Crossing-free monochromatic trees for bicolored point sets

José Fernández Goycoolea*[1], Luis H. Herrera†[2], Pablo Pérez-Lantero‡[3], and Carlos Seara§[4]

[1]Departamento de Matemática y Física, Universidad de Magallanes, Punta Arenas, Chile.
[2]Departamento de Informática y Computación, Universidad Tecnológica Metropolitana, Santiago, Chile.
[3]Departamento de Matemática y Ciencia de la Computación, Universidad de Santiago de Chile, Santiago, Chile.
[4]Departament de Matemàtiques, Universitat Politècnica de Catalunya, Barcelona, Spain.

Abstract

We study the problem of connecting the points of a bicolored point set $S = R \cup B$ by monochromatic non-overlapping geometric trees. As has been done for similar geometric problems, we characterize the minimum number of trees required in terms of the number τ of non-monochromatic edges in the convex hull of S. A novel algorithm to construct this forest is presented; the construction aims to maintain the trees' diameter long and also provides an intuitive reason why the number of trees required depends only on the border configuration and not on the interior points.

1 Introduction

Among the many geometrical problems surveyed by Kano and Urrutia [3] for discrete colored point sets, one surprising fact is that in some strong results obtained for a bicolored point set in the plane, the conditions required depend only on the properties of the configuration of the points in the boundary of its convex hull, and not on the distribution of the points on the interior. More precisely, for a bicolored set $S = R \cup B \subset \mathbf{R}^2$ of n points in the plane, of which the ones in R are red and the ones in B are blue, the number τ of segments of the convex hull of S having end points of different colors governs the minimum number of crossings needed to draw both: an alternating 1-plane Hamiltonian cycle [2], and a couple of monochromatic simple covering geometric trees ($\max\{\tau/2 - 1, 0\}$ crossings) [4].

Here, we focus our attention on the problem of connecting bichromatic point sets by non-crossing monochromatic geometric trees with the initial aim of characterizing the minimum number of trees required. This problem has been previously studied by Aichholzer et al. [1]; there an $O(n \log n)$-time algorithm that finds a single spanning tree for B and a forest of $\max\{\tau/2, 1\}$ trees covering R is given. This algorithm is based on a triangulation of the point set and permits the connection of the red parts to reach Tokunaga's minimum number of crossings for a couple of monochromatic simple covering geometric trees [4].

Our contribution to this problem is first to prove that no forest of red and blue trees of less than $\max\{\tau/2 + 1, 2\}$ members can be constructed without crossings and then to present an alternative algorithm that makes use of successive convex hull layers to construct a forest composed of caterpillar trees or trees formed by linkage of caterpillar parts. An implementation strategy that allows our algorithm to run in $O(n \log n)$ time is presented.

2 Monochromatic forest cardinality

For an arbitrary set $X \subset \mathbf{R}^2$, the notation $\overline{X}$ stands for the convex hull of X, and ∂X is the boundary of X. X is monochromatic and red if $X \cap S \subset R$ (resp. blue if $X \cap S \subset B$).

A geometric graph $G = (V, E)$ is a graph embedded in the plane with its vertices being a finite set $V \subset \mathbf{R}^2$ of points and its edge set E consisting of closed line segments whose end points lie in V. G is monochromatic if its segments are monochromatic; if it is simple and acyclic it is a V-tree.

In this context, Tokunaga's Theorem [4] relates the minimum number of intersections between a R-tree and a B-tree, denoted by $f(R, B)$, with the number $\tau = \tau(R, B)$ of closed segments of $\partial\overline{S}$ that are non-monochromatic. The relation is:

$$f(R,B) = \begin{cases} 0 & \text{if } \tau = 0, \\ \frac{1}{2}\tau(R,B) - 1 & \text{if } \tau > 0. \end{cases} \qquad (1)$$

We prove the following simple lemma.

Lemma 1 *For any couple of simple geometric trees T_R and T_B having the minimum number of intersections possible, and hence realizing Equation* (1)*, every edge of T_R intersects at most one edge of T_B. And vice versa, every edge of T_B intersects at most one edge of T_R.*

Then, let $h(R, B)$ denote the cardinality of a smallest *monochromatic forest* of S, that is, a set of

*Email: jose.fernandezg@umag.cl
†Email: luis.herrerab@utem.cl
‡Email: pablo.perez.l@usach.cl
§Email: carlos.seara@upc.edu

monochromatic pairwise non-overlapping simple geometric trees $T_i = (V_i, E_i)$ such that $\bigcup_{i=1}^{k} V_i = S$.

At first glance, it is conceivable that for a specific configuration of points, a monochromatic forest of less than $2 + f(R, B) = \max\{\tau/2 + 1, 2\}$ trees could be found. This would be possible if, for connecting the forest into only two trees T_R and T_B, at least one double crossing (a single edge crossing two edges) were to be always required. The next result, proven by induction on n shows this is not possible.

