

Alternating Paths and Cycles of Minimum Length

William S. Evans¹, Giuseppe Liotta², Henk Meijer³, and Stephen Wismath⁴(✉)

¹ University of British Columbia, Vancouver, Canada

² Università degli Studi di Perugia, Perugia, Italy

³ U. C. Roosevelt, Middelburg, The Netherlands

⁴ University of Lethbridge, Lethbridge, Canada

wismath@uleth.ca

Abstract. Let R be a set of n red points and B be a set of n blue points in the Euclidean plane. We study the problem of computing a planar drawing of a cycle of minimum length that contains vertices at points $R \cup B$ and alternates colors. When these points are collinear, we describe a $\Theta(n \log n)$ -time algorithm to find such a shortest alternating cycle where every edge has at most two bends. We extend our approach to compute shortest alternating paths in $O(n^2)$ time with two bends per edge and to compute shortest alternating cycles on 3-colored point-sets in $O(n^2)$ time with $O(n)$ bends per edge. We also prove that for arbitrary k -colored point-sets, the problem of computing an alternating shortest cycle is NP-hard, where k is any positive integer constant.

1 Introduction

A recent paper by Chan et al. [5] studies the problem of computing a planar drawing of an n -vertex planar graph such that the vertex locations are given as part of the input and the drawing has minimum total edge length. The problem is known to be NP-hard [4] in general and Chan et al. describe different polynomial time approximation algorithms for paths, matchings, and general planar graphs. They also give a polynomial time exact algorithm for paths on fixed positions that lie on a line, which computes a planar drawing where all edges are monotone in a common direction and each edge can be represented by a poly-line having $O(n)$ bends.

In this paper we consider a variant of the problem by Chan et al. where the position for each vertex is not fixed, but it can be chosen by the algorithm as one in a given subset of a point set. To be precise, we are given a k -colored graph (i.e., a graph where each vertex is one of k different colors) and we want to compute a planar drawing of the graph on a given k -colored point-set so that vertices are mapped to distinct points of the same color and the total edge length is minimized.

The research reported in this paper started at the 2015 Bertinoro workshop, sponsored by the EuroGIGA Project. Research also supported by NSERC, and by MIUR of Italy under project AlgoDEEP prot. 2008TFBWL4.

We mainly focus on drawing shortest alternating 2-colored (bicolored) paths and cycles and collinear point-sets. But we also consider the case of more than two colors and the case that the points are non-collinear. Our main results are:

- Let R be a set of n red points and B be a set of n blue points such that $R \cup B$ is a set of distinct and collinear points. We describe a $\Theta(n \log n)$ -time sweep-line algorithm to compute a planar drawing of an alternating cycle of minimum length on $R \cup B$ such that every edge is a poly-line with at most two bends.
- We adapt the approach for cycles to the problem of computing a shortest alternating path on a bicolored set of collinear points. We describe an $O(n^2)$ -time algorithm that solves the problem by computing drawings with at most two bends per edge.
- We extend the study to 3-colored collinear point-sets and describe an $O(n^2)$ -time algorithm to compute shortest alternating cycles (visiting the colors in cyclic order) such that every edge has $O(n)$ bends.
- We consider non-collinear point-sets and prove that computing a shortest alternating cycle is NP-hard in the general case of k -colored point-sets, where $k \geq 1$ is a given constant.

From a technical point of view, our drawing algorithms are based on the idea of computing an alternating topological book embedding of a path or cycle such that the number of edges that are intersected by any cut is minimum. This approach seems to be specific for two and three colors, since we also present an example with four colors where an alternating cycle of minimum length cannot match the cut lower bound that we use for fewer than four colors.

