OI-OPPA. European Social Fund
Prague & EU: We invest in your future.

Smallest Enclosing Circle
(nejmenší uzavírající kruh)

Luboš Vonásek

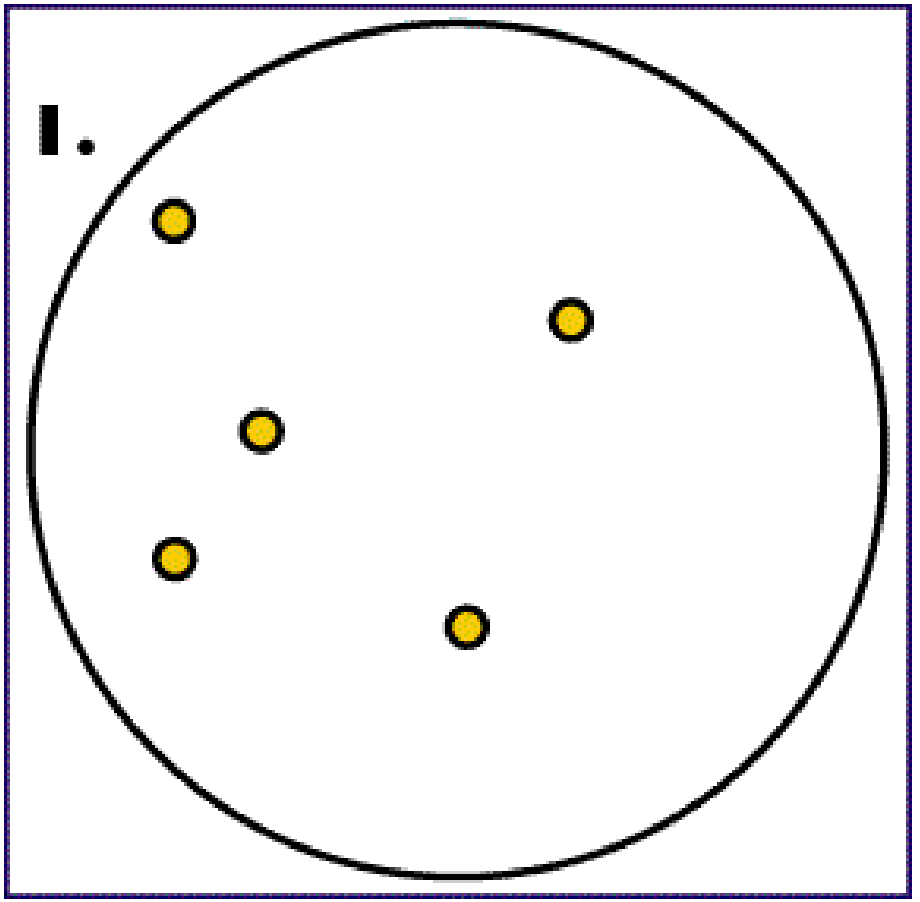
e-mail:
vonaslub@fel.cvut.cz

Smallest Enclosing Circle
(nejmenší uzavírající kruh)

- Popis problému
- Historie řešení problému
- Moderní řešení problému
- Porovnání algoritmů

Popis problému

- Množina bodů, hledání centra blízké pro všechny body



- V praxi:
- Kam dát bombu
- Kde postavit poštu
- Kde postavit nemocnici

[1]

