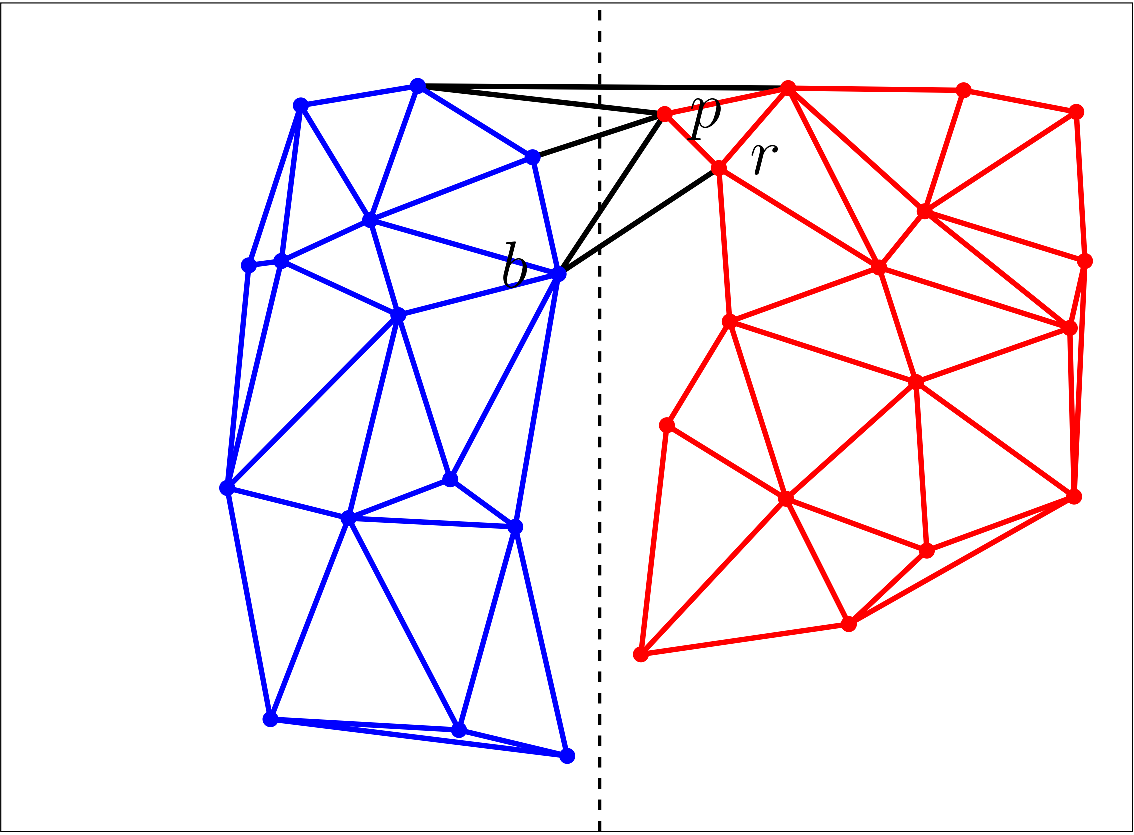

pas de rouges
p
r
r_{next}
b
b_{next}

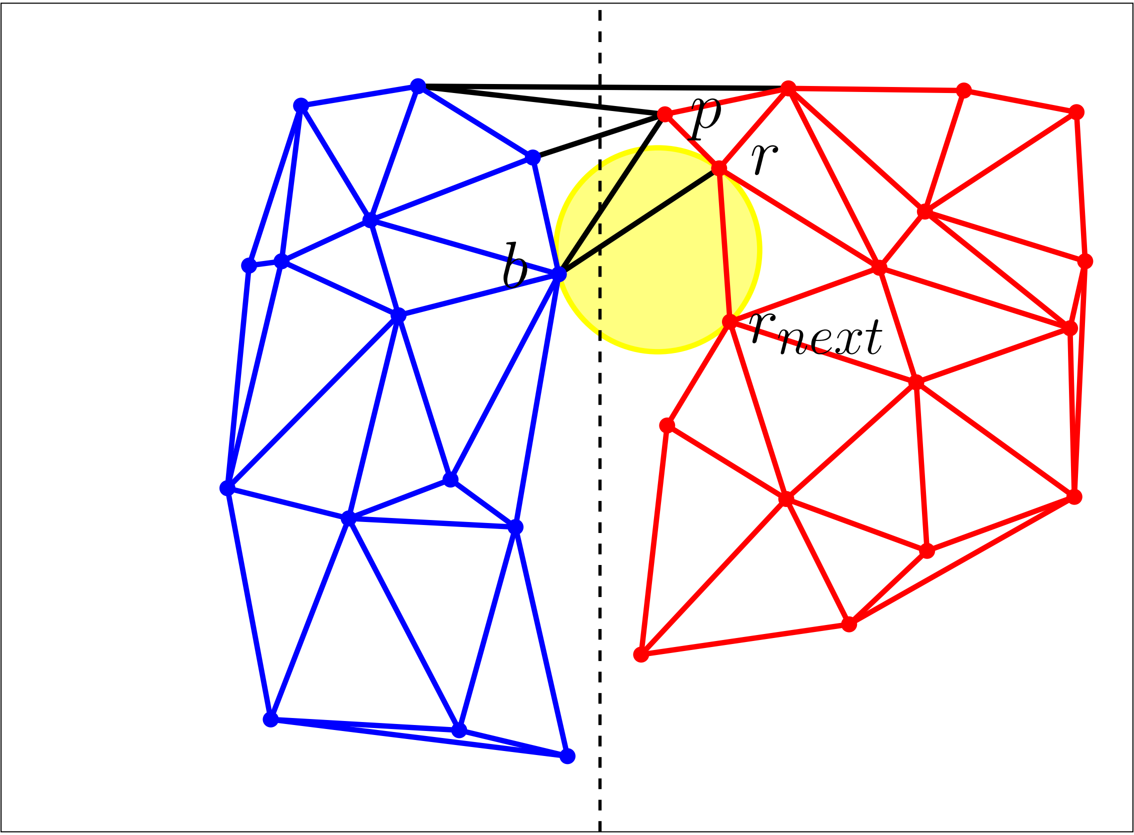

pas de bleus
p
r
r_{next}
b
b_{next}

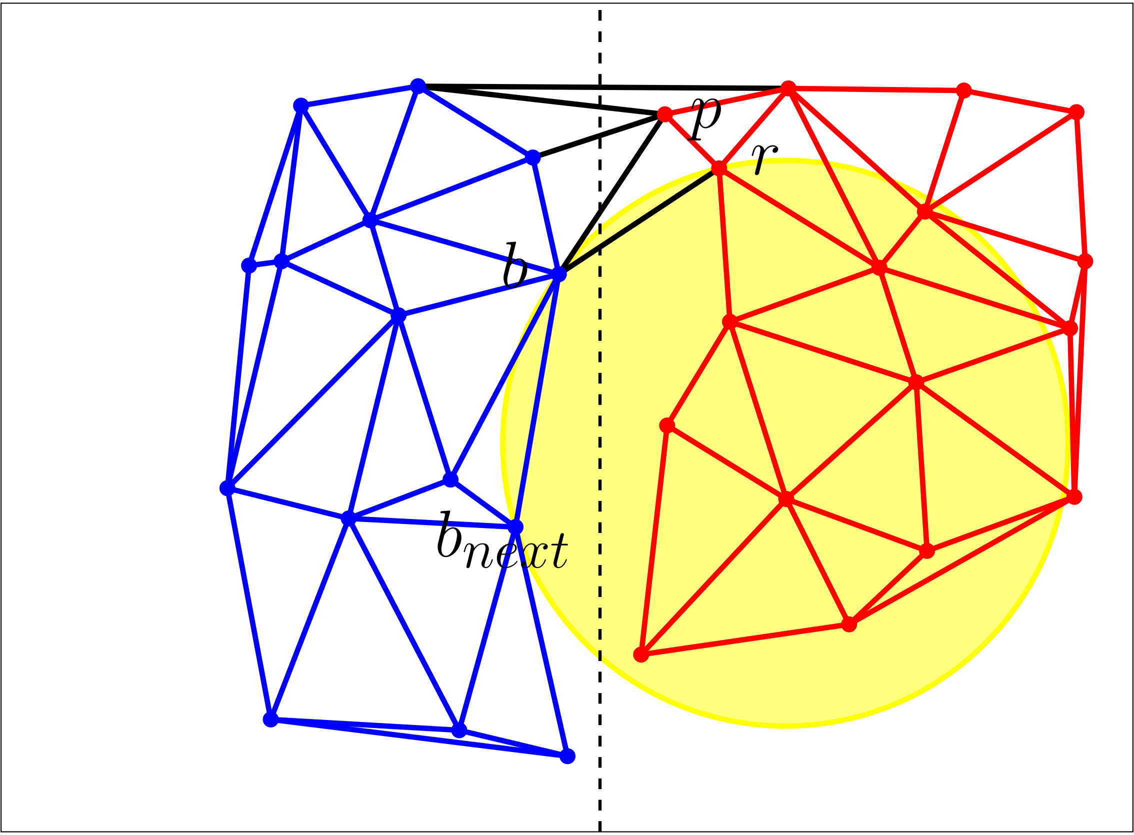

pas de points
p
r
r_{next}
b
b_{next}



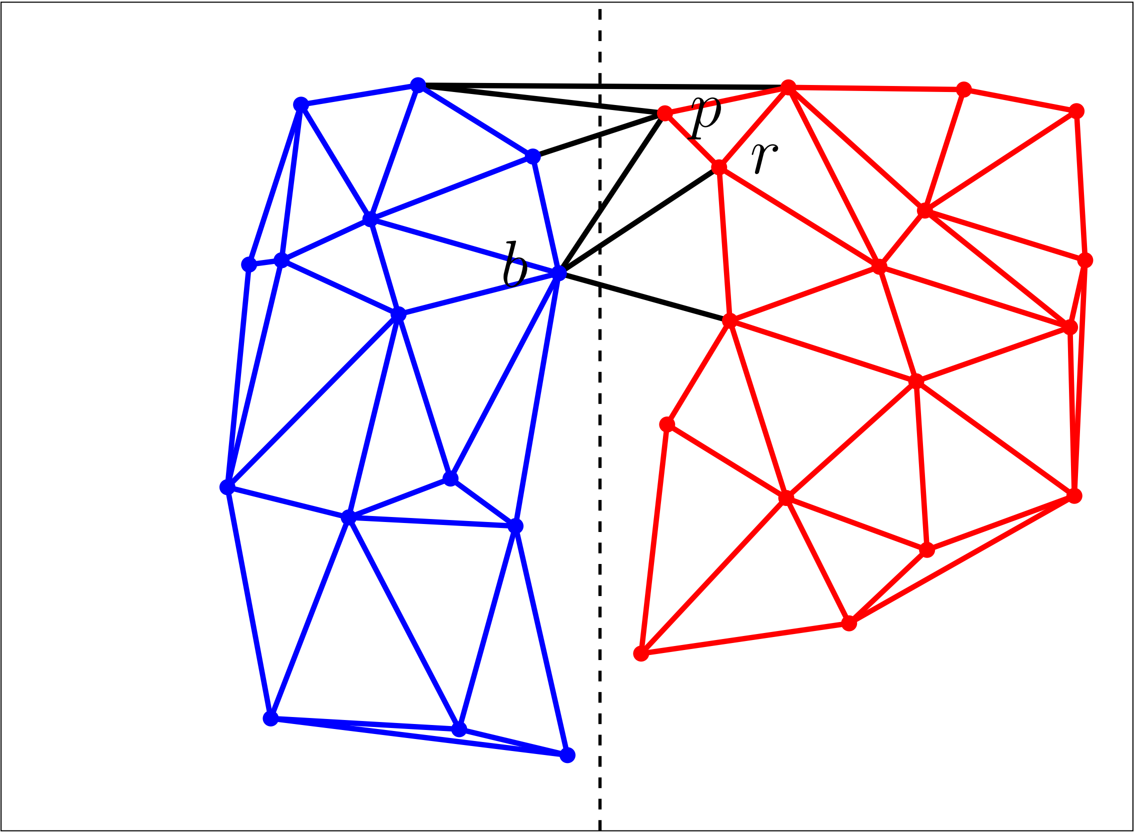
r
p
r_{next}
b
b_{next}

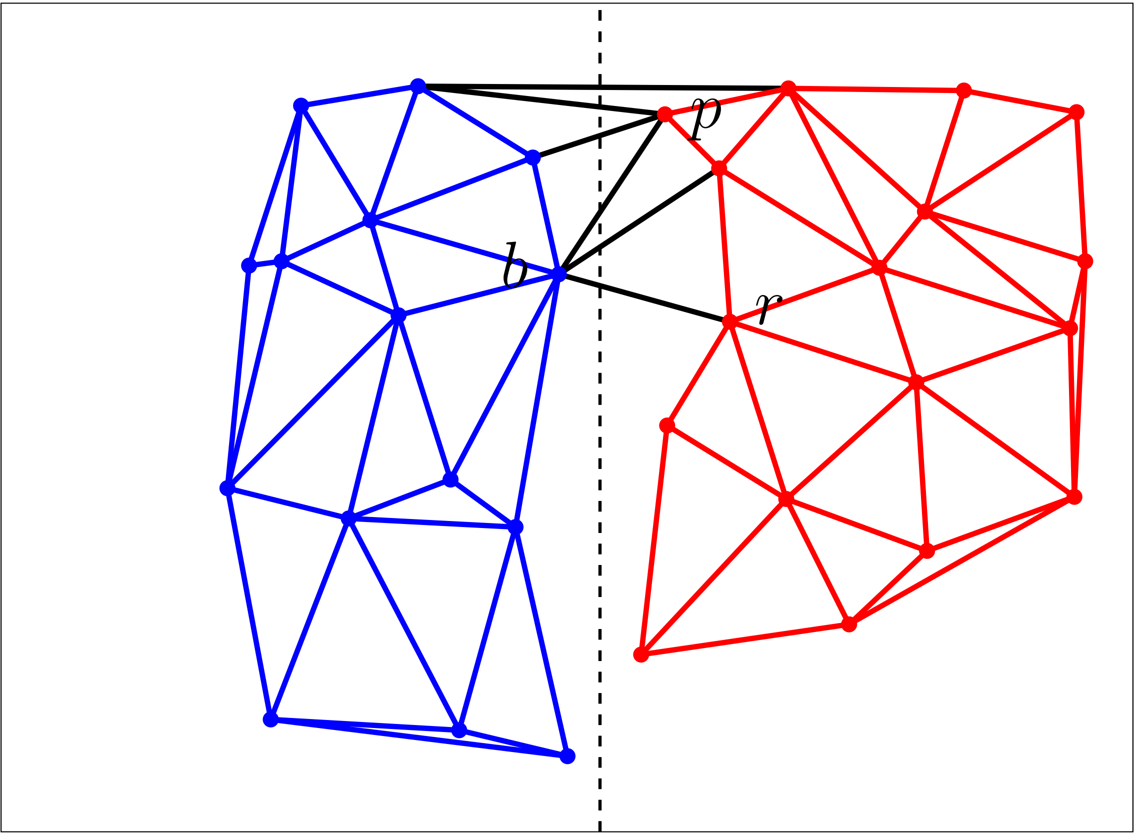

p
r
b

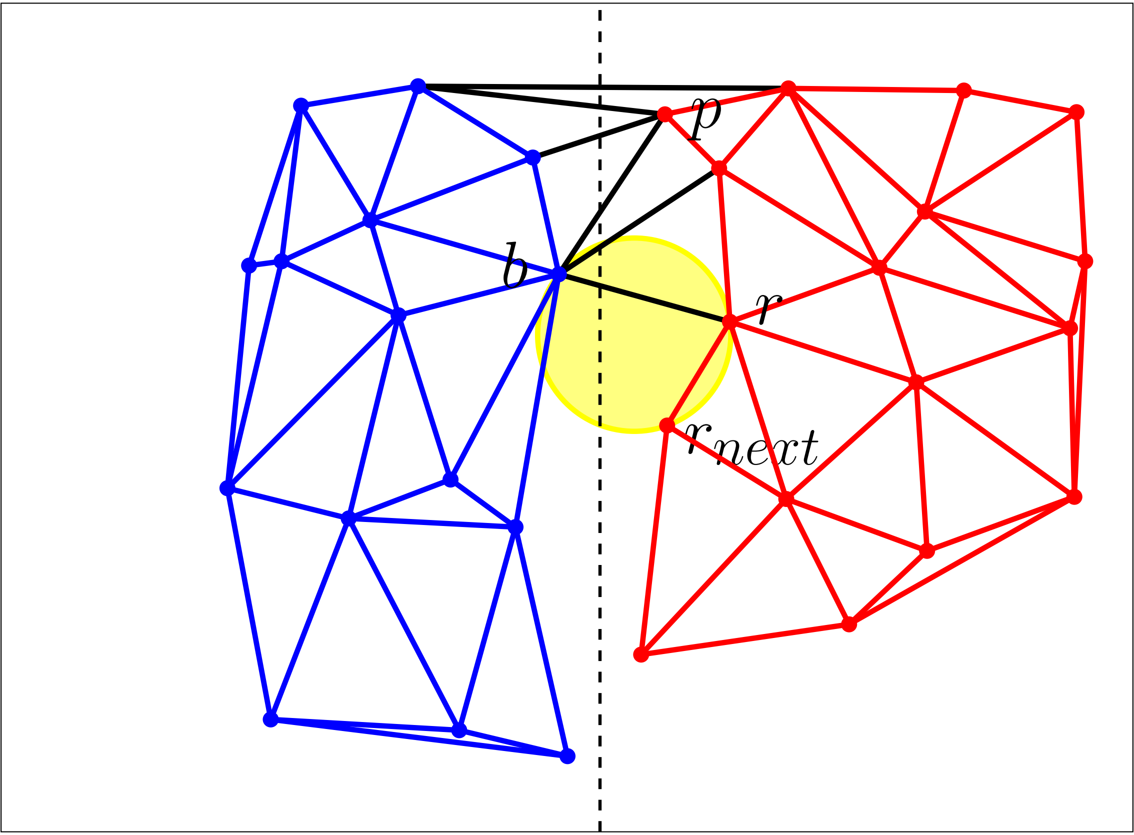

p
r
b
r_{next}

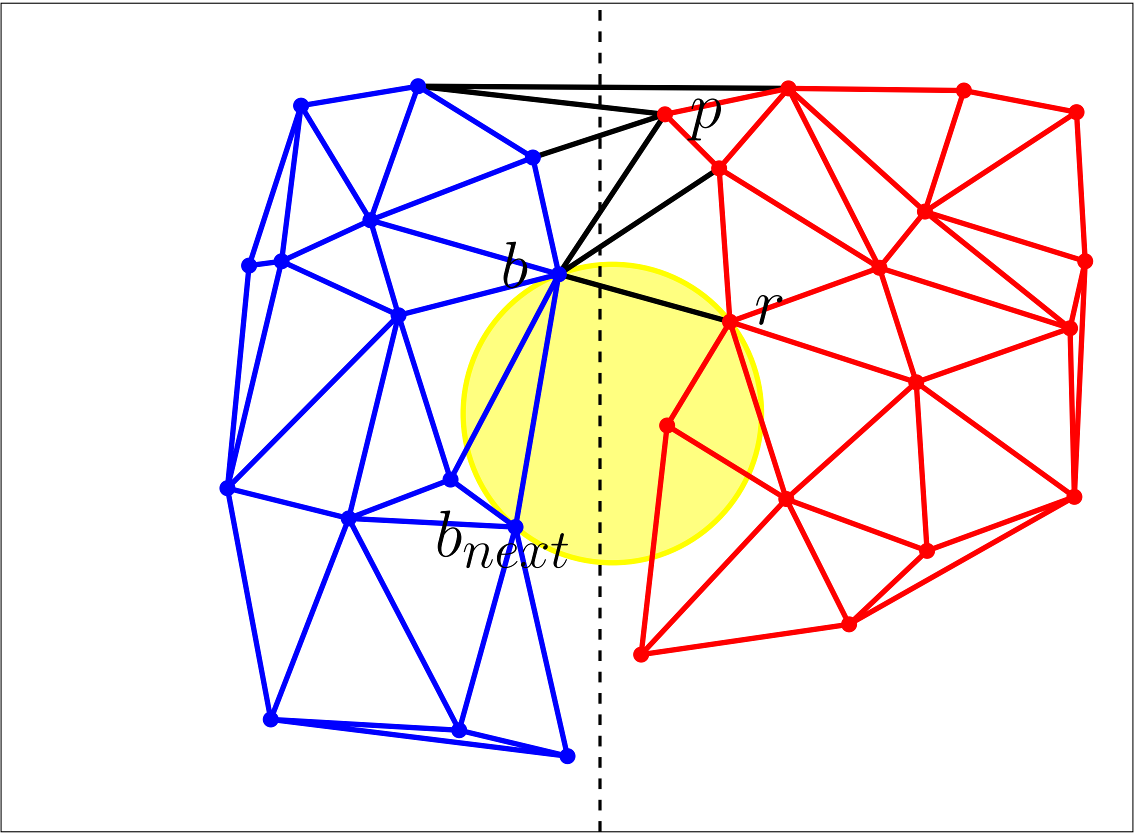


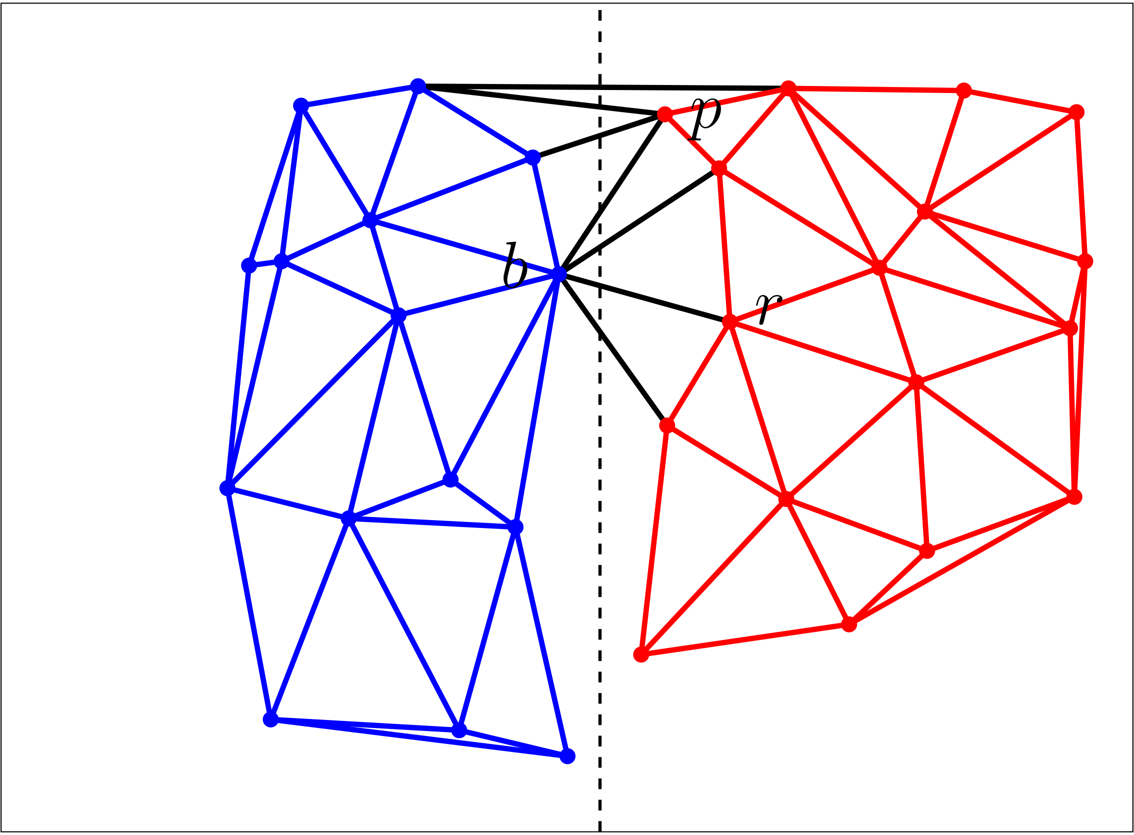
p
r
b
b_{next}

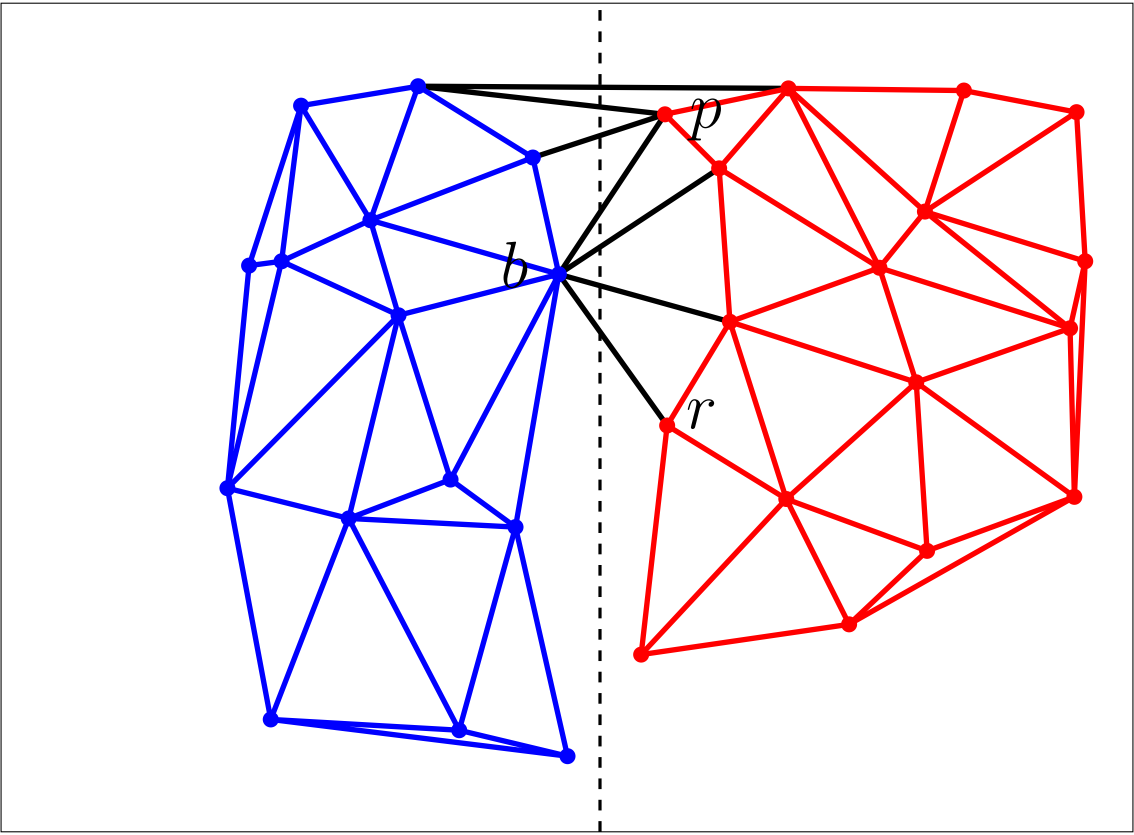

p
r
b

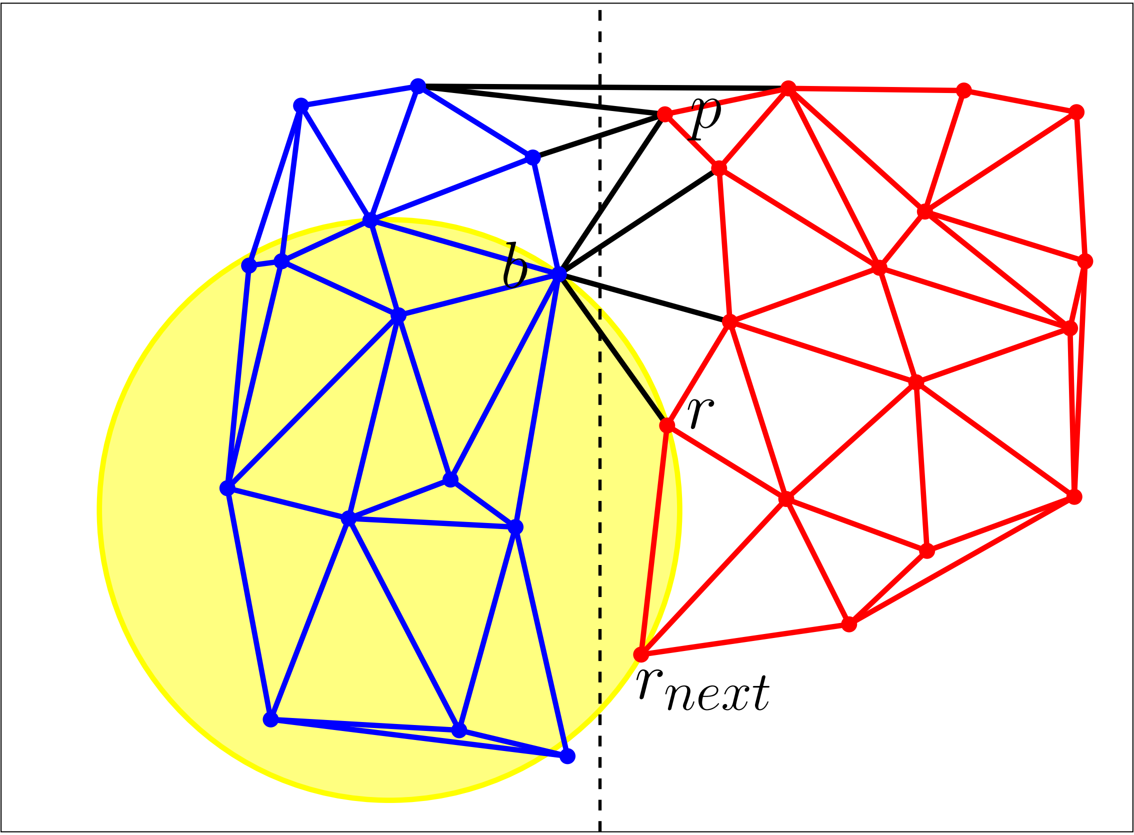

