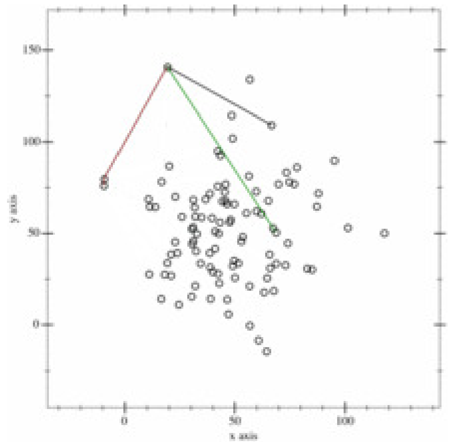
Gift wrapping algorithm

From Wikipedia, the free encyclopedia

In computational geometry, the **gift wrapping algorithm** is an algorithm for computing the convex hull of a given set of points.



Animation of the gift wrapping algorithm. The red lines are already placed lines, the Black line is the current best guess for the new line, and the green line is the next guess

Contents

- 1 Planar case
- 2 Algorithm
- 3 Pseudocode
- 4 Complexity
- 5 References

Planar case

In the two-dimensional case the algorithm is also known as **Jarvis march**, after R. A. Jarvis, who published it in 1973; it has O(nh) time complexity, where n is the number of points and h is the number of points on the convex hull. Its real-life performance compared with other convex hull algorithms is favorable when n is small or h is expected to be very small with respect to n. In general cases the algorithm is outperformed by many others.

Algorithm

For the sake of simplicity, the description below assumes that the points are in general position, i.e., no three points are collinear. The algorithm may be easily modified to deal with collinearity, including the choice whether it should report only extreme points (vertices of the convex hull) or all points that lie on the convex hull. Also, the complete implementation must deal with degenerate cases when the convex hull has only 1 or 2 vertices, as well as with the issues of limited arithmetic precision, both of computer computations and input data.

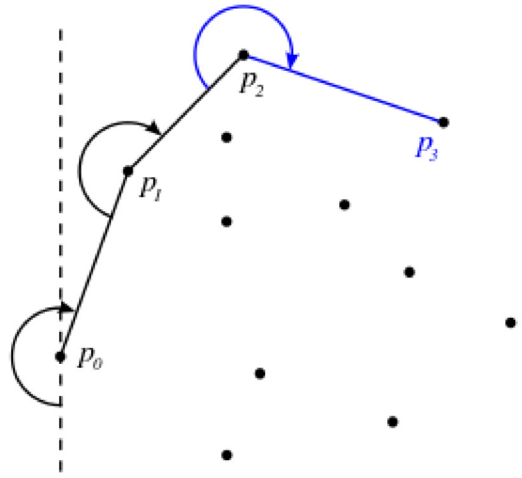
The gift wrapping algorithm begins with i=0 and a point p_0 known to be on the convex hull, e.g., the leftmost point, and selects the point p_{i+1} such that all points are to the right of the line $p_i\, p_{i+1}$. This point may be found in O(n) time by comparing polar angles of all points with respect to point p_i taken for the center of polar coordinates. Letting i=i+1, and repeating with until one reaches p_h=p_0 again yields the convex hull in h steps. In two dimensions, the gift wrapping algorithm is similar to the process of winding a string (or wrapping paper) around the set of points.

The approach can be extended to higher dimensions.

Pseudocode

```
jarvis(S)
   pointOnHull = leftmost point in S
   i = 0
   repeat
      P[i] = pointOnHull
      endpoint = S[0]      // initial endpoint for a candidate edge on the hull
      for j from 1 to |S|
         if (endpoint == pointOnHull) or (S[j] is on left of line from P[i] to endpoint)
            endpoint = S[j]   // found greater left turn, update endpoint
      i = i+1
      pointOnHull = endpoint
   until endpoint == P[0]      // wrapped around to first hull point
```

Complexity



Jarvis's march computing the convex hull.

The inner loop checks every point in the set *S*, and the outer loop repeats for each point on the hull. Hence the total run time is $O(nh)$. The run time depends on the size of the output, so Jarvis's march is an output-sensitive algorithm.

However, because the running time depends linearly on the number of hull vertices, it is only faster than $O(n \log n)$ algorithms such as Graham scan when the number h of hull vertices is smaller than log *n*. Chan's algorithm, another convex hull algorithm, combines the logarithmic dependence of Graham scan with the output sensitivity of the gift wrapping algorithm, achieving an asymptotic running time $O(n \log h)$ that improves on both Graham scan and gift wrapping.

References

- Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2001) [1990]. "33.3: Finding the convex hull". *Introduction to Algorithms* (2nd ed.). MIT Press and McGraw-Hill. pp. 955–956. ISBN 0-262-03293-7.
- Jarvis, R. A. (1973). "On the identification of the convex hull of a finite set of points in the plane". *Information Processing Letters*. **2**: 18–21. doi:10.1016/0020-0190(73)90020-3 (https://doi.org/10.1016%2F0020-0190%2873%2990020-3).