Historie řešení problému

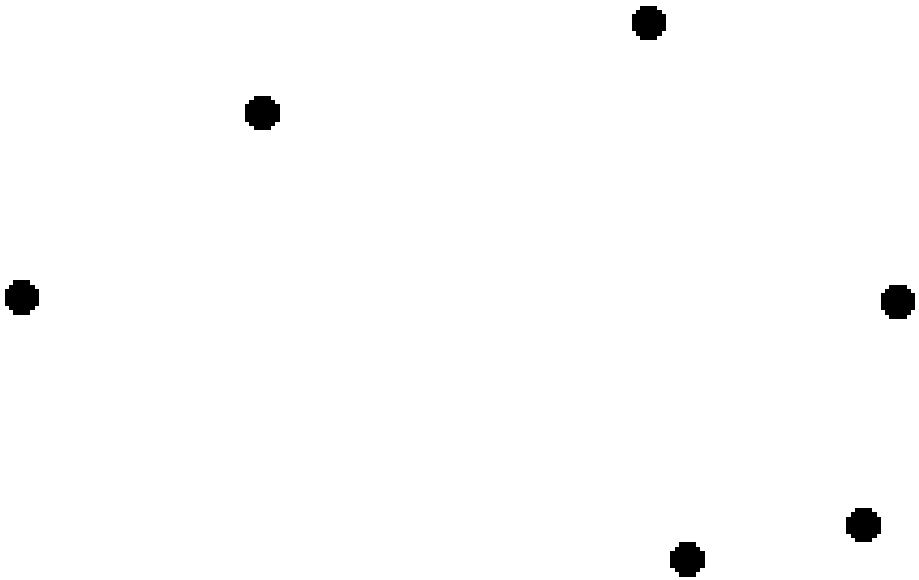
- <1970 - algoritmus neznámého autora $O(n^4)$
- 1972 - Elzinga and Hearn $O(n^2)$
- 1975 - Shamos and Hoey O(n x log(n))
- 1977 - Preparata O(n x log(n))
- 1978 - Shamos O(n x log(n))
- 1983 - Megiddo O(n)
- 1990 - Ritter O(n)

Moderní řešení problému
Ritterův algoritmus

- Lineární časová složitost
- Snadno rozšířitelné do 3D, 4D
- Výsledkem není optimální řešení, ale řešení velmi blízké optimálnímu

