Convex Drawing for c-Planar Biconnected Clustered Graphs

Hiroshi Nagamochi[1] and Katsutoshi Kuroya[2]

[1] Toyohashi University of Technology, Toyohashi, Aichi 441-8580, Japan
naga@ics.tut.ac.jp
[2] Hitachi Advanced Digital, Inc. 292 Yoshida-cho, Totsuka-ku, Yokohama 244, Japan

Abstract. In a graph, a cluster is a set of vertices, and two clusters are said to be non-intersecting if they are disjoint or one of them is contained in the other. A clustered graph is a graph with a set of non-intersecting clusters. In this paper, we assume that the graph is planar, each non leaf cluster has exactly two child clusters in the tree representation of non-intersecting clusters, and each cluster induces a biconnected subgraph. Then we show that such a clustered graph admits a drawing in the plane such that (i) edges are drawn as straight line segments with no crossing between two edges, and (ii) the boundary of the biconnected subgraph induced by each cluster is convex polygon.

1 Introduction

A *clustered graph* $C = (G, T)$ consists of an undirected graph $G = (V, E)$ and a rooted tree $T = (\mathcal{V}, \mathcal{E})$ such that each node $c \in \mathcal{V}$, called a *cluster*, corresponds to a subset of V, denoted by $V(c)$. For a cluster c, $Ch(c)$ denotes the set of children of c, and $parent(c)$ denotes the parent of c if c is not the root. We assume that, for each non-leaf node c, it holds $V(c) = \cup_{c' \in Ch(c)} V(c')$. (Notice that a leaf cluster c may contain more than one vertex.) The subgraph of G induced by $V(c)$ for a cluster c is denoted by $G(c)$.

A clustered graph can be used to draw graphs with large size such as WWW connection graphs or VLSI schematics. The vertex set of such a graph is clustered to display a part of the graph [4]. On the other hand, there are already clustered graphs in applications such as statistics (e.g. [9]) and linguistics (e.g., [2]). Drawing clustered graphs in an understandable way is important to visualize these structures. See [1,12] for recent developments in graph drawings. In this paper, we consider how to draw clustered graphs nicely in the plane.

In a *drawing* of a clustered graph $C = (G, T)$, graph G is drawn as points and curves as usual. For each node c of T, the cluster is drawn as a simple closed region R_c that contains the drawing of $G(c)$, such that:

(1) the regions for all child clusters of c are completely contained in the interior of R_c;
(2) the regions for all other clusters are completely contained in the exterior of R_c;

G. Liotta (Ed.): GD 2003, LNCS 2912, pp. 369–380, 2004.

(3) if there is an edge e between two vertices of $V(c)$ then the drawing of e is completely contained in R_c.

We say that the drawing of edge e and region R have an *edge-region crossing* if the drawing of e crosses the boundary of R more than once. A drawing of a clustered graph is *compound planar* (c-planar, for short) if there are no edge crossings or edge-region crossings. If a clustered graph C has a c-planar drawing then it is called *c-planar* [8]. Fig. 1(a) shows an example of a c-planar drawing of a c-planar clustered graph. It is known that, if each cluster induces a connected subgraph, then testing whether a given clustered graph C has a c-planar drawing or not can be done in linear time [3]. However, the complexity status of the testing problem is unknown (see [10] for a recent progress on this issue). In this paper, we are given a clustered graph equipped with a c-planar drawing.

One of the fundamental questions in planar clustered graph drawing is: does every c-planar clustered graph admit a planar drawing such that edges are drawn as straight-line segments and clusters are drawn as convex polygons? The question has been solved affirmatively by Eades et al. [5,6,7]. To determine the xy-coordinates of the vertices in a c-planar graph which has been embedded in the plane, their method first computes y-coordinates of vertices based on an extended numbering of the st-numbering, and then determines an adequate x-coordinate of each vertex. The idea behind this is that after fixing the y-coordinates of the vertices any two disjoint clusters are separable by a horizontal line (a line parallel with the x-axis) no matter how their x-coordinates will be determined; x-coordinates can be determined so as to draw each edge as a straight line segment without taking into account the cluster structure any more. As a result, in the obtained drawing, clusters are arranged in a special way. Fig. 1(b) shows an output of their method applied to the clustered graph in Fig. 1(a). The characteristic of their drawing may be favorable to some purpose, but is rather disadvantageous to obtain a drawing in which clusters are required to be packed compactly in the plane. In this paper, we propose a new way of drawing c-planar clustered graphs. Our method is based on a divide-and-conquer approach without relaying on any special numbering on vertices, and may produce a drawing that has no structure biased in a certain direction.

2 Preliminaries

For a c-planar clustered graph $C = (G, T)$, we assume that the embedding of G of a c-planar drawing of C is given, and C is stored in an $O(n)$ space as follows, where n denotes the number of vertices in G.