Theorem 2 *If $R = \varnothing$ or $B = \varnothing$, then $h(R, B) = 1$. Otherwise, the following relation holds:*

$$h(R, B) = f(R, B) + 2 = \begin{cases} 2 & \text{if } \tau = 0, \\ \frac{1}{2}\tau + 1 & \text{if } \tau > 0. \end{cases}$$

The following section briefly describes the algorithm that constructs the monochromatic forest.

3 Monochromatic Forest Construction Algorithm

The algorithm starts from an empty forest $F = \varnothing$ and *adds points of S to the forest* until no more points remain. As an initialization step, the points of $S \cap \partial\overline{S}$ and the value $\tau = \tau(R, B)$ are computed. Then there are three different cases: case-1 is when $\tau = 0$, case-2 is when $\tau = 2$, and case-3 is when $\tau \geq 4$.

Case-2 gets the bulk of the work; it either terminates or is reduced to two caterpillar trees and a sub-problem, which is itself also in case-2. For a given input triplet (S, R, B), a sub-problem is another triplet (S', R', B') composed of a subset $S' \subset S$ and its induced coloring $R' = R \cap S'$ and $B' = B \cap S'$ for which the algorithm can be applied recursively.

In the nontrivial instances of case-1 there are still points of both colors, and the convex hull of one of the colors is contained in the convex hull of the other, say $\overline{B} \subset \overline{R}$. Then, the general idea is to construct an outer red caterpillar tree using all but one segment of $\partial\overline{R}$ as a backbone through a sweeping procedure and leaving a subproblem in case-2.

In case-2, the general idea is to construct two outer caterpillar trees that include the two monochromatic runs of $\partial\overline{S}$ as backbones with sweeping procedures analogous to that of case-1, and then join them to a central sub-problem that contains the inner points $S' = S \cap \overline{B} \cap \overline{R}$ and is again in case-2. The construction needs some care to avoid configurations where just the inner points are not a case-2 subproblem. The result consists of two caterpillar trees that *wind themselves into each other*.

In case-3 the idea is to iteratively solve sub-problems from S that are themselves in case-2. The boundary $\partial\overline{S}$ is composed of τ runs connected by τ non-monochromatic segments. The runs alternate in color, so starting at an arbitrary run, say it is red, the farthest points of the two neighboring blue runs b_1 and b_2 are selected to divide an initial subproblem. Then, additional subproblems are processed similarly as shown in the example of Figure 1.

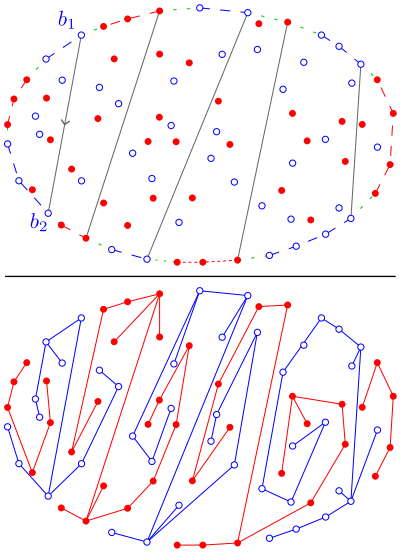


Figure 1: Case-3 example.

An example of the type of caterpillar trees obtained by the algorithm for a set with $|R| = |B| = 55$ in case-2 is shown in Figure 2.

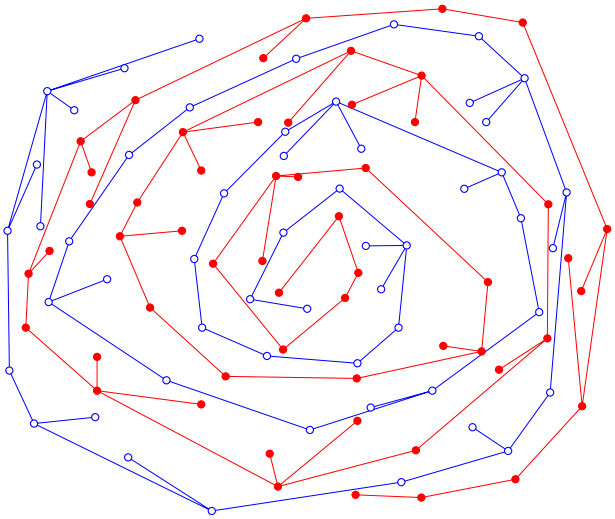


Figure 2: Case-2 example, caterpillars.

