On Embedding a Graph on Two Sets of Points

Emilio Di Giacomo Giuseppe Liotta Francesco Trotta*

1 Introduction

Let $S_0, S_1, \ldots, S_{k-1}$ be k sets of points such that the points of S_i are colored with color i ($i = 0, \ldots, k-1$). Let G be a planar graph such that $|S_i|$ vertices of G have color i, for every $0 \leq i \leq k-1$. A *k-chromatic point-set embedding* of G on $S = S_0 \cup S_1 \cup \cdots \cup S_{k-1}$ is a crossing-free drawing of G such that each vertex colored i is mapped to a point of S_i, and each edge is a polygonal curve.

If $k = 1$ ("monochromatic" case), all the vertices and all the points have the same color, and therefore any vertex can be mapped on any point. In this case the algorithm that computes the drawing of G on S can choose the mapping between vertices and points as it is more convenient. Kauffman and Wiese [4] prove that every planar graph admits a monochromatic point-set embedding with at most two bends per edge on any given set of points.

If $k = n$, where n is the number of vertices of G, each vertex of G must be drawn on the unique point with the same color. Therefore the mapping between vertices and points is given as a part of the input and the drawing algorithm cannot change it. Pach and Wenger [5] prove that every planar graph has an n-chromatic point-set embedding on any given set of points such that each edge has $O(n)$ bends; they also prove that this bound is asymptotically optimal in the worst case even if G is a path.

Given the two results above, a natural question arises: If two bends per edge are necessary and sufficient to solve the monochromatic point-set embedding problem [4] while $O(n)$ bends per edge are necessary and sufficient for the n-chromatic case [5], how many bends per edge do we need if the number of colors is a constant larger than one?

In this paper we continue the study, initiated in a previous work [1], of the apparently simple case of two colors. The two colors will be referred in the following as *red* and *blue* and the two set of points S_0 and S_1 will be denoted as R and B. Notice that, by the result of Pach and Wenger [5], every planar graph has a bi-chromatic point-set embedding (2CPSE) with $O(n)$ bends per edge (arbitrarily map every red/blue vertex to a red/blue point and then use the drawing technique of [5]). Therefore the question that we ask is whether $O(n)$ bends per edge is also a lower bound for the 2CPSE problem and/or there are cases where a constant number of bends can be achieved. In [1] the simple family of bi-chromatic paths is considered and it is proved that every bi-chromatic path admits a 2CPSE on any two sets of red and blue points with at most one bend per edge.

The main results in this paper can be listed as follows. **(a)** In Section 3 we show that there exists a 2-colored tri-connected planar graph G with $n \geq 8$ vertices and two sets of points R and B such that every 2CPSE of G on $R \cup B$ has one edge that requires at least $\lceil \frac{n}{6} \rceil - 1$ bends. This proves that bi-chromatic point-set embeddability of a generic planar graph requires $O(n)$ bends per edge. Motivated by this result we investigate subclasses of planar graphs.

(b) In Section 4 we prove that every 2-colored caterpillar admits a 2CPSE on any two sets of red and blue points such that every edge of the drawing has at most two bends. We also prove that for properly 2-colored caterpillars (i. e. colored in such a way that no two adjacent vertices have the same color) the number of bends per edge can be reduced to one, which is worst-case optimal since not all bi-chromatic paths admit a 2CPSE with no bends per edge [3].

2 Preliminaries

Let $G = (V, E)$ be a planar graph. A *2-coloring* of G is a partition of V into two disjoint sets V_b and V_r, the *blue vertices* and the *red vertices* respectively. A 2-coloring is *proper* if for every edge $(u, v) \in E$ we have $u \in V_b$ and $v \in V_r$. Let R be a set of red points in the plane and let B be a set of blue points in the plane. We say that $S = B \cup R$ is *equipollent with* G if $|B| = |V_b|$ and $|R| = |V_r|$. A 2-colored planar graph G is *bi-chromatic point-set embeddable* if it has a 2CPSE on *any* set of points equipollent with G. We denote by $c(x)$ the color of x, where x can be a vertex, a point or a set of vertices/points with the same color.