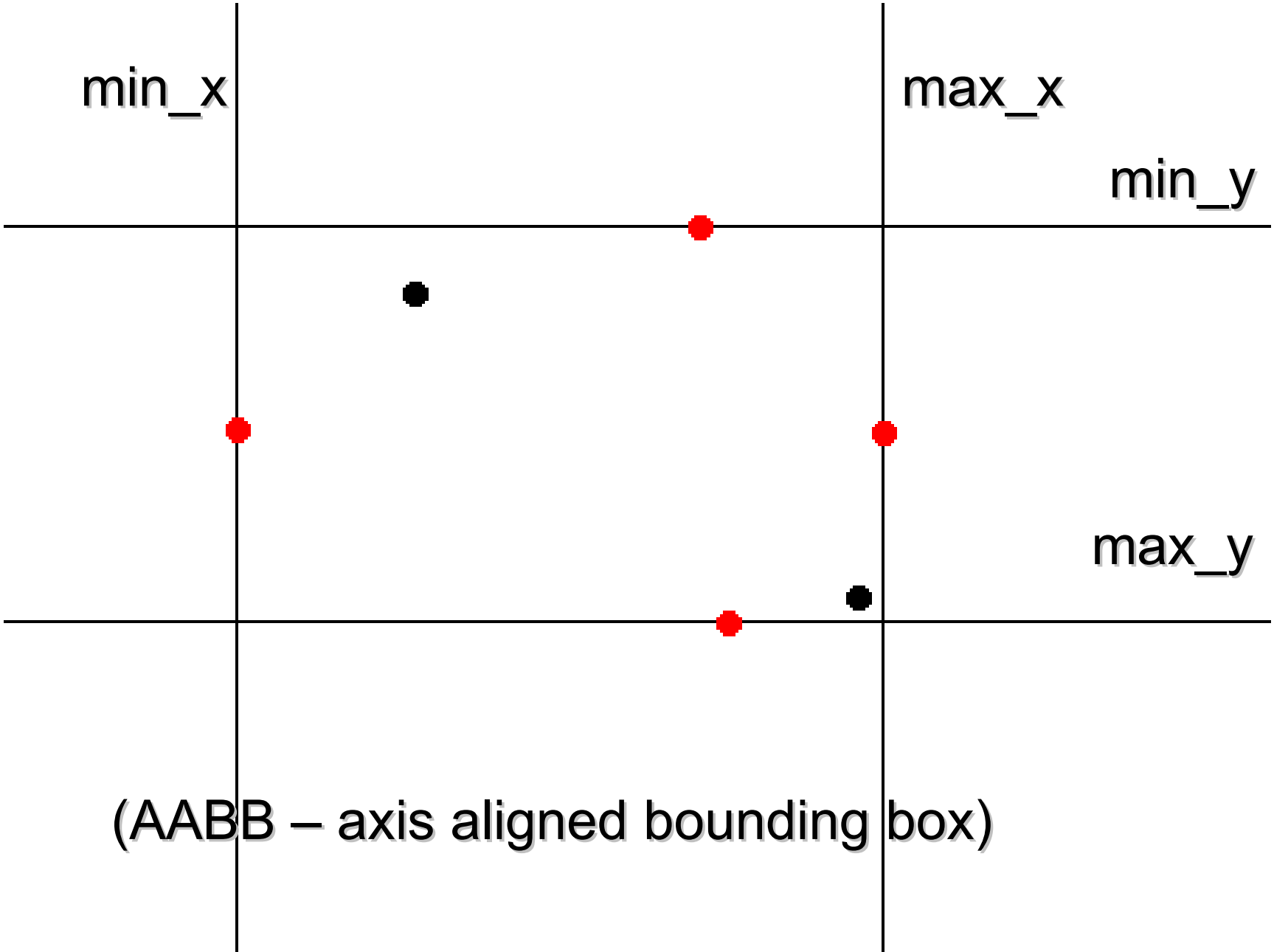
Moderní řešení problému

Ritterův algoritmus



Moderní řešení problému

