15-451 Lecture 23 — Smallest Enclosing Circle

Problem: Given $n \geq 2$ points in the plane, find the smallest enclosing circle that contains these points.

Ground Rules:

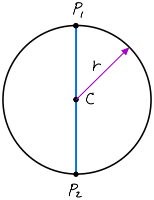
- The points are pairs of real numbers $p_i = (x_i, y_i)$.
- Arithmetic on reals can be done in $O(1)$ time.

Let $\text{SEC}(p_1, \ldots, p_n)$ denote the smallest enclosing circle of points $p_1, \ldots, p_n$.

In this lecture we present a randomized incremental algorithm to compute the $\text{SEC}(p_1, \ldots, p_n)$ in expected $O(n)$ time.

Base Cases

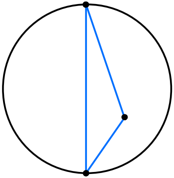
$n = 2$:



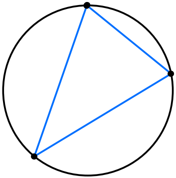
$$C = \frac{1}{2}(P_1 + P_2)$$

$$r = \frac{1}{2}|P_1 - P_2|$$

$n = 3$ Obtuse triangle:



$n = 3$ Acute triangle:

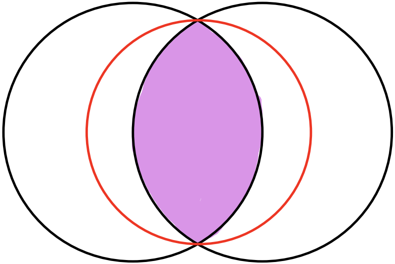


(Explain how to find the center, and why this is optimum.)

Uniqueness of the SEC

Theorem 1: The SEC($p_1, \ldots, p_n$) is unique.

Proof: By contradiction. Suppose there are two distinct SECs.



The picture shows that if there are two distinct SECs then there's a smaller SEC than the one asserted to be the smallest. □

Characterization of the SEC

If we take any pair of points and halve the distance between them, that's obviously a lower bound on the radius of the SEC.

If we take any triple of points the form an acute triangle and take the circumradius of it, that's also obviously a lower bound on the radius of the SEC.

It turns out that if we take the maximum of all these lower bounds, this will in fact equal the radius SEC. We will now prove this.

Characterization of the SEC

Theorem 2: For $n \geq 3$ the SEC$(p_1, \ldots, p_n)$ is the SEC(p_i, p_j, p_k) for some set of distinct indices i, j, k.

This means that we can find the SEC of all n points by just computing the SEC(p_i, p_j, p_k) for all i, j, k, and taking the one of maximum radius r. Because given that triple of points there is only one circle of radius r that can contain them. So by the theorem it must contain all the points.

This immediately gives an $O(n^3)$ algorithm for the problem.

Helly's Theorem

Our proof will make use of this theorem:

Helly's Theorem: Let $X_1, \ldots, X_n$ be a finite collection of convex subsets of R^d, with $n \geq d+1$. If the intersection of every $d+1$ of these subsets is non-empty, then the whole collection has a non-empty intersection; that is,

$$\bigcap_{j=1}^{n} X_j \neq \varnothing.$$

We only need this for $d = 2$ so it says that if in a collection of convex 2D sets every three intersect, then they all intersect.

Proof of Theorem 2

Proof: Compute the radius r which is the maximum of the radii of all the SECs of the triples of points p_i, p_j, p_k.

Now consider the ball B_i centered at p_i of radius r. It must be the case that $B_i \bigcap B_j \bigcap B_k \neq \varnothing$ for all i, j, k.

Therefore it follows from Helly's theorem that all the balls intersect. Pick the circle with radius r whose center is in the mutual intersection of all the balls. By construction all the points must be in that circle.

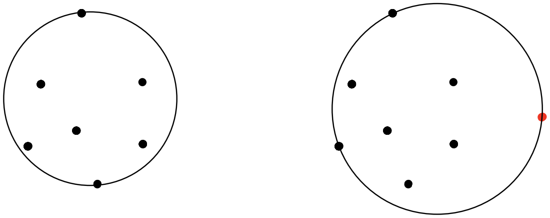
Furthermore let (p_i, p_j, p_k) be three points whose SEC has radius r. This SEC is the unique circle of radius r containing p_i, p_j, p_k. Therefore it must also be the circle found by the application of Helly's Theorem.