Let G be a planar graph. A *2-page book embedding* of G is a crossing-free drawing of G such that: (i) the vertices of G are represented as points of a straight line called *spine*, and (ii) each edge is drawn as a simple Jordan curve completely contained in one of the two half-planes (*pages*) defined by the spine. A *subdivision* of a graph $G = (V, E)$ is a graph obtained from G by replacing each edge $(u, v) \in E$ by a path with at least one edge whose endpoints are u and v. Inter-

*Dipartimento di Ingegneria Elettronica e dell'Informazione, Università degli Studi di Perugia, {digiacomo, liotta, francesco.trotta}@diei.unipg.it

nal vertices on this path are called *division* vertices. A 2-*page topological book embedding* of G is a 2-page book embedding of a subdivision of G. If the number of division vertices for each edge is at most d we say that G has 2-*page topological book embedding with at most d divisions*.

A *red-blue sequence* σ is a set of collinear points such that each point is either red or blue. Let G be a 2-colored planar graph and let σ be a red-blue sequence equipollent with G. A 2-page (topological) book embedding of G *consistent with* σ is a 2-page (topological) book embedding of G such that each vertex v of G is represented by a point p of σ and $c(v) = c(p)$. A 2-colored planar graph G is 2-*page bi-chromatic (topological) book embeddable* if, for any red-blue sequence σ equipollent with G, G has a 2-page (topological) book embedding consistent with σ. If the number of division vertices for each edge is at most d, we say that G is 2-*page bi-chromatic topological book embeddable with at most d divisions*.

3 Curve Complexity of 2CPSEs

One can compute a drawing with $O(n)$ bends per edge by mapping every vertex $v \in G$ to a point $p \in S$ such that $c(v) = c(p)$ and then use the algorithm in [5]. Therefore every 2-colored planar graph is bi-chromatic point-set embeddable with $O(n)$ bends per edge. In this section we prove that such number of bends per edge can also be necessary for some configurations.

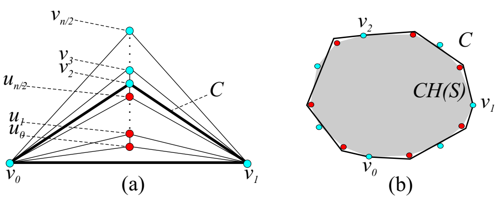

Figure 1: (a) The graph G^* whose 2CPSE on the set of points (b) requires $\lceil \frac{n}{6} \rceil - 1$ bends per edge .

Let G^* be one of the graph in the class of tri-connected bi-chromatic planar graphs shown in Figure 1 (a) and let S be a convex set equipollent with G^* such that the red and blue points alternate along the shape of its convex hull. We can prove that in any 2CPSE of G^* on S one of the three edges of the cycle C highlighted in Figure 1 has at least $\lceil \frac{n}{6} \rceil - 1$ bends (see [2]). The following theorem therefore holds.

Theorem 1 *A 2-colored planar graph is bi-chromatic point-set embeddable with $O(n)$ bends per edge, which is worst-case optimal.*

Theorem 1 naturally raises the question about whether there exist families of 2-colored planar graphs that are bi-chromatic point-set embeddable with a constant number of bends. In next section we affirmatively aswer this question for caterpillars.

4 Computing 2CPSEs with at Most Two Bends per Edge

Bi-chromatic point-set embeddability can be studied by modelling the problem as a 2-page bi-chromatic topological book embeddability problem based on the following theorem, whose proof is omitted for brevity (see [2]).