A graph $G = (V, E)$ is represented by a set of adjacency lists $L(v)$ for all vertices $v \in V$, and in an adjacency list $L(v)$, all the edges incident to v appear in the same order that they appear around v in clockwise order (that is how the embedding of G in the drawing is represented). Each edge $e = (v, u)$ is equipped pointers that indicate the cells for e in the lists $L(v)$ and $L(u)$. Also we assume that, for two nodes $c, c' \in T$, their least common ancestor, denoted by $\mathrm{LCA}(c, c')$, can be found in $O(1)$ time after an $O(n)$ time preprocessing is

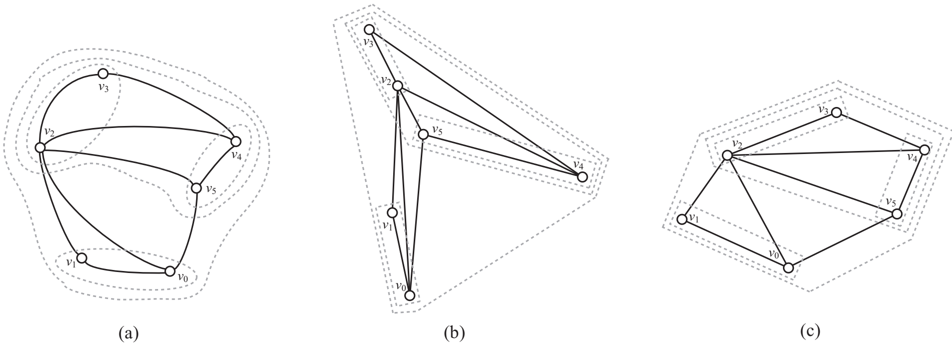

Fig. 1. Three drawings of a clustered graph C, which has five clusters $c_1 = \{v_0, v_1, v_2, v_3, v_4, v_5\}$, $c_2 = \{v_0, v_1\}$, $c_3 = \{v_2, v_3, v_4, v_5\}$, $c_4 = \{v_2, v_3\}$ and $c_5 = \{v_4, v_5\}$ each surrounded by a broken line: (a) a c-planar drawing, (b) a straight line cluster drawing, and (c) a c-planar convex cluster drawing.

applied to a rooted tree $T = (\mathcal{V}, \mathcal{E})$ [11]. We assume that in a clustered graph $C = (G, T)$, every non-leaf node of tree T has at least two children. Hence the size of $T = (\mathcal{V}, \mathcal{E})$ is $O(|\mathcal{V}| + |\mathcal{E}|) = O(n)$.

Definition 1. *For a c-planar clustered graph $G = (G, T)$, a drawing of G that satisfies the followings is called a* c-planar straight line cluster drawing *of C.*

- *Each vertex is drawn as a point and each edge is drawn as a straight line segment between two points drawn for its end vertices.*
- *For each cluster c, let the region R_c of c be the convex hull of the points drawn for the vertices in $V(c)$.*
- *For each cluster c, the edges in $G(c)$ are drawn in the interior of R_c, and the vertices and edges in $G - V(c)$ are drawn in the exterior of R_c.*
- *There are no edge crossings or edge-region crossings.*

Storing a region R_c for a cluster c in a given drawing may take $\Omega(|V(c)|)$ space, and just computing all regions would take $\Omega(n^2)$ space and time (even though computing the points for the vertices in G in a c-planar straight line cluster drawing may be done in linear time). In the above definition, we do not need explicitly store a data structure for the region R_c for each cluster c, which can be obtained by the convex hull of the points drawn for the vertices in $V(c)$. For example, the region R_c for cluster $c = \{v_0, v_1, \ldots, v_5\}$ is given by the convex hull with corner points v_0, v_1, v_3, v_4.

A clustered graph $C = (G, T)$ is a *connected clustered graph* if each cluster induces a connected subgraph of G. Eades et al. [7] proved the following result.

Theorem 1. [7] *Let $C = (G, T)$ be a c-planar connected clustered graph. Then there always exists a c-planar straight line cluster drawing of C, and such a drawing of C with n vertices can be constructed in $O(n)$ time.* □

Definition 2. *For a c-planar clustered graph $G = (G, T)$, a c-planar straight line clustered drawing of G is called a* c-planar convex cluster drawing *of C if the following holds for each cluster c, whose region R_c is defined to be the convex hull of the points drawn for the vertices in $V(c)$.*

– *The boundary of the embedding of $G(c)$ is a strict convex polygon (i.e., a convex polygon such that the internal angle at each corner is less than π).*

A connected clustered graph $C = (G, T)$ is a *biconnected clustered graph* if each cluster with more than two vertices induces a biconnected subgraph of G. In this paper, we show the next result.