p
b
r


p
b
r
r_{next}

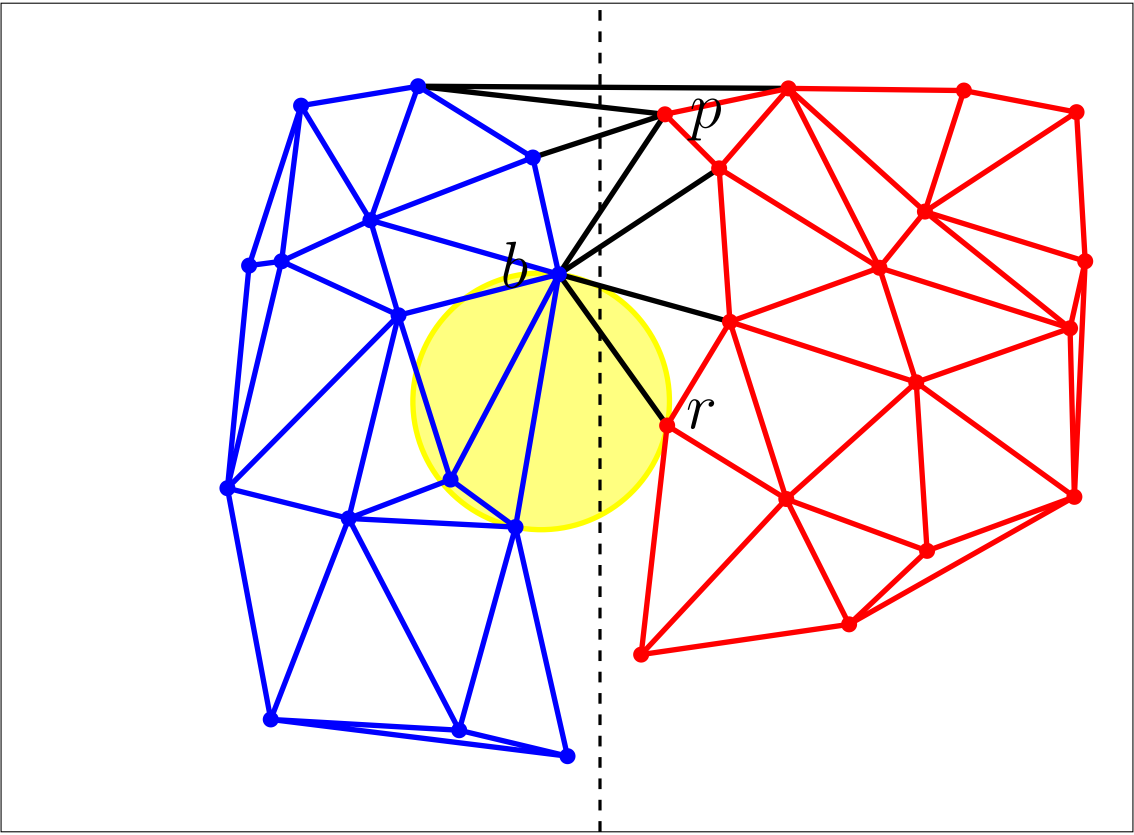

p
b
r
b_{next}

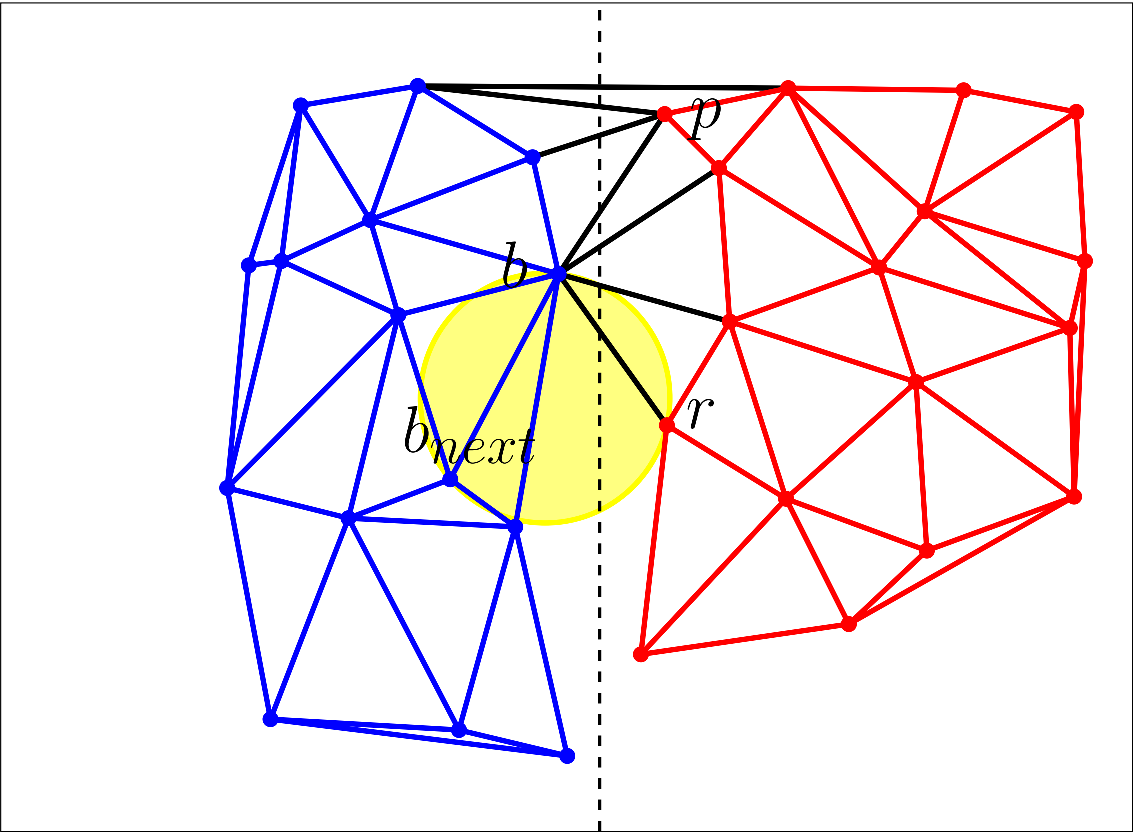

p
b
r

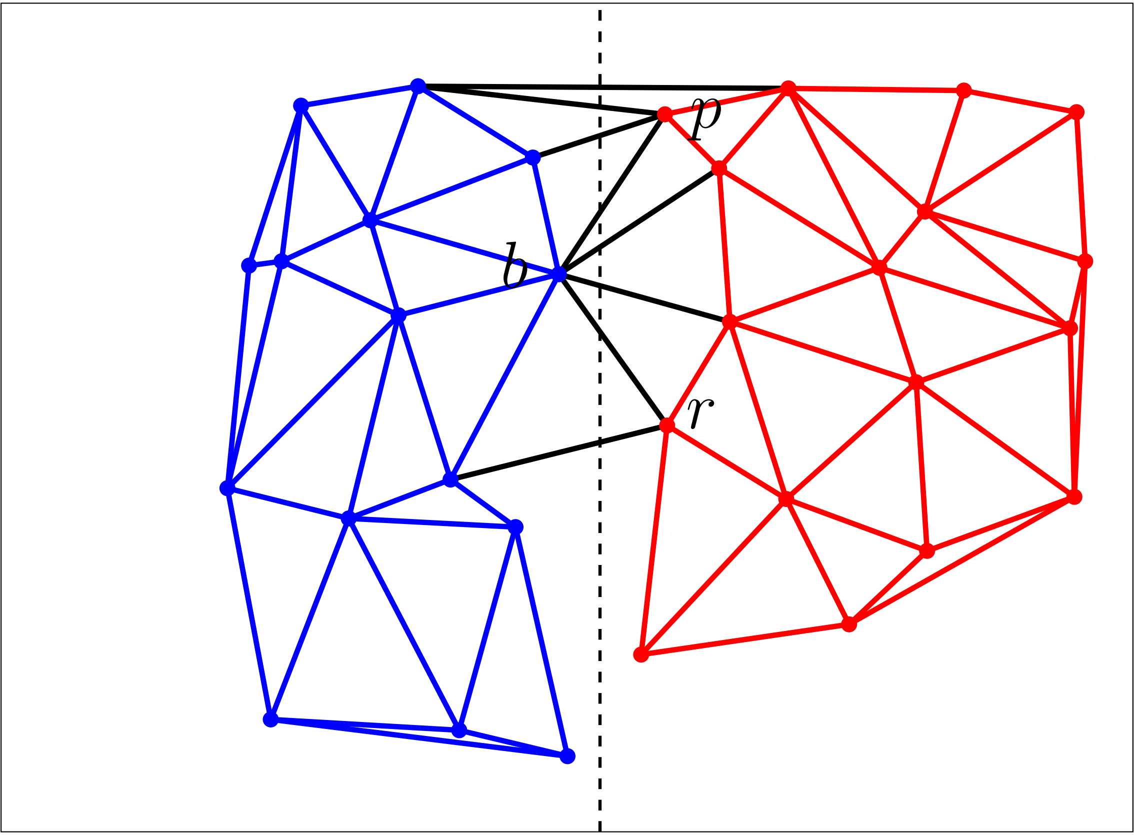

p
b
r

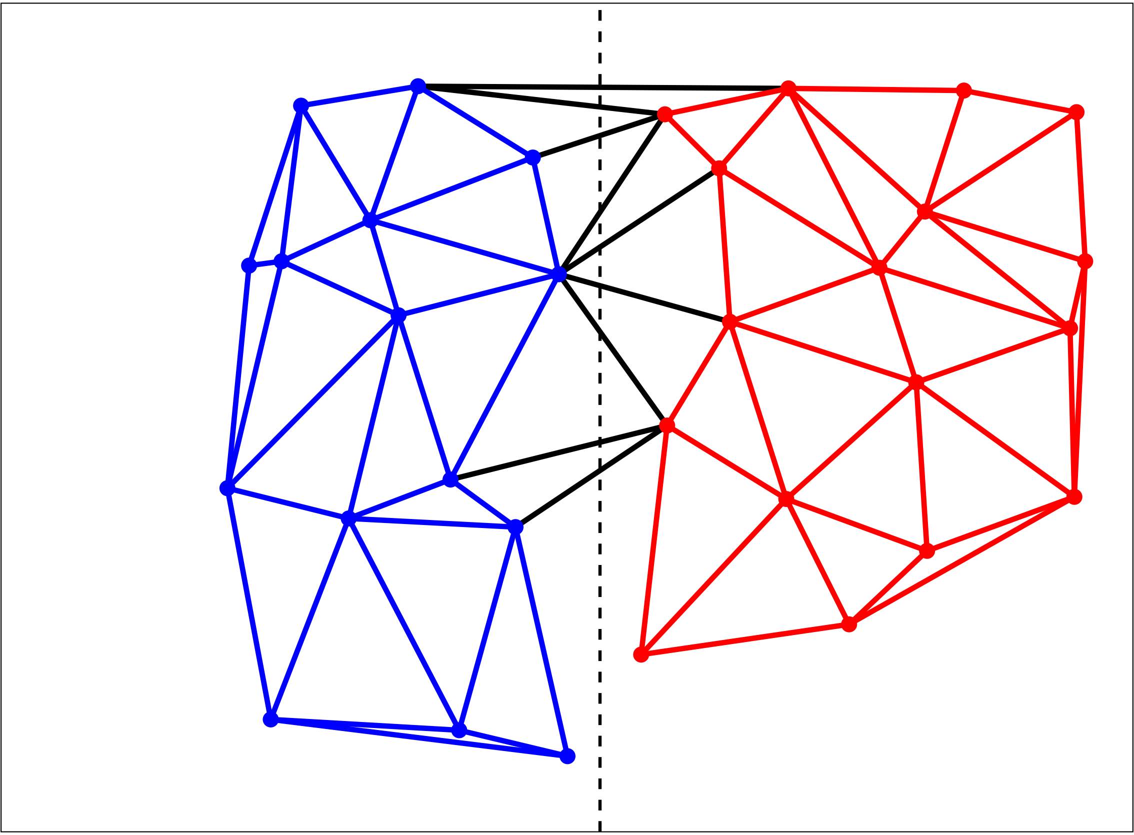


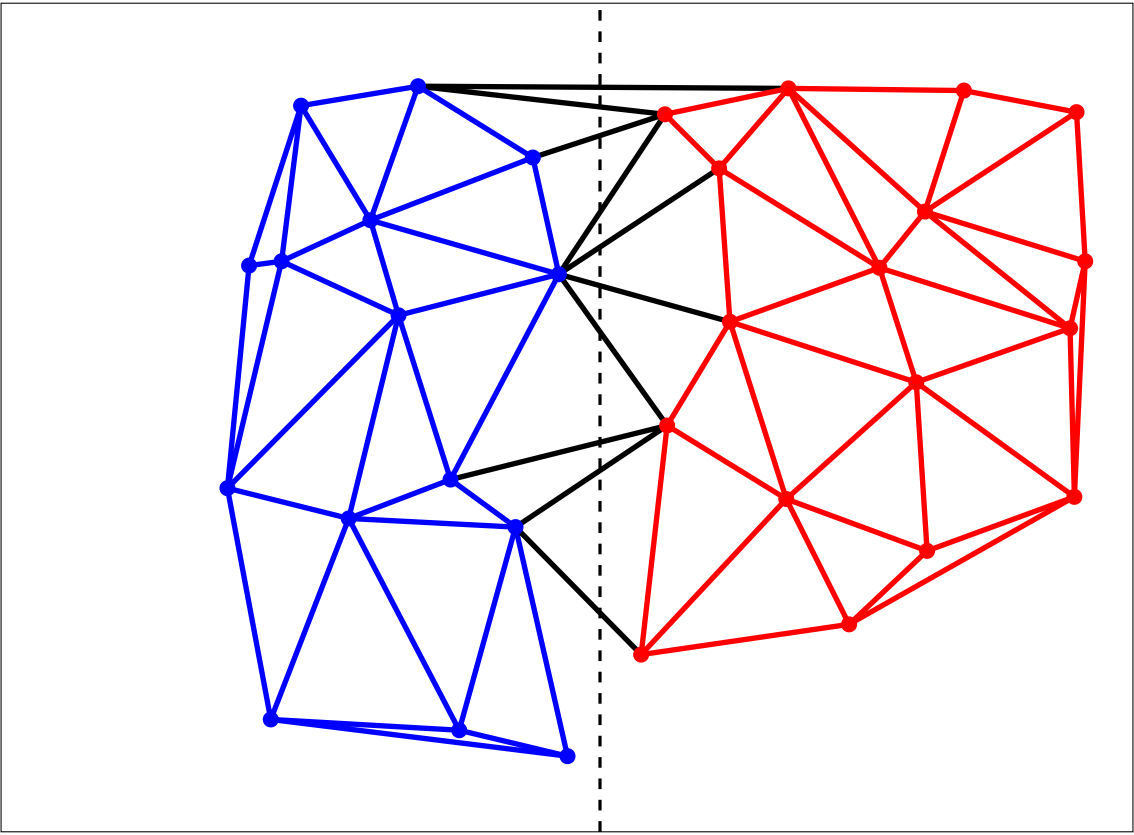
p
b
r
r_{next}

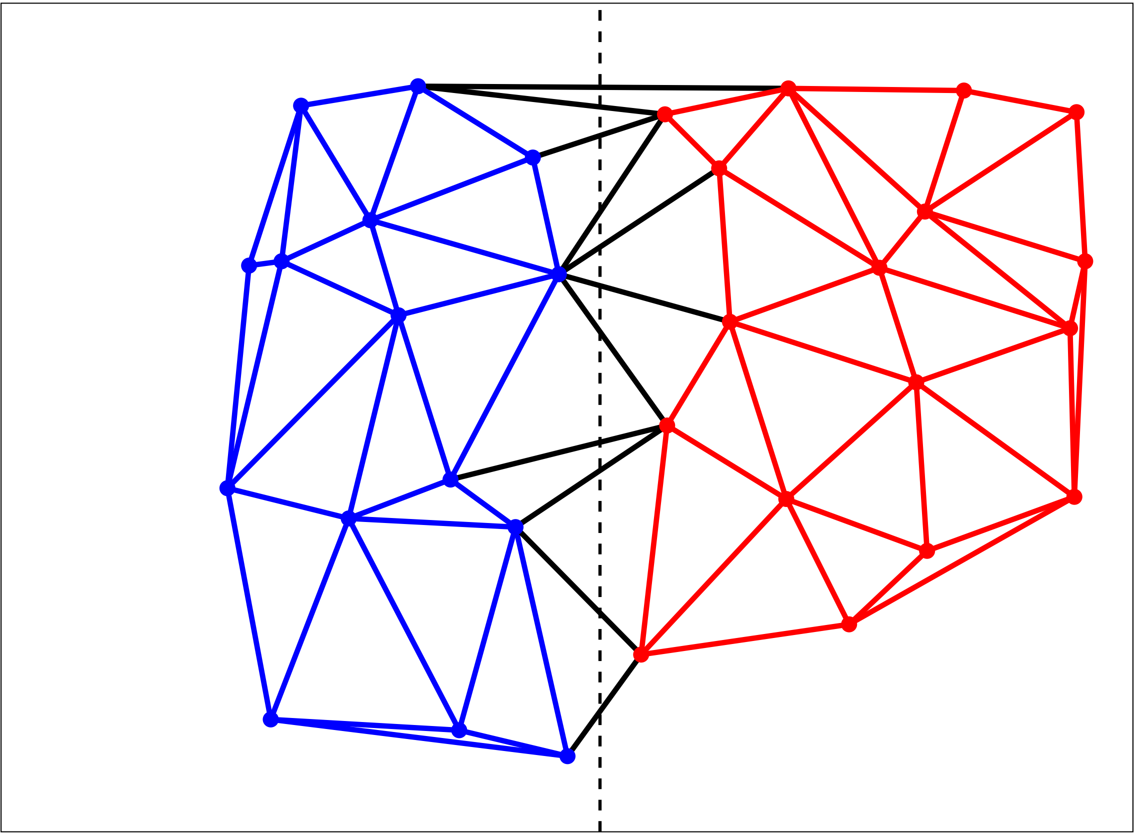

p
b
r

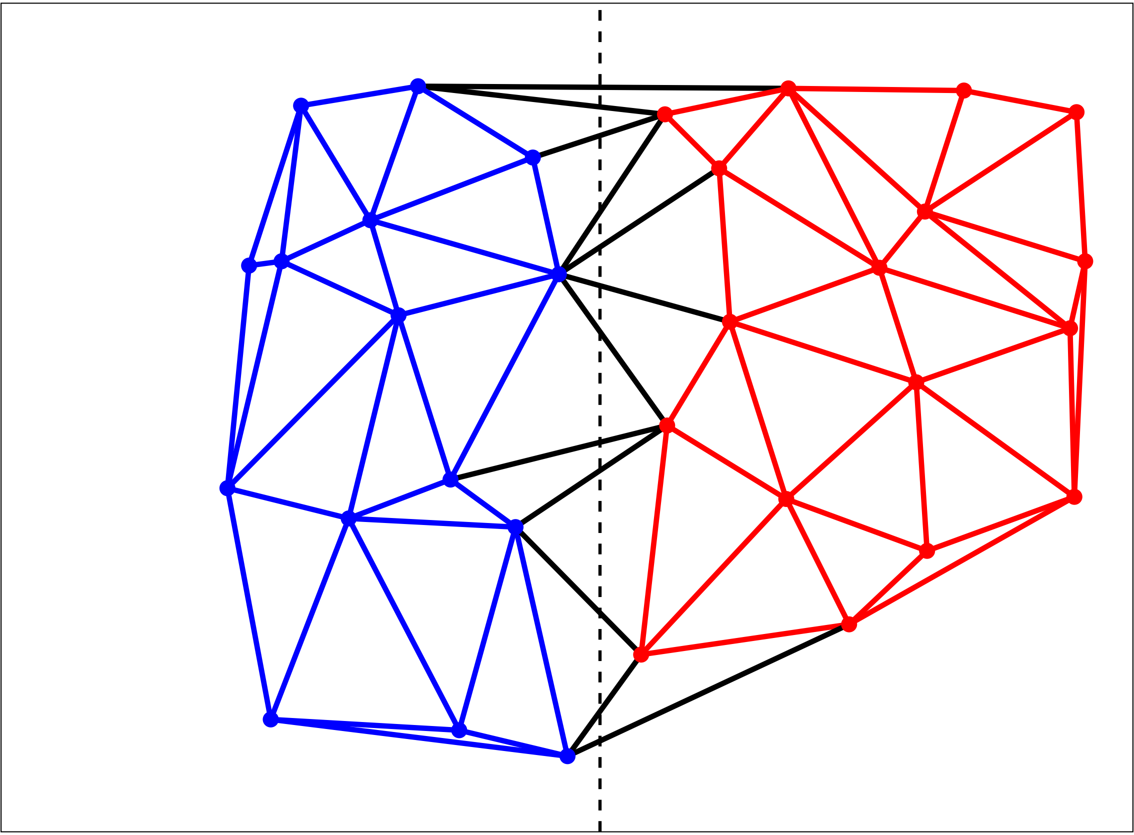

p
b
r
b_{next}


p
b
r









Complexité

Complexité

A chaque étape de la recherche de r_{next}

Complexité

A chaque étape de la recherche de r_{next}

On efface une arête rouge

Complexité

A chaque étape de la recherche de r_{next}