□

Towards an Incremental Algorithm

Suppose we've computed the SEC of points $p_1, \ldots p_{n-1}$. What happens when we add another point p_n to the set?

There are two cases. (1) p_n is in the SEC of the previous $n-1$ points. (2) It's not. Here's an example:



It turns out that in case (2) the new point must be on the boundary of the SEC of all the points.

Towards an Incremental Algorithm

So we have the following theorem:

Theorem 3: For a set of points $\{p_1, \ldots, p_n\}$, let $C_1 = \text{SEC}(p_1, \ldots, p_{n-1})$ and $C_2 = \text{SEC}(p_1, \ldots, p_n)$. If $C_1 \neq C_2$ then p_n is on the boundary of C_2.

Proof: By the uniqueness of the SEC (Theorem 1) we know that $\text{radius}(C_1) < \text{radius}(C_2)$. But we know from Theorem 2 that these radii are just the maximum over all triples of points in their respective sets. The only way that C_2 can have a larger radius than C_1 is if p_n is involved in the triple causing that to happen. Therefore p_n must be on C_2. □

A Randomized Incremental Algorithm

```
SEC([p_1, p_2, ..., p_n]) = {
    Randomly permute the input points, so [p_1, ..., p_n]
    is a random permutation of the given points.

    Let C be the smallest circle enclosing p_1 and p_2.
    (This is just the circle for which p_1 and p_2 form a diameter.)

    for i = 3 to n do
        // at this point C is the smallest enclosing circle for [p_1, ..., p_{i-1}]
        if p_i is not in C then C ← SEC1([p_1, ..., p_{i-1}], p_i)
    done

    return C
}
```

Here $\mathrm{SEC1}([p_1, \ldots, p_{i-1}], p_i)$ computes the SEC of $\{p_1, \ldots, p_i\}$, and can make use of the fact that p_i is on the boundary of the SEC of $\{p_1, \ldots, p_i\}$.

Analyzing the Algorithm

Assuming that SEC1 (called on a set of i points) is $O(i)$ expected time, then SEC$(p_1, \ldots, p_n)$ is $O(n)$ expected time.

Proof: Backward analysis. Recall from Theorem 2 that there are two or three points that determine the SEC of all the points. Unless we choose to delete one of these points the SEC will not change. So $C_i \neq C_{i-1}$ with probability at most $\frac{3}{i}$.

$$E[\text{Cost of step } i] = O(i) * \frac{3}{i} + O(1) * \frac{i-3}{i} = O(1)$$

□

SEC1

```
SEC1([p_1, p_2, ..., p_n], q) = {
    // We know that the point q is on the SEC containing p_1, ... p_n, q.
    Randomly permute the input points, so [p_1, ..., p_n]
    is a random permutation of the given points.

    Let C be the smallest circle enclosing p_1 and q.

    for i = 2 to n do
        // At this point C is the smallest enclosing circle of [p_1, ..., p_{i-1}, q]
        // that also passes through q.
        if p_i is not in C then C ← SEC2([p_1, ..., p_{i-1}], p_i, q)
    done

    return C
}
```

Here SEC2($[p_1, \ldots, p_{i-1}], p_i, q$) computes the smallest circle through p_i and q containing $\{p_1, \ldots, p_{i-1}\}$ It is $O(i)$ time.

Proof that it's $O(i)$ expected time is the same as that for SEC.

SEC2

```
SEC2([p_1, p_2, ..., p_n], q_1, q_2) = {
    // The job of SEC2 is to compute the smallest circle containing
    // [p_1, ... p_n, q_1, q_2] that passes through q_1 and q_2.
    // We know such a circle exists.

    Let C be the smallest circle enclosing q_1 and q_2.

    for i = 1 to n do
        // At this point C is the smallest enclosing circle of [p_1, ..., p_{i-1}, q_1, q_2]
        // that also passes through q_1 and q_2
        if p_i is not in C then C ← Circle through points (p_i, q_1, q_2)
    done

    return C
}
```

This is a deterministic linear time algorithm.

Final Notes

It can be proven that there is no need to compute a random permutation inside of SEC1. One random permutation at the beginning in SEC sufficies.

The code can be consolidated into a single recursive function, where an additional argument passes the points that must be on the boundary.

The same algorithm can be easily extended to d dimensions. In this case, as in Seidel's algorithm, the bound is $O(n\, d!)$, so the dependence on n is linear and the dependence on d is exponential.