UNIVERSITÀ DEGLI STUDI DI ROMA TRE
Dipartimento di Informatica e Automazione
Via della Vasca Navale, 79 – 00146 Roma, Italy

How to Draw a Clustered Tree

GIUSEPPE DI BATTISTA, GUIDO DROVANDI AND FABRIZIO FRATI

RT-DIA-115-2007 **Ottobre 2007**

Dipartimento di Informatica e Automazione – Università di Roma Tre, Italy.

ABSTRACT

The visualization of clustered graphs is a classical algorithmic topic that has several practical applications and is attracting increasing research interest. In this paper we deal with the visualization of clustered trees, a problem that is somehow foundational with respect to the one of visualizing a general clustered graph. We show many, in our opinion, surprising results that put in evidence how drawing clustered trees has many sharp differences with respect to drawing "plain" trees. We study a wide class of drawing standards, giving both negative and positive results. Namely, we show that there are clustered trees that do not have any drawing in certain standards and others that require exponential area. On the contrary, for many drawing conventions there are efficient algorithms that allow to draw clustered trees with polynomial asymptotic optimal area.

1 Introduction and Overview

The problem of drawing trees is a classical topic of investigation in algorithmics. Contributions on that field span almost three decades, from the groundbreaking work of Valiant [14] to the recent papers of Garg and Rusu that investigate how to obtain optimal area drawings with prescribed aspect ratio [12]. Algorithms for drawing trees have been proposed within a wide spectrum of drawing conventions. To give a few examples, upward drawings have been studied in [11], straight-line upward drawings in [13, 1], and straight-line orthogonal drawings in [3, 2].

Despite such a large amount of investigation on algorithms for drawing trees no contribution has been presented in the literature on how to draw *clustered trees*.

A *clustered graph* is a pair $C = (G, T)$, where G is a graph and T is a rooted tree such that the leaves of T are the vertices of G. Graph G and tree T are called *underlying graph* and *inclusion tree*, respectively. Fig. 1 shows a clustered graph. A *clustered tree* is a clustered graph whose underlying graph is a tree. The clustered graph in Fig. 2 is a clustered tree. Each internal node ν of T corresponds to the subset $V(\nu)$ of the vertices of G (called *cluster*) that are the leaves of the subtree rooted at ν. The subgraph of G induced by $V(\nu)$ is denoted by $G(\nu)$, where ν is a cluster of T. If each cluster induces a connected subgraph of G, then C is c-*connected*. The clustered tree in Fig. 2 is not c-connected.

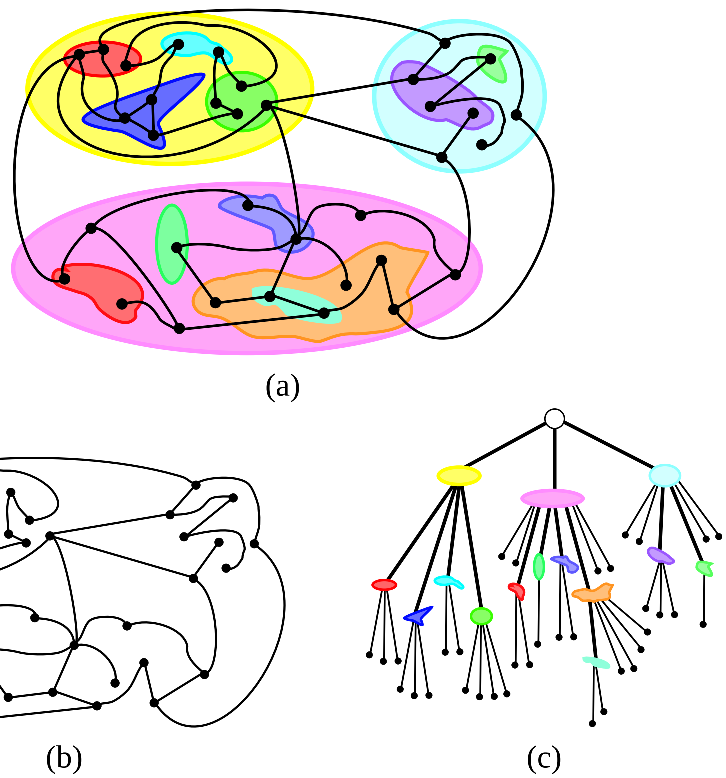


Figure 1: (a) A c-planar drawing of a clustered graph $C = (G, T)$. (b) The underlying graph G of C. (c) The inclusion tree T of C.

clustered graphs are widely used in applications where it is needed at the same time to show relationships between entities and to group entities with semantic affinities. For example, in a social network representing the working relationships between the employees of a company it might be desirable to group into clusters the people of each department.

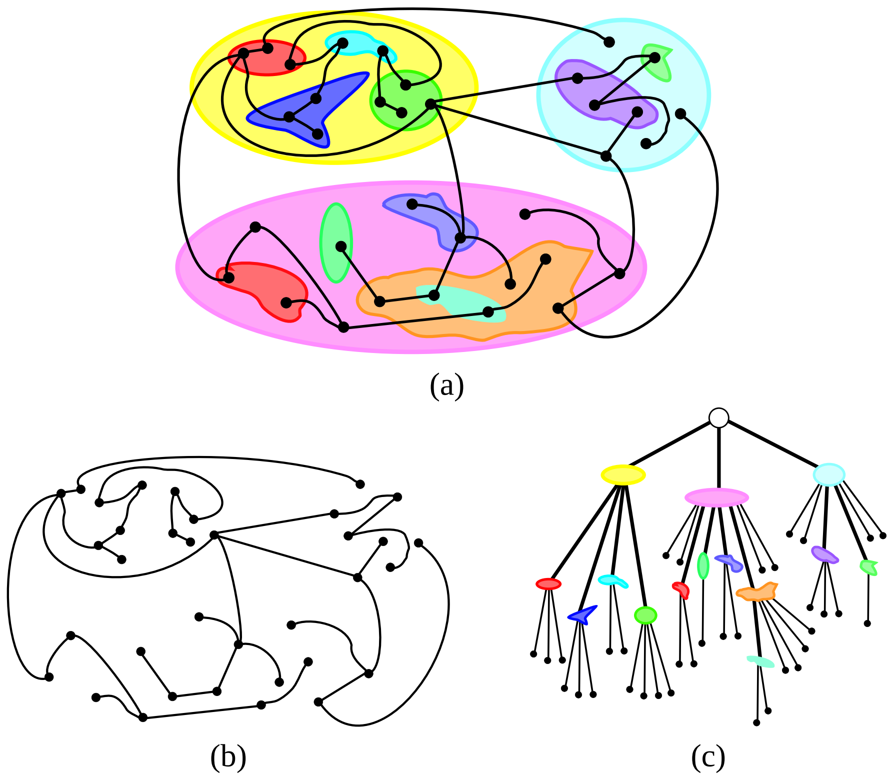


Figure 2: (a) A c-planar drawing of a clustered tree $C = (G, T)$. (b) The underlying graph G of C. Note that G is a tree. (c) The inclusion tree T of C.

Visualizing clustered graphs turns out to be a difficult problem, due to the simultaneous need for a readable drawing of the underlying structure and for a good rendering of the recursive clustering relationships. As for the visualization of graphs, the most important aesthetic criterion for a drawing of a clustered graph to be "nice" is commonly regarded to be the *planarity*. However, the classical concept of planarity needs a refinement in the contest of clustered graphs, in order to deal also with the clustering structure.

A drawing of a clustered graph $C = (G, T)$ consists of a drawing of G and of a representation of each node ν of T as a simple closed region $R(\nu)$ such that: (i) $R(\nu)$ contains the drawing of $G(\nu)$; (ii) $R(\nu)$ contains a region $R(\mu)$ iff μ is a descendant of ν in T; and (iii) the borders of any two regions do not intersect. Consider an edge e and a node ν of T. If e crosses the boundary of $R(\nu)$ more than once, we say that edge e and region $R(\nu)$ have an *edge-region crossing*. A drawing of a clustered graph is *c-planar* if it does not have edge crossings or edge-region crossings and a graph is c-planar if it has a c-planar drawing.

A number of papers have been presented for constructing c-planar drawings of clustered graphs within many drawing conventions. Namely, Eades, Feng, and Nagamochi in [7] show how to construct $O(n^2)$ area c-planar orthogonal drawings of clustered graphs with maximum degree 4. Eades et al. [6] present an algorithm for constructing c-planar straight-line drawings of clustered graphs, where clusters are drawn as convex regions. Such an algorithm requires, in general, exponential area. However, in [9] it is shown that such a bound is asymptotically optimal in the worst case.

In this paper we look for algorithms for constructing c-planar drawings of clustered trees in efficient area, by considering the most investigated drawing standards for the underlying tree (see, e.g., [2, 1, 11, 12]). We deal both with c-connected and non-c-connected clustered trees and consider drawings in which the clusters are represented by rectangles (R-drawings), by convex polygons (C-drawings), and also by eventually non-convex polygons (NC-drawings). In most cases we are able to find asymptotically optimal area bounds. After preliminaries (Section 2), in Section 3 we deal with c-connected clustered trees and show that quadratic area is achievable for many drawing styles, namely strictly upward order-preserving poly-line R-drawings, strictly upward non-order-preserving straight-line R-drawings, and upward orthogonal order-preserving R-drawings (if the underlying graph is a binary tree). Such results are interesting to compare with the results in the above mentioned [9], where it is shown that for general c-connected clustered graphs exponential area can be needed. Furthermore, such bounds are asymptotically optimal in the worst-case. On the other hand, we show that orthogonal straight-line R-drawings are generally not realizable. In Section 4 we deal with non-c-connected clustered trees: we show that straight-line C-drawings generally require exponential area, and that poly-line order-preserving drawings can be realized in optimal quadratic area. Moreover, we show that upward drawings of non-c-connected clustered trees aren't generally feasible. In Section 5 we show that if the clusters can be represented by non-convex regions, then polynomial area is achievable in many cases.