On efface une arête rouge

A chaque étape de la recherche de b_{next}

Complexité

A chaque étape de la recherche de r_{next}

On efface une arête rouge

A chaque étape de la recherche de b_{next}

On efface une arête bleue

Complexité

A chaque étape de la recherche de r_{next}

On efface une arête rouge

A chaque étape de la recherche de b_{next}

On efface une arête bleue

Choisir entre r_{next} et b_{next}

Complexité

A chaque étape de la recherche de r_{next}

On efface une arête rouge

A chaque étape de la recherche de b_{next}

On efface une arête bleue

Choisir entre r_{next} et b_{next}

On trace une arête noire

Complexité

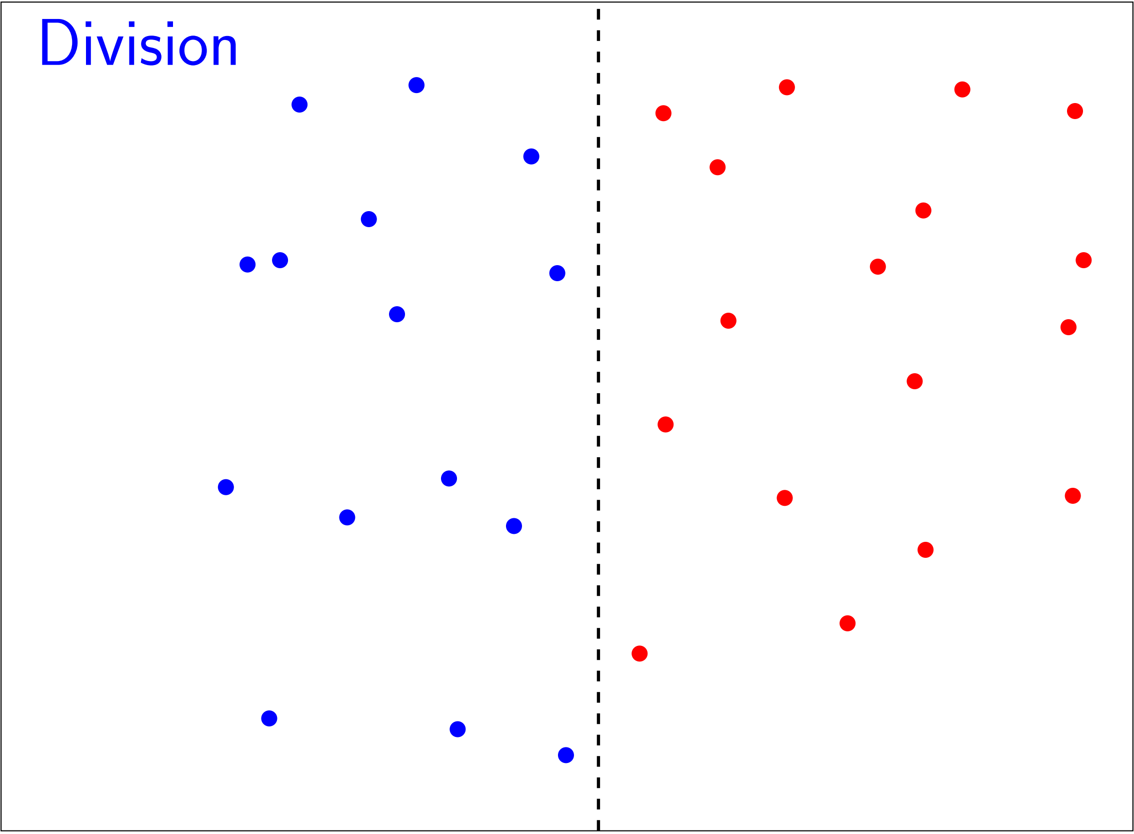
$$\text{Complexité} \leq \begin{array}{l} \sharp \text{ arêtes rouges} \\ + \sharp \text{ arêtes bleues} \\ + \sharp \text{ arêtes noires} \end{array}$$

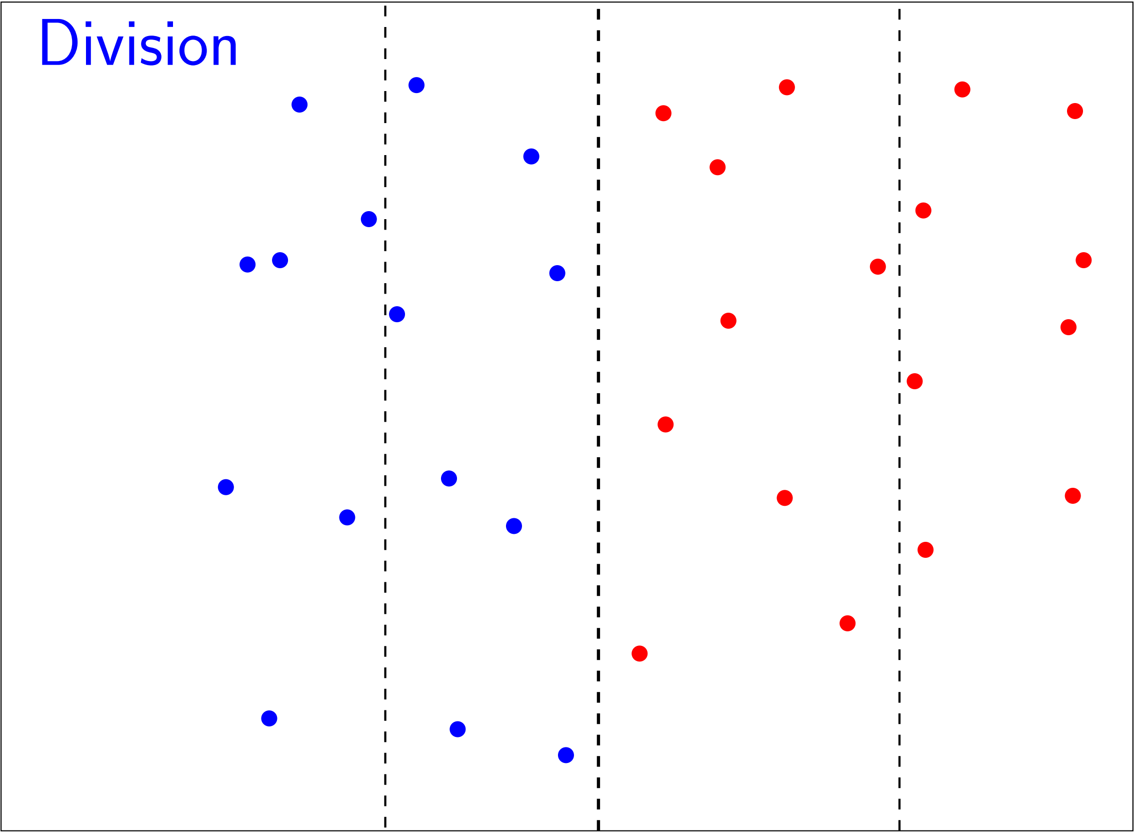
Complexité

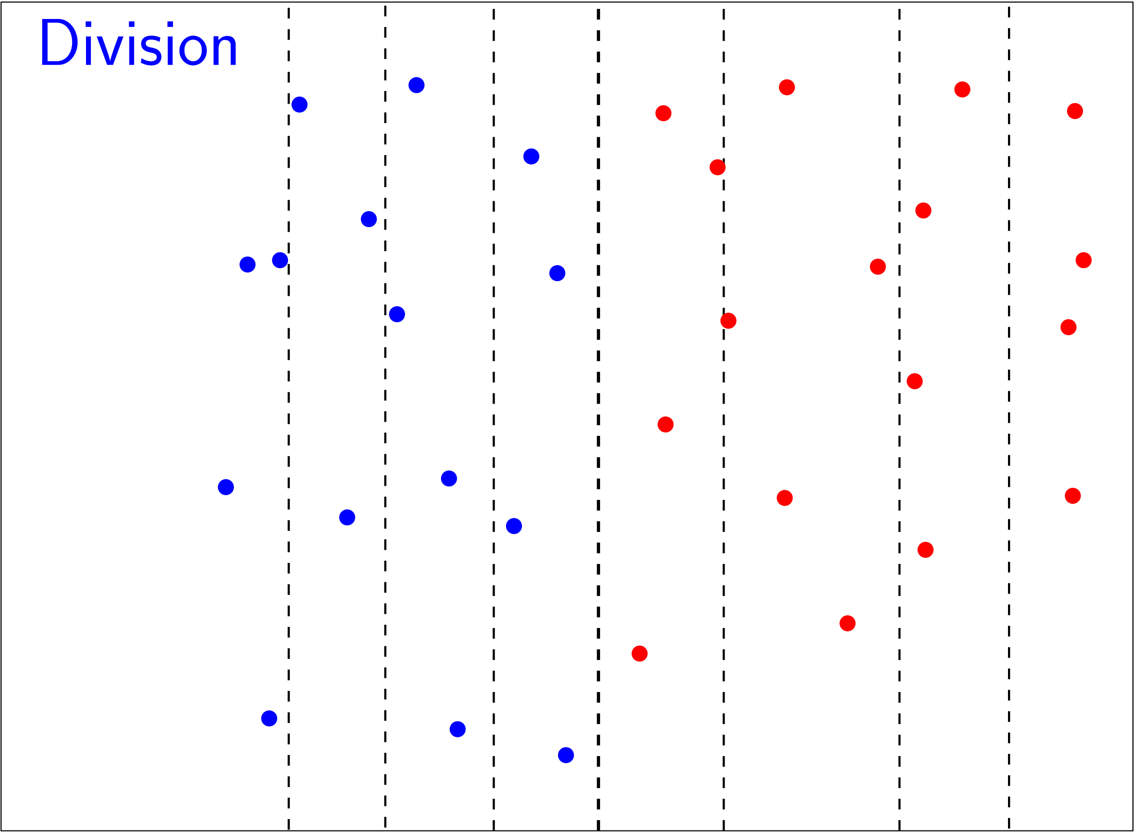
Complexité $\leq$ $\sharp$ arêtes rouges
$+\sharp$ arêtes bleues
$+\sharp$ arêtes noires

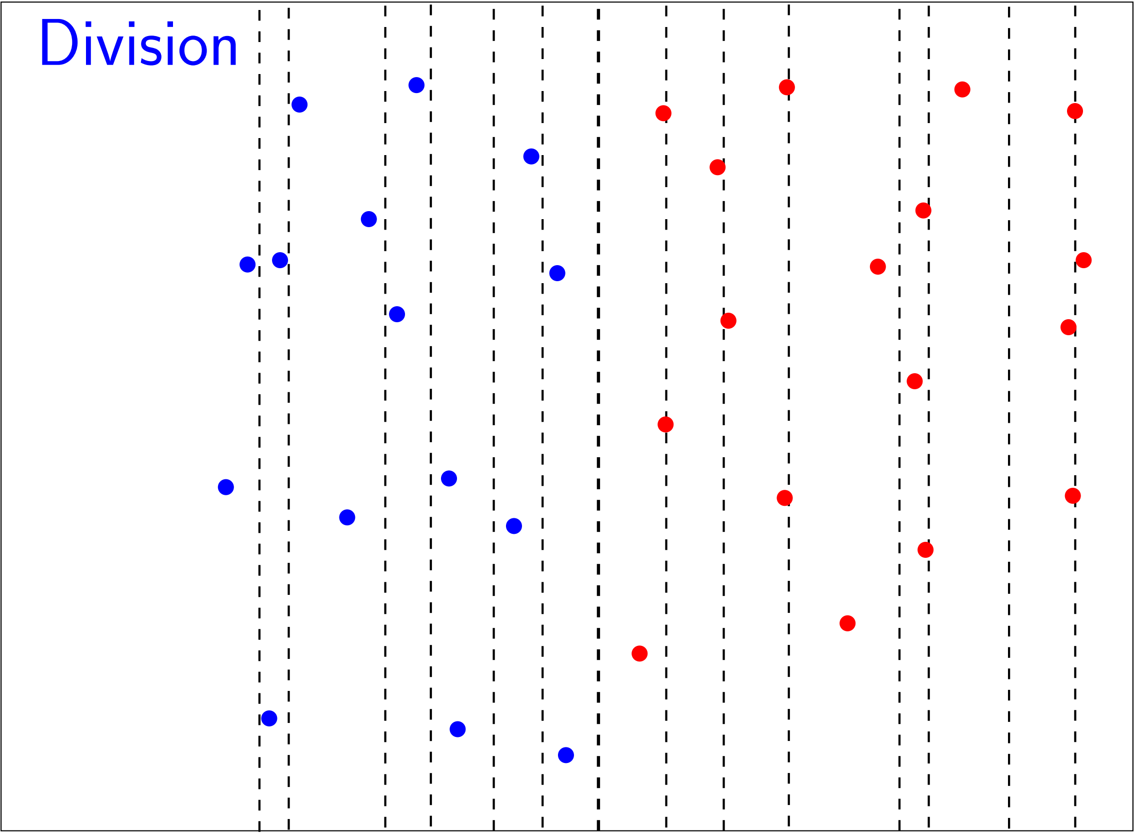
$$\leq 3n + 3n = O(n)$$

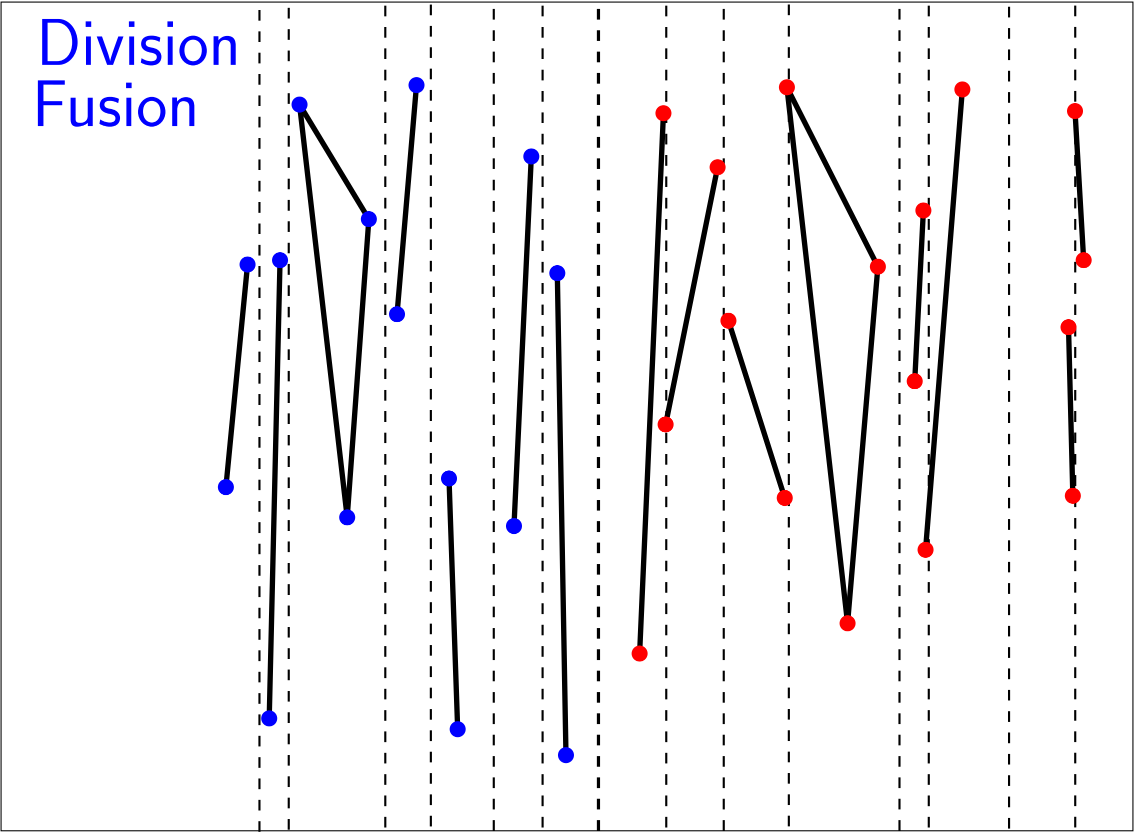
Division-Fusion $\Longrightarrow O(n \log n)$