1.1 Related Work and Paper Organization

The problem of computing a planar alternating path or a planar alternating cycle on $R \cup B$ has a long tradition in graph drawing and computational geometry. While the interested reader may refer to the survey by Kaneko and Kano [11] for a list of early references, we briefly recall here some of the milestone results. Akiyama and Urrutia [2] study straight-line alternating paths when $R \cup B$ is in convex position; they exhibit a set of sixteen points for which a straight-line alternating path does not exist and present an $O(n^2)$ -time algorithm to test when a straight-line alternating path on points in convex position exists. Abellanas et al. [1] show that if either the convex hull of $R \cup B$ consists of all red points and no blue points or there exists a line that separates all blue points from red ones, then a straight-line alternating path always exists. Kaneko, Kano, and Suzuki [12] characterize those point sets in general position for which a straight-line alternating path always exists: If $R \cup B$ consists of at most twelve points or if it consists of exactly fourteen points, then a straight-line alternating path always exists; for all other cases, there exist configurations of red and blue points for which a straight-line alternating path does not exist. These early results about straight-line alternating paths have motivated further research on computing

alternating paths and cycles when the edges can bend. Di Giacomo et al. [8] proved that every point set admits an alternating path and an alternating cycle with at most one bend per edge; the result is based on projecting the points on a horizontal line and then computing a book embedding on this line before mapping the edges back to the original points.

The results above have motivated further research where either bicolored graph families other than paths or cycles have been studied or more than two colors have been considered, or both. For graph families other than paths or cycles, the input is a bicolored planar graph G together with a bicolored set of points in the plane and the goal is to compute a planar drawing of G such that every red vertex is mapped to a red point and every blue vertex is mapped to a blue point and either each edge is a straight-line segment or it has a constant number of bends. See, for example, [6, 9, 10, 13, 14]. For more than two colors (see, e.g., [7, 8]), the input is k point sets each with the same cardinality and the goal is to compute an alternating path/cycle on the entire set of points; that is, a planar drawing of a path/cycle containing the given points and such that the i th point on the cycle comes from the $(i \bmod k)$ th set. In the extreme case, one is given n colors modeled as n numbers from 1 to n and the goal is to compute a planar drawing of a path or cycle that touches the vertices in increasing order. In other words, the n -colored version of the problem is the same as asking for a planar drawing of a graph where the location of the vertices is specified as part of the input. A seminal result in this context is due to Pach and Wenger [15] who prove that linearly many bends per edge are always sufficient and sometimes necessary for n -colored paths and n -colored point-sets in convex position. Their drawing technique applies to general n -colored planar graphs and the number of bends per edge was improved by Badent et al. [3].

The remainder of the paper is organized as follows. An overview of our algorithmic approach is presented in Sect. 2. Section 3 describes the algorithm for shortest alternating cycles on collinear red-blue points. Shortest alternating paths on collinear red-blue points are studied in Sect. 4. Shortest alternating cycles on more than two colors and the proof of hardness for general k -colored point-sets is in Sect. 5. Finally, open problems are listed in Sect. 6. For reasons of space, some proofs have been sketched and will be available in the full version of the paper.

2 Overview of the Algorithmic Approach

Let R be a set of n distinct red points and B be a set of n distinct blue points in the Euclidean plane. An *alternating cycle* (*alternating path*) is a drawing of a cycle (path) such that the vertex set of the drawing is the set $R \cup B$ and such that no two vertices having the same color are adjacent. The drawing is *planar* if no two edges cross. The length of the cycle (path) is the sum of the lengths of its edges. A *shortest* alternating cycle (path) is one of minimum length. In this paper we are interested in computing shortest planar alternating cycles (paths). Since the problem is NP-hard for general point sets (Sect. 5), we focus on collinear point-sets and assume that the line through the point set, called the *spine*, is horizontal.

A set of n blue points and n red points on a line define $2n + 1$ intervals, two of which are infinite. Assume we have a (not necessarily optimal) planar drawing of an alternating cycle (path). Consider a vertical line in any interval and count how many edges of the cycle (path) are intersected by the line. If we multiply the length of each finite interval by the number of edges that are intersected by a vertical line through the interval and then sum up all the obtained numbers, we obtain a lower bound on the length of the cycle. Therefore, we aim at computing an alternating cycle (path) C such that for any vertical line ℓ , the number of edges of C cut by ℓ is the minimum over all alternating cycles (paths). In addition, no two edges of C cross and every edge is a poly-line consisting of at most three segments (i.e., it has at most two bends). For brevity, in what follows we will often say alternating cycle (path) to mean planar alternating cycle (path).