A summary of the results presented in this paper is given in Tables 1 and 2, where "UB" and "LB" stand for *Upper Bound* and *Lower Bound*, respectively. "Upward" means *upward* when referred to orthogonal drawings and means *strictly upward* otherwise. If the straight-line column does not have a "✓", then the drawing is poly-line. Orthogonal drawings are referred to binary trees. An "X" means that in general a drawing with the corresponding features does not exist. Observe that an area upper bound obtained within a certain drawing convention (say, upward straight-line) for R-drawings is also an upper bound for C-drawings and for NC-drawings. On the contrary, a lower bound for NC-drawings implies a lower bound for C-drawings and for R-drawings.

				R-Drawings				C-Drawings				NC-Drawings			
upward	straight-line	ordered	orthogonal	UB	ref.	LB	ref.	UB	ref.	LB	ref.	UB	ref.	LB	ref.
✓	✓			$O(n^2)$	Th. 2	$\Omega(n^2)$	Le. 1	$O(n^2)$	Th. 2	$\Omega(n^2)$	Le. 1	$O(n^2)$	Th. 2	$\Omega(n^2)$	Le. 1
✓	✓	✓		?	-	$\Omega(n^2)$	Le. 1	?	-	$\Omega(n^2)$	Le. 1	$O(n^4)$	Th. 9	$\Omega(n^2)$	Le. 1
✓		✓		$O(n^2)$	Th. 1	$\Omega(n^2)$	Le. 1	$O(n^2)$	Th. 1	$\Omega(n^2)$	Le. 1	$O(n^2)$	Th. 1	$\Omega(n^2)$	Le. 1
✓		✓	✓	$O(n^2)$	Th. 4	$\Omega(n^2)$	Le. 1	$O(n^2)$	Th. 4	$\Omega(n^2)$	Le. 1	$O(n^2)$	Th. 4	$\Omega(n^2)$	Le. 1
	✓		✓	**X**		Th. 5		?	-	$\Omega(n^2)$	Le. 1	$O(n^3 \log n)$	Th. 10	$\Omega(n^2)$	Le. 1
		✓	✓	$O(n^2)$	[7]	$\Omega(n^2)$	Le. 1	$O(n^2)$	[7]	$\Omega(n^2)$	Le. 1	$O(n^2)$	[7]	$\Omega(n^2)$	Le. 1

Table 1: Summary of the results on minimum area drawings of c-connected clustered trees.

				R-Drawings				C-Drawings				NC-Drawings			
upward	straight-line	ordered	orthogonal	UB	ref.	LB	ref.	UB	ref.	LB	ref.	UB	ref.	LB	ref.
✓				**X**		Th. 6		**X**		Th. 6		?	-	$\Omega(n^2)$	Le. 1
	✓			?	-	$\Omega(2^n)$	Th. 7	$O(2^n)$	[6]	$\Omega(2^n)$	Th. 7	?	-	$\Omega(n^2)$	Le. 1
		✓	✓	$O(n^2)$	[7]	$\Omega(n^2)$	Le. 1	$O(n^2)$	[7]	$\Omega(n^2)$	Le. 1	$O(n^2)$	[7]	$\Omega(n^2)$	Le. 1
		✓		$O(n^2)$	Th. 8	$\Omega(n^2)$	Le. 1	$O(n^2)$	Th. 8	$\Omega(n^2)$	Le. 1	$O(n^2)$	Th. 8	$\Omega(n^2)$	Le. 1

Table 2: Summary of the results on minimum area drawings of non-c-connected clustered trees.

2 Preliminaries

We assume familiarity with trees, clustered graphs, and graph drawing (see, e.g., [4]).

A *grid drawing* of a graph is a mapping of each vertex v to a point $(x(v), y(v))$ in the plane, where $x(v)$ and $y(v)$ are integers, and of each edge to a Jordan curve between the endpoints of the edge. A *planar drawing* is such that no two edges intersect. A *planar graph* is a graph that admits a planar drawing. A *poly-line* drawing is such that the edges are sequences of rectilinear segments. An *orthogonal drawing* is such that the edges are sequences of axis-parallel rectilinear segments. A *straight-line* drawing is such that all edges are rectilinear segments. The smallest rectangle with sides parallel to the axes that covers a drawing completely is called *bounding box* of the drawing. The *height* (*width*) of a drawing is one plus the height (resp. width) of its bounding box. The *area* a drawing is the product of its height by its width.

A *rooted tree* is a tree with one distinguished node called *root*. A *binary tree* is a rooted tree such that each node has at most 2 children. A *complete binary tree* is a rooted binary tree such that each non-leaf node has exactly two children and such that each path from the root to a leaf has the same number of nodes. For an underlying tree G (for an inclusion tree T) the subtree of G (the subtree of T) rooted at a vertex v is denoted by $G(v)$ (by $T(v)$).

A drawing of a tree is *upward* (*strictly upward*) if every node is placed not below (above) its children and each edge is represented by a curve non-increasing (monotonically decreasing) in the vertical direction. A drawing of a tree is *order-preserving* if the order of the edges incident on each node is the same of one specified in advance.

We define the following drawing conventions for clustered graphs. A polygon with vertices having integer coordinates is a *lattice polygon*.

Definition 1 *A drawing of a clustered tree $C = (G, T)$ is an NC-drawing (*Non-Convex-*drawing) if it is c-planar, the vertices of G and the bends on the edges of G (if any) have integer coordinates, and the border of each cluster is a lattice polygon.*

Definition 2 *A drawing of a clustered tree $C = (G, T)$ is an C-drawing (*Convex-*drawing) if it is c-planar, the vertices of G and the bends on the edges of G (if any) have integer coordinates, and the border of each cluster is a convex lattice polygon.*

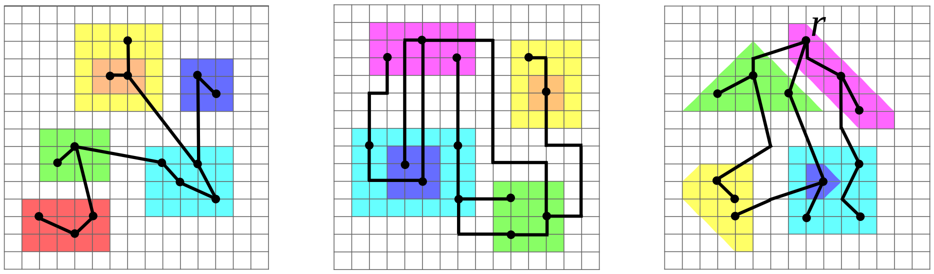
Definition 3 *A drawing of a clustered tree $C = (G, T)$ is an R-drawing (*Rectangle-*drawing) if it is c-planar, the vertices of G and the bends on the edges of G (if any) have integer coordinates, and the border of each cluster is an axis-parallel rectangle with corners having integer coordinates.*

Notice that by definition an R-drawing is also a C-drawing and a C-drawing is also an NC-drawing. Figure 3 shows example of R-drawings and C-drawings within different drawing standards for the underlying tree.

The following lemma is easy to prove.

Lemma 1 *There exist n-vertex clustered trees requiring $\Omega(n^2)$ area in any NC-drawing.*

Proof: Consider a clustered tree $C = (G, T)$ such that T has height h. We show by induction on h that C requires $2h$ width in any NC-drawing. If $h = 1$ then the border of the only cluster of T must be drawn as a simple lattice polygon and hence it must intersect at least two vertical grid lines of the plane. Suppose by inductive hypothesis that $2h - 2$ is the minimum



(a) (b) (c)

Figure 3: (a) A straight-line R-drawing. (b) An orthogonal R-drawing. (c) A poly-line upward C-drawing.

width of a NC-drawing of a clustered tree $C' = (G', T')$ such that the height of T' is $h - 1$. Consider a clustered tree $C = (G, T)$ such that the height of T is h. Clearly, there exists a subtree $T(\nu)$ of T rooted at a child ν of the root r of T that has height $h - 1$. By inductive hypothesis every NC-drawing Γ' of $C' = (G(\nu), T(\nu))$ requires $2h - 2$ width. Draw the polygon P representing r. By definition of NC-drawing, P surrounds Γ' and its vertices have integer coordinates, so P must touch the vertical grid line one unit to the left of the leftmost vertical line intersecting Γ' and must touch the vertical grid line one unit to the right of the rightmost vertical line intersecting Γ'. It follows that C requires the width of Γ' plus two units, so it requires $2h$ width. Analogously, the minimum height of an NC-drawing of a clustered tree with height h is $2h$. Since there exist clustered trees $C = (G, T)$ such that T has height $\Omega(n)$, the lemma follows. □

3 R-Drawings of C-Connected C-Trees

We show that quadratic area is sufficient (and necessary) to construct R-drawings of c-connected clustered trees in which the underlying tree is represented within several drawing standards. Namely, we present an algorithm for constructing $\Theta(n^2)$ area strictly upward order-preserving poly-line R-drawings of n-node c-connected clustered trees. Then, we show how to slightly modify such an algorithm to obtain different kinds of drawings.

Let $C = (G, T)$ be a c-connected clustered tree and suppose G is rooted at any node r. For each cluster of T we add dummy vertices and edges to G as follows. Consider any arbitrary order of the clusters. For each cluster, vertices and edges are added to the clustered tree obtained from the augmentations performed when considering the previous clusters. For each cluster μ inducing a subtree $G(\mu)$ of G, consider the root r_μ of $G(\mu)$.

If the parent p_μ of r_μ exists in the augmented graph G (see Fig. 4.a), then insert a dummy node s_μ splitting edge (p_μ, r_μ) into edges (p_μ, s_μ) and (s_μ, r_μ). Node s_μ is the parent of r_μ and the child of p_μ. If r_μ is the root of the augmented graph G (Fig. 4.b), then insert a dummy node s_μ and an edge (s_μ, r_μ). Node s_μ is the parent of r_μ (and hence the root of the newly augmented graph G). In any case, add also two dummy vertices c^1_μ and c^2_μ as children of s_μ and a dummy node c^3_μ as child of c^2_μ. The counter-clockwise order of the children of s_μ is c^1_μ, r_μ, and c^2_μ (Fig. 4.c and 4.d). Vertices s_μ, c^1_μ, c^2_μ, and c^3_μ belong to cluster μ. After having performed the described augmentation on each cluster, we obtain a clustered tree $C' = (G', T')$. We call r' the root of G'.