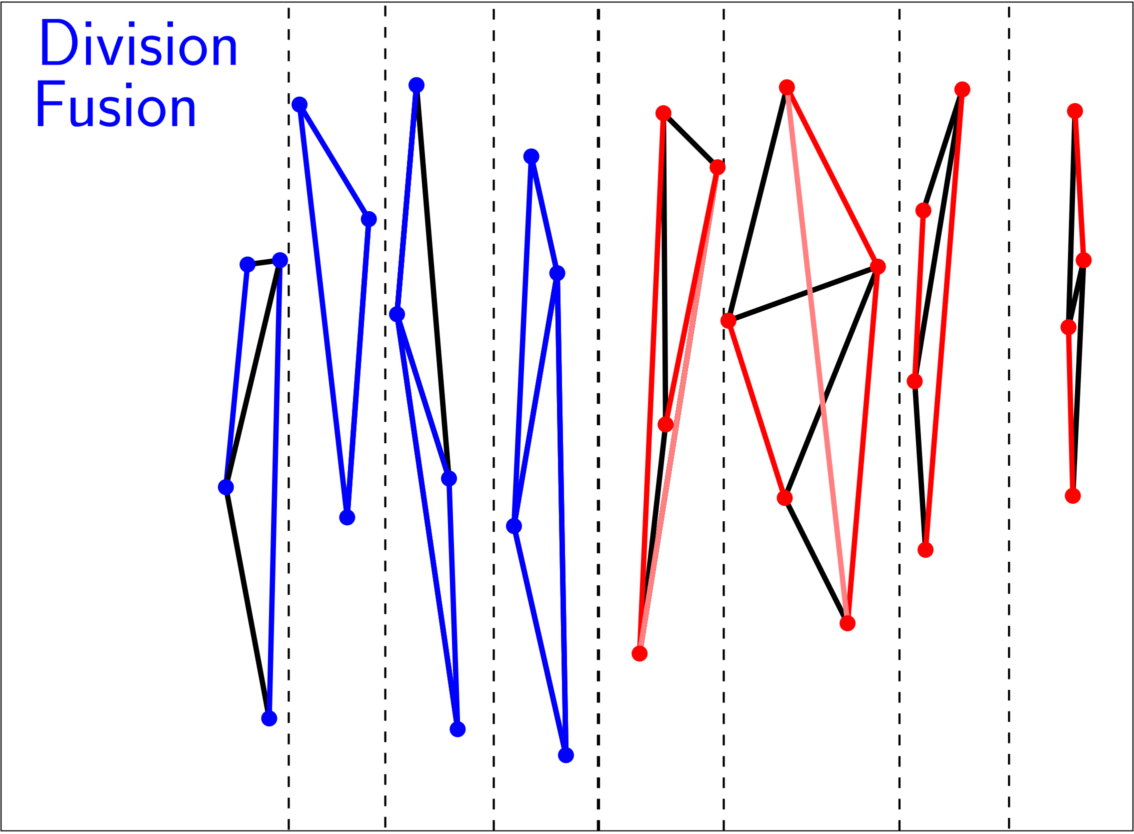

Division

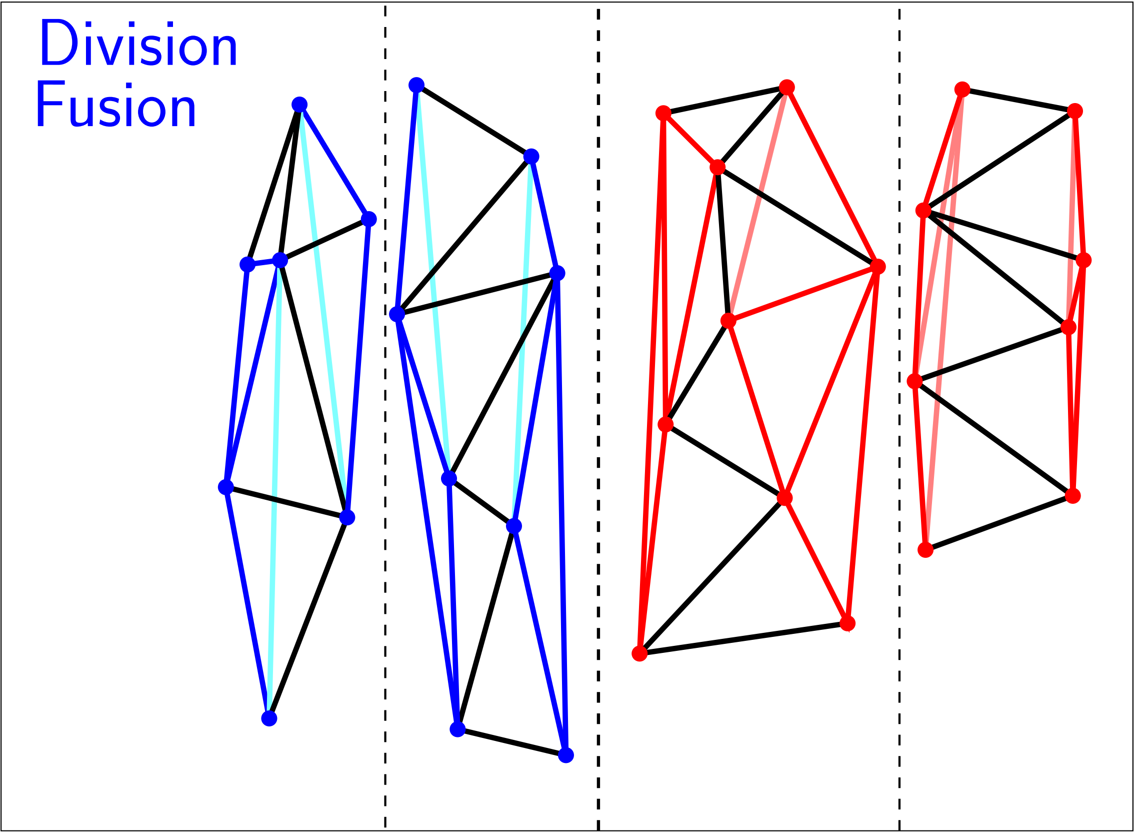

Division

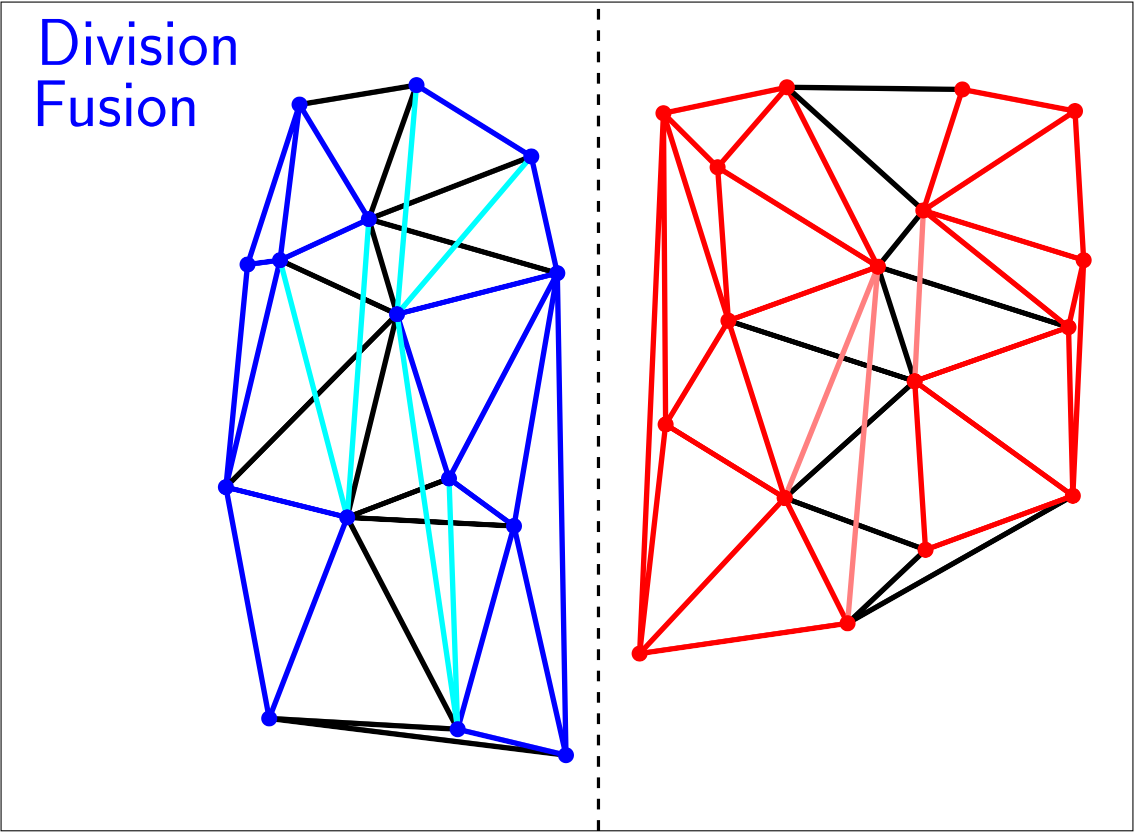

Division

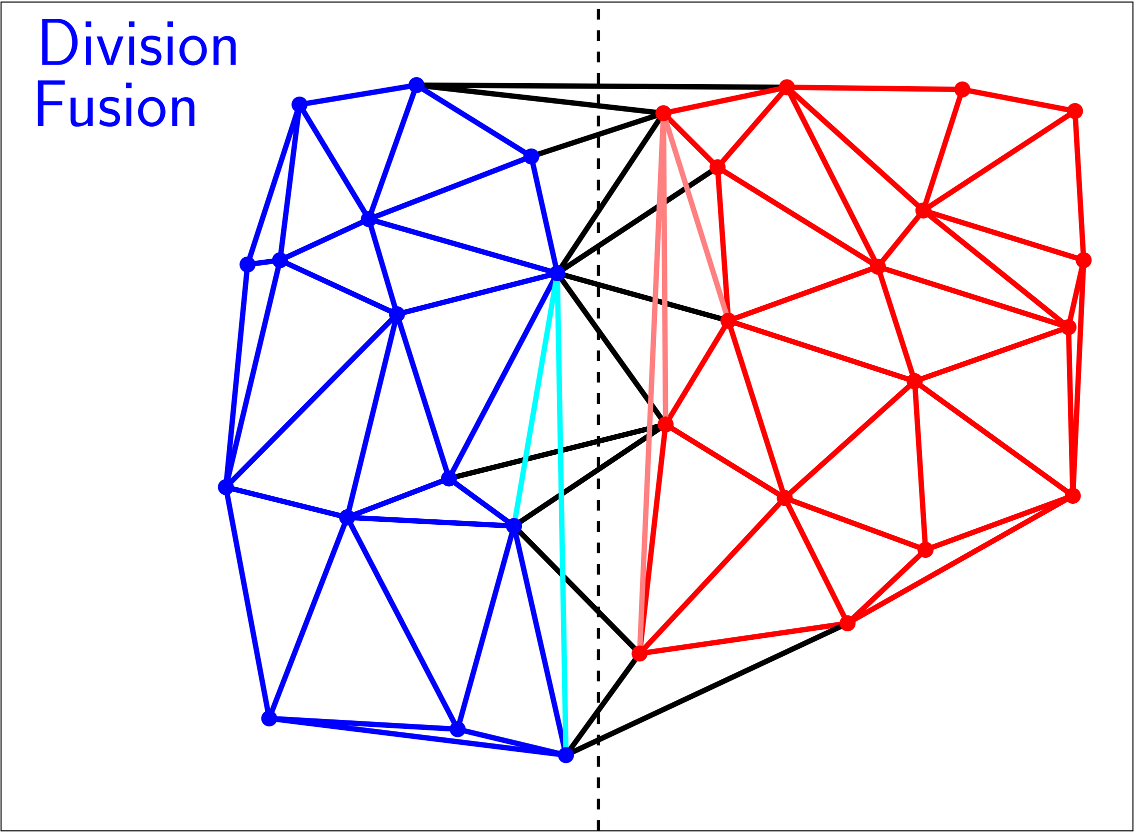

Division

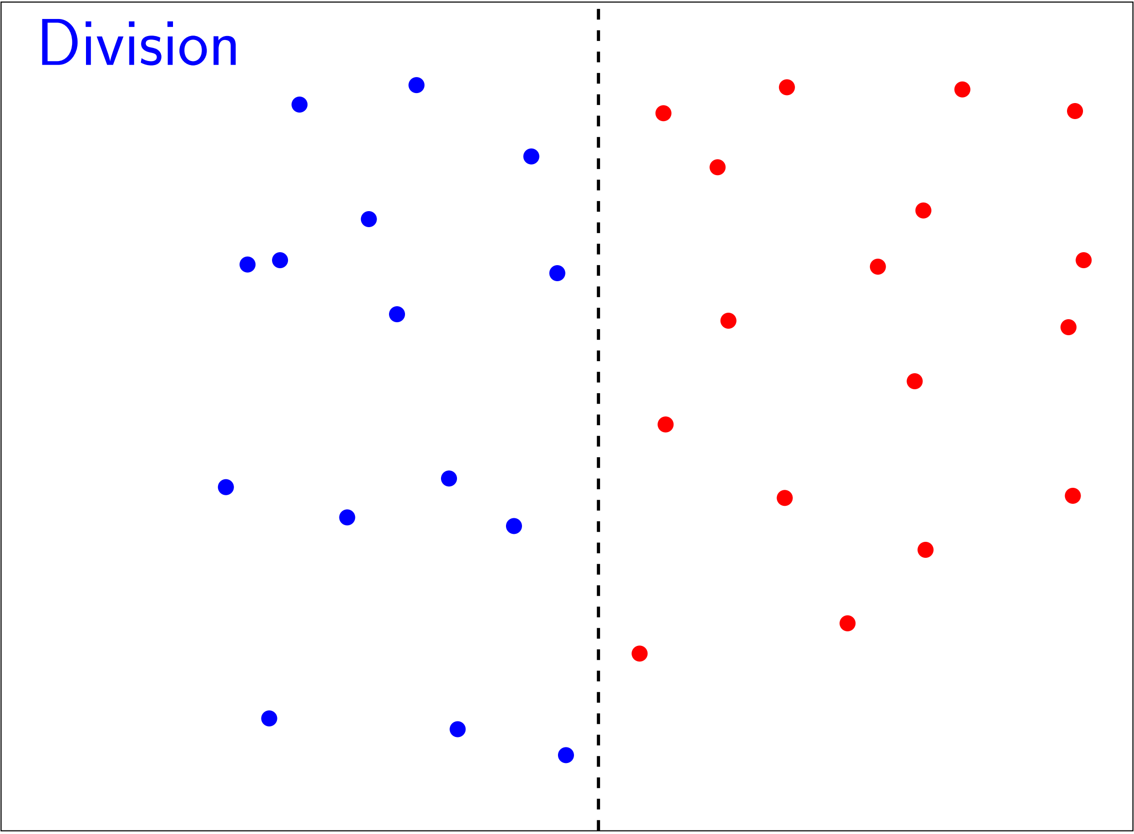

Division
Fusion

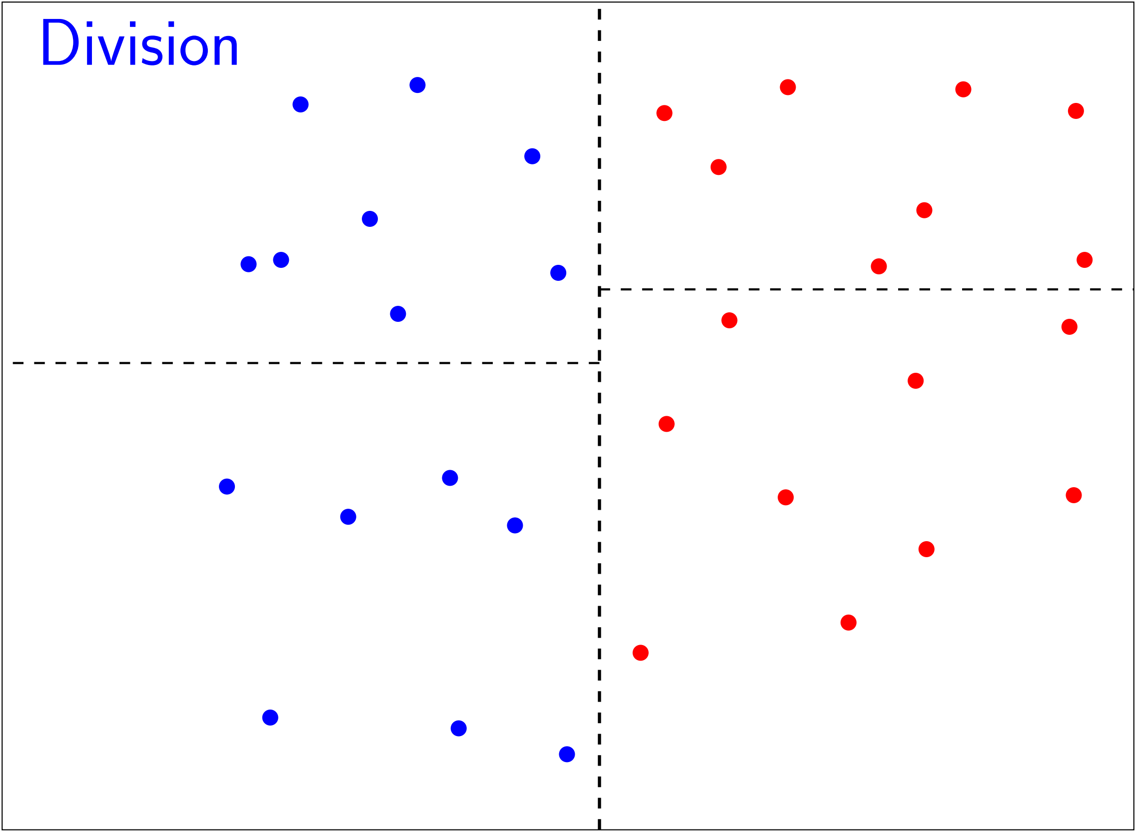

Division
Fusion

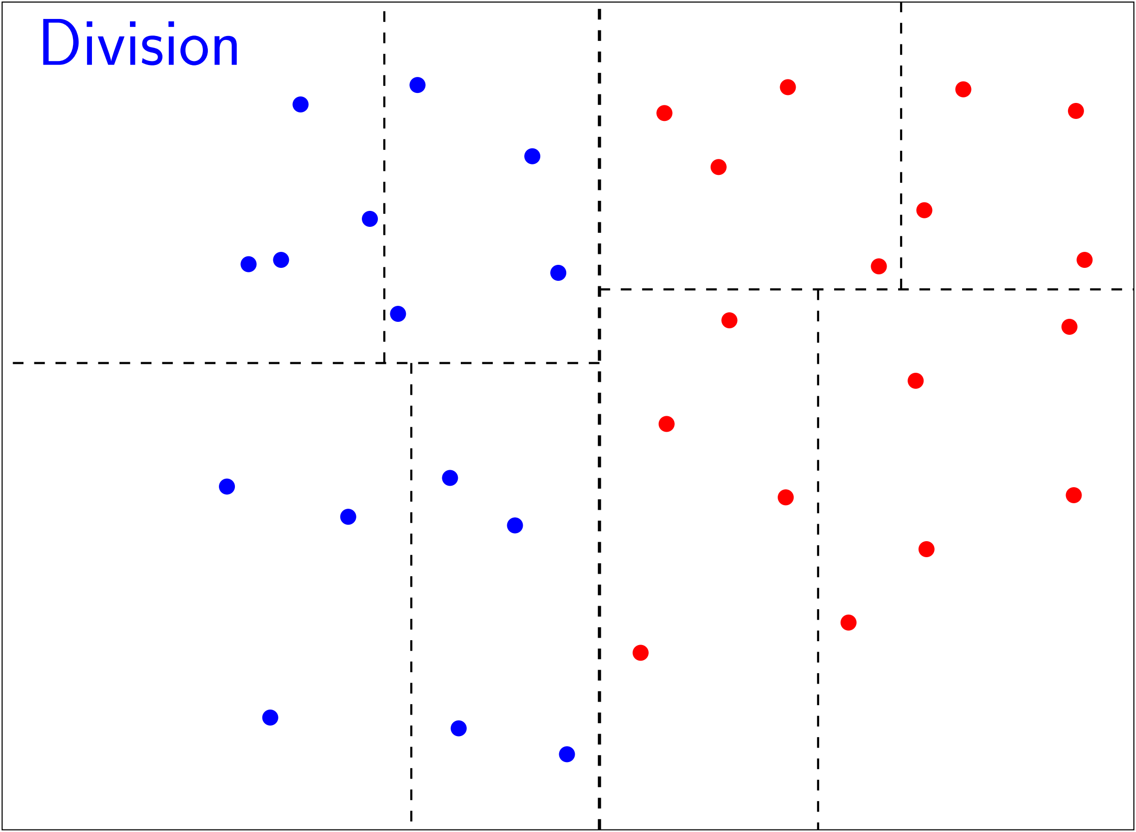

Division
Fusion

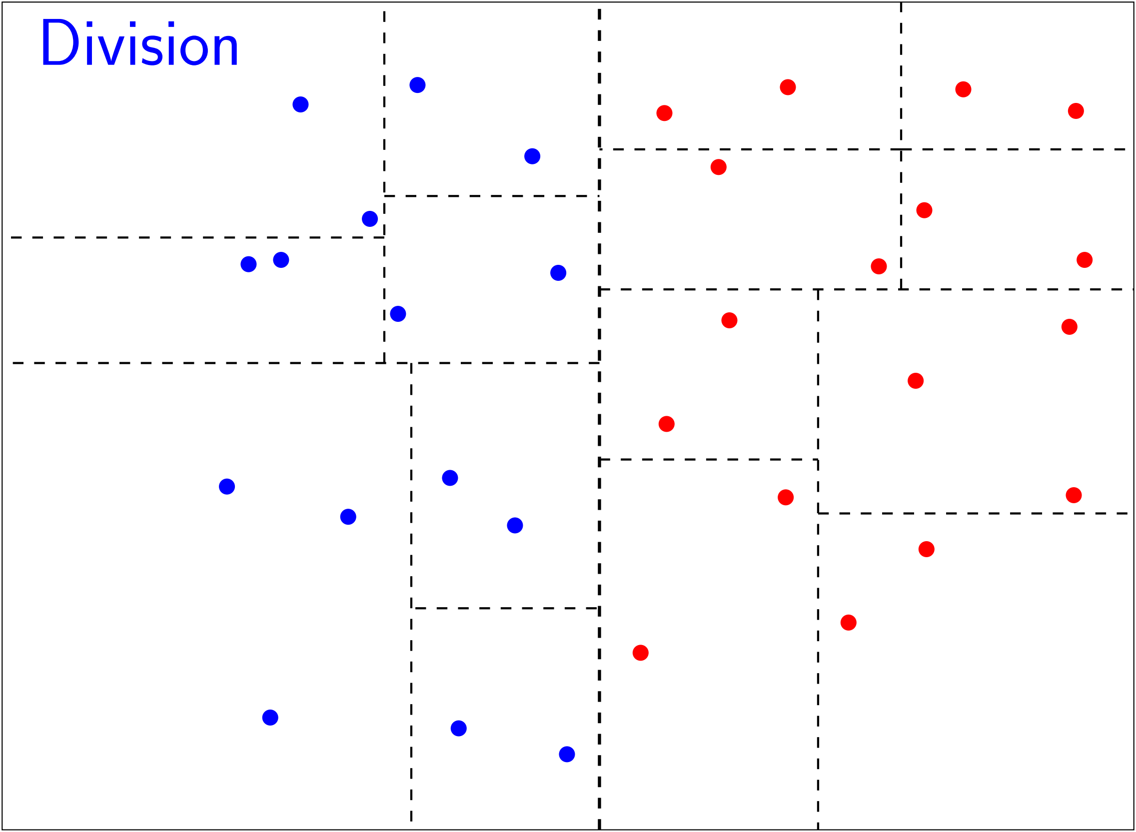

Division
Fusion

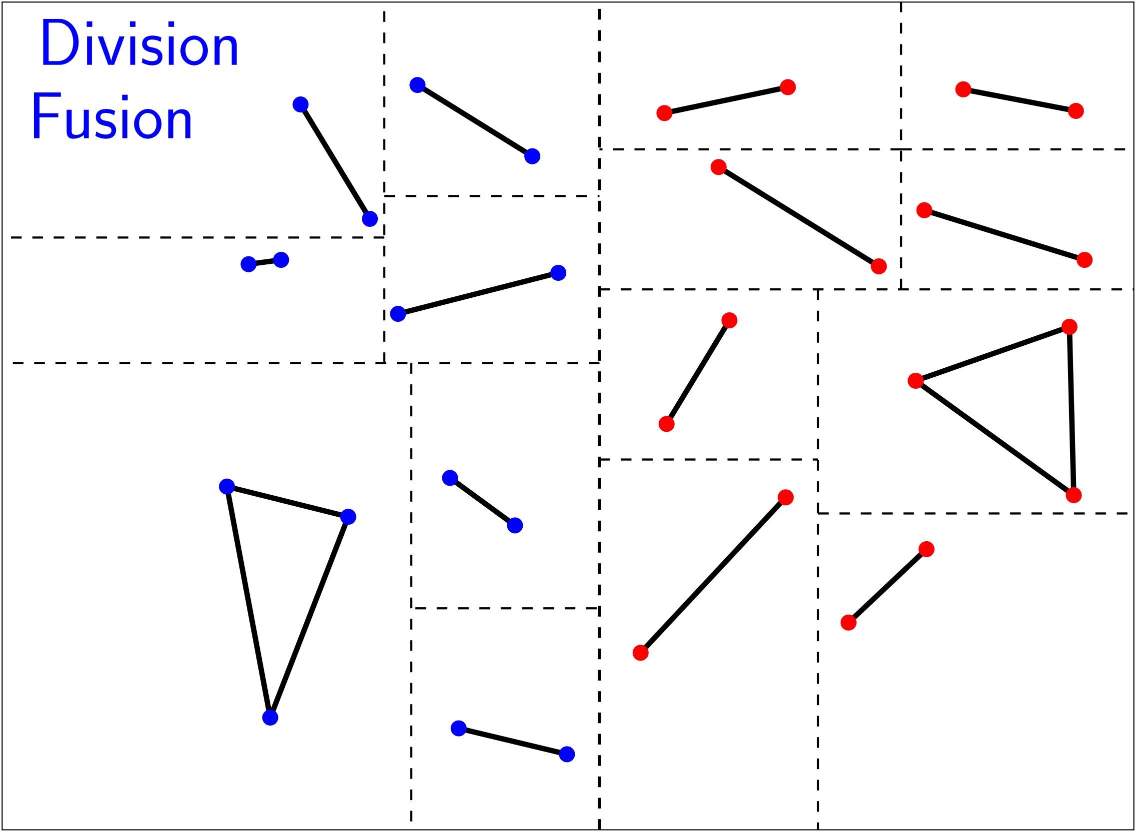

Division
Fusion

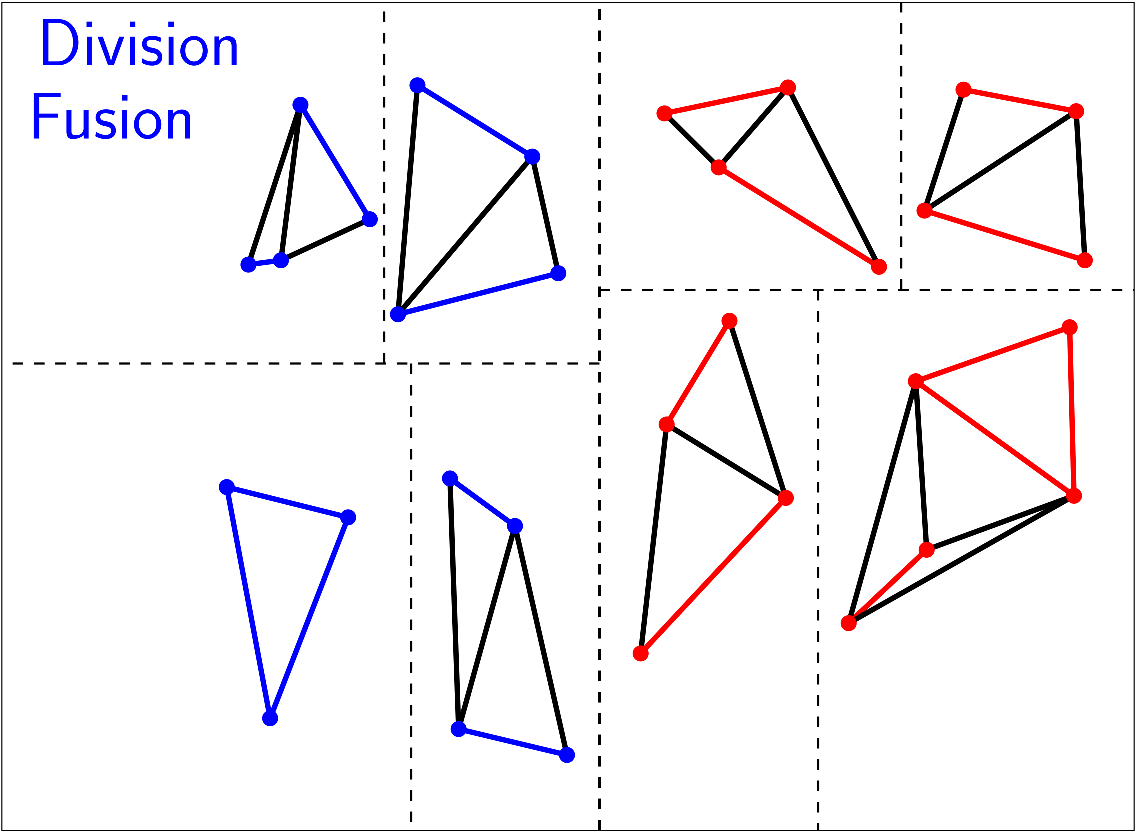

Division

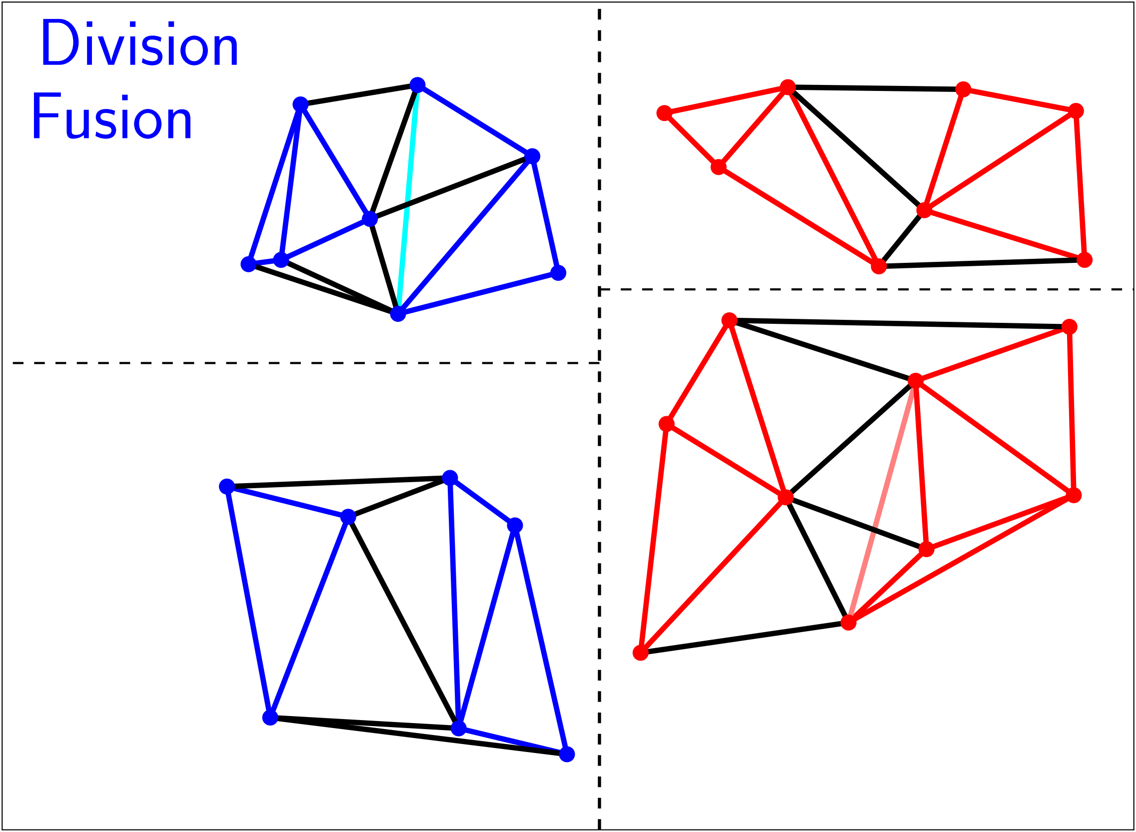

Division

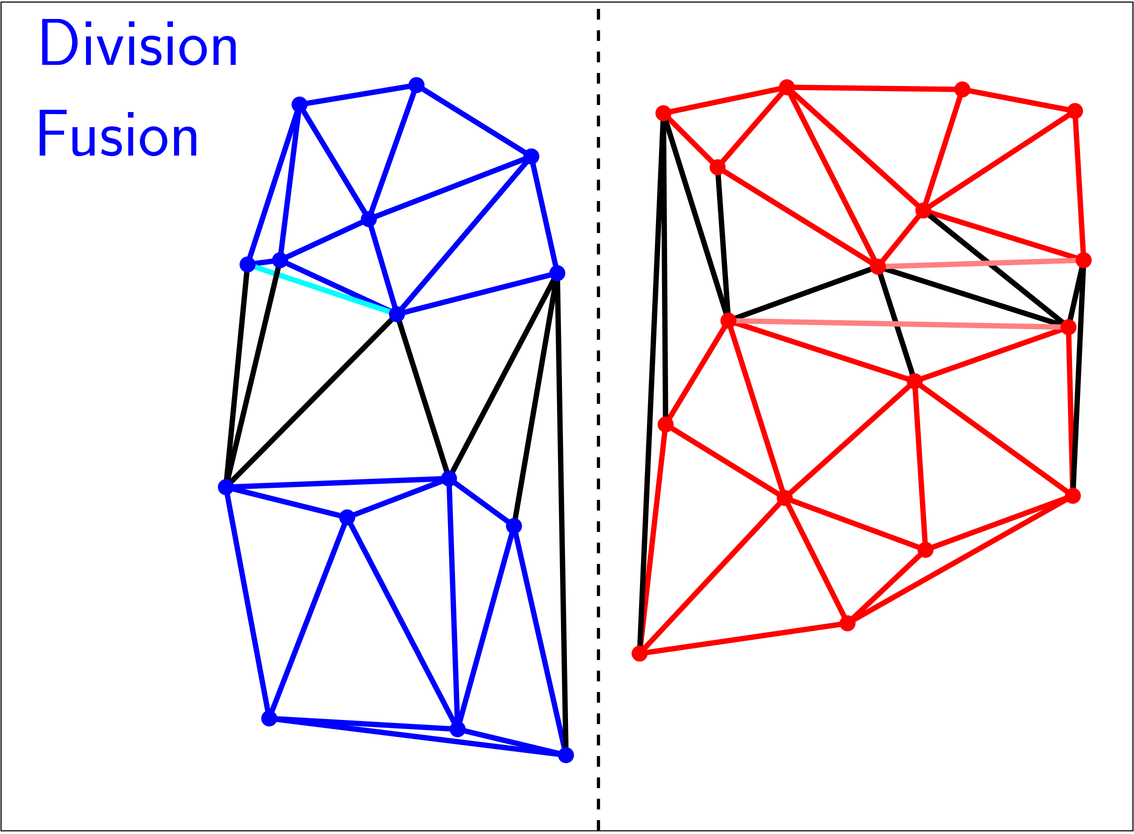

Division

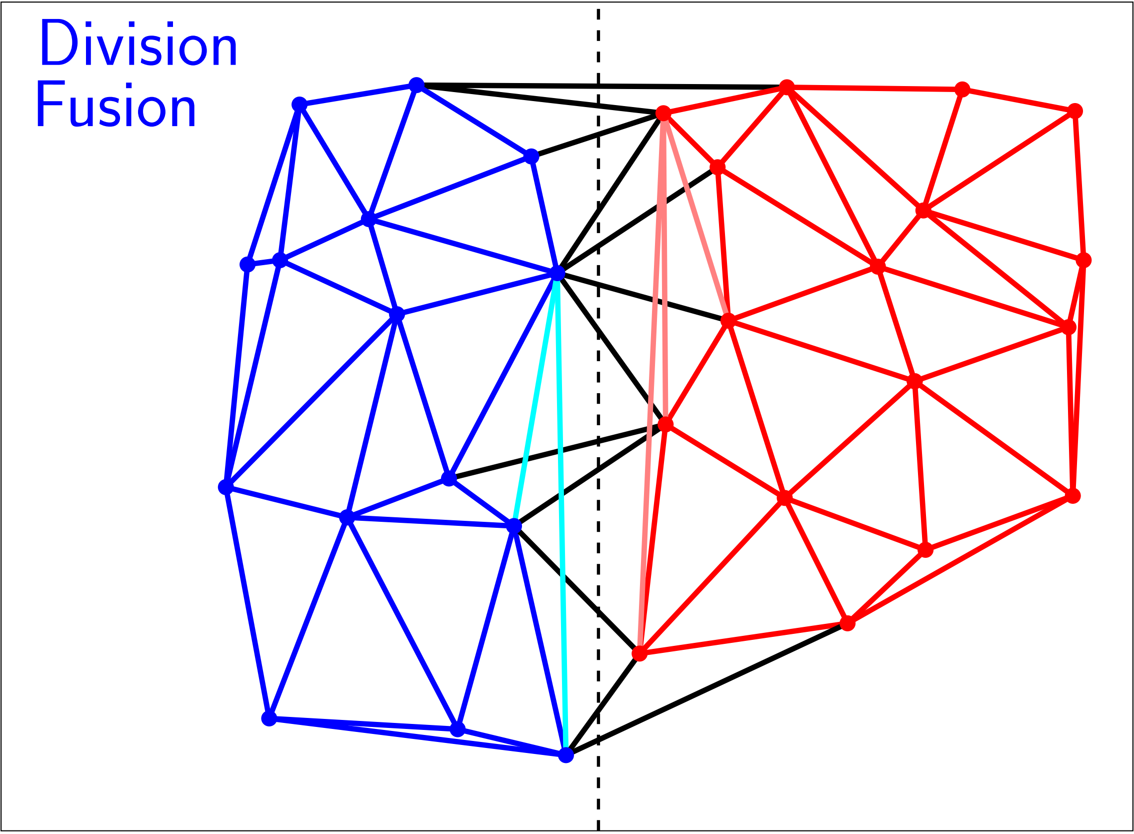

Division


Division
Fusion


Division
Fusion


Division
Fusion


Division
Fusion


Division
Fusion

Complexité

Complexité

Fusion

Complexité

Fusion $O(n)$

Complexité

Fusion $O(n)$

Division ?

Complexité

Fusion $O(n)$

Division ? médian ?