Theorem 2 *Let G be a 2-colored planar graph. Then G is bi-chromatic point-set embeddable with at most two bends per edge if and only if it is 2-page bi-chromatic topological book embeddable with at most one division.*

A *caterpillar* G is graph that consists of a path, called the *body* of G, and of a (possibly empty) set of vertices adjacent to the body and having degree one. Based on Theorem 2, we describe a drawing algorithm that computes a 2-page bi-chromatic topological book embedding of G consistent with σ and with at most one division.

We need some additional notation. We denote as $v_0, v_1, \ldots, v_h$ the vertices of the body of G. We denote as G_0 the graph consisting of vertex v_0 and as G_k $(k = 1, \ldots, h)$ the subgraph of G induced by $v_0, v_1, \ldots, v_{k-1}$ and by their adjacent vertices. We denote as $\mathcal{N}_k$ the set of vertices of $G_{k+1} \setminus G_k$, i.e. all vertices adjacent to v_k in G except v_{k-1}; also, we denote as $\mathcal{N}_k^-$ the set $\mathcal{N}_k \setminus \{v_{k+1}\}$, i.e. all the leaves of G adjacent to v_k. We define $\mathcal{N}_{-1}$ as the graph consisting of vertex v_0.

We are going to describe a drawing algorithm, called `Cater-Draw` that is recursive with the number of vertices in the body of G. At Step k of the recursion the subgraph induced by $\mathcal{N}_{k-1}$ is added to the current drawing. The output of the algorithm after k steps is a drawing γ_k that maintains a set of invariant properties. In the next sections we denote as $\sigma_k \subseteq \sigma$ the red-blue subsequence consisting of all points representing the vertices of G_k in γ_k. The rightmost point of σ_k is denoted as ρ_k. The set of all points of $\sigma \setminus \sigma_k$ that are to the left of ρ_k in σ is denoted as NB_k. The set of vertices that are after ρ_k in σ and whose color is c $(c \in \{b, r\})$ is denoted as F_k^c.

Let l be the line through the points of σ_k and let q be any point of l (q may or may not be an element of σ) and let π be either the top or the bottom half-plane (page) defined by l. We say that q is *accessible from* π *in* γ_k if there is no edge (u, v) in G_k such that q is between the point representing u and the point representing v in γ_k. For a vertex v of G, we often denote as $p(v)$ the point of σ that represents v.

4.1 The Drawing algorithm

At each step of Algorithm `Cater-Draw` the following three invariant properties are maintained. **Property 1**: γ_k is a 2-page bi-chromatic topological book embedding with at most one division. **Property 2**: All the points in NB_k have the same color, and each of them is accessible from one of the two pages. **Property 3**: Point $p(v_k)$ is accessible from one of the two pages.