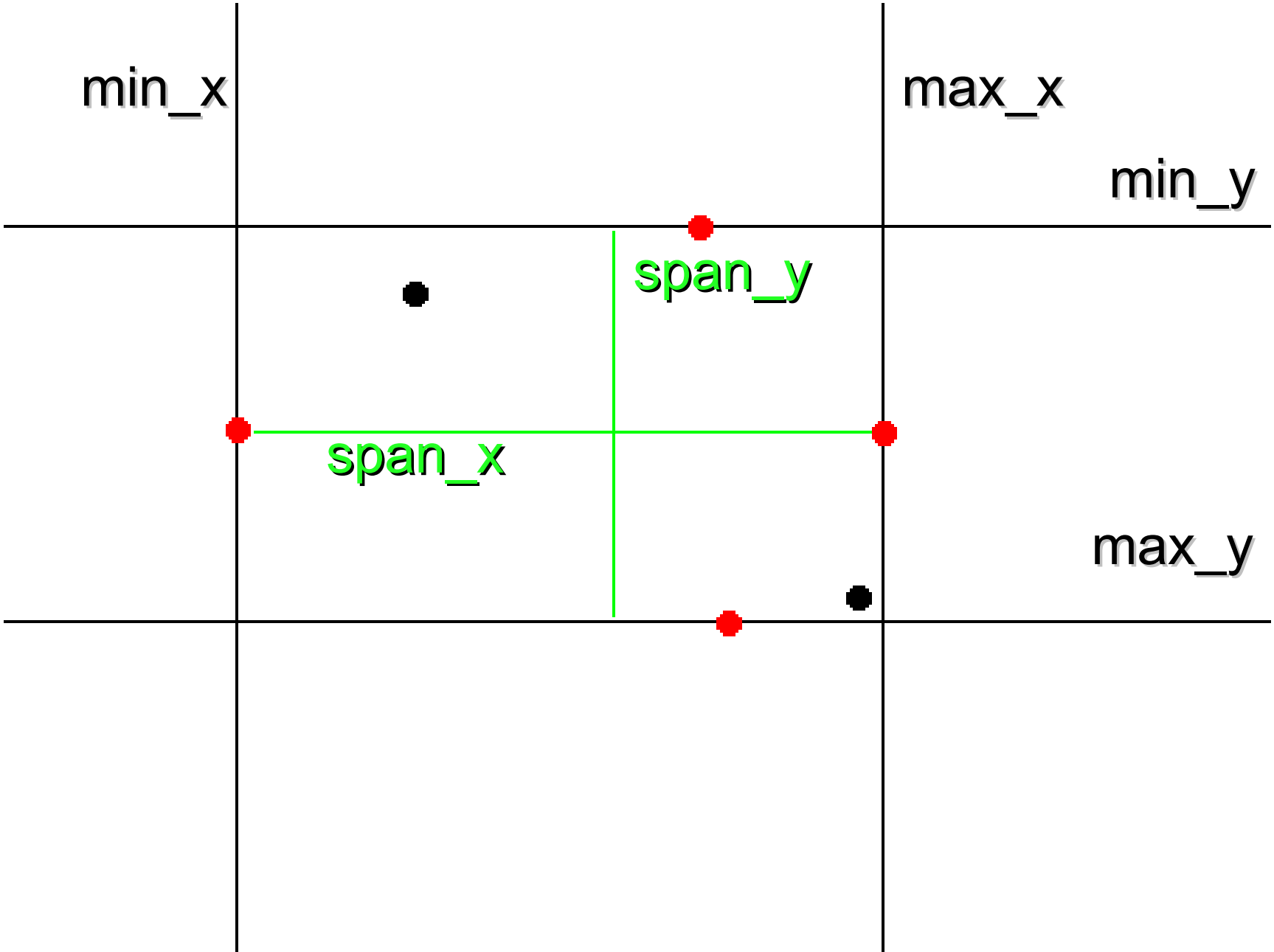
Ritterův algoritmus



(AABB – axis aligned bounding box)