Based on the observation above, the problem turns into the computation of a special type of topological book embedding, such that every edge can cross the spine at most once and such that the number of edges that span any interval between two consecutive points along the spine is minimum. Every edge of such a topological book embedding can be represented as a poly-line with at most two bends. We recall that a topological book embedding is a planar drawing of a graph such that all vertices are points of a line called the *spine* and the edges are simple Jordan arcs.

It is worth remarking that we are interested in solving the combinatorial problem of finding an order in which a shortest alternating cycle (path) visits the colored points, and the embedding of its edges. Once this is found, a planar alternating cycle (path) of minimum length can be computed by making the edges “as flat as possible” around the spine, that is by making the distance between each edge and the spine tend to zero. Hence when we say that we “compute the shortest cycle (path)”, we mean that we compute an ordering and embedding for which such a cycle (path) exists.

3 Shortest Alternating Cycle on Collinear Red-Blue Points

Following the approach of Sect. 2, we start by giving a lower bound on the number of edges of any alternating cycle intersected by a vertical line. Next, we present a sweep-line algorithm to compute a topological book embedding such that every interval is spanned by the minimum number of edges and such that every edge crosses the spine at most once.

A Lower Bound Lemma. The following lemma establishes the lower bound that will be used to prove the optimality of the alternating cycles.

Lemma 1. *Let R be a set of n red points and let B be a set of n blue points such that all points are distinct and lie on the x -axis. Let ℓ be a vertical line that intersects the x -axis between two points of $R \cup B$. If there are r red points and b blue points to the left of ℓ , then any alternating cycle on $R \cup B$ crosses ℓ at least $2 \max\{1, |r - b|\}$ times.*

Proof. Let C be an alternating cycle on $R \cup B$ and ℓ^- be the halfplane to the left of ℓ . In each component of $C \cap \ell^-$, the number of points of one color can be at most one more than the number of points of another color. Thus, the minimum number of components of C to the left of ℓ is $|r - b|$. If the line ℓ lies between two vertices of C (i.e. it is not to the left of the leftmost vertex of C and it is not to the right of the rightmost vertex of C), then the number of components to the left of ℓ is also at least one, and the number of edges of C that intersect ℓ is twice the number of components in $C \cap \ell^-$. \square

A Sweep-Line Algorithm. We now describe a sweep-line algorithm that computes the shortest alternating cycle of a set of n red points and n blue points lying on the horizontal line $y = 0$, called the *spine*. We call our algorithm **Spine-Sweep**.

Spine-Sweep first orders the points by increasing x -coordinate and then it sweeps a vertical cut line ℓ across the points. The algorithm maintains a set of disjoint curves to the left of ℓ , each of which has both endpoints, called *terminals*, on ℓ . These curves are the connected components of the intersection of some shortest alternating cycle with the halfplane, ℓ^- , to the left of ℓ . The terminals are colored red or blue depending on the color of the closest colored point on the curve. Terminals above the spine are *positive* and those below are *negative*; this is called the *sign of the terminal*. If both terminals of a component have the same sign, then this is the *sign of the component*, otherwise the component *straddles* the spine. The *distance* of a component to the spine is the minimum number of terminals between one of its terminals and the spine. Two terminals are *adjacent* if the segment connecting them contains no other terminals. Note that these definitions are with respect to the current sweep-line ℓ ; the distance of a component to the spine, for example, may change as the line ℓ moves.

During the sweep, components are created or merged when ℓ encounters a colored point. By carefully selecting which components to create and merge and how to merge them, the algorithm maintains the following invariants:

- P1. If there is exactly one component and its terminals have different colors, then its terminals have different signs.
- P2. If there are more than two terminals, then they all have the same color.
- P3. The two closest components to the spine do not have the same sign.