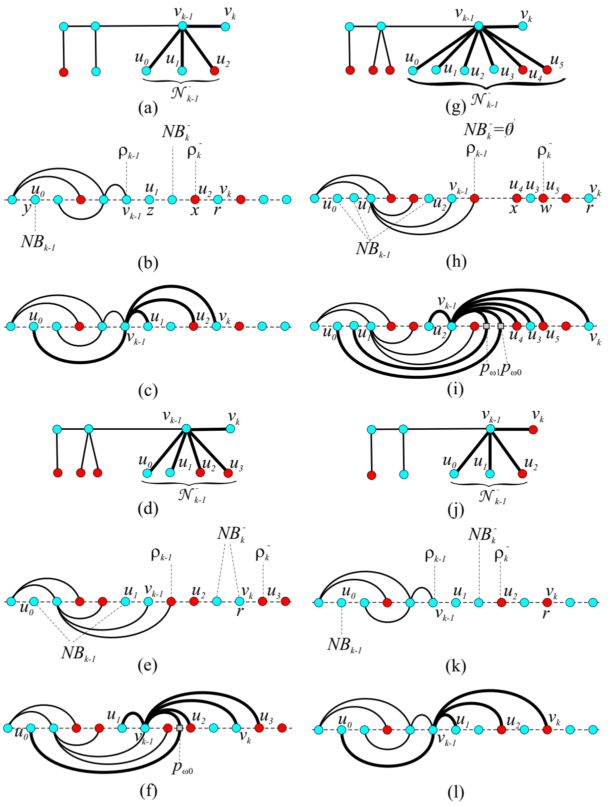

Figure 2: Different examples for Step k of `Algorithm Cater-Draw`. (a),(d),(g),(j): Different examples for graph G_k. (b),(e),(h),(k): Mapping the vertices of $\mathcal{N}_{k-1}$ to points of σ. (c), (f), (i), (l) Drawing edges connecting v_{k-1} to the vertices of $\mathcal{N}_{k-1}$.

At Step 0 of the algorithm, $\mathcal{N}_{-1}$ (i.e. vertex v_0) is drawn as the leftmost point p_j of σ such that $c(v_0) = c(p_j)$. The output of Step 0 is a drawing γ_0 consisting of a single vertex and for which it is immediate to see that all above invariants are satisfied. At Step $k > 0$, we assume by induction that γ_{k-1} satisfies the invariants and show how to add $\mathcal{N}_{k-1}$ and the corresponding edges in order to compute γ_k.

The algorithm draws first the vertices of $\mathcal{N}^-_{k-1}$ and then adds v_k. In the description of Algorithm `Cater-Draw` we shall often refer to Figure 2 for different examples. For example, in the caterpillar of Figure 2 (a) vertex v_{k-1} is blue, $\mathcal{N}^-_{k-1}$ consists of two blue vertices (vertices u_0 and u_1) and one red vertex (vertex u_2): Algorithm `Cater-Draw` will first draw u_0, u_1, u_2, and finally draw v_k.

Since Property 2 is satisfied by γ_{k-1} we have that all points in NB_{k-1} have the same color; we denote as c_1 such a color and we denote the other color as c_2. We draw first the vertices of $\mathcal{N}^-_{k-1}$ whose color is c_2, i.e. the vertices of $\mathcal{N}^-_{k-1} \cap V_{c_2}$. Assume n_2 is the number of such vertices (for example $n_2 = 1$ in Figure 2 (a) and $n_2 = 2$ in Figure 2 (g)). We map the vertices of $\mathcal{N}^-_{k-1} \cap V_{c_2}$ to the first n_2 points of color c_2 that are to the right of ρ_{k-1}; more formally, the vertices of $\mathcal{N}^-_{k-1} \cap V_{c_2}$ are mapped to the leftmost n_2 points of $F^{c_2}_{k-1}$. For example, the red vertex u_2 of $\mathcal{N}^-_{k-1}$ in Figure 2 (a) is mapped to the first red point that follows ρ_{k-1} in Figure 2 (b), i.e. u_2 is mapped to point x. Vertices u_4 and u_5 of Figure 2 (g) are mapped to points x and w of Figure 2 (h).

We now draw the vertices of $\mathcal{N}^-_{k-1}$ that have color c_1. Let $n_1 = |\mathcal{N}^-_{k-1} \cap V_{c_1}|$. If $n_1 \leq |NB_{k-1}|$ the vertices of the set $\mathcal{N}^-_{k-1} \cap V_{c_1}$ are mapped to the rightmost n_1 points of NB_{k-1} (that is, the last n_1 points of NB_{k-1} encountered when walking along the spine of γ_{k-1} from left to right). For example, see the drawing of vertices u_0, u_1, and u_2 in Figure 2 (h). If $n_1 > |NB_{k-1}|$ the vertices of $\mathcal{N}^-_{k-1}$ that have color c_1 are mapped to all vertices of NB_{k-1} and to the $n_1 - |NB_{k-1}|$ leftmost vertices of $F^{c_1}_{k-1}$. For example in Figure 2 (a) $\mathcal{N}^-_{k-1}$ has the two blue vertices u_0 and u_1 but NB_{k-1} has only one blue point (point y); hence we map u_0 to y and u_1 to the first blue point after ρ_{k-1}, i.e. we map u_1 to point z.