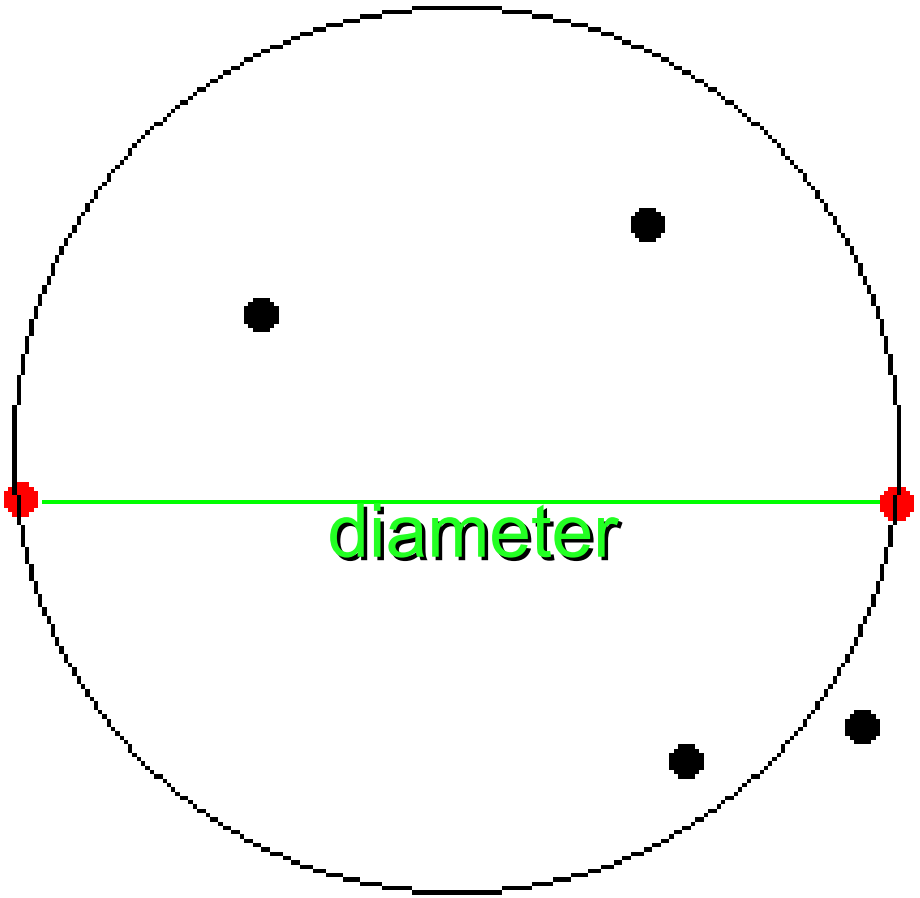
Moderní řešení problému

Ritterův algoritmus



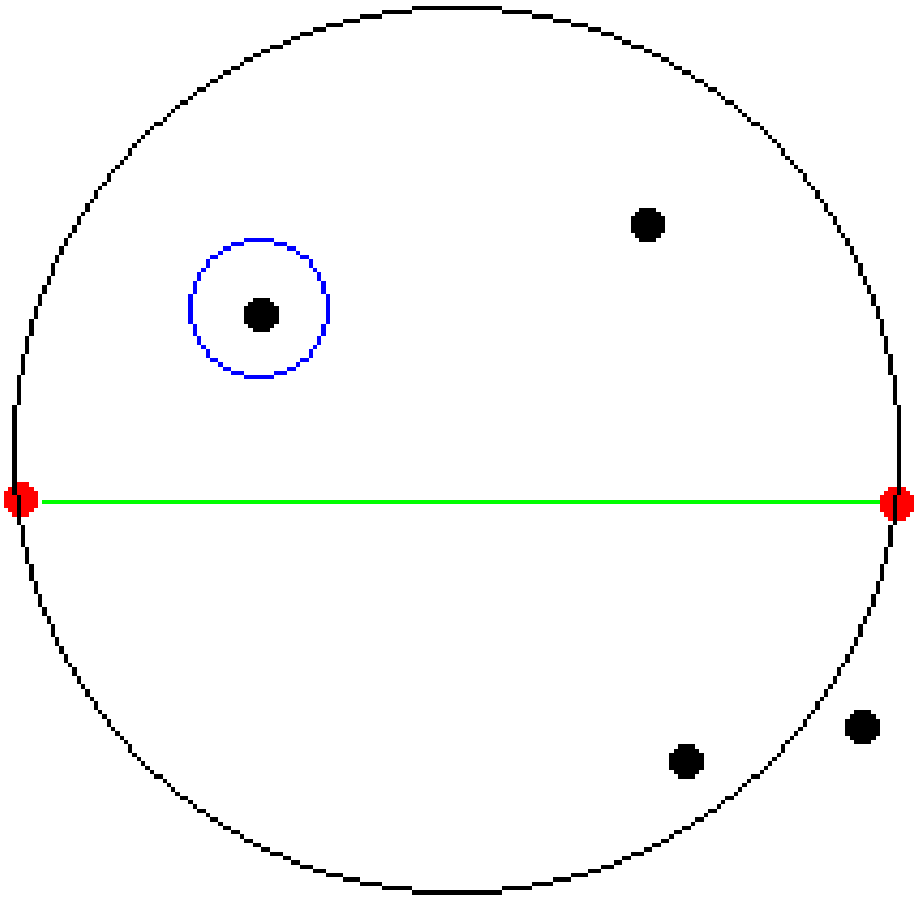
Moderní řešení problému

Ritterův algoritmus



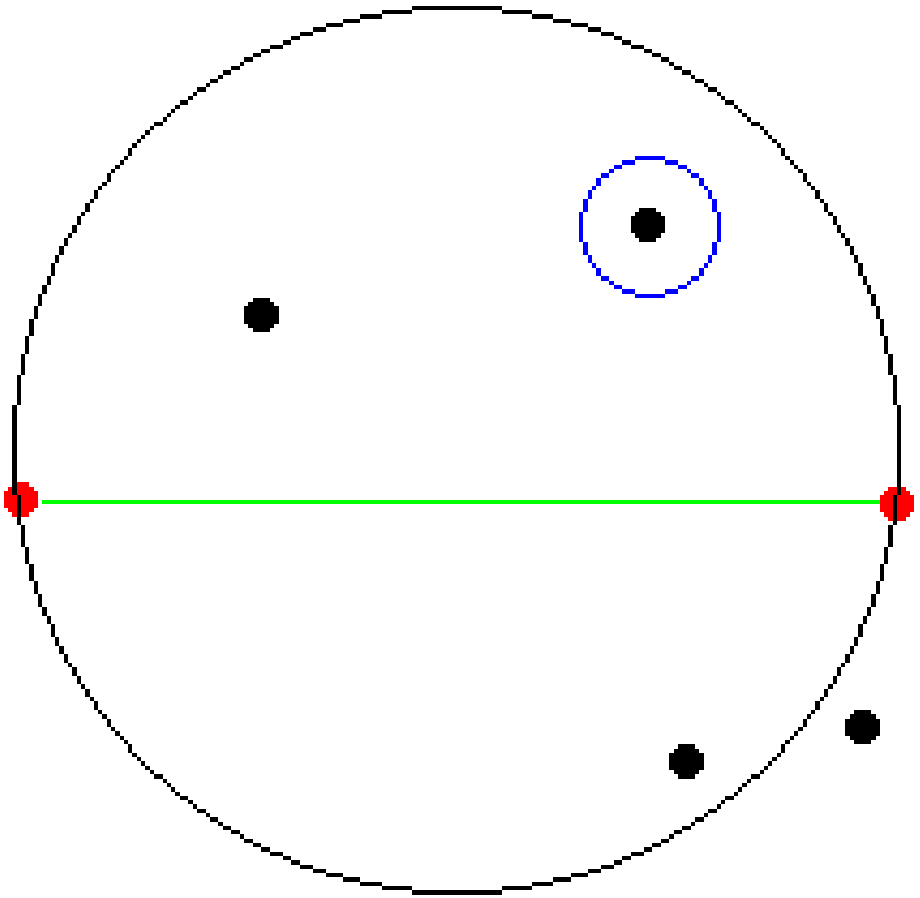
Moderní řešení problému

Ritterův algoritmus



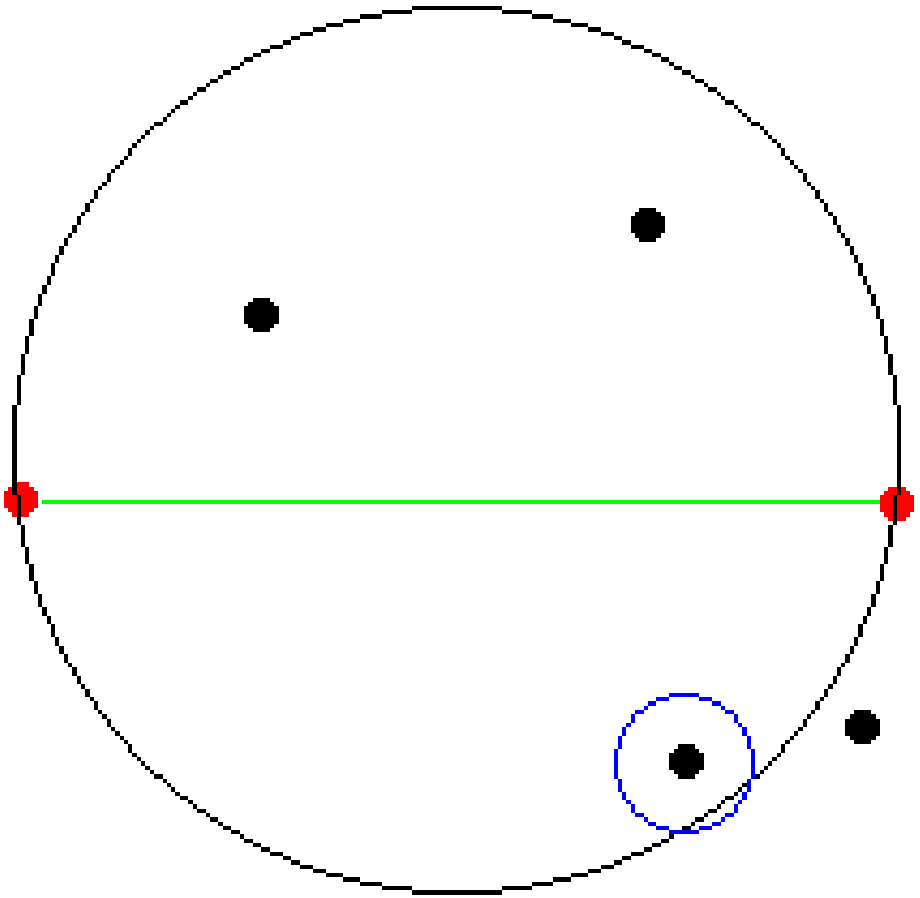
Moderní řešení problému

Ritterův algoritmus



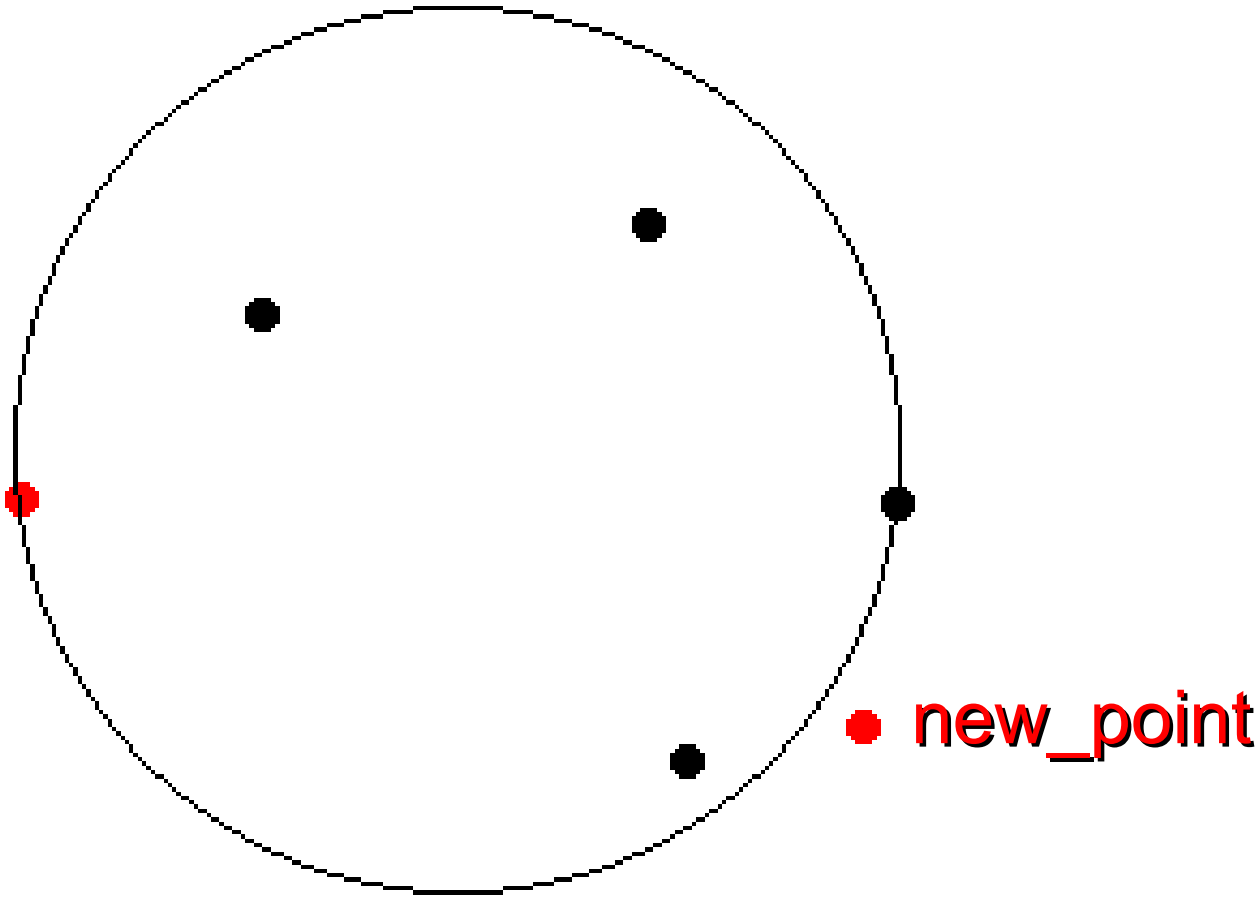
Moderní řešení problému

Ritterův algoritmus



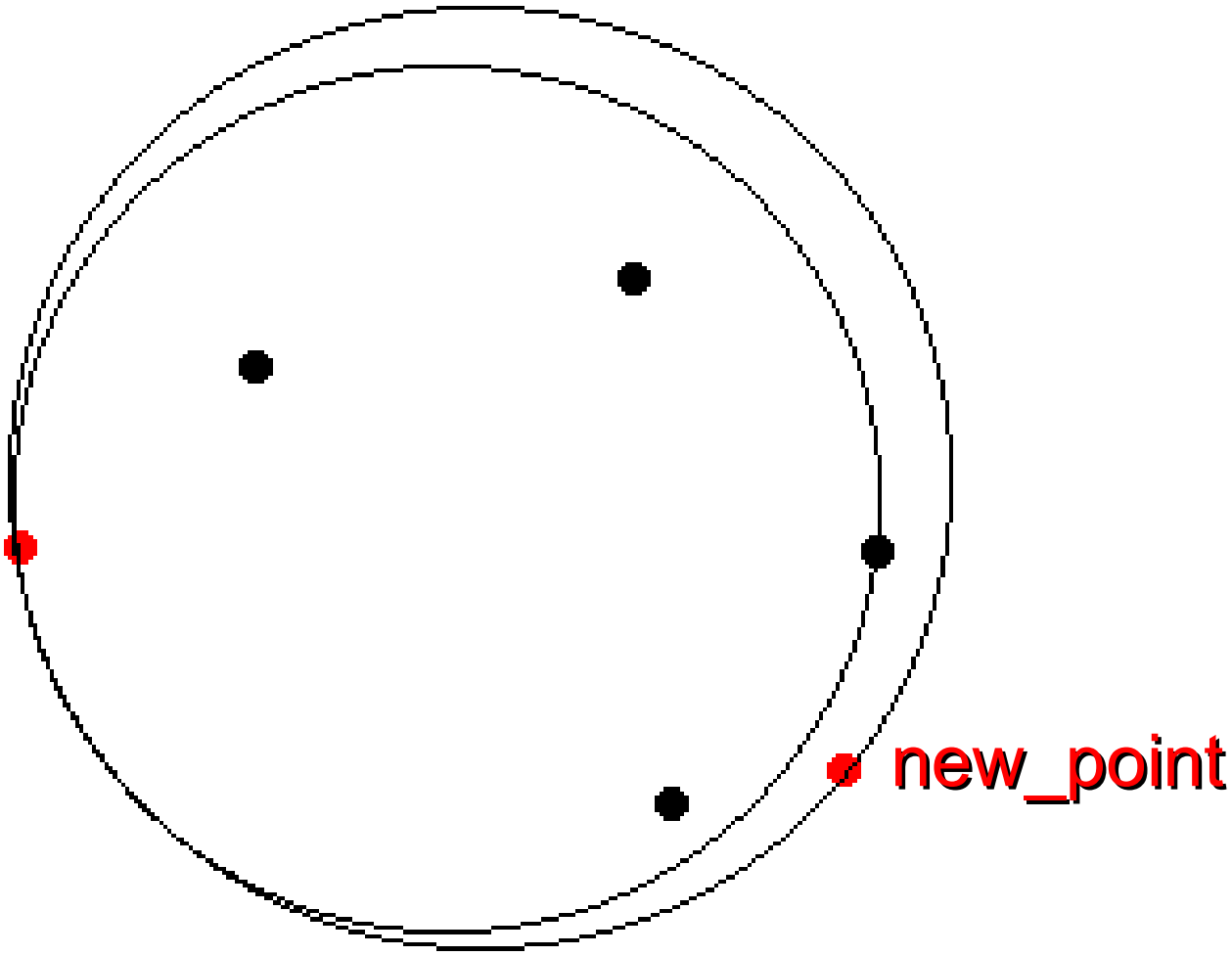
Moderní řešení problému

Ritterův algoritmus



Moderní řešení problému