When the algorithm encounters a colored point, p , it either *forks* a new component, if p 's color matches the color of all terminals (or there are no terminals), otherwise it *merges* p with one or two existing components creating a single new component. We describe these two cases under the assumption that the encountered point p is blue. Symmetric operations hold if p is red. In the next two figures, the terminals are drawn as squares; also, light/dark vertices are red/blue.

Fork. If there are no red terminals, we create a new component containing (blue) p that straddles the spine and has adjacent (blue) terminals. See Fig. 1(a).

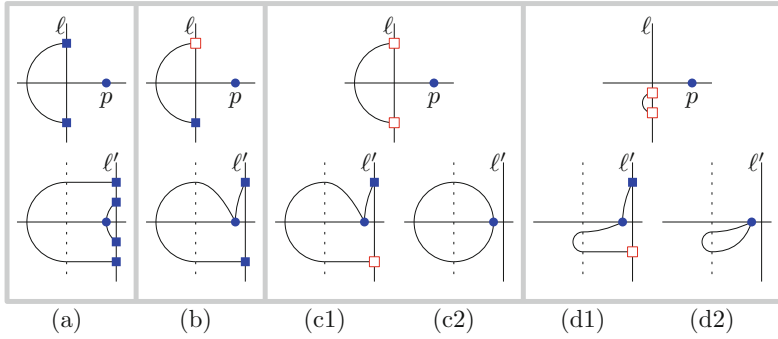


Fig. 1. The top row shows the initial configurations (the symmetric versions of (b) and (d) are not shown). The bottom row shows the possible configurations after merging. (a) Fork. One component is shown in the initial configuration but there may be many or none as long as all terminals are blue. The two new terminals are closest to the spine. (b-d) Merge with one component. All components are shown. Cases (c2) and (d2) only occur when p is the last colored point (Color figure online)

Merge. If there are red terminals, we create a new component that contains (blue) p . If there is only one component, we add p to that component by extending the edge from the closest red terminal to p . If p is not the last colored point, we add a new edge from p to a new (blue) terminal so that the new component straddles the spine. If p is the last colored point, we extend the edge from the other (red) terminal to p . See Fig. 1.

If there are at least two components, then by property P2 all their terminals are red. Let K and J be the two closest components to the spine. We extend the edges from a terminal from K and a terminal from J to p . We choose the terminals and route the edges from all terminals to ensure that our invariant properties remain true.

By property P3 and the fact that components do not intersect, the configuration of the two closest components to the spine is one of the four shown schematically in the top row of Fig. 2. For two of these configurations, (c) and (d) in Fig. 2, we extend the edges from the closest terminals to the spine from K and J to p , and extend all other terminals horizontally. In the other two configurations, we choose how to merge based on the sign of the closest component to the spine after the merge that is not the newly merged component. If this component is negative, we merge to form configuration (a1) or (b1) in Fig. 2. Otherwise, we merge to form configuration (a2) or (b2) in Fig. 2. This is used to preserve property P3.

Notice that forming configuration (b1) in Fig. 2 causes an edge of the alternating cycle (shown in bold) to cross the spine. We ensure that this edge is not forced to cross the spine again which implies that each edge of the alternating cycle produced by the algorithm can be drawn with at most two bends.

Main Theorem. We prove that **Spine-Sweep** computes an alternating cycle C such that each edge crosses the spine at most once and no two edges cross each

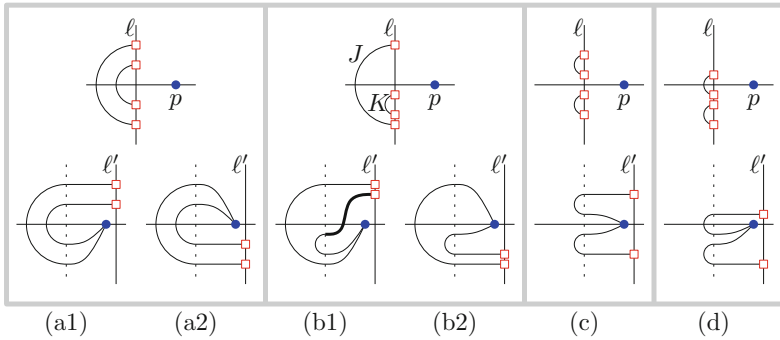


Fig. 2. Merge with two components. The top row shows the four basic configurations (the symmetric versions of (b) and (d) are not shown). The bottom row shows the possible configurations after merging (Color figure online).