Theorem 2. *Let $C = (G, T)$ be a c-planar biconnected clustered graph such that T is a binary tree. Then there always exists a c-planar convex cluster drawing of C, and such a drawing of C with n vertices and n' clusters can be constructed in $O(n + n' \log^2 n')$ time.* □

The clustered graph in Fig. 1(a) is a c-planar biconnected cluster graph, but the drawing in Fig. 1(b) is not a convex cluster drawing. Fig. 1(c) shows a convex cluster drawing of the c-planar biconnected cluster graph C in Fig. 1(a). Note that in a convex cluster drawing one can indicate the region R_c for a particular cluster c by emphasizing the boundary of $G(c)$ (say, by allocating a different color to the edges in the boundary), without drawing the boundary of R_c.

For a clustered graph $C = (G, T)$ in Theorem 2, we introduce some terminology. Fix an embedding of G in a c-planar drawing of C. For each cluster c, let $\mathrm{ex}(c)$ denote the set of vertices on the boundary of the outer face of $G(c)$. When $\mathrm{ex}(c)$ is represented by its members $\{v_1, v_2, \ldots, v_n\}$, we assume that $v_1, v_2, \ldots, v_n$ appear in clockwise order along the boundary of c.

Let c be a non-root cluster c with $|V(c)| \geq 3$, and c' and c'' be its child clusters of c. The set of edges commonly used in the boundaries of $G(c)$ and $G(c')$ forms a single path, and hence the set of edges that are used in the boundary of $G(c')$ but not in that of $G(c)$ forms a single path, say P_c. The set of vertices in P_c is denoted by $\lambda(c')$. The end vertices of P_c are denoted by $q_1(c')$ and $q_2(c')$, where we assume that P_c is a path that goes from $q_1(c')$ to $q_2(c')$ in clockwise order along the boundary of $G(c')$. We call these vertices $q_1(c')$ and $q_2(c')$ *joint vertices* of c'. The edge set $\{(v, w) \in E | v \in \lambda(c'), w \in \lambda(c'')\}$ is denoted by $\Delta(c)$ (see Fig. 2).

In what follows, for notational simplicity, a point embedded from a vertex v in a drawing of a clustered graph may be also denoted by v. For two points p_1 and p_2, the line segment between them is denoted by p_1p_2, and the line passing through them is denoted by $\ell(p_1, p_2)$.

2.1 Basic Transformations

In this subsection, we review some operations that can transform a c-planar convex cluster drawing into another c-planar convex cluster drawing. We use the following five types of operations.

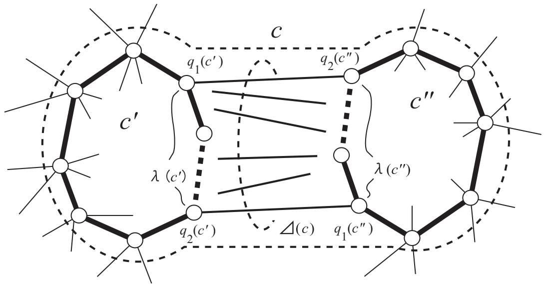

Fig. 2. Definition of $\Delta(c)$ of a cluster c, $\lambda(c')$, $q_1(c')$ and $q_2(c')$ for a child cluster c' of c'' (thick lines show cycles $\mathrm{ex}(c')$ and $\mathrm{ex}(c'')$ for the two child clusters).

Let $p = (p_x, p_y)$ be a point with an x-coordinate p_x and a y-coordinate p_y in the plane.

- Translation with respect to a vector $\boldsymbol{a} = (a_x, a_y)$: Move p to a new point $p' = (p + a_x, p + a_y)$.
- Rotation with respect to a real θ and a reference point r: Rotate p in clockwise order by the angle θ around r.
- Scaling with respect to a reference point r and $\gamma > 0$: Scale the line segment between r and p by factor γ fixing the end point r, i.e., move p to a new point $p' = (r_x + \gamma \cdot (p_x - r_x), r_y + \gamma \cdot (p_y - r_y))$.
- One-dimensional scaling with respect to a reference line ℓ, and a real $\gamma > 0$: Let r^p be the point on ℓ that is closest to p (i.e., $pr^p \perp \ell$), and move p to a new point $p' = (r^p_x + \gamma(p_x - r^p_x), r^p_y + \gamma(p_y - r^p_y))$.
- Shearing with respect to a reference line ℓ with a head and a tail, and a real $\gamma > 0$: Let $\boldsymbol{a}$ be the unit vector in the direction from the tail to the head of ℓ, and h be the distance of p from ℓ. Then move p to a new point $p' = (p_x + \gamma \cdot h \cdot a_x,\ p_y + \gamma \cdot h \cdot a_y)$ if p is on the left side with respect to ℓ with the head upward, or to a new point $p' = (p_x - \gamma \cdot h \cdot a_x,\ p_y - \gamma \cdot h \cdot a_y)$ otherwise.

For a point p in the plane, let $f(p)$ denote the point obtained by an operation f in the above. Any operation f in the above is an affine transformation, by which a given point $p = (p_x, p_y)$ is projected to a point $f(p) = (p'_x, p'_y)$ by $(p'_x, p'_y)^t := A \cdot (p_x, p_y)^t + b$ for a 2×2 matrix A and a vector b.