Ritterův algoritmus



Moderní řešení problému
Ritterův algoritmus

Ritterův algoritmus

```
point2d<T> min_x, min_y, max_x, max_y;
for(InputIterator it = begin; it != end; ++it) {
    if (*it.x < min_x.x) min_x = *it;
    if (*it.x > max_x.x) max_x = *it;
    if (*it.y < min_y.y) min_y = *it;
    if (*it.y > max_y.y) max_y = *it;
}

T max_span = distance(max_y, min_y);
circle = make_circle(min_y, max_y);
if (distance(max_x, min_x) > distance(max_y, min_y)) {
    max_span = distance(max_x, min_x);
    circle = make_circle(min_x, max_x);
}

for(InputIterator it = begin; it != end; ++it) {
    if (lay_distance_from_point_to_circle_center(*it,circle) > sqr(circle.radius)) {
        T dist = sqrt(lay_distance_from_point_to_circle_center(*it,circle));
        circle.radius = (circle.radius + dist) * T(0.5);
        circle = make_circle((circle.radius*circle.x + (dist-circle.radius) * *it.x)/dist,
                             (circle.radius*circle.y + (dist-circle.radius) * *it.y)/dist,
                              circle.radius);
    }
}
```

Porovnání algoritmů

1983 - Megiddo O(n)	1990 - Ritter O(n)
+optimální výsledek	-neoptimální výsledek
-těžce rozšířitelné do 3D	+snadno rozšířitelné do 3D
-na webu nejspíš není zdrojový kód algoritmu	+implementace algoritmu je volně k dispozici

Zdroje

[1] http://www.cs.mcgill.ca/~cs507/projects/1998/jacob/problem.html

[2] http://www.personal.kent.edu/~rmuhamma/Compgeometry/MyCG/
CG-Applets/Center/centercli.htm

[3] http://www.codeproject.com/Articles/22568/
Computational-Geometry-C-and-Wykobi

Smallest Enclosing Circle
(nejmenší uzavírající kruh)

Děkuji za pozornost

OI-OPPA. European Social Fund
Prague & EU: We invest in your future.