other. Also, any vertical line ℓ that intersects C , does so exactly $2 \max\{1, |r - b|\}$ times, where r and b denote the number of red and blue points to the left of ℓ . Therefore, by using Lemma 1 and the observations that: (i) each fork/merge operation can be executed in constant time (for example by using a stack to maintain the components sorted according to their distance from the spine); and (ii) the red and blue points must be sorted in increasing x -order, we obtain the following.

Theorem 1. *Let R be a set of n red points and let B be a set of n blue points such that all points are distinct and collinear. There exists an optimal $\Theta(n \log n)$ -time algorithm that computes a planar alternating cycle of minimum length with at most two bends per edge.*

Note that the time complexity of Theorem 1 is worst-case optimal. Namely, if the red and blue points alternate along the spine, computing a shortest alternating cycle is equivalent to computing a circular sorting of the point set.

4 Shortest Alternating Paths on Collinear Red-Blue Points

We can also obtain a shortest alternating path on a set of n red and n blue points that are all distinct and collinear, provided we are given the endpoints. The approach is the same as in the cycle case: we prove a lower bound on the number of times any alternating path with these endpoints intersects a vertical line ℓ and use essentially the same algorithm to find an alternating path that matches the bound. The lower bound is complicated slightly by the path endpoints.

Lemma 2. *Let R be a set of n red points and let B be a set of n blue points such that all points are distinct and lie on the x -axis. Let ℓ be a vertical line that intersects the x -axis between two points of $R \cup B$. If there are r red points and*

b blue points to the left of ℓ , then the number of times any alternating path on $R \cup B$ crosses ℓ is at least:

$$\begin{aligned} & 2 \max\{1, |r - b|\} \text{ if both path endpoints are on the same side of } \ell, \\ & 1 + 2 \max\{b - r, r - b - 1\} \text{ if only the red path endpoint is left of } \ell, \\ & 1 + 2 \max\{r - b, b - r - 1\} \text{ if only the blue path endpoint is left of } \ell. \end{aligned}$$

Proof. If both path endpoints are on the same side of ℓ , the proof is the same as in the cycle case. If only the red path endpoint is left of ℓ , then its component can have at most one more red point than blue points, and at most zero more blue points than red points. Thus, if $r > b$ this component can account for one of the excess $r - b$ red points, while if $b \geq r$, it cannot account for any of the excess $b - r$ blue points. This component crosses ℓ once; all others cross twice. If only the blue path endpoint is left of ℓ , a symmetric argument applies. \square

To find a shortest alternating path between two given endpoints, we can use a modification of **Spine-Sweep**. More precisely, a fork operation on a vertex of degree one gives rise to a component with only one terminal and a merge operation on a vertex of degree one joins the closest terminal of the closest component. The main difference with the approach described in the previous section is that we may have an odd number of terminals during some steps of the sweep-line procedure, which however does not change the reasoning behind either the proof of correctness or the time complexity.

Lemma 3. *Given a set R of n red points and a set B of n blue points such that all points are distinct and collinear, and given $u \in R$ and $v \in B$, there exists a $\Theta(n \log n)$ -time algorithm that computes a planar alternating path of minimum length with at most two bends per edge that starts at u and ends at v .*

Suppose we want to find the shortest alternating path but it may start and end at any pair of points. While one might think that a shortest alternating path always starts at the leftmost red (blue) point and always ends at the rightmost blue (red) point, this is not always the case. For example, the point set of Fig. 3 has an alternating path in Fig. 3(a) whose length is minimal if it is required that both endpoints are extremal but it is not as short as the one in Fig. 3(b).