We now construct a strictly upward drawing of G'. Denote by $p(v)$ the parent of a node v.

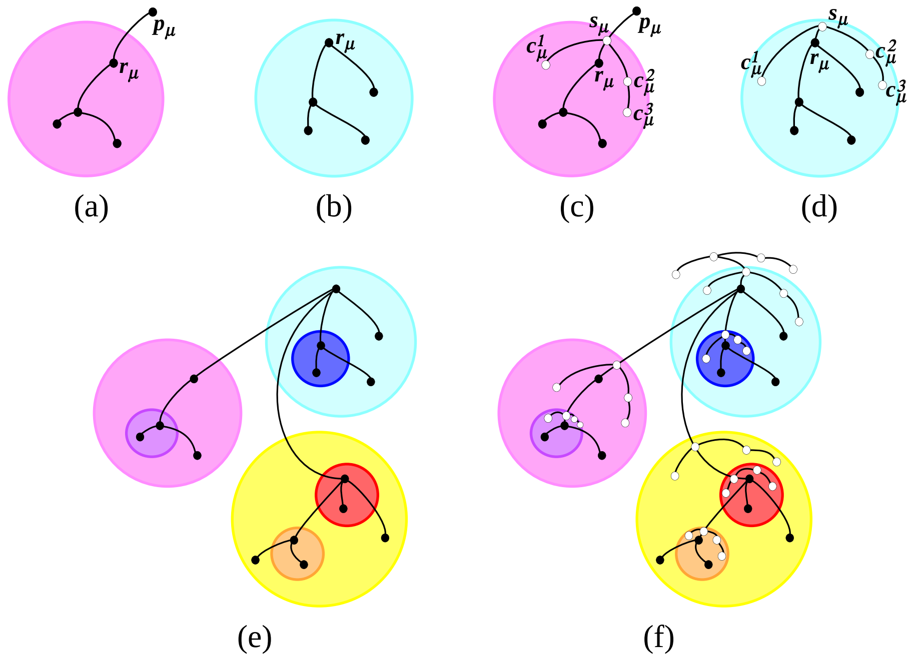


Figure 4: (a) A cluster μ and the subgraph $G(\mu)$ of G induced by μ. The root of $G(\mu)$ is not the root of the current underlying tree G. (b) A cluster μ and the subgraph $G(\mu)$ of G induced by μ. The root of $G(\mu)$ is the root of the current underlying tree G. (c)–(d) $G(\mu)$ augmented with dummy vertices and edges. (e) A clustered tree $C = (G, T)$. (f) Clustered tree $C' = (G', T')$ obtained by the augmentation of C with dummy vertices and edges.

First, assign an x-coordinate to each node in G' by means of a depth-first traversal of G'. Set $x(r') = 1$. Then, suppose that the x-coordinate has been already assigned to a node v. Let $v_1, v_2, \ldots, v_m$ be the children of v in counter-clockwise order. Set $x(v_1) = x(v)$; for each child v_i of v, $i = 2, \ldots, m$, set $x(v_i) = 1 + \max_{u \in G'(v_{i-1})}\{x(u)\}$ (see Fig. 5).

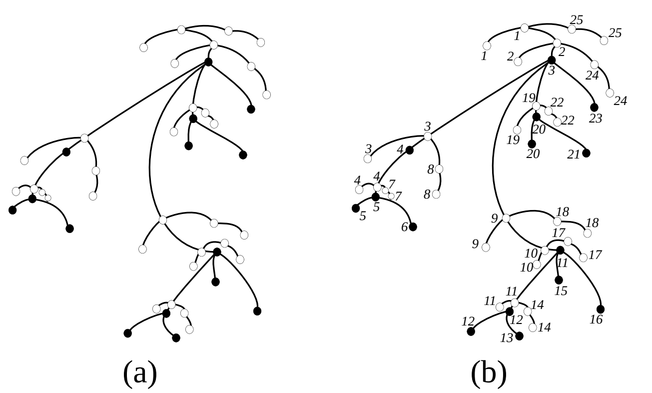


Figure 5: (a) The underlying tree G' of the clustered tree C' of Fig. 4. (b) The x-coordinates assignment to the vertices of G'.

Concerning the y-coordinates, we perform a traversal of G' based on the clustering hierarchy of C'. Such a visit has the following properties: (i) we maintain a *current cluster* such that when node $\mu \in T'$ is the current cluster, all nodes belonging to $V'(\mu)$ are visited before all not yet visited nodes not belonging to $V'(\mu)$; (ii) when current cluster is node μ, the next visited node is the first not yet visited node in a depth-first traversal of $G'(\mu)$; and (iii) when a node v is visited the current cluster is set equal to the smallest cluster containing v and containing at least one not yet visited node.

The y-coordinates assignment is as follows (see Fig. 6): Set $y(r') = 1$. Let μ_r be the smallest cluster containing r'. Set the current cluster to be μ_r. Now suppose that node $\mu \in T'$ is the current cluster.

- If there is more than one node in $V'(\mu)$ that is not yet visited, then consider the first not yet visited node $v \in V'(\mu)$ that is encountered in a depth-first traversal of $G'(\mu)$.
  - If the smallest cluster ν containing v is the same cluster or is a descendant of the smallest cluster containing $p(v)$, then set $y(v) = y(p(v)) - 1$; cluster ν is the new current cluster.
  - Otherwise (the smallest cluster ν containing v is not the same cluster or a descendant of the smallest cluster containing $p(v)$) set $y(v)$ equal to the minimum y-coordinate of a node in the biggest cluster containing $p(v)$ and not containing v minus one; cluster ν is the new current cluster.
- If there is exactly one node in $V'(\mu)$ that is not yet visited, then such node is c^3_μ; set $y(c^3_\mu)$ equal to the minimum y-coordinate of a node in $V'(\mu)$ minus one; set the current cluster to be μ.
- If all nodes in $V'(\mu)$ are already visited, then set the current cluster to be the parent of μ.

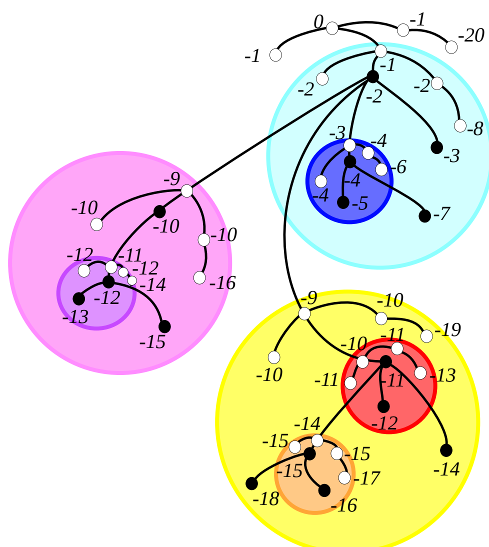


Figure 6: The y-coordinates assignment to the vertices of the underlying tree G' of the clustered tree C' of Fig. 4.

For each cluster μ remove vertices s_μ, c^1_μ, c^2_μ, and c^3_μ and their incident edges, and insert a rectangle R_μ:$[x(s_\mu), x(c^3_\mu)] \times [y(c^3_\mu), y(s_\mu)]$ representing μ in the final drawing. Draw the edges of G: For each edge $(p(v), v)$ in G, if $y(v) = y(p(v)) - 1$ then draw a straight-line segment between $p(v)$ and v, otherwise ($y(v) < y(p(v)) - 1$) draw a polygonal line composed of two segments, the first between $p(v)$ and point $(x(v), y(p(v)) - 1)$, and the second between point $(x(v), y(p(v)) - 1)$ and v. Figure 7 shows a drawing constructed by the described algorithm. We have:

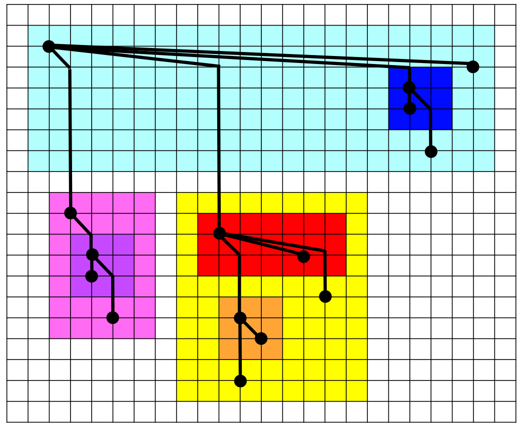


Figure 7: The drawing of the clustered tree C of Fig. 4 constructed by the algorithm described in Section 3

Theorem 1 *For every n-node c-connected clustered tree $C = (G, T)$ a $\Theta(n^2)$ area strictly upward order-preserving poly-line R-drawing can be constructed in $O(n^2)$ time.*

Proof: The drawing Γ obtained by applying the previous described algorithm is strictly upward, order-preserving, and poly-line by construction. All the vertices, bends, and corners of the rectangles representing clusters have integer coordinates. Further, each cluster μ contains all the vertices and clusters belonging to $T(\mu)$. This is because the coordinates of a cluster μ are obtained from the coordinates of the dummy vertices s_μ and c^3_μ: Since such vertices are respectively the first and the last vertex considered in the coordinates' assignment for the vertices in $V'(\mu)$, then they have respectively the smallest x- and y-coordinate and the greatest x- and y-coordinate among all real and dummy vertices in $V'(\mu)$. Hence, the rectangle delimited by such coordinates contains all the vertices and clusters belonging to $T(\mu)$.