Complexité

Fusion $O(n)$

Division ? médian ?

plus difficile

Complexité

Fusion $O(n)$

Division ? médian ?

$O(n)$

Complexité

		Points aléatoires
Fusion	$O(n)$	$O(\sqrt{n})$
Division ?	médian ?	$O(1)$
	$O(n)$	

$$f(n) = O(\sqrt{n}) + 2f(\tfrac{n}{2})$$

$$f(n) = \sqrt{n} + 2f(\tfrac{n}{2})$$

$$\sqrt{n} + 2\left(\sqrt{\tfrac{n}{2}} + f(\tfrac{n}{4})\right)$$

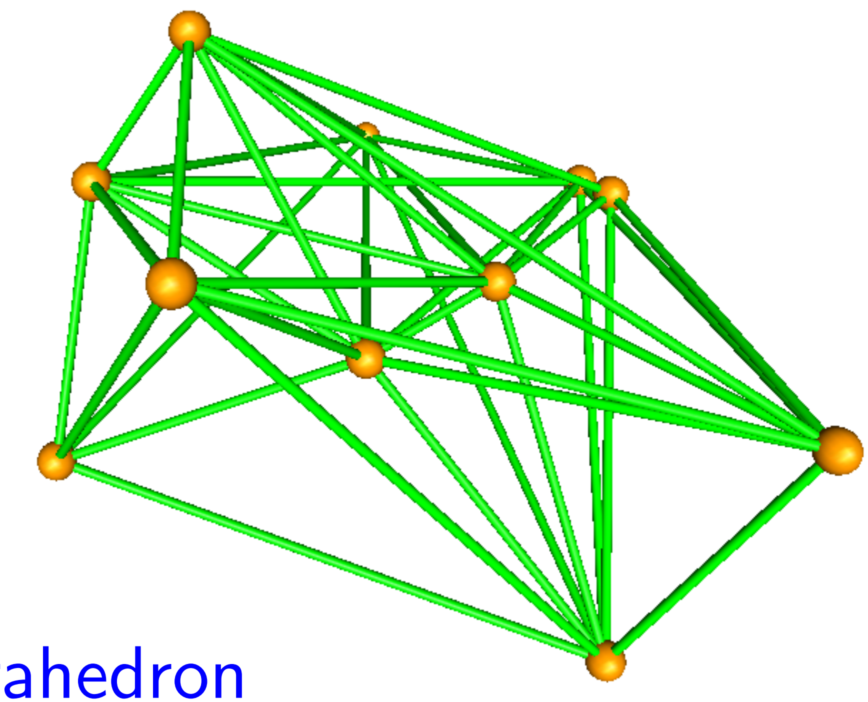
$$\sqrt{n} + 2\left(\sqrt{\tfrac{n}{2}} + 2\left(\sqrt{\tfrac{n}{4}} + f(\tfrac{n}{8})\right)\right)$$

$$\underbrace{\sqrt{n} + 2\sqrt{\tfrac{n}{2}} + \ldots + \tfrac{n}{2}\sqrt{2} + n\sqrt{1}}_{\log_2 n}$$

$$f(n) \leq n + 2^{-\frac{1}{2}}n + \ldots + 2^{-\frac{i}{2}}n + \ldots$$

$$\boxed{f(n) = O(n)}$$

Delaunay Triangulation: 3D



Same as 2D

Dual Voronoi diagram

Empty sphere property

Triangle ⟶ Tetrahedron

Duality with 4D convex hull

Incremental algorithm (find the hole and star)

Delaunay

Same as 2D

Dual

Empty

Tri