For a set P of points, let $f(P)$ denote the set $\{f(p) \mid p \in P\}$. We see that for a set P of points on a line segment p_1p_2, $f(P)$ is a set of a line segment $f(p_1)f(p_2)$. Also, we see that for a set P of corner points of a convex polygon, $f(P)$ is a set of corner points of some convex polygon, where the points $f(p) \in f(P)$ appear in clockwise order in the same order that they appear around P. This also implies that, after any transformation, no two edges create a corner point, i.e., no two edges cross. From these observations, we have the next.

Lemma 1. *Let $\mathcal{D}$ be a c-planar convex cluster drawing of a c-planar clustered graph C. Then the drawing $\mathcal{D}'$ obtained from $\mathcal{D}$ by applying any of the above five transformations is also a c-planar convex cluster drawing of C.* □

Note that a point $p'' = (p''_x, p''_y)$ obtained from a given point p by a sequence of operations $f_1, f_2, \ldots, f_k$ is given by $(p''_x, p''_y)^t := A'' \cdot (p_x, p_y)^t + b''$ for adequate matrix A'' and vector b''.

2.2 Drawing for c-Planar Biconnected Clustered Graphs

In the next section, we prove that every c-planar biconnected clustered graph $C = (G, T)$ such that T is a binary tree admits a c-planar convex cluster drawing. Our proof is algorithmic and can be implemented to run in polynomial time.

We construct a drawing of a given c-planar cluster graph $C = (G, T)$ by a divide-and-conquer approach. Supposing that, for a non-leaf cluster c, c-planar convex cluster drawings $\mathcal{D}(c')$ and $\mathcal{D}(c'')$ for its child clusters c' and c'' are obtained, we consider how to combine them to obtain a c-planar convex cluster drawing $\mathcal{D}(c)$ for their parent cluster c, where we may transform the two drawings if necessary. However, as shown in Fig. 3(a), in general two c-planar convex cluster drawings cannot be transformed by any of the above five operations to get a desired drawing. To overcome this, we impose a technical constraint on c-planar convex cluster drawings.

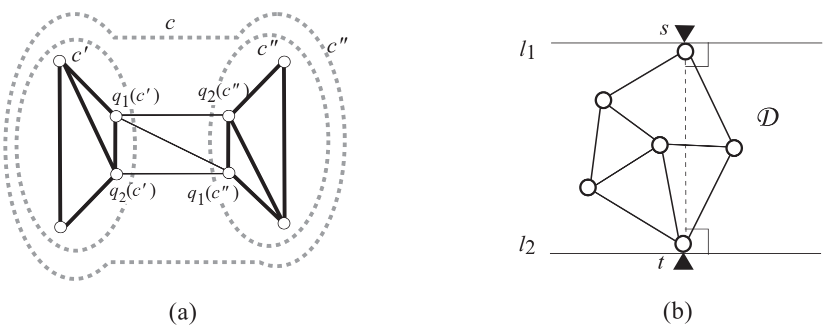

Fig. 3. (a) c-planar convex cluster drawings for child clusters c' and c'', depicted by thick lines, which cannot be combined into a c-planar convex cluster drawing for the parent cluster c; (b) a supported c-planar convex cluster drawing with support vertices s and t.

Let $\mathcal{D}$ be a c-planar convex cluster drawing of a clustered graph C. We say that a line *supports* $\mathcal{D}$ if it passes through a corner point of the boundary and all other vertices are situated in one of the half planes divided by the line. We also say that two parallel lines ℓ_1 and ℓ_2 ($\ell_1 \neq \ell_2$) *support* $\mathcal{D}$ if each of the lines supports the drawing, where the two corner points s and t (resp., their corresponding vertices in V) are called *support corners* of the drawing (resp., *support vertices*) of the clustered graph. We further say that ℓ_1 and ℓ_2 *properly support* $\mathcal{D}$ if $\ell_1 \perp st$.

Definition 3. *A c-planar convex cluster drawing is called a* supported c-planar convex cluster drawing *if there are two parallel lines ℓ_1 and ℓ_2 that properly support the drawing.*

Fig. 3(b) illustrates an example of a supported c-planar convex cluster drawing Now we formulate the following problem:

Problem 1. Input: A c-planar clustered graph $C = (G, T)$ with a planar graph G embedded in the plane and a binary tree T, and two distinct vertices s and t on the boundary of G.
Output: A supported c-planar convex cluster drawing $\mathcal{D}(C)$ of C with support vertices s and t.

For a leaf c in T, which has no child cluster, a supported c-planar convex cluster drawing $\mathcal{D}(c)$ of $G(c)$ can be easily constructed by a conventional convex drawing algorithm [13]. Also if $G(c)$ has at most three vertices, then such a drawing $\mathcal{D}(C)$ is trivially obtained. In the next section, we prove that the problem can be solved by a divide-and-conquer method.

3 Divide and Conquer

Let $C = (G, T)$ be a c-planar clustered graph with a planar graph G and a binary tree T. The clustered graph induced from C by a cluster c is denoted by $C(c) = (G(c), T(c))$, where $T(c)$ is a subtree of T rooted at node c.