To find the best endpoints, we may use the fact that our algorithm matches the lower bound described in Lemma 2. Let r_i and b_i be the number of red and blue points, respectively, among the first (leftmost) i colored points in $R \cup B$. (Note: $r_i + b_i = i$.) Let $c_i = 2 \max\{1, |r_i - b_i|\}$, $s_i = 1 + 2 \max\{b_i - r_i, r_i - b_i - 1\}$, and $t_i = 1 + 2 \max\{r_i - b_i, b_i - r_i - 1\}$. Let d_i be the distance between the i th and $(i + 1)$ st colored points. If the path starts at the j th and ends at the k th colored point, then its minimum length is the sum of the lower bounds from Lemma 2 weighted by the distance between adjacent colored points:

$$P[j, k] = \begin{cases} \sum_{i=1}^{j-1} d_i c_i + \sum_{i=j}^{k-1} d_i s_i + \sum_{i=k}^{n-1} d_i c_i & \text{if } j \text{ th point is red} \\ \sum_{i=1}^{j-1} d_i c_i + \sum_{i=j}^{k-1} d_i t_i + \sum_{i=k}^{n-1} d_i c_i & \text{if } j \text{ th point is blue} \end{cases}.$$

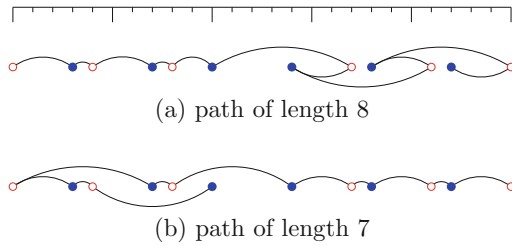


Fig. 3. Alternating paths. (Note the scale – fractional units are used) (Color figure online)

We can find the indices of different colored points $1 \leq j < k \leq n$ that minimize $P[j, k]$ in $O(n^2)$ time by calculating $c_i, s_i, t_i,$ and d_i for all i in linear time; tabulating the partial sums $\sum_{i=1}^{j-1} d_i c_i$ (for all j) and $\sum_{i=k}^{n-1} d_i c_i$ (for all k) in linear time; and tabulating $\sum_{i=j}^{k-1} d_i s_i$ and $\sum_{i=j}^{k-1} d_i t_i$ (for all pairs $1 \leq j < k \leq n$) in quadratic time. Once we know the endpoints of the shortest alternating path, we can find the actual path using Lemma 3.

Theorem 2. *Given a set R of n red points and a set B of n blue points such that all points are distinct and collinear, there exists an $O(n^2)$ -time algorithm that computes a planar alternating path of minimum length such that no two edges cross and each edge has at most two bends.*

5 Extensions and Generalizations

In this section we discuss how to extend the described approaches to more than two colors and we consider the case where the points are not collinear.

Shortest Alternating Paths and Cycles with More Than Two Colors.

Consider 3-colored collinear point-sets. We use the colors red, green and blue, denoted by r, g and b . An alternating cycle is a cycle that connects points in the order $rgbrgbr$ etc. For ease of presentation we consider the cycle oriented in this direction. The following lemma and theorems are the 3-color version of Lemma 1 and of the approaches illustrated in the previous sections for 2-colored point-sets. Proofs are omitted.

Lemma 4. *Let R, G and B be sets of n red, n green and n blue points that are all distinct and lie on the x -axis. Any alternating cycle C on $R \cup G \cup B$, crosses a vertical line, ℓ , between two colored points at least $2 \max\{1, |r - g|, |g - b|, |b - r|\}$ times, when there are r red, g green and b blue points to the left of ℓ .*

Theorem 3. *Let R, G and B be sets of n red, n green and n blue points such that all points are distinct and lie on the x -axis. A shortest planar alternating path (cycle) having $O(n)$ bends per edge can be computed in $O(n^2)$ time.*

It is natural to ask whether one can construct alternating shortest paths and cycles for collinear point-sets having more than three colors with an approach that computes drawings which satisfy generalizations of Lemmas 1 and 4. It is not hard to see that this is not the case even for 4-colored point-sets.