Convex hull

Higher dimensions

Dehn Sommerville relations $f_i = \sharp(\text{faces of dim } i)$

Euler: $f_0 - f_1 + f_2 - \ldots f_{d-1} = (-1)^{d-1} + 1$

$$\sum_j = k^{d-1} - 1^j \binom{j+1}{k+1} f_j = (-1)^{d-1} f_k$$

$-1 \leq k \leq d-2$ $\qquad f_{-1} = f_d = 1$

$\left\lfloor \frac{d+1}{2} \right\rfloor$ independent equations

Duality with 4D convex hull

Incremental algorithm (find the hole and star)

Delaunay

Same as 2D

Dual

Empty

Tri

Convex hull

Higher dimensions

Dehn Sommerville relations $f_i = \sharp(\text{faces of dim } i)$

Euler: $f_0 - f_1 + f_2 - \dots f_{d-1} = (-1)^{d-1} + 1$

$$\sum_j = k^{d-1} - 1^j \begin{pmatrix} j+1 \\ k+1 \end{pmatrix} f_j = (-1)^{d-1} f_k$$

$-1 \leq k \leq d-2$ $\qquad f_{-1} = f_d = 1$

$\left\lfloor \frac{d+1}{2} \right\rfloor$ independent equations

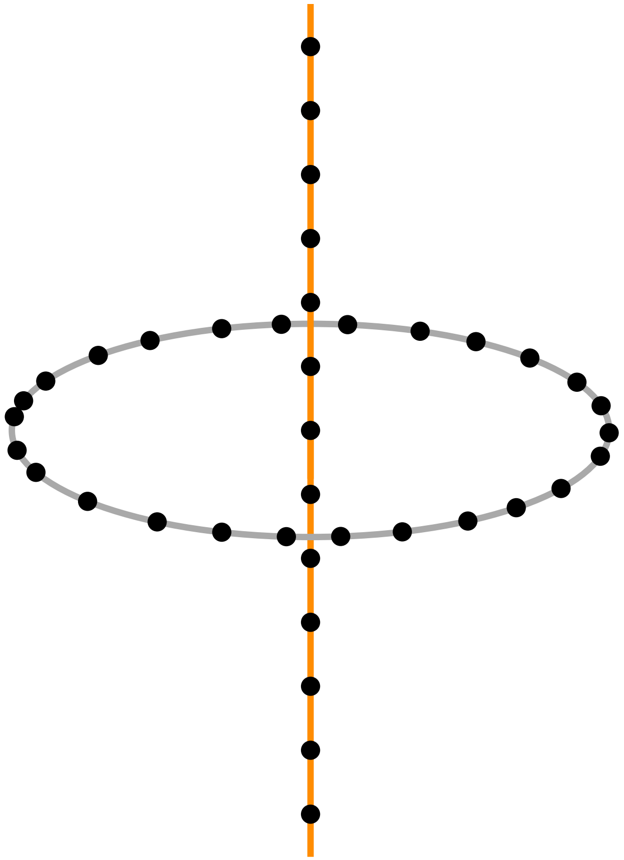
Duality with 4D convex hull

quadratic

Incremental algorithm (find the hole and star)