Let c_r denote the root cluster of C. We choose two arbitrary vertices $\{s_r, t_r\}$ on the boundary of the outer face of G as a set $S(c_r)$ of the support vertices of C. To obtain a supported c-planar convex cluster drawing $\mathcal{D}(C)$ of C with support vertices s_r and t_r, we apply the next recursive procedure to (c_r, s_r, t_r).

DRAW(c, s, t)
if c is a leaf in T or $|V(c)| \leq 3$ holds **then**
  Return a supported c-planar convex cluster drawing $\mathcal{D}(c)$ of $C(c)$;
else
  Let c' and c'' be the two child clusters of c;
  Determine adequate pairs of support vertices $S(c') = \{s', t'\}$ and $S(c'') = \{s'', t''\}$ for induces clustered graphs $C(c')$ and $C(c'')$;
  Call DRAW(c', s', t') and DRAW(c'', s'', t'') to compute supported c-planar convex cluster drawings $\mathcal{D}(c')$ of $C(c')$ and $\mathcal{D}(c'')$ of $C(c'')$;
  Transform $\mathcal{D}(c')$ and $\mathcal{D}(c'')$ into another c-planar convex cluster drawings respectively by affine transformations $f_{c''}$ and $f_{c''}$ (which are obtained by sequences of basic transformations) before combining them into a supported c-planar convex cluster drawing $\mathcal{D}(c)$ of $C(c)$;

Return $\mathcal{D}(c)$.
/* end if */

3.1 Dividing Phase

In the dividing phase of our divide-and-conquer procedure DRAW, support vertices of a cluster are determined from the support vertices of its parent cluster in the following way. Let c be a cluster and $S(c) = \{s, t\}$ be the pair of support vertices. We choose the sets of support vertices $S(c') = \{s', t'\}$ and $S(c'') = \{s'', t''\}$ of its child clusters c' and c''. We distinguish the next two cases.

Case-1. One of $G(c')$ and $G(c'')$ contains both support vertices in $S(c)$; $S(c) \subseteq V(c')$ is assumed without loss of generality (see Fig. 4(a)). Define the pairs $S(c')$ and $S(c'')$ of support vertices of c' and c'' by

$$s' := s, \quad t' := t, \quad s'' := q_2(c'') \text{ and } t'' := q_1(c'').$$

Case-2. One of the support vertices in $S(c)$ belongs to $G(c')$ and the other $G(c'')$ (see Fig. 4(b)). Define the pairs $S(c')$ and $S(c'')$ of support vertices of c' and c'' by

$$s' := q_1(c'), \quad t' := q_2(c'), \quad s'' := q_2(c'') \text{ and } t'' = q_1(c'').$$

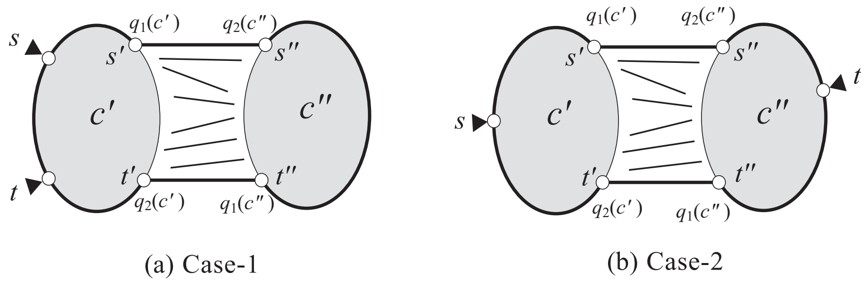

Fig. 4. Illustration for support vertices, where (a) and (b) indicate Cases-1 and 2, respectively.

3.2 Combining Phase

In this subsection, we consider the combining phase of DRAW. Suppose that, for two child clusters c' and c'' of a cluster c, we have obtained their supported c-planar convex cluster drawings $\mathcal{D}(c')$ and $\mathcal{D}(c'')$ with the support vertices which have been determined during the dividing phase. We may transform each of the drawings $\mathcal{D}(c')$ and $\mathcal{D}(c'')$ by some of the basic transformations.

In what follows, we further assume that $|V(c')| \geq 3$ and $|V(c'')| \geq 3$ (the case of $|V(c')| \leq 2$ or $|V(c'')| \leq 2$ can be treated with a slight modification of the subsequent argument). It suffices to show that two drawings for child clusters can be combined so as to meet the following three conditions:

(i) The boundary formed for $G(c)$ is a strict convex polygon.
(ii) Edges in $\Delta(c)$ (i.e., those between $G(c')$ and $G(c'')$) are drawn by line segments without creating any intersection with each other or with the boundaries of $G(c')$ and $G(c'')$.
(iii) There exits a pair of parallel lines that support the drawing for $G(c)$.

