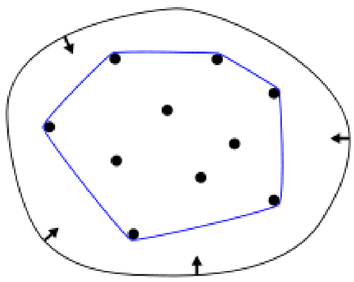
Convex Hull Algorithms

Hariank Muthakana

April 10, 2015

1 Introduction

The *convex hull* problem is an important computational geometry problem. Given a set of points in a plane, we want to find a convex polygon that contains all of them. A useful analogy is visualizing the points as "pins" and wrapping a rubber band around them:

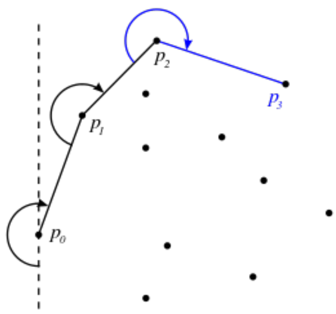


Convex hulls have many geometric applications, but also have uses in optimization, image processing, and even quantum computing.

2 Gift Wrapping

The first, and (conceptually) simplest convex hull algorithm is the *gift wrapping* algorithm, also known as the Jarvis march.

We start with a point we know is on the hull - for example, the leftmost point. This is the current point, p_i. At each step, we consider all other points in the set and find the polar angles of the segments formed. Using these angles, we pick the point p_{i+1} such that all the other points are to the right of the segment.

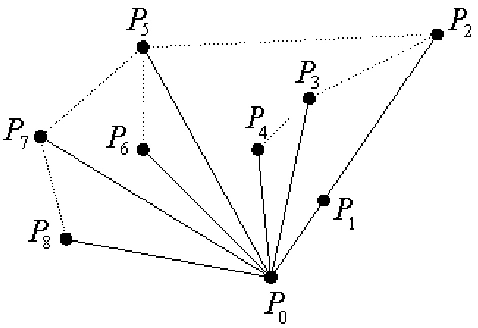


The gift wrapping algorithm gets its name from the way the points are processed, like wrapping a string around the set. This process is continued until the original leftmost point is reached again. At each step, all the other points must be considered in $O(n)$. If we call the number of points in the hull h, the algorithm has a total complexity of $O(nh)$.

3 Graham Scan

Gift wrapping is simple, but it is quadratic in the worst case. The *Graham scan* algorithm is more complicated but allows for a better time complexity.

In Graham scan, we will again be working with angles. The first step is to pick an "origin". This is the point we will be drawing segments from - we can use either the lowest point in the set, or a "center point" found by averaging all the points. We then find the angles to all the other points and sort them in counterclockwise order.

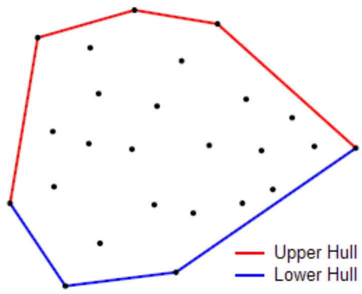


Iterating through the points in this order, we consider the angle formed with the previous two points. If the angle $p_i p_{i-1} p_{i-2}$ is not valid, we delete p_{i-1} and check again. Once the correct angle is found, we increment i. Note that this condition can be computed quickly using cross products. We continue until the original point is reached, checking the angle $p_n p_0 p_1$.

The initial angle sort takes $O(n \log n)$ time. Surprisingly, the "turn checking" loop is only $O(n)$, since each point is only processed at most twice. Therefore, the overall complexity is $O(n \log n)$

4 Monotone Chain

Another method to find the convex hull is the *monotone chain* algorithm. This is similar to Graham scan, but there are two key difference - sorting by x-coordinate rather than angle, and building the upper and lower hulls separately.



We begin by sorting the points left to right. Consider the lower hull first. Iterating from left to right, we run Graham scan, eliminating points that contribute clockwise turns. For the upper hull, we iterate from *right to left* and do the same thing. Finally, we remove the repeated leftmost and rightmost points.

Although we have to consider both halves separately, monotone chain allows us to avoid calculating angles. Monotone chain is the same complexity as Graham scan, $O(n \log n)$, for similar reasons.

5 Output Sensitivity

We saw that gift wrapping had an interesting difference from both Graham scan and monotone chain - the complexity involved h, the number of points on the hull. Gift wrapping is therefore an *output-sensitive* algorithm, because its complexity is dependent on the size of the output.

This distinction is important when considering random sets of points - for many practical situations, there are many unused points (that is, h is much less than n) . Some other convex hull algorithms such as Kirkpatrick-Seidel and Chan's algorithm are also output-sensitive, and have an optimal complexity of $O(n \log h)$.

6 Problems

1. (Training Pages) Given a collection of polygon houses, calculate the minimum length boundary that encloses all but at most one of them.

2. (USACO JAN14 Gold) Cow curling is a popular cold-weather sport played in the Moolympics. Like regular curling, the sport involves two teams, each of which slides N heavy stones ($3 \leq N \leq 50,000$) across a sheet of ice. At the end of the game, there are $2N$ stones on the ice, each located at a distinct point.

   Scoring in the cow version of curling is a bit curious, however. A stone is said to be "captured" if it is contained inside a triangle whose corners are stones owned by the opponent (a stone on the boundary of such a triangle also counts as being captured). The score for a team is the number of opponent stones that are captured. Compute the final score of a cow curling match, given the locations of all $2N$ stones.