Once all vertices of $\mathcal{N}^-_{k-1}$ have been mapped to points, Algorithm `Cater-Draw` draws v_k. Let $\sigma^-_k \subseteq \sigma$ be the red-blue sequence consisting of all points used to construct the drawing up to now. Let ρ^-_k be the rightmost point of σ^-_k. For example, in Figure 2 (b) ρ^-_k is point x. Let NB^-_k be the set of points of $\sigma \setminus \sigma^-_k$ that are to the left of ρ^-_k. It can be proved that all points of NB^-_k share the same color $c(NB^-_k)$. For example, in Figure 2 (b) the set NB^-_k consists of one blue point. As other examples, in Figure 2 (h) NB^-_k is the empty set, in Figure 2 (e), NB^-_k consists of two blue points (note that all points of NB^-_{k-1} belong to σ^-_k), and in Figure 2 (k) NB^-_k consists of a blue point.

In order to draw v_k, Algorithm `Cater-Draw` distinguishes among different cases.

Case A: If the first point that follows ρ^-_k has the same color as v_k, then we take it; more formally, if $c(v_k) = c(next(\rho^-_k))$ then we map v_k to $next(\rho^-_k)$. For example, vertex v_k of Figure 2(a) is mapped to point r of Figure 2 (b).

Case B: If $NB^-_k \neq \emptyset$ and $c(v_k) = c(NB^-_k)$ then we map v_k to the rightmost point of NB^-_k. For example, we map vertex v_k of Figure 2 (d) to point r of Figure 2 (e).

Case C: Otherwise, v_k is mapped to the "first" point of σ to the right of ρ_k^- and having the wanted color. More formally, v_k is mapped to the point $r \in \sigma$ to the right of ρ_k^- such that $c(r) = c(v_k)$ and there are no other points between ρ_k^- and r having color $c(v_k)$. For example, in Figure 2 (h) where $NB_k^- = \emptyset$ and $c(v_k) \neq c(next(\rho_k^-))$, we have that v_k is mapped to point r. In Figure 2 (k) $NB_k^- \neq \emptyset$ but $c(v_k) \neq c(next(\rho_k^-))$ and $c(v_k) \neq c(NB_k^-)$; hence v_k is mapped to the highlighted red point r.

Once all vertices of $\mathcal{N}_{k-1}$ have been mapped to points, Algorithm `Cater-Draw` draws the edges connecting v_{k-1} to the vertices of $\mathcal{N}_{k-1}$. We define where to draw the division vertices along the edges and to which page each edge (or portion of edge between consecutive division vertices) is assigned. We distinguish two cases.

Case 1: $p(v_{k-1}) = \rho_{k-1}$**.** Refer to Figures 2 (c) and (l). The edges connecting v_{k-1} to the vertices mapped to points that follow ρ_{k-1} (i.e. the points of of $F_{k-1}^{c_1} \cup F_{k-1}^{c_2}$) are assigned to a same page that can be arbitrarily chosen. Indeed note that since $p(v_{k-1})$ is the right-most vertex drawn so-far, $p(v_{k-1})$ is accessible from both pages and so any of the two pages can be chosen to draw edges that connect $p(v_{k-1})$ to points on its right.

Let u be a vertex mapped to a point p of NB_{k-1}; by Property 2, p is accessible from a page π. Edge (v_{k-1}, u) is assigned to page π. See for example edge (v_{k-1}, u_0) in Figure 2 (c) and edge (v_{k-1}, u_0) in Figure 2 (l).

Case 2: $p(v_{k-1}) \neq \rho_{k-1}$**.** Since γ_{k-1} satisfies by inductive hypothesis the invariants, we have that by Property 3 $p(v_{k-1})$ is accessible from one page, say π. The edges connecting v_{k-1} to points to the right of ρ_{k-1} (i.e. to points of $F_{k-1}^{c_1} \cup F_{k-1}^{c_2}$) are assigned to π. See for example edge (v_{k-1}, u_2) in Figure 2 (f) and edge (v_{k-1}, v_k) in Figure 2 (i).