For the same example of Figure 2, the obtained caterpillars can be *flattened* in a secondary process producing two paths, as is shown in Figure 3. Although for *well-blended* homogeneously distributed configurations of points, this can usually be done, it is not always possible. Example configurations of points in case-2 can be devised for which two non-overlapping paths do not exist.

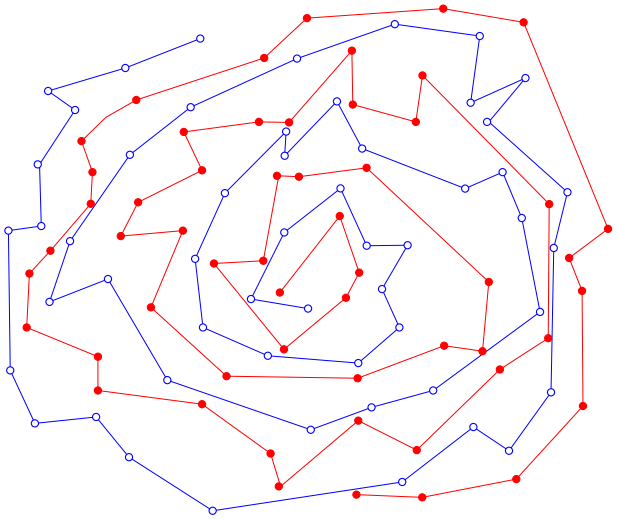


Figure 3: Case-2 example, paths.

Using the Delaunay triangulations of R and B as *routing graphs* data-structures an implementation of the algorithm can achieve the following result.

Theorem 3 *Let $S = R \cup B$ be a bicolored set of n points in the plane in general position. If $\tau(R, B) \leq 2$, then two disjoint non-crossing monochromatic caterpillar trees can be constructed in $O(n \log n)$ time and $O(n)$ space. If $\tau(R, B) > 2$, a forest of $\tau(R, B)/2 + 1$ non-overlapping monochromatic trees composed of concatenated caterpillars can be constructed in $O(n \log n)$ time and $O(n)$ space.*

To conclude, an additional example, again in case-2, but for a smaller set $|R| = |B| = 20$ is presented in Figure 4. There, the results obtained from the novel *large-diameter* algorithm (label A) are compared to the *small-diameter* algorithm (label B) from [1].

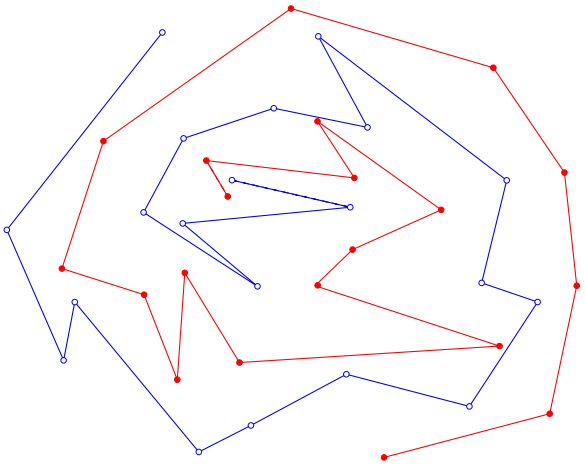
References

[1] O. Aichholzer, F. Aurenhammer, T. Hackl, and C. Huemer. Connecting colored point sets. *Discrete Applied Mathematics*, 155(3):271–278, 2007.

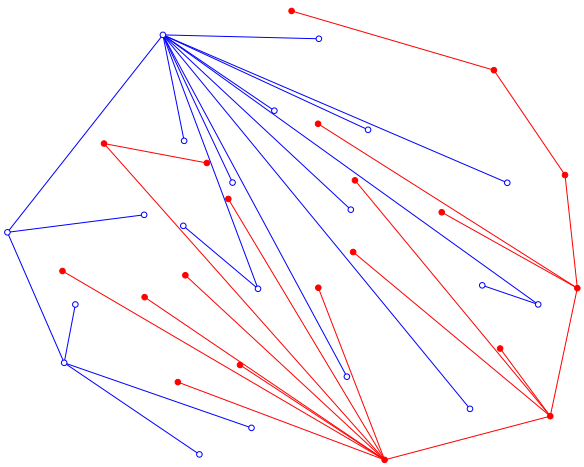
[2] M. Claverol, A. García, D. Garijo, C. Seara, and J. Tejel. On Hamiltonian alternating cycles and paths. *Computational Geometry*, 68:146–166, 2018.

[3] M. Kano and J. Urrutia. Discrete geometry on red and blue points in the plane—A survey. *Graphs and Combinatorics*, 37:1–53, 2021.

[4] S. Tokunaga. Intersection number of two connected geometric graphs. *Information Processing Letters*, 59(6):331–333, 1996.



A



B

Figure 4: Algorithm comparison example.