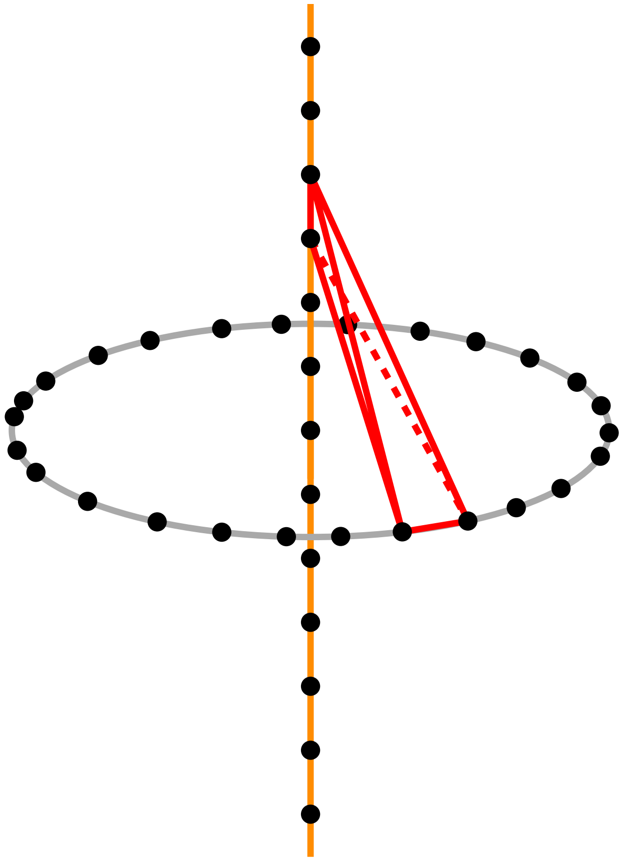
Delaunay Triangulation: 3D

Quadratic examples



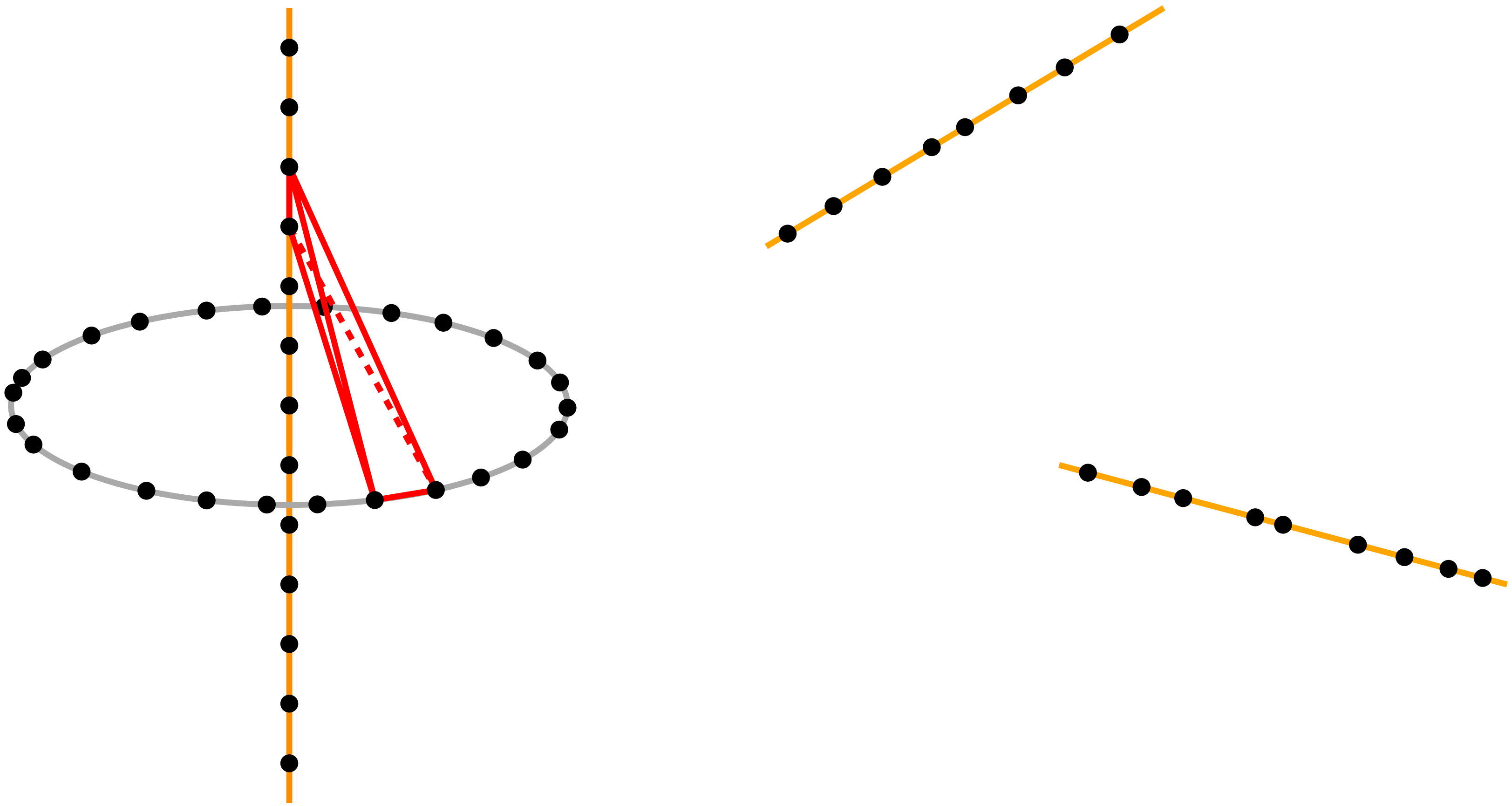
Delaunay Triangulation: 3D

Quadratic examples



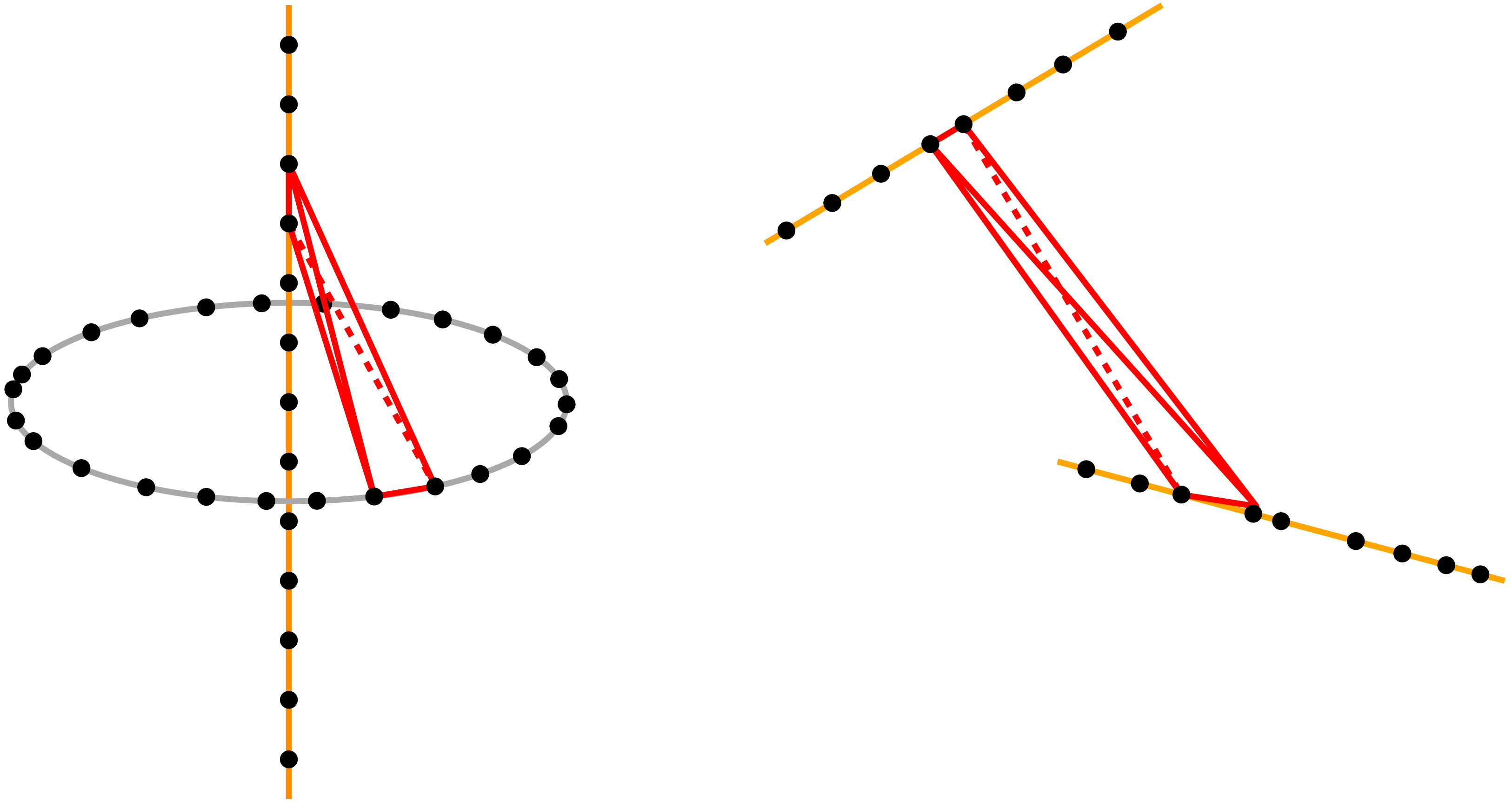
Delaunay Triangulation: 3D

Quadratic examples



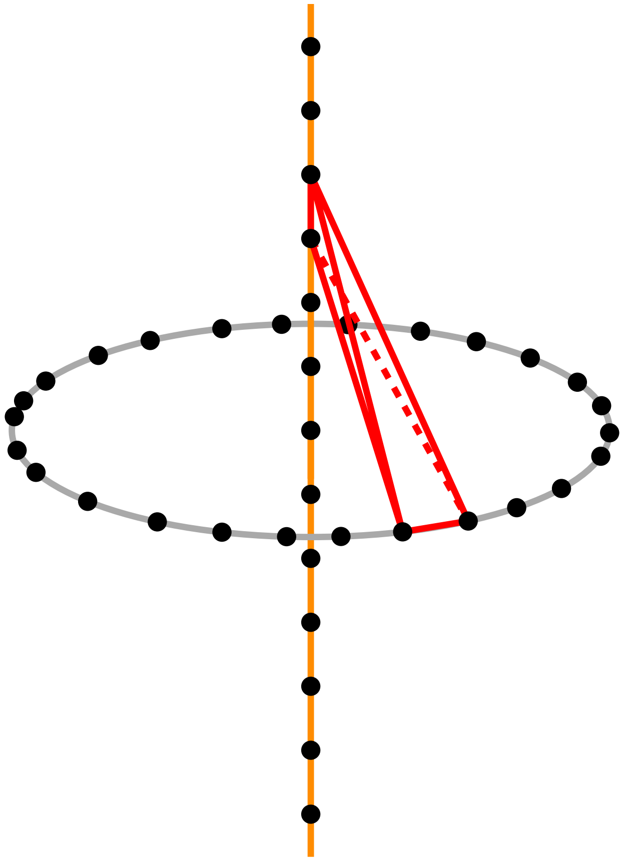
Delaunay Triangulation: 3D

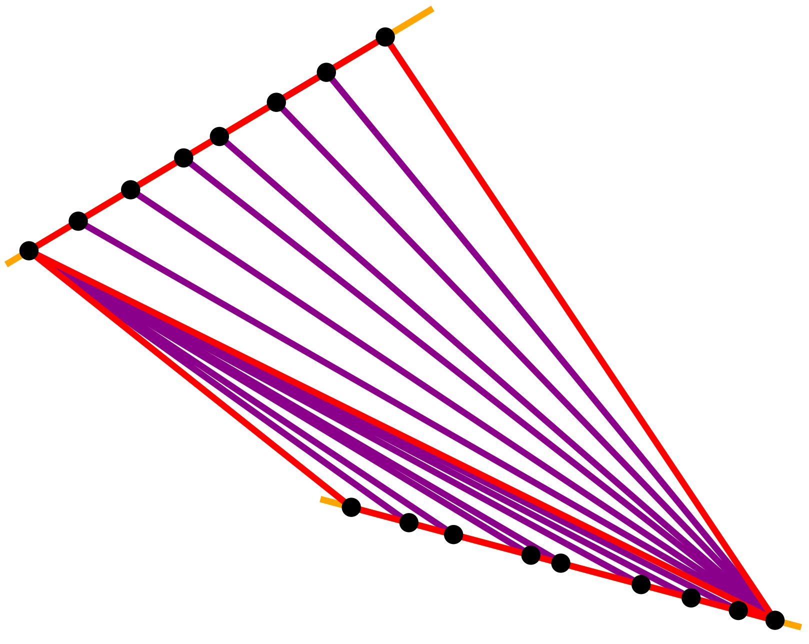
Quadratic examples



Delaunay Triangulation: 3D

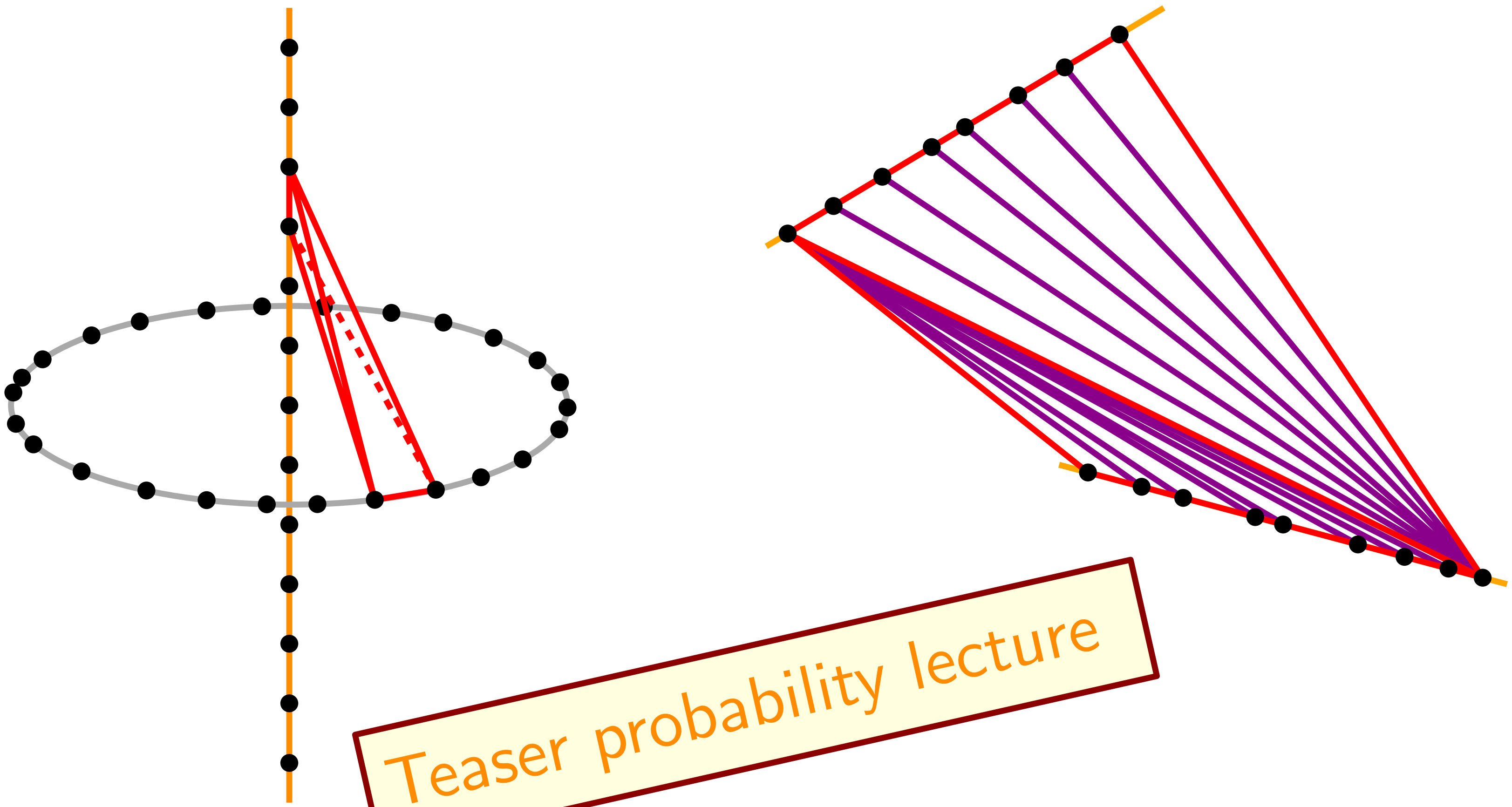
Quadratic examples





Delaunay Triangulation: 3D

Quadratic examples



Teaser probability lecture

Better results for random points

Delaunay Triangulation: 3D

Algorithms

4D convex hull duality

~~Flip~~

Incremental

Delaunay Triangulation: 3D

Algorithms

4D convex hull duality	$O(f \log n + n^{\frac{4}{3}})$ or $\Theta(n^2)$
~~Flip~~	
Incremental	$\Theta(n^3)$
	practical

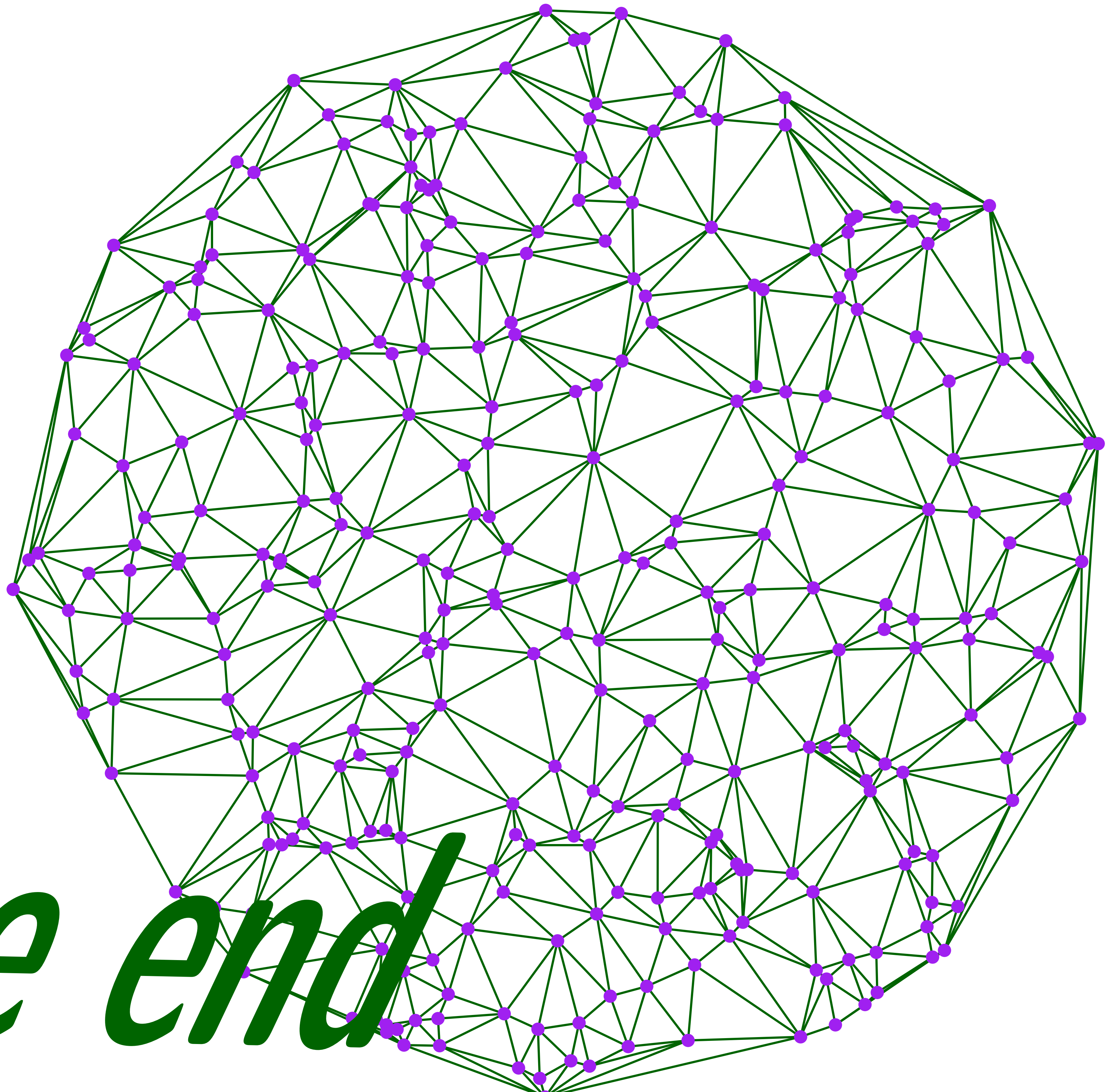
Teaser randomization lecture

Delaunay Triangulation: higher dimensions

$d+1$ convex hull duality $\quad O\left(n^{\lfloor \frac{d+1}{2} \rfloor}\right)$

Incremental practical $\quad O(n)$ for random points

coeff exponential in d



The end