Univ. u Novom Sadu
Zb. Rad. Prirod.–Mat. Fak.
Ser. Mat. 21, 2 (1991), 157-160

Review of Research
Faculty of Science
Mathematics Series

ON PAIRING POINTS IN THE PLANE

Ratko Tošić
Institute of Mathematics, University of Novi Sad,
Trg Dositeja Obradovića 4, 21000 Novi Sad, Yugoslavia

Ivan Stojmenović
Computer Science Dept. University of Ottawa
Ottawa, Ontario, Canada K1N9B4

Abstract

We present a simple $O(nlog^2n)$ time algorithm for pairing points in the plane.

AMS Mathematics Subject Classification (1980): 52C99
Key words and phrases: Pairing algorithm, convex hull.

1. Introduction

Suppose we are given $2n$ distinct points in the plane, no three of which are collinear, n of which are colored blue and the remaining n are colored red. The problem we consider is that of finding one to one correspondence between red and blue points such that if we join every pair of corresponding points by a straight line segment, then no two of the resulting n segments intersect.

Atallah in [1] gave an $O(nlog^2n)$ time algorithm for finding the desired n straight line segments.

We present another solution to the same problem and with the same running time. We believe that our algorithm is simpler to implement.

2. Atallah's solution

In [1] it is assumed that the red points are stored in an augmented tree structure (call it TR) according to their x coordinate. Structure TR also contains a description of the convex hull (call it HR) of the red points, and it supports insertion and deletion operations in time $O(log^2 n)$ [4]. A similar structure is available for the blue points (call it TB) and we call the convex hull of the blue points HB). Creating TR and TB takes $O(n * log(n))$ time [4]. Atallah's algorithm runs as follows:

Step 1: Test whether HR and HB are disjoint. Case 1: HR and HB are disjoint. Then repeat the following until no more points remain: Find a line PQ tangent to both HR and HB, which is one of the desired n segments and delete P from TR and Q from TB.

Case 2: HR and HB are not disjoint. Then let a and b be respectively the smallest and largest x coordinates among the red points, and let v and w be respectively the smallest and largest $x-$ coordinate among the blue points. If none of the intervalss $[a, b]$ and $[v, w]$ contains the other than use the algorithm described in [1] to find a common tangent PQ to HR and HB such thatHR and HB are on the same side of this tangent, join P and Q by a straight line segment, delete P from TR, Q from TB and go to Step 1. Otherwise one of the intervals $[a, b]$ and $[v, w]$ contains the other and there must exist a vertical line (call it L) with the property that there are points both to its left and to its right, with as many red points as blue ones to its left, and similarly to its right. Do the following:

(i) Find the line L by performing two "scans" of the remaining points (both red and blue). We alternate between one scan which starts at the leftmost point and moves rightward, and the other which starts at the rightmost point and moves leftward. Both scans "look" for line L until one of them finds it.

(ii) Find TR(1), TR(2), TB(1), TB(2), where TR(1) and TB(1) are the augmented tree structures of the (respectively) red and blue points to the left of L, and TR(2) and TB(2) are analogously defined for points to the right of L. TR(1) and TR(2) (TB(1) and TB(2)) are obtained by splittiong TR (TB) about the x component of the vertical line L.

(iii) Recursively solve each of the two subproblems consisting of the points to the left of L (using TR(1) and TB(1)) and of the points to the

right of L (using TR(2) and TB(2)), then stop.

Atallah [1] has showed that the algorithm runs in time $O(nlog^2n)$.

3. New technique for pairing points

We describe new technique for pairing points which runs with the same efficiency as Atallah's one and is simpler to expose and implement.

Initial step 1. Create an augmented tree structure T [4] according to the x coordinates of both red and blue points, which contains also a description of the convex hull H of all points.

Let pred(X) and succ(X) be the predecessor and successor of a point X on H in counterclockwise order. Let P be a point on H and let R = P.

Do the following steps until no more points remain.

Step 1. Examine points R and succ(R). If they are colored by different colors then join them by straight line segment, do P ← pred(P) if R = P, R ← pred(R), delete examined points from T, find new H and go to step 1. If R and succ(R) are colored by the same color then let R ← succ(R). If scan is finished (i.e. P = succ (R)) then go to step 2. Otherwise go to Step 1.

Step 2. (i) Find a vertical line L with the property that there are points both to its left and to its right, with as many red points as blue ones to its left, and similarly to its right. This can be done in the same way as in [1].

(ii) Find T(1) and T(2), the augmented tree structures of the (respectively) points to the left and to the right of L. This can be done by splitting T about the $x-$ component of the vertical line L.

(iii) Recursively solve each of the two subproblems consisting of the points to the left of L (using T(1)) and of the points to the right of L (using T(2)), then stop.

Correctness of the aalgorithm is obvious. We show that it runs in time $O(nlog^2n)$. Cost of Initial step is $O(nlogn)$ [4]. The total number of examined pairs (R, succ(R)) in Step 1 is not greater than the number of points on H (where H is taken at the end of Step 1) plus the number of straight line segments obtained in the "walk". The total cost (including recursive calls) of deleting straight line segments in Step 1 is $O(nlog^2n)$. The total cost of Step 2 is the same as Steps (i) - (iii) in Atallah's algorithm, i.e. $O(nlog^2n)$.

4. Conclusion

Both Atallah's and ours algorithms could improve their running time to $O(nlog^2n)$ by designing a $O(log^2n)$ time technique for splitting an augmented tree structure (describing also the convex hull) into two parts separated by a vertical line.

References

[1] Atallah, M.J. : A matching problem in the plane, J. Comput. System Sci. 31 (1985), 63-70.

[2] Cole,R., Sharir, M., Yap, C.K.: On k-hulls and related problems, in: "proc. 16th Annual ACM Symposium on Theory of Computing", Washington D.C., pp 154-166, 1984.

[3] Iri, M., Murota, K., Matsui, S.: Linear - time approximation algorithms for finding the minimum - weight matching on a plane, Inform. Proc. Lett., 12, 4 (1981), 206-209.

[4] Overmars, M.H., van Leeuwen, J.: Maintenance of configurations in the plane, J. Comput. System Sci. 23 (1981), 166-204.

[5] Papadimitriou, C.H., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity, Prentice - Hall, Engewood Cliffs, N.J., 1982.

REZIME

O SPAJANJU TAČAKA U RAVNI

U radu je izložen jedan algoritam za sparivanje tačaka u ravni. Algoritam ima vremensku složenost $O(nlog^2n)$ i jednostavan je za implementaciju.

Received by the editors June 13, 1988