In (iii), the parallel lines do not necessarily *properly* support the drawing since we can always make them properly support the drawing by applying a shearing operation with respect to one of the lines.

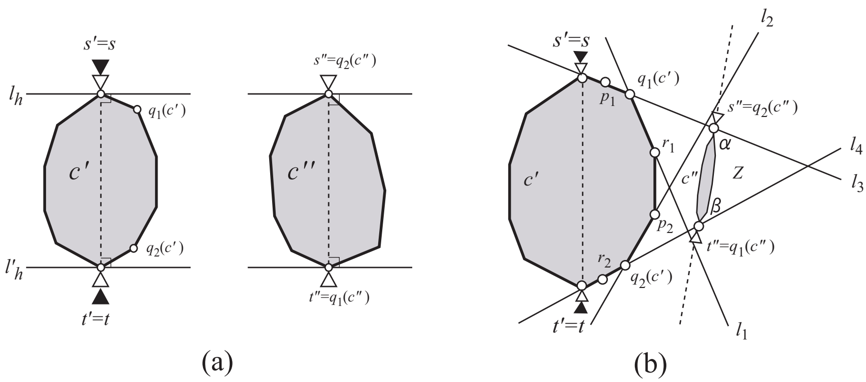

Fig. 5. (a) Given supported c-planar convex cluster drawings $\mathcal{D}(c')$ and $\mathcal{D}(c'')$ in Case 1, and (b) Combining supported c-planar convex cluster drawings $\mathcal{D}(c')$ and $\mathcal{D}(c'')$ in Case 1.

We fix the drawing $\mathcal{D}(c')$ in the xy-plane so that its supporting lines ℓ_h and ℓ'_h are parallel with the x-axis (hence the line segment $s't'$ is parallel with the y-axis). Although the other drawing $\mathcal{D}(c'')$ may be transformed before being combined with $\mathcal{D}(c')$, we temporarily fix $\mathcal{D}(c'')$ so that its supporting lines are parallel with the x-axis. We assume without loss of generality that ℓ_h and ℓ'_h pass through point s' and t', respectively, and that the y-coordinate of s' (resp., $q_1(c')$, s'', $q_2(c'')$) is larger than that of t' (resp., $q_2(c')$, t'', $q_1(c'')$).

Let $p_1, r_1 \in \mathrm{ex}(c')$ (resp., $p_2, r_2 \in \mathrm{ex}(c')$) be the vertices adjacent to $q_1(c')$ (resp., $q_2(c')$) such that these vertices appear along the boundary of $\mathcal{D}(c')$ in the order of $p_1, q_1(c'), r_1, p_2, q_2(c')$ and r_2. Define lines $\ell_1 = \ell(q_1(c'), r_1)$, $\ell_2 = \ell(q_2(c'), p_2)$, $\ell_3 = \ell(p_1, q_1(c'))$ and $\ell_4 = \ell(r_2, q_2(c'))$, where for a technical reason we set $\ell_3 = \ell_h$ if $q_1(c') = s'$ (resp., $\ell_4 = \ell'_h$ if $q_1(c') = t'$).

The other drawing $\mathcal{D}(c'')$ will be situated in a region Z which is defined below distinguishing Cases-1 and 2.

Case-1 (See Fig. 5(a).) The Z is set to be the region that is above lines ℓ_1 and ℓ_4 and below lines ℓ_2 and ℓ_3. We now show how to put $\mathcal{D}(c'')$ inside the Z. Choose an internal point α (resp., β) of the line segment induced by Z from ℓ_3 (resp., ℓ_4).

By applying translation, rotation and scaling operations, we put $\mathcal{D}(c'')$ in Z in such a way that $q_2(c'')$ and $q_1(c'')$ fall on the points α and β, respectively. Let $\mathcal{D}_1$ be the drawing transformed from $\mathcal{D}(c'')$ in this way. Finally we apply to $\mathcal{D}_1$ a one-dimensional scaling with respect to line $\ell(\alpha, \beta)$ and a sufficiently small real γ so that all points (except for $q_2(c'')$ and $q_1(c'')$) in the drawing $\mathcal{D}_2$ transformed from $\mathcal{D}_1$ are situated properly inside Z (see Fig.5(b)). Then a drawing $\mathcal{D}(c)$ for the parent cluster c is set to be the union of $\mathcal{D}(c')$ and $\mathcal{D}_2$. It is not difficult to see that the resulting drawing $\mathcal{D}(c)$ for $C(c)$ satisfies all conditions (i),(ii) and (iii) for supporting lines ℓ_h and ℓ'_h.