Assume that we have eight points and that we want to embed an alternating cycle with four colors. Figure 4 shows that for any sweep-line there is a solution that crosses this sweep-line exactly twice. So all lower-bounds are 2. However it is not hard to see that there is no embedding that satisfies all lower-bounds simultaneously.

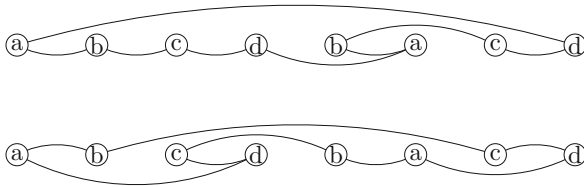


Fig. 4. 4-colored cycles on a set of eight points.

Non-collinear Point Sets. We show that finding the shortest alternating cycle is NP-hard by showing that deciding if there is a shortest alternating cycle of length less than L is NP-hard. Our reduction is from the EXACT COVER problem: Given a family F of subsets of a finite set U , is there a subfamily F' of F , consisting of disjoint sets, such that $\bigcup_{S \in F'} S = U$.

Theorem 4. *Given a k -colored point-set for constant $k \geq 1$, it is NP-hard to find the shortest planar alternating cycle.*

Proof. If $k = 1$, shortest planar alternating cycle is Euclidean TSP, which is NP-hard [16]. If $k = 2$, we describe a polynomial time reduction from EXACT COVER that is a slight modification of the reduction by Papadimitriou [16] showing that Euclidean TSP is NP-hard. Let P be the point set obtained from Papadimitriou’s reduction from the EXACT COVER instance, rotated slightly so that no points share the same x - or y -coordinate. If the EXACT COVER instance is solvable, the shortest tour of P has length L (see [16]), while if it is not solvable, the shortest tour of P has length at least $L + \sqrt{a^2 + 1} - a$ where $a = 20$. Choose $0 < \epsilon \leq (\sqrt{a^2 + 1} - a)/(10(n + 1))$ to be smaller than half the smallest difference between the x - or y -coordinates of points in P . Let $R = (P + (-\epsilon, -\epsilon)) \cup (P + (\epsilon, \epsilon))$ and $B = (P + (-\epsilon, \epsilon)) \cup (P + (\epsilon, -\epsilon))$ (where $P + (x, y) = \{(p_x + x, p_y + y) | p \in P\}$). That is, each point $p \in P$ becomes a cluster of four points (two red and two blue) forming the corners of a square $S(p)$ of side-length 2ϵ centered at p .

If the EXACT COVER instance is solvable, there is a planar alternating tour of $R \cup B$ of length at most $L + 10\epsilon n < L + \sqrt{a^2 + 1} - a$. The alternating tour follows the shortest tour of P from cluster to cluster. Within the cluster for p , it

follows three of the four sides of $S(p)$, leaving one side whose endpoints connect to the two neighbors of the cluster. It is not hard to verify that one may choose such a side for each cluster so that the resulting alternating tour is planar. Its length is at most $L + (6 + \sqrt{2})\epsilon n < L + 10\epsilon n < L + \sqrt{a^2 + 1} - a$. If the instance is not solvable, any alternating tour of $R \cup B$ is at least as long as the shortest tour of $R = P$, which has length at least $L + \sqrt{a^2 + 1} - a$. Thus $R \cup B$ has an alternating tour of length at most $L + \sqrt{a^2 + 1} - a$ if and only if the EXACT COVER instance is solvable.