The planarity of the drawing of G is proved as follows. Consider any pair of edges $e_1 = (u_1, v_1)$ and $e_2 = (u_2, v_2)$ such that u_1 and u_2 are parents of v_1 and v_2, respectively. Assume, w.l.o.g., that if u_1 and u_2 are on the same path from the root to a leaf then u_1 is an ancestor of u_2. We distinguish three cases:

- If v_1 is an ancestor of u_2 or coincides with u_2, then u_1 is an ancestor of u_2 and edges e_1 and e_2 do not cross by the upwardness of Γ.

- If u_1 and u_2 coincide, consider the at most two segments s^a_1 and s^b_1 (resp. s^a_2 and s^b_2) representing edge e_1 (resp. edge e_2) in Γ, where s^a_1 connects u_1 with $(x(v_1), y(u_1) - 1)$ and s^b_1 connects $(x(v_1), y(u_1) - 1)$ with v_1 (resp. where s^a_2 connects $u_1 = u_2$ with $(x(v_2), y(u_1) - 1)$ and s^b_2 connects $(x(v_2), y(u_1) - 1)$ with v_2). Segment s^a_1 cannot cross s^b_2 and segment s^a_2 cannot cross s^b_1, since they lie in disjoint y-intervals; segment s^a_1 cannot cross s^a_2 since they are incident to the same vertex and have different slopes; segment s^b_1 cannot cross s^b_2 since they lie on different vertical lines.

- If u_1 is an ancestor of u_2 but v_1 is not, then consider the at most two segments s^a_1 and s^b_1 representing edge e_1 in Γ, where s^a_1 connects u_1 with $(x(v_1), y(u_1) - 1)$ and s^b_1 connects $(x(v_1), y(u_1) - 1)$ with v_1. Segment s^a_1 can not cross e_2 since they lie in disjoint y-intervals. Segment s^b_1 can not cross e_2 since they lie in disjoint x-intervals.

- If u_1 is not ancestor of u_2 and if they are not the same node, then they are not on the same path from the root to a leaf. Consider the lowest common ancestor lca of u_1 and u_2. Edges e_1 and e_2 belong to different subtrees among the subtrees rooted at the children of lca and hence they lie in in disjoint x-intervals.

Region-region crossings do not occur: Namely, consider two distinct clusters μ and ν that are not ancestor one of the other in T. If the roots $r(\mu)$ and $r(\nu)$ of $G(\mu)$ and $G(\nu)$, respectively, are such that $r(\mu)$ is an ancestor of $r(\nu)$ in G or viceversa, then the regions representing μ and ν in Γ lie in disjoint y-intervals. Otherwise ($r(\mu)$ and $r(\nu)$ are not on the same path from the root to a leaf in G) the regions representing μ and ν in Γ lie in disjoint x-intervals.

Concerning edge-region crossings, if the y-interval of an edge $e = (u, v)$ has intersection with the y-extension of a cluster μ three cases are possible:

- Edge e belongs to $G(\mu)$. In this case e is internal to the region $R(\mu)$ representing μ in Γ, since by construction the drawing of e in Γ is contained inside the smallest rectangle $R(e)$ with sides parallel to the axis that contains both u and v. Clearly $R(e)$ is internal to $R(\mu)$.

- Edge e does not belong to $G(\mu)$, but it is incident to a vertex of $G(\mu)$. Since $R(\mu)$ is a rectangle with sides parallel to the axes, since the second segment of e, if any, is vertical, then e crosses the border of μ exactly once.

- If edge e does not belong to $G(\mu)$ and is not incident to a vertex of $G(\mu)$, then there exists a node $\overline{r}$ in G such that e is contained in a subtree $G(v_i) \cup \overline{r}$ and the vertices of μ are contained in a subtree $G(v_j) \cup \overline{r}$, with v_i and v_j distinct children of $\overline{r}$. This implies that e and $R(\mu)$ lie in disjoint x-intervals.

Concerning the area requirement, observe that for every horizontal or vertical line intersecting Γ there is at least one node of G or one side of a rectangle representing a cluster. Since there are n vertices and $O(n)$ sides, both the height and the width of Γ are at most linear, so the area upper bound follows. By Lemma 1, quadratic area is necessary in the worst case. Concerning the running time, it's easy to see that the algorithm can be implemented to run in $O(n^2)$ time. $\square$

The above described algorithm can be slightly modified in order to produce R-drawings within different drawing conventions for the underlying tree.

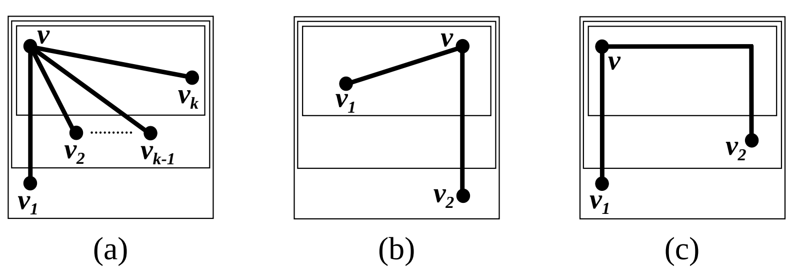


Figure 8: (a) Strictly upward non-order-preserving straight-line R-drawings of c-connected clustered trees. (b) Strictly upward order-preserving straight-line R-drawings of c-connected binary clustered trees. (c) Upward orthogonal order-preserving straight-line R-drawings of c-connected binary clustered trees.

Theorem 2 *For every n-node c-connected clustered tree $C = (G, T)$ a $\Theta(n^2)$ area strictly upward non-order-preserving straight-line R-drawing can be constructed in $O(n^2)$ time.*

Proof sketch: First, counter-clockwise order the children of each node v so that a child v_i of v coming before a child v_{i+1} of v is such that the smallest cluster containing both v_i and v is an ancestor or is the same cluster of the smallest cluster containing both v_{i+1} and v. Second, augment $C = (G, T)$ to a clustered tree $C' = (G', T')$ and assign x- and y-coordinates to the nodes of G' as in the previous algorithm. The x- and y-coordinates' assignment of the algorithm and the order of the children of each node ensure that $x(v_{i+1}) > x(v_i)$ and $y(v_{i+1}) \geq y(v_i)$, for each pair of consecutive nodes v_i and v_{i+1} of a node v (see Fig. 8.a). Hence, after replacing dummy vertices with rectangles representing clusters, the edges connecting each node to its children can be drawn as straight-line segments, without introducing crossings. □

Theorem 3 *For every n-node c-connected binary clustered tree $C = (G, T)$ a $\Theta(n^2)$ area strictly upward order-preserving straight-line R-drawing can be constructed in $O(n^2)$ time.*

Proof sketch: In this case the previous described algorithm loses the invariant that the root v of a subtree $G(v)$ is the leftmost node in $G(v)$. Hence, the x-coordinates assignment changes slightly, while the y-coordinates assignment remains the same. Compute the x-coordinate of each node as in the previous algorithm. For each node $v \in G$ let v_1 and v_2 be the children of v and let μ_1 (μ_2) be the smallest cluster containing both v and v_1 (containing both v and v_2). If μ_1 is an ancestor or is the same cluster of μ_2 nothing changes. Otherwise (if μ_2 is an ancestor of μ_1), decrease by the same value $\overline{x}(v_2)$ the x-coordinate of each node in $G(v_1)$, of each node in $G(v_2)$, and of each corner of each cluster containing a node in $G(v_1)$ or in $G(v_2)$, so that v_2 gets the same x-coordinate of v (see Fig. 8.b). It's easy to see that the edges of G can be drawn as straight-line segments without introducing crossings. □

Theorem 4 *For every n-node c-connected binary clustered tree $C = (G, T)$ a $\Theta(n^2)$ area upward orthogonal order-preserving R-drawing can be constructed in $O(n^2)$ time.*

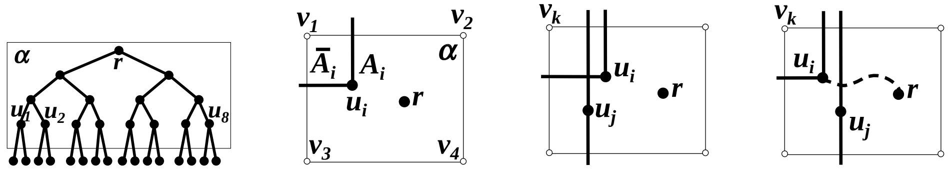
Proof sketch: What changes in this case with respect to the original formulation of the algorithm is just the position of the bend on every edge between a node v and its second child v_2, in the counter-clockwise order of the children of v. Namely, such an edge must be orthogonal, hence it goes horizontally till reaching $x(v_2)$, and then goes down to v_2 vertically (see Fig. 8.c). □

Contrasting with the above positive results, we prove the following theorem, that also contrasts with the fact that each binary tree has an orthogonal straight-line drawing [4].

Theorem 5 *There exists a c-connected c-planar binary clustered tree that does not admit any orthogonal straight-line R-drawing.*

Proof sketch: Consider the clustered tree $C = (G, T)$ defined as follows: G is a complete rooted binary tree with 31 vertices; all the non-leaf vertices of G belong to the same cluster α that is the only non-root cluster, and all the leaves of G do not belong to α. It's easy to see that C is c-planar. Fig. 9.a shows a c-planar drawing of C.

Consider any orthogonal straight-line R-drawing Γ of C. Consider the rectangle A representing α in Γ. Let r be the root of G, let $u_1, \dots, u_8$ be the 8 vertices that are leaves in $G(\alpha)$, and let $v_1, \dots, v_4$ be the corners of A.



(a) (b) (c) (d)

Figure 9: (a) A c-planar drawing of C. (b) Regions A_i and $\overline{A}_i$. (c) The edges connecting u_i to its children and the edges connecting u_j to its children cross. (d) $\overline{A}_i$ is enclosed inside $\overline{A}_j$.