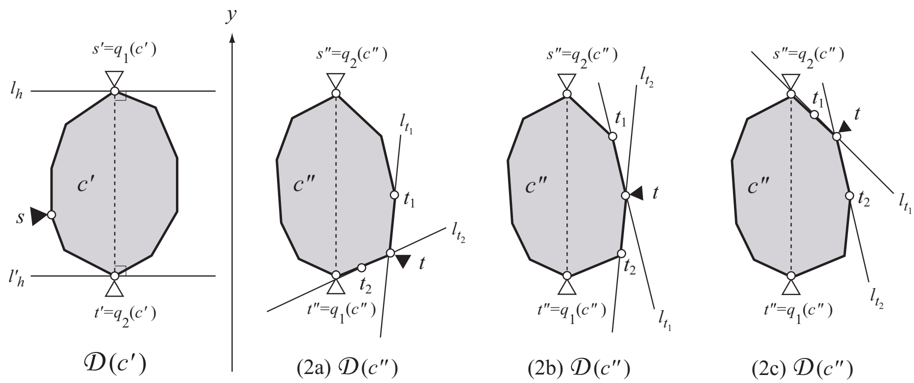

Fig. 6. Illustration for supported c-planar convex cluster drawings $\mathcal{D}(c')$ and $\mathcal{D}(c'')$ in Case 2, where (2a), (2b) and (2c) show three subcases for $\mathcal{D}(c'')$.

Case-2 (See Fig. 6.) Assume that two support vertices s and t in $S(c)$ belong to c' and c'', respectively. Let s_1 and s_2 be the vertices in $ex(c')$ that appear respectively before and after s when we visit the boundary of $G(c')$ in clockwise order. Let ℓ_{s_i} $(i = 1, 2)$ be the line $\ell(s, s_i)$.

Let t_1 and t_2 be the vertices in $ex(c'')$ that appear respectively before and after t when we visit the boundary of $G(c'')$ in clockwise order, and let $\ell_{t_i} = \ell(t, t_i)$ $(i = 1, 2)$. Then we define lines ℓ^{ref} and ℓ^{tan} by distinguishing the following three subcases, where $g(\ell)$ for a line ℓ denotes the gradient of ℓ.

(2a) $g(\ell_{t_1}) > 0$ or $\ell_{t_1} \parallel s''t''$ (see Fig. 6(2a)). Then define ℓ^{ref} (resp., ℓ^{tan}) to be a line parallel with ℓ_{s_1} (resp., ℓ_{s_2}).
(2b) $g(\ell_{t_2}) > 0$ and $g(\ell_{t_1}) < 0$ (see Fig. 6(2b)). Then define ℓ^{ref} (resp., ℓ^{tan}) to be a line parallel with ℓ_{s_1} (resp., ℓ_{s_2}).
(2c) $g(\ell_{t_2}) < 0$ or $\ell_{t_2} \parallel s''t''$ (see Fig. 6(2c)). Then define ℓ^{ref} (resp., ℓ^{tan}) to be a line parallel with ℓ_{s_2} (resp., ℓ_{s_1}).

We consider case (2a) (the rest of the cases can be treated similarly). Let α (resp., β) be the intersection of ℓ_{ref} and ℓ_h (resp., ℓ_{ref} and ℓ'_h). We choose ℓ_{ref} so that both α and β are strictly above ℓ_1 and below ℓ_2. Let $\ell'_1 = \ell(q_1(c'), \beta)$ and $\ell'_2 = \ell(q_2(c'), \alpha)$. Then the Z is defined to be the region that is above ℓ'_1 and ℓ'_h and below ℓ'_2 and ℓ_h.

We now show how to put the drawing $\mathcal{D}(c'')$ within the region Z. We first transform $\mathcal{D}(c'')$ by applying translation, rotation and scaling operations so that points $q_2(c'')$ and $q_1(c'')$ fall on α and β, respectively. We next apply one-dimensional scaling operation to the resulting drawing $\mathcal{D}_1$ so that all the points in the drawing except for $\alpha = q_2(c'')$ and $\beta = q_1(c'')$ are properly contained in Z and ℓ^{tan} has a gradient between those of ℓ_{s_1} and ℓ_{s_2} (see Fig. 7). Since line $\ell^{ref} = \ell(\alpha, \beta)$ is parallel with ℓ_{s_2}, line $\ell_{t_1} = \ell(t, t_1)$ will have the gradient between those of ℓ_{s_1} and ℓ_{s_2} if $\mathcal{D}(c'')$ gets enough close to line segment $\alpha\beta$. Let $\mathcal{D}_2$ be the resulting drawing.

As observed in Case 1, it is not difficult to see that the drawing $\mathcal{D}(c)$ obtained by combining $\mathcal{D}(c')$ and $\mathcal{D}_2$ meets conditions (i), (ii) and (iii) with lines ℓ_{s_2} and ℓ^{tan} in $\mathcal{D}_2$.