If $k > 2$, the reduction is the same except that inside each square are $2(k-2)$ points (two of each color other than red and blue). These points lie on the diagonal that connects the red corners of the square, with one point of color i at distance $i\epsilon/(k-2)$ from each corner, for $i = 1, 2, \dots, k-2$. (Red is color 0 and blue is color $k-1$.) The resulting alternating tour, which uses paths of diagonal points in place of the two red-to-blue sides in each square, has length at most $L + (5\sqrt{2} + 2)\epsilon n < L + \sqrt{a^2 + 1} - a$. \square

6 Open Problems

The research in this paper suggests several open problems. We conclude the paper by listing some of those that in our opinion are among the most interesting. (1) Can the time complexity of Theorem 2 be improved? (2) Can the bend-complexity of Theorem 3 be improved? (3) The problem of computing shortest alternating cycles on collinear k -colored point-sets is open for $k > 3$. (4) Study the problem of drawing not necessarily alternating shortest bicolored cycles/paths on collinear bicolored point-sets. That is, we are given a cycle/path where any blue (red) vertex may have a neighbor of its same color and we want to draw the cycle/path using the points of $R \cup B$ such that the total edge length is minimized.

References

1. Abellanas, M., Garcia-Lopez, J., Hernández-Peñalver, G., Noy, M., Ramos, P.A.: Bipartite embeddings of trees in the plane. *Discr. Appl. Math.* **93**(2–3), 141–148 (1999)
2. Akiyama, J., Urrutia, J.: Simple alternating path problem. *Discr. Math.* **84**, 101–103 (1990)
3. Badent, M., Di Giacomo, E., Liotta, G.: Drawing colored graphs on colored points. *Theor. Comput. Sci.* **408**(2–3), 129–142 (2008)
4. Bastert, O., Fekete, S.P.: Geometrische Verdrahtungsprobleme. Technical Report 96–247, Universität zu Köln (1996)
5. Chan, T.M., Hoffmann, H.-F., Kiazzyk, S., Lubiw, A.: Minimum length embedding of planar graphs at fixed vertex locations. In: Wismath, S., Wolff, A. (eds.) GD 2013. LNCS, vol. 8242, pp. 376–387. Springer, Heidelberg (2013)
6. Frati, F., Glisse, M., Lenhart, W.J., Liotta, G., Mchedlidze, T., Nishat, R.I.: Point-set embeddability of 2-colored trees. In: Didimo, W., Patrignani, M. (eds.) GD 2012. LNCS, vol. 7704, pp. 291–302. Springer, Heidelberg (2013)

7. Di Giacomo, E., Didimo, W., Liotta, G., Meijer, H., Trotta, F., Wismath, S.K.: k -colored point-set embeddability of outerplanar graphs. *J. Graph Alg. and Appl.* **12**(1), 29–49 (2008)
8. Di Giacomo, E., Liotta, G., Trotta, F.: Drawing colored graphs with constrained vertex positions and few bends per edge. *Algorithmica* **57**(4), 796–818 (2010)
9. Kaneko, A., Kano, M.: Straight-line embeddings of two rooted trees in the plane. *Disc. Comp. Geometry* **21**(4), 603–613 (1999)
10. Kaneko, A., Kano, M.: Straight line embeddings of rooted star forests in the plane. *Discr. Appl. Math.* **101**, 167–175 (2000)
11. Kaneko, A., Kano, M.: Discrete geometry on red and blue points in the plane - a survey-. In: Aronov, B., Basu, S., Pach, J., Sharir, M. (eds.) *Discrete and Computational Geometry. Algorithms and Combinatorics*, vol. 25. Springer, New York (2003)
12. Kaneko, A., Kano, M., Suzuki, K.: Path coverings of two sets of points in the plane. In: Pach, J. (ed.) *Towards a Theory of Geometric Graph*, vol. 342. American Mathematical Society, Providence (2004)
13. Kaneko, A., Kano, M., Tokunaga, S.: Straight-line embeddings of three rooted trees in the plane. In: *Canadian Conference on Computational Geometry, CCCG 1998* (1998)
14. Kaneko, A., Kano, M., Yoshimoto, K.: Alternating hamilton cycles with minimum number of crossing in the plane. *Int. J. Comp. Geometry Appl.* **10**, 73–78 (2000)
15. Pach, J., Wenger, R.: Embedding planar graphs at fixed vertex locations. *Graphs Comb.* **17**, 717–728 (2001)
16. Papadimitriou, C.H.: The Euclidean traveling salesman problem is NP-complete. *Theor. Comp. Sci.* **4**, 237–244 (1977)