The two edges connecting a node u_i and its children divide A in two regions A_i and $\overline{A}_i$ the first containing r and the second not. Since Γ is a straight-line orthogonal drawing, then both A_i and $\overline{A}_i$ contain at least one corner v_k (see Fig. 9.b). If two regions $\overline{A}_i$ and $\overline{A}_j$, with $i \neq j$, contain the same corner v_k, then either the edges connecting u_i to its children and the edges connecting u_j to its children cross (see Fig. 9.c), or $\overline{A}_i$ ($\overline{A}_j$) is enclosed inside $\overline{A}_j$ (resp. $\overline{A}_i$), and so the path connecting r and u_i (resp. connecting r and u_j) crosses one of the edges connecting u_j and its children (resp. connecting u_i and its children) (see Fig. 9.d).

Since (i) each region $\overline{A}_i$ contains at least one corner v_k of A, (ii) any two regions $\overline{A}_i$ and $\overline{A}_j$, with $i \neq j$, cannot contain the same corner v_k, and (iii) there are four corners v_k and eight regions $\overline{A}_i$, then Γ cannot be an orthogonal straight-line R-drawing of C. $\square$

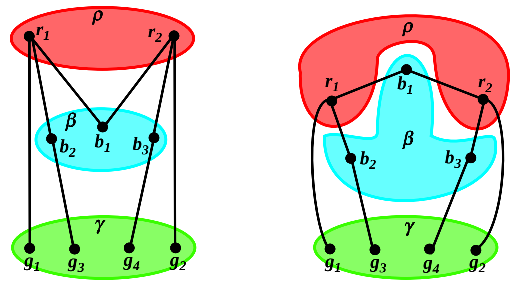
4 R-Drawings and C-Drawings of Non-C-Connected C-Trees

In this section we consider R-drawings and C-drawings of non-c-connected clustered trees. We have that most of the positive results presented for c-connected trees are not achievable for non-c-connected trees, that seem to have the same area requirement of general clustered graphs. We begin by showing that upward drawings of non-c-connected clustered trees are generally not feasible.

Theorem 6 *There exists a non-c-connected c-planar clustered tree that does not admit any upward C-drawing.*

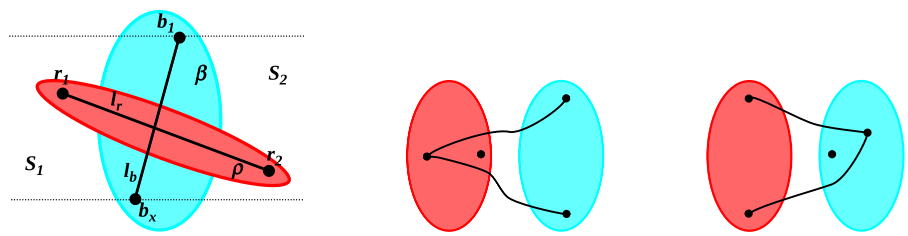
Proof: Consider the clustered tree $C = (G, T)$ defined as follows: G has root b_1, that has two children r_1 and r_2. Node r_1 (node r_2) has two children b_2 and g_1 (b_3 and g_2). Node b_2 (node b_3) has a child g_3 (resp. g_4). Vertices b_i, with $i \in \{1, 2, 3\}$, belong to cluster β, vertices r_i, with $i \in \{1, 2\}$ belong to cluster ρ, and vertices g_i, with $i \in \{1, 2, 3, 4\}$ belong to cluster γ. The inclusion tree T has root α that has three children β, ρ, and γ. It's easy to see that C is c-planar. Fig. 10 shows two c-planar drawings of C.

Suppose that an upward C-drawing Γ of C exists. Let b_x be the one between b_2 and b_3 that has minimum y-coordinate. Consider the horizontal strip S delimited by the horizontal lines through b_1 and through b_x. Let l_b be the segment connecting b_1 and b_x and let l_r be the segment connecting r_1 and r_2. Segment l_b divides S in two parts S_1 and S_2. The upwardness of Γ implies that $y(b_x) \leq y(r_1), y(r_2) \leq y(b_1)$. Hence, if r_1 and r_2 are not both in S_1 or both in S_2 segment l_b crosses segment l_r. However, the convexity of β and ρ implies that l_b and l_r belong entirely to the regions representing β and ρ in Γ, respectively. It follows that vertices r_1 and r_2 are both on the same of the parts S_1 and S_2 of S cut by l_b, otherwise β would cross ρ (see Fig. 11.a).



(a) (b)

Figure 10: (a) A non-upward c-planar drawing of C in which each cluster is represented by a convex region. (b) An upward c-planar drawing of C in which each cluster is represented by a non-convex region.

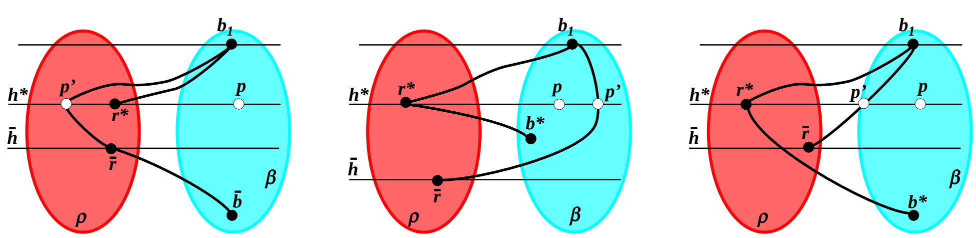


(a) (b) (c)

Figure 11: (a) Placing nodes r_1 and r_2 one in S_1 and one in S_2 leads a region-region crossing.

We claim that either there exists a node r_i, with $i \in \{1,2\}$, that is enclosed inside a region R delimited by cluster β and by edges (b_j, r_k) (see Fig. 11.b), with $j \in \{1,2,3\}$, $k \in \{1,2\}$, and $k \neq i$, or there exists a node b_i, with $i \in \{2,3\}$ that is enclosed inside a region R delimited by cluster ρ and by edges (b_j, r_k) (see Fig. 11.c), with $j \in \{1,2,3\}$, $k \in \{1,2\}$, and $j \neq i$.

Consider the horizontal line h^* through the one between r_1 and r_2 that has greater y-coordinate. Consider any intersection point p between h^* and β. Let r^* (resp. $\overline{r}$) be the one between r_1 and r_2 that has greater (resp. smaller) y-coordinate. If $y(r_1) = y(r_2)$, then let r^* (resp. $\overline{r}$) be the one between r_1 and r_2 that is closer (resp. farther) to p. Let b^* (resp. $\overline{b}$) be the only child of r^* (resp. $\overline{r}$) in β.



(a) (b) (c)

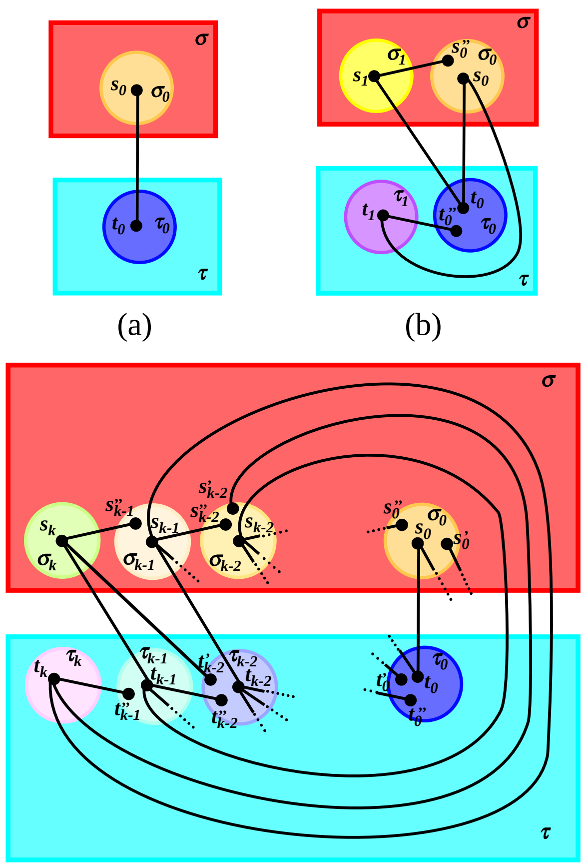
Figure 12: (a) Point p' lies outside β and r^* is closer than p' to p. (b) Point p' lies inside β or r^* is farther than p' to p, and edge (r^*, b^*) has no intersection with $\overline{h}$. (c) Point p' lies inside β or r^* is farther than p' to p, and edge (r^*, b^*) has intersection with $\overline{h}$.

Consider any intersection point p' between h^* and edge $(b_1, \overline{r})$.

- If p' is outside β and r^* is closer than p' to p, then r^* is closed inside the region R delimited by cluster β, by edge $(b_1, \overline{r})$, and by edge $(\overline{r}, \overline{b})$ (Fig. 12.a).

- If p' is inside β or if r^* is farther than p' to p, let $\overline{h}$ be the horizontal line through $\overline{r}$.
  - If edge (r^*, b^*) has no intersection with $\overline{h}$, then b^* is closed inside the region R delimited by cluster ρ, by edge (b_1, r^*), and by edge $(b_1, \overline{r})$ (see Fig. 12.b).
  - If edge (r^*, b^*) has intersection with $\overline{h}$, then $\overline{r}$ is closed inside the region R delimited by cluster β, by edge (b_1, r^*), and by edge (r^*, b^*) (Fig. 12.c).

Observe that every node b_i or r_j, with $i \in \{2,3\}$ and $j \in \{1,2\}$, has a child g_k, with $k \in \{1,2,3,4\}$, belonging to cluster γ. Hence, the child g_k of the node b_i or r_j that is closed inside region R must lie inside R, as well, since placing g_k outside R would imply an edge crossing or an edge-region crossing. Moreover, the child g'_k of the node b_i that has minimum y-coordinate among the vertices of cluster β lies outside R, with $k \in \{3,4\}$ and $i \in \{2,3\}$. It follows that γ crosses region R, implying an edge-region crossing or a region-region crossing. $\square$



(c)

Figure 13: Inductive construction of the clustered tree C_k. (a) C_0. (b) C_1. (c) C_k.

Now we show that straight-line drawings of non-c-connected clustered trees may require exponential area. Let $C_k = (G_k, T_k)$ be the family of non-c-connected c-planar clustered trees inductively defined as follows.

- Clustered tree C_0 (see Fig. 13.a): Tree G_0 has vertices s_0 and t_0 and edge (s_0, t_0). The inclusion tree T_0 has a root node with two children σ and τ. Node σ (node τ) has one child σ_0 (resp. τ_0), where $s_0 \in V(\sigma_0)$ (resp. $t_0 \in V(\tau_0)$).

- Clustered tree C_1 (see Fig. 13.b): Tree G_1 is obtained from G_0 by adding vertices s_1, t_1, s''_0, and t''_0 and edges (s_1, t_0), (s_1, s''_0), (t_1, s_0) and (t_1, t''_0). The inclusion tree T_1 is obtained from T_0 by adding σ_1 to the children of σ and τ_1 to the children of τ, where $s''_0 \in V(\sigma_0)$, $s_1 \in V(\sigma_1)$, $t''_0 \in V(\tau_0)$, and $t_1 \in V(\tau_1)$.
- Clustered tree C_k, with $k > 1$ (see Fig. 13.c): Tree G_k is obtained from G_{k-1} by adding vertices s_k, t_k, s''_{k-1}, t''_{k-1}, s'_{k-2}, and t'_{k-2}, and edges (s_k, t_{k-1}), (s_k, s''_{k-1}), (t_k, s_{k-1}), (t_k, t''_{k-1}), (s_k, t'_{k-2}), and (t_k, s'_{k-2}). The inclusion tree T_k is obtained from T_{k-1} by adding σ_k to the children of σ and τ_k to the children of τ, where $s'_{k-2} \in V(\sigma_{k-2})$, $s''_{k-1} \in V(\sigma_{k-1})$, $s_k \in V(\sigma_k)$, $t'_{k-2} \in V(\tau_{k-2})$, $t''_{k-1} \in V(\tau_{k-1})$, and $t_k \in V(\tau_k)$.

It is easy to see (Fig. 14.a) that C_k is c-planar. Also, $G(\sigma)$, $G(\tau)$, $G(\sigma_i)$, and $G(\tau_i)$, with $i = 0, \ldots, k-1$, are not connected. For simplifying the notation, in the following we assume k is odd.

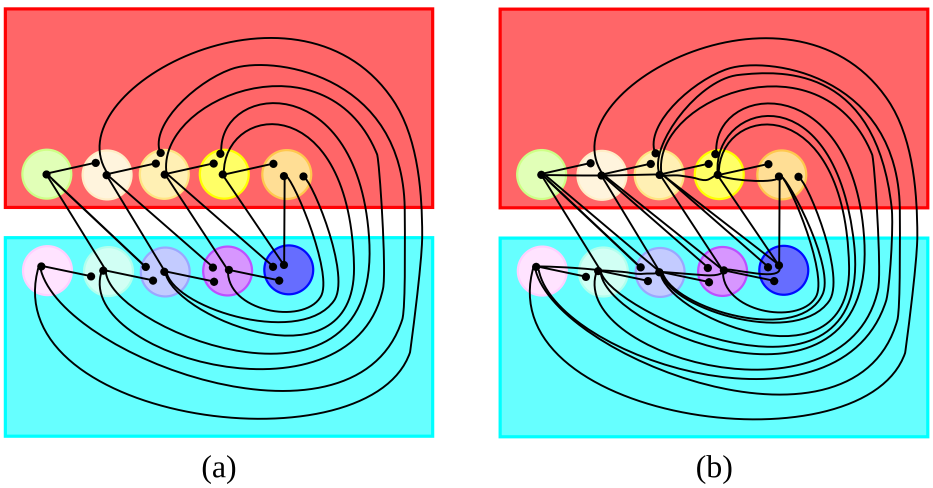


Figure 14: (a) A c-planar drawing of C_5. (b) A c-planar drawing of C_5 augmented as in Lemma 2.

We have the following lemma (see Fig. 14.b):

Lemma 2 *In any c-planar drawing of C_k polygonal lines $l(s_0, s_1)$ connecting s_0 to s_1, $l(t_0, t_1)$ connecting t_0 to t_1 and, for $i = 2, \ldots, k$, $l(s_{i-1}, s_i)$ connecting s_{i-1} to s_i, $l(t_{i-1}, t_i)$ connecting t_{i-1} to t_i, $l(t_{i-2}, s_i)$ connecting t_{i-2} to s_i, and $l(s_{i-2}, t_i)$ connecting s_{i-2} to t_i can be drawn such that they do not cross between themselves, do not cross any edge of G_k, and: (1) $l(s_0, s_1)$ crosses only the border of clusters σ_0 and σ_1; (2) $l(t_0, t_1)$ crosses only the border of clusters τ_0 and τ_1; (3) $l(s_{i-1}, s_i)$ crosses only the border of clusters σ_{i-1} and σ_i; (4) $l(t_{i-1}, t_i)$ crosses only the border of clusters τ_{i-1} and τ_i; (5) $l(t_{i-2}, s_i)$ crosses only the border of clusters τ_{i-2}, τ, σ_i, and σ; and (6) $l(s_{i-2}, t_i)$ crosses only the border of clusters σ_{i-2}, σ, τ_i, and τ.*

Proof: We only show how to draw line $l(s_{i-1}, s_i)$; the other lines are drawn analogously. Consider any c-planar drawing Γ_k of C_k. Polygonal line $l(s_{i-1}, s_i)$ is composed of two parts: the first part is a segment between s_i and a point p_i arbitrarily close to s''_{i-1}; such a segment can be drawn arbitrarily close to segment (s_i, s''_{i-1}), so that it does not intersect any edge of G_k. Moreover, since (s_i, s''_{i-1}) crosses only the borders of clusters σ_{i-1} and σ_i, then (s_i, p_i) crosses only the borders of clusters σ_{i-1} and σ_i, as well. The second part of $l(s_{i-1}, s_i)$ is a polygonal line

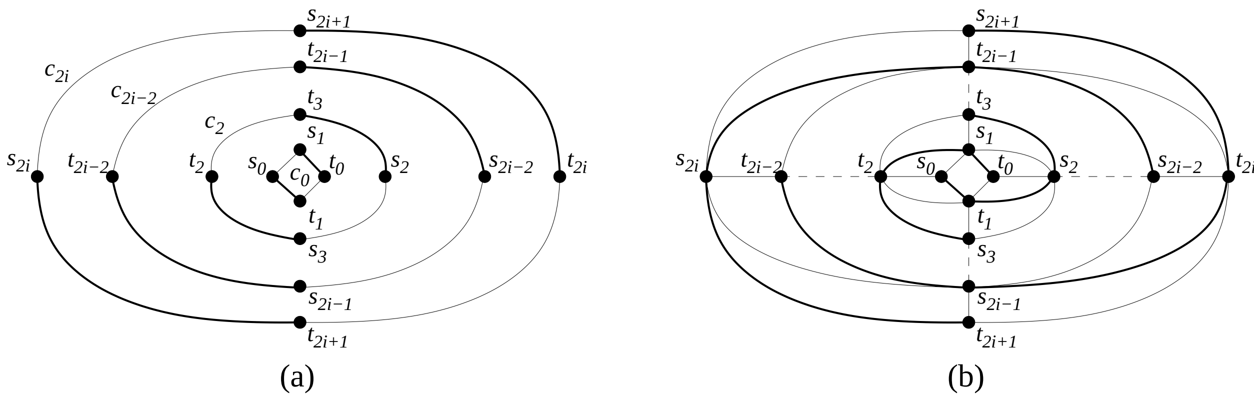


Figure 15: Graph G'_k. (a) Cycles c_{2i} and (b) their interconnections. Thick edges and thin edges distinguish among those edges that are common to G_k and those edges that are not common to G_k, respectively.

between p_i and s_{i-1}. Such points lie both inside the region representing σ_{i-1} in Γ_k. Since such a region contains only vertices s_{i-1}, s'_{i-1}, and s''_{i-1} that are not adjacent and contains entirely at most one polygonal line between a point arbitrarily close to s'_{i-1} and s_{i-1} (such a line is part of $l(s_{i-1}, t_{i+1})$), then we can connect p_i and s_{i-1} with a polygonal line without creating crossings. □

Consider any straight-line C-drawing of clustered tree C_k. Augment it by the polygonal lines of Lemma 2. Now remove the vertices of G_k with apex "prime" or "double prime" and their incident edges. The resulting clustered graph $C'_k = (G'_k, T'_k)$ is exactly the one defined in [9] to prove an exponential area lower bound for straight-line C-drawings of clustered graphs.

More formally, graph G'_k is defined as follows (see Fig. 15). For $i = 0, 1, \ldots, \frac{k-1}{2}$, let c_{2i} be the simple cycle composed of edges (s_{2i}, s_{2i+1}), (t_{2i}, t_{2i+1}), (s_{2i}, t_{2i+1}), and (s_{2i+1}, t_{2i}). For $i = 0, 1, \ldots, \frac{k-3}{2}$, cycle c_{2i} is connected to c_{2i+2} by edges (s_{2i}, t_{2i+2}), (t_{2i}, s_{2i+2}), (s_{2i+1}, t_{2i+2}), (t_{2i+1}, s_{2i+2}), (s_{2i+1}, s_{2i+2}), (t_{2i+1}, t_{2i+2}), (s_{2i+1}, t_{2i+3}), and (t_{2i+1}, s_{2i+3}). The graph resulting from the connection of all the c_{2i} is G'_k. The inclusion tree T'_k is the subtree of T_k restricted to the vertices of G'_k.

In order to study the c-planar drawings of C_k, we study the ones of C'_k. Notice that Lemma 2 does not directly extend the exponential area lower bound from the straight-line C-drawings of C'_k to the ones of C_k, since, even if by Lemma 2 we can obtain a C-drawing of C'_k by augmenting any straight-line C-drawing of C_k, the edges necessary for such an augmentation (i.e., the polygonal lines of Lemma 2) are not forced to be drawn as straight lines.

Observe that, by the c-planarity of C_k and by Lemma 2, clustered tree C'_k is c-planar. Also, it is easy to see that G'_k is triconnected.

Since G'_k is triconnected, all the plane embeddings of G'_k differ only for the external face. Consider any face f of G'_k as external. Three cases are possible: (i) f coincides with c_0; (ii) f coincides with c_{k-1}; (iii) otherwise, let c_{2h} and c_{2h+2} be the cycles that contain the vertices of f. Selecting f as external face induces a nesting of the cycles c_{2i} of G'_k. In case (i) c_{2i+2} is contained into c_{2i}, for $i = 0, 1, \ldots, \frac{k-1}{2}$. In case (ii) c_{2i} is contained into c_{2i+2}, for $i = 0, 1, \ldots, \frac{k-1}{2}$. In case (iii) c_{2i} is contained into c_{2i+2}, for $i = 0, 1, \ldots, h$, and c_{2i+2} is contained into c_{2i}, for $i = h+1, h+2, \ldots, \frac{k-1}{2}$. In all three cases there is a nesting composed of at least $\lceil (k-1)/4 \rceil$ cycles. The area lower bound for C'_k will be obtained by considering the area requirement of such a nesting.

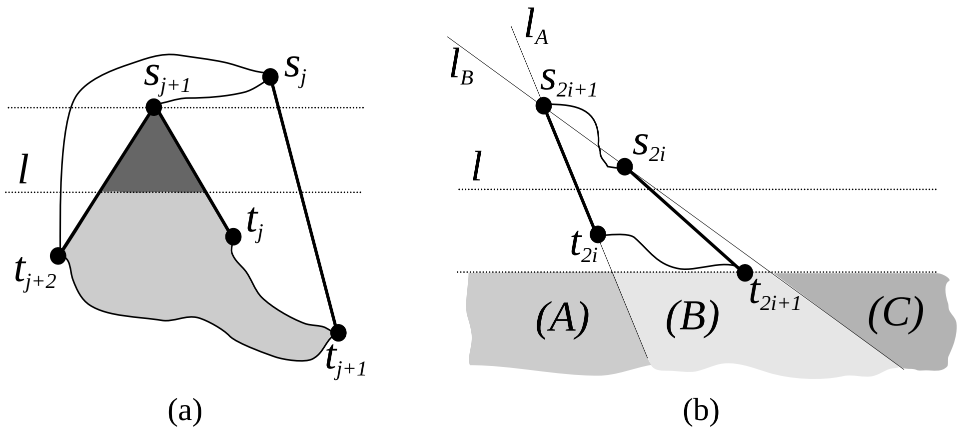


Figure 16: Illustrations for the proof of Lemma 3. (a) $y(s_{2i}) \geq y(s_{2i+1})$. The grey region (both the dark and the light grey) is R_j. The dark grey region is I_j. (b) Possible placements for vertex t_{2i+2}. Region (A), Region (B), and Region (C) have different shades of grey.

The following lemma is a generalization of Theorem 4 in [9]. In that paper the drawings of C'_k are studied where all the edges are straight-lines, while in the following lemma only the edges of G'_k that are also edges of G_k are required to be straight.

Lemma 3 *Any c-planar drawing of C'_k such that the edges of G_k are straight-line segments and the clusters are represented by convex polygons requires $\Omega(b^n)$ area, with $b > 1$.*

Proof: Consider any c-planar drawing Γ' of C'_k, in which the edges of G_k are straight-line segments and the clusters are represented by convex polygons. As already discussed, in Γ' there is a nesting of at least $\lceil (k-1)/4 \rceil$ cycles. Rename the vertices of G'_k (and of G_k) according to such a nesting. Namely, call c_0 the most nested cycle, and s_0, t_0, s_1, and t_1 its vertices. Call c_2 the cycle surrounding c_0, and s_2, t_2, s_3, and t_3 its vertices, etc. The outermost cycle is denoted by c_{2d}. Observe that $d = O(k) = O(n)$.

Let Γ'_{2i} denote the part of Γ' embedded inside c_{2i}, including such a cycle. Notice that, among the edges (s_{2i}, s_{2i+1}), (t_{2i}, t_{2i+1}), (s_{2i}, t_{2i+1}), and (t_{2i}, s_{2i+1}) composing cycle c_{2i}, only edges (s_{2i}, t_{2i+1}) and (t_{2i}, s_{2i+1}) are necessarily straight-lines in Γ'. Because of the convexity of the regions $R(\sigma)$ and $R(\tau)$ representing σ and τ, respectively, there exists a line l in Γ' separating $R(\sigma)$ and $R(\tau)$. Suppose w.l.o.g. that l is horizontal and that $R(\sigma)$ is above $R(\tau)$. Denote by H^+ and by H^- the half-planes above and below l, respectively. We argue that the area of Γ'_{2i+2} is at least twice the one of Γ'_{2i}, for $0 \leq i \leq d-1$. The thesis follows from this argument.

First, we show that $y(s_j) < y(s_{j+1})$, $0 \leq j \leq 2d-2$. Suppose, for a contradiction, that $y(s_j) \geq y(s_{j+1})$ (Fig. 16.a). We claim that s_j is outside the region R_j delimited by edges (s_{j+1}, t_j), (s_{j+1}, t_{j+2}), (t_{j+1}, t_j) and (t_{j+1}, t_{j+2}). Namely, s_j lies in H^+, hence if R_j contains s_j in its interior, then the intersection I_j between R_j and H^+ contains s_j in its interior, as well. However, since edges (s_{j+1}, t_j) and (s_{j+1}, t_{j+2}) are straight-lines and since edges (t_{j+1}, t_j) and (t_{j+1}, t_{j+2}) cannot intersect l by the supposed c-planarity of Γ', I_j is a triangle whose uppermost vertex is s_{j+1}. Since $y(s_j) \geq y(s_{j+1})$, I_j cannot contain s_j in its interior and R_j cannot contain s_j in its interior, as well. Since s_j is outside R_j, if j is even (if j is odd) cycle c_{j+2} cannot be external to cycle c_j (resp. cycle c_{j+1} cannot be external to cycle c_{j-1}), contradicting the assumption that cycle c_{2i+2} is drawn externally with respect to cycle c_{2i}. An analogous proof shows that $y(t_j) > y(t_{j+1})$, for $0 \leq j \leq 2d-2$.

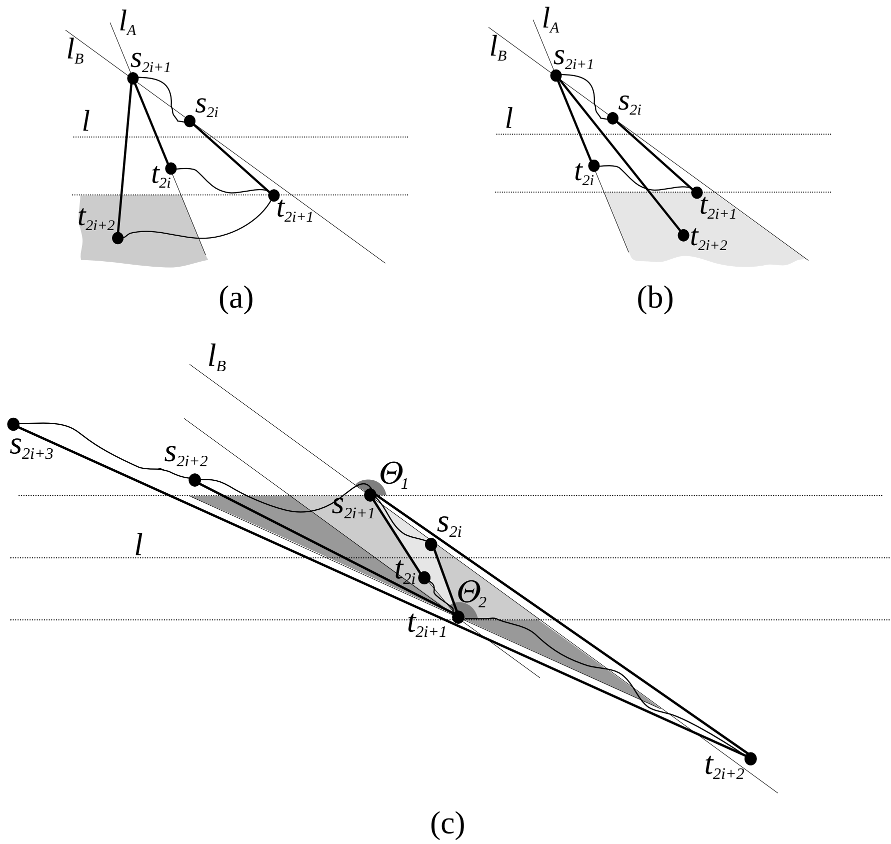


Figure 17: (a) If vertex t_{2i+2} is placed inside Region (A), then the embedding of G'_k changes. (b) If vertex t_{2i+2} is placed inside Region (B), then Γ' has a crossing. (c) Placement of vertex t_{2i+2} inside Region (C). Parallelogram P^* is composed of the regions colored by light shades of grey. Triangle T^* is composed of all regions colored by grey.

Consider the placement of vertex t_{2i+2} in Γ'. Let l_A be the line through t_{2i} and s_{2i+1} and let l_B be the line through s_{2i} and s_{2i+1}. By the above discussion, $y(t_{2i+2}) < y(t_{2i+1})$ holds. Hence, vertex t_{2i+2} can only lie in one of the following regions (see Fig. 16.b). Region (A), that is the intersection region between the half-plane delimited by l_A and not including s_{2i}, and the half-plane $y < y(t_{2i+1})$. Region (B), that is the intersection region between the half-plane delimited by l_A and including s_{2i}, the half-plane delimited by l_B and including t_{2i}, and the half-plane $y < y(t_{2i+1})$. Region (C), that is the intersection region between the half-plane delimited by l_B and not including t_{2i}, and the half-plane $y < y(t_{2i+1})$.

If vertex t_{2i+2} is placed inside Region (A), then vertex t_{2i} is enclosed inside the cycle $\mathcal{C}$ composed of edges (s_{2i+1}, s_{2i}), (s_{2i}, t_{2i+1}), (t_{2i+1}, t_{2i+2}), and (t_{2i+2}, s_{2i+1}) (see Fig. 17.a). Since vertex s_{2i+2} has to be connected to vertex t_{2i}, then, by the supposed planarity of Γ', s_{2i+2} is enclosed inside $\mathcal{C}$, as well. However, this contradicts the assumption that cycle c_{2i+2} is external with respect to cycle c_{2i}.

If vertex t_{2i+2} is placed inside Region (B) then Γ' has a crossing. Namely, edge (s_{2i+1}, t_{2i+2}) is a straight-line segment in Γ'. Hence, if vertex t_{2i+2} is placed inside Region (B) edge (s_{2i+1}, t_{2i+2}) crosses either edge (t_{2i+1}, t_{2i}) or edge (s_{2i}, t_{2i+1}). In fact such edges separate vertex s_{2i+1} from Region (B) (see Fig. 17.b).

Hence, we have that Region (C) is the only possible placement of t_{2i+2}. This geometric constraint on the placement of t_{2i+2} is exactly the same that was exploited in [5] to prove an exponential area lower bound for straight-line upward drawings of planar directed graphs.

Let Θ_1 be the angle formed by line l_B and by the x-axis and let Θ_2 be the angle formed by the line through t_{2i} and t_{2i+1} and by the x-axis. In [5] it is shown that (suppose that $\Theta_1 \geq \Theta_2$ and see Fig. 17.c): (i) the parallelogram P^* delimited by the horizontal lines through s_{2i+1} and t_{2i+1}, by l_B and by the line through t_{2i+1} parallel to l_B has area at least twice the area of the cycle composed of edges (t_{2i}, s_{2i+1}), (s_{2i+1}, t_{2i+1}), (t_{2i+1}, s_{2i}), and (s_{2i}, t_{2i}); (ii) the triangle T^* delimited by the horizontal line through s_{2i+1}, by l_B and by the line through t_{2i+1} parallel to edge (t_{2i+2}, s_{2i+3}) contains P; (iii) the drawing of the cycle composed of edges (t_{2i+2}, s_{2i+3}), (s_{2i+3}, s_{2i+2}), (s_{2i+2}, t_{2i+3}), and (t_{2i+3}, t_{2i+2}) contains T^*. Properties symmetric to properties (i), (ii), and (iii) hold if $\Theta_1 < \Theta_2$.

Such arguments straightforwardly apply here (by suitably replacing the area of the drawing with the convex-hull area of the vertices) and this concludes the proof. □

Lemma 4 *If there exists a straight-line C-drawing of C_k with area A, then there exists a c-planar drawing of C'_k such that the edges of G_k are straight-line segments, the clusters are represented by convex polygons, and the area is less or equal than A.*

Proof: Consider any C-drawing of C_k with area A. It can be augmented without increasing the area by inserting the polygonal lines of Lemma 2, still remaining c-planar. At this point the vertices that do not belong to G'_k and their incident edges can be removed obtaining a c-planar drawing of C'_k with area less or equal than A. □

From the above lemmas we have:

Theorem 7 *There exists an n-vertex non-c-connected c-planar clustered tree requiring $\Omega(b^n)$ area in any straight-line C-drawing, with $b > 1$.*

The above lower bound is matched by an exponential upper bound. Namely, one can augment the non-c-connected c-planar clustered tree in a c-connected c-planar clustered graph, that admits an exponential area C-drawing, by the results in [6]. If we relax the straight-line constraints, then better results can be obtained:

Theorem 8 *There exists an algorithm that computes an order-preserving 2-bends poly-line R-drawing requiring $\Theta(n^2)$ area of every non-c-connected c-planar clustered tree.*

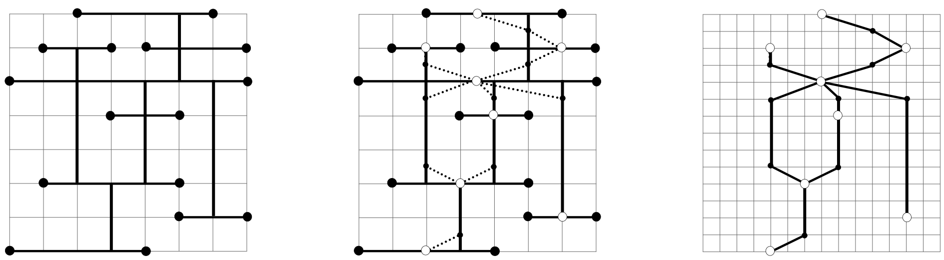
Proof: The proof is strongly based on the results of Eades et al. in [7]. Namely, in [7] it is shown an algorithm for computing an orthogonal c-planar drawing of a clustered graph $C = (G, T)$ such that G has maximum degree 4. As a first step, such an algorithm computes an $O(n^2)$ area c-planar drawing of C where the drawing of G is a *visibility representation* and where each cluster is drawn as a rectangle having sides parallel to the axes and having corners with integer coordinates. A visibility representation of a planar graph (see Fig 18) is such that each vertex is represented by an horizontal segment and each edge is represented by a vertical segment between two points of the segments representing its end-vertices. No two segments of the drawing cross or overlap. As noticed in [7], a visibility representation with the above described features can be constructed for a clustered graph whichever is the maximum degree of its underlying graph. We use such results as follows:

1. According to Theorem 2 of [10], augment the input clustered tree $C = (G, T)$ to a c-connected c-planar clustered graph $C' = (G', T)$ with the same number of vertices and with some extra dummy edges.

2. Compute a visibility representation Γ' of C' by using the algorithm in [7].

3. Turn Γ' in a poly-line drawing of C': For each vertex v represented in Γ' by an horizontal segment $S(v)$ with endpoints $(x_1(v), y(v))$ and $(x_2(v), y(v))$, with $x_1(v) < x_2(v)$, remove $S(v)$ from Γ' and insert a point $p(v) = (x(v), y(v))$, with $x(v)$ integer such that $x_1(v) \leq x(v) \leq x_2(v)$; for each edge $e = (u, v)$ of G represented in Γ' by a vertical segment $S(e)$ lying on the line $x = x(e)$, assuming w.l.o.g. that u has been mapped to a point $p(u) = (x(u), y(u))$, that v has been mapped to a point $p(v) = (x(v), y(v))$, and that $y(u) < y(v)$, remove $S(e)$ from Γ' and insert a polygonal line composed of three segments, the first between $p(u)$ and point $(x(e), y(u) + 1/2)$, the second between point $(x(e), y(u) + 1/2)$ and point $(x(e), y(v) - 1/2)$ (this segment is not drawn if the vertical distance between u and v is one unit), and the third between point $(x(e), y(u) + 1/2)$ and $p(v)$.

4. Remove the dummy edges from Γ' and multiply by two the coordinates of each vertex, bend, and rectangle's vertex of the drawing, so that each coordinate is integer, obtaining an R-drawing Γ of the clustered tree C.



(a) (b) (c)

Figure 18: (a) A visibility representation $VR(G)$ of a planar graph G. (b) Inserting vertices (big white circles) and bends (small black circles) to turn the visibility representation $VR(G)$ in a poly-line drawing. (c) The poly-line drawing resulting from $VR(G)$, where the coordinates of each vertex and bend are doubled.

It is easy to observe that turning a visibility drawing in a poly-line drawing as described at Step 3 of the previous algorithm preserves the c-planarity of the drawing.

Clearly, the coordinates' doubling performed at Step 4 of the previous algorithm does not increase the asymptotic value of the area of Γ, that hence remains quadratic. Moreover, by Lemma 1, such area requirement is optimal. Notice that at most two bends per edge are introduced by our algorithm.

Concerning the running time of the describe algorithm, it has been observed in [7] that, supposing the c-connected clustered graph C' to be given, then a c-planar visibility representation Γ' of C' can be computed in linear time. Further, it's easy to see that turning the visibility representation in a poly-line drawing can be performed in linear time, as well. Hence the total running time is linear if C' is given. If not, then the running time can not be assumed neither polynomial, since the complexity of providing a c-connected c-planar clustered graph containing a non-c-connected clustered tree as subgraph is unknown, as far as we know. □

5 NC-Drawings of C-Trees

In this section we consider c-planar drawings of clustered trees, assuming that each cluster is drawn as a simple, eventually non-convex, lattice polygon. We show that polynomial area is sufficient for strictly upward order-preserving straight-line NC-drawings of c-connected clustered trees. Notice that in the same drawing convention whether R- and C-drawings require polynomial or exponential area is open.

We show an inductive algorithm to construct a strictly upward order-preserving straight-line NC-drawing of a c-connected clustered tree $C = (G, T)$. Let r be the root of G and let $G(r_1)$, $G(r_2)$, ..., $G(r_k)$ be the subtrees of G rooted at the children r_1, r_2, ..., r_k of r, respectively. Suppose that, for each $C_i = (G(r_i), T_i)$, where $1 \leq i \leq k$ and where T_i is the subtree of T induced by the clusters containing at least one vertex of $G(r_i)$, a strictly upward NC-drawing Γ_i can be constructed. Suppose also that each cluster μ of T_i is represented in Γ_i by a polygonal line composed by four parts (see Fig. 19): An horizontal segment $T(\mu)$ delimiting the top side of the cluster and lying on the line $y = y_T(\mu)$, two vertical segments $L(\mu)$ and $R(\mu)$ delimiting the left and right sides of the cluster and lying on the lines $x = x_L(\mu)$ and $x = x_R(\mu)$, respectively, and one polygonal line $B(\mu)$ monotonically increasing in the x-direction delimiting the bottom side of the cluster.