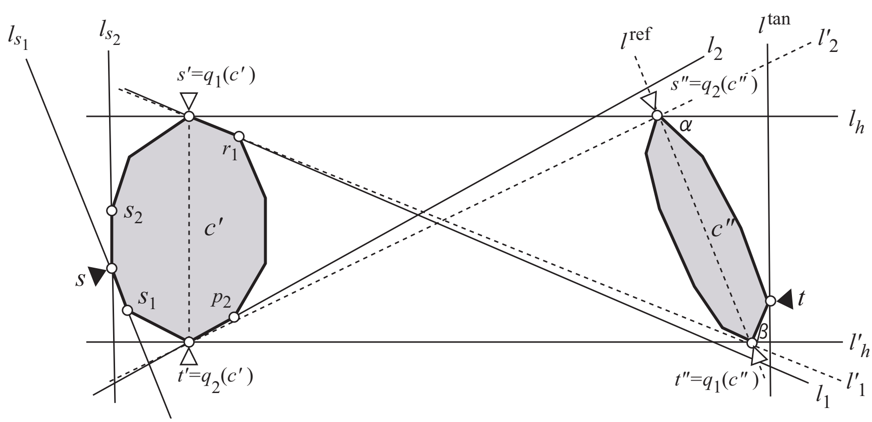

Fig. 7. Combining supported c-planar convex cluster drawings $\mathcal{D}(c')$ and $\mathcal{D}(c'')$ in Case 2.

Each iteration in the dividing and combining phases of DRAW can be executed in $O(n)$ time. There are $O(|\mathcal{V}|) = O(n')$ such iterations. A naive implementation of DRAW takes $O(n'n)$ time. This can be reduced to $O(n+n'\log^2 n')$. Note that, to determine $f_{c'}$ and $f_{c''}$ in an iteration of the combining phase, we only need to know the positions of joint nodes of c' and c'' and their neighbors on $\mathrm{ex}(c')$ and $\mathrm{ex}(c'')$. All such nodes can be identified in linear time before resorting procedure DRAW. The position of such a node, say $q_1(c')$ in drawing $\mathcal{D}(c')$ can be obtained by $f_{\langle c_1,c'\rangle}(q_1(c'))$ after computing the synthesized affine transformation

$$f_{\langle c_1,c'\rangle} \equiv f_{c_k} \circ f_{c_{k-1}} \circ \cdots \circ f_{c_2} \circ f_{c_1},$$

where c_1 is the leaf cluster containing $q_1(c')$ and $c_1, c_2, \ldots, c_k, c_{k+1} (= c')$ are the clusters that appear in this order from c to c' in T. The computation can be executed in $O(n'\log^2 n')$ time by a technique of pointer jumping and a decomposition of T into disjoint paths (the detail is omitted). Once the affine transformations f_c, $c \in \mathcal{V} - \{c_r\}$ have been determined, the final drawing $\mathcal{D}(c_r)$ can be constructed by transforming the points in each leaf cluster c' by $f_{\langle c',c_r\rangle}$ in $O(n + n'\log^2 n')$ time.

References

1. G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Graph drawing*, Prentice-Hall, 1999.
2. V. Batagelj, D. Kerzic and T. Pisanski, Automatic clustering of languages, *Computational Linguistics*, 18(3) 339–352, 1992.
3. E. Dahlhaus, Linear time algorithm to recognize clustered planar graphs and its parallelization (extended abstract), *Latin America symposium on theoretical informatics* (LATIN'98), Lecture Notes in Computer Science, 1380, 239–248, 1998.
4. C. A. Duncan, M. T. Goodrich, and S. G. Kobourov, Planarity-preserving clustering and embedding for large planar graphs, *Graph Drawing* (GD'99), Lecture Notes in Computer Science, 1731, 186–196, 1999.
5. P. Eades, Q.-W. Feng and H. Nagamochi, Drawing clustered graphs on an orthogonal grid, *Journal of Graph Algorithms and Application*, 3(4), 3–29, 1999.
6. P. Eades and Q.-W. Feng, Drawing clustered graphs on orthogonal grid, *Graph Drawing* (GD'97), Lecture Notes in Computer Science, 1353, 146–157, 1997.
7. P. Eades, Q.-W. Feng, X. Lin, and H. Nagamochi, Straight-line drawing algorithms for hierarchical graphs and clustered graphs, *Technical Report Newcastle University* (submitted to a journal), 1999.
8. Q.-W. Feng, R.-F. Cohen and P. Eades, Planarity for clustered graphs, *Algorithms* (ESA'95), Lecture Notes in Computer Science, 979, 213–226, 1995.
9. E. Godehardt, *Graphs as Structural Models*, Advances in System Analysis 4, Vieweg, 1998.
10. C. Gutwenger, M. Jünger, S. Leipert, P. Mutzel, M. Percan and R. Weiskircher, Advances in c-planarity testing of clustered graphs (extended abstract), *Graph Drawing* (GD 2002), Lecture Notes in Computer Science, 2528, 220–235, 2002.
11. D. Harel and R. Tarjan, Fast algorithms for finding nearest common ancestors, *SIAM J. Computing*, 13, 338–355, 1984.
12. M. Kaufmann and D. Wagner, Drawing graphs, *Lecture Notes in Computer Science*, 2025, Springer, 2001.
13. T. Nishizeki and N. Chiba, *Planar Graphs: Theory and Algorithms*, North-Holland Mathematics Studies 140/32, 1988.