Let u be a vertex mapped to a point p of NB_{k-1}; by Property 2, p is accessible from a page π'. There are two subcases.

Case 2.a: π **coincides with** π'**.** If π' coincides with π then edge (v_{k-1}, u) is assigned to page π. See for example edge (v_{k-1}, u_1) in Figure 2 (f) and edge (v_{k-1}, u_2) in Figure 2 (i).

Case 2.b: π **and** π' **are distinct.** If the two pages π and π' are different, it is not possible to connect v_{k-1} and u with a curve entirely contained in one page without creating a crossing. See for example the points v_{k-1} and u_0 in Figure 2 (e), where v_{k-1} is accessible from the top page, u_0 is accessible from the bottom page and any curve connecting these two points without crossing the edges of γ_{k-1} must cross the spine.

In this case, Algorithm `Cater-Draw` splits edge (v_{k-1}, u) by means of a division vertex ω, then adds a dummy point p_ω between ρ_{k-1} and $next(\rho_{k-1})$; finally, edge (v_{k-1}, ω) is assigned to π and edge (ω, u) is assigned to π'. See for example the drawing of edge (v_{k-1}, u_0) in Figure 2 (f).

If we have more than one vertex not accessible from π, some additional care is needed when placing the dummy points. Let $u_0, u_1, \ldots, u_{d-1}$ be the vertices of NB_{k-1} that are not accessible from π and assume that they are encountered in this order from left to right in σ. Let ω_i be the division point of edge (v_{k-1}, u_i) $(i = 0, \ldots, d-1)$. In order to avoid crossings the dummy points p_{ω_i} must be placed in the order $p_{\omega_{d-1}}, \ldots, p_{\omega_1}, p_{\omega_0}$ so that edges (u_i, ω_i) do not cross each other $(i = 0, \ldots, d-1)$. See for example the edges (v_{k-1}, u_0) , (v_{k-1}, u_1) and the dummy points p_{ω_1}, p_{w_0} in Figure 2 (i): Note that if we swapped p_{ω_1} and p_{ω_0} in the drawing, the two edges (v_{k-1}, u_0) and (v_{k-1}, u_1) would cross each other.

It can be proved that algorithm `Cater-Draw` correctly computes a 2-page bi-chromatic topological book embedding of a 2-colored caterpillar consistent with any given red-blue sequence (see [2]). Based on this result and on Theorem 2 the following theorem holds.

Theorem 3 *Any* 2*-colored caterpillar is bi-chromatic point-set embeddable with at most two bends per edge.*

It is possible to prove that if a caterpillar is properly 2-colored, the number of bends per edge can be reduced to one. This is worst-case optimal since not all properly 2-colored simple paths are bi-chromatic point-set embeddable with zero bends per edge [3].

Theorem 4 *Any properly* 2*-colored caterpillar is bi-chromatic point-set embeddable with at most one bend per edge, which is worst-case optimal.*

References

[1] E. Di Giacomo, G. Liotta, and F. Trotta. How to embed a path onto two sets of points. In *Proc. GD 2005*. Springer, 2005. To appear.

[2] E. Di Giacomo, G. Liotta, and F. Trotta. On embedding a graph on two sets of points. Technical Report RT-005-05, DIEI Università di Perugia, 2005.

[3] A. Kaneko, M. Kano, and K. Suzuki. Path coverings of two sets of points in the plane. In *Towards a Theory of Geometric Graph*, volume 342. American Math. Society, 2004.

[4] M. Kaufmann and R. Wiese. Embedding vertices at points: Few bends suffice for planar graphs. *JGAA*, 6(1):115–129, 2002.

[5] J. Pach and R. Wenger. Embedding planar graphs at fixed vertex locations. *Graph and Combinatorics*, 17:717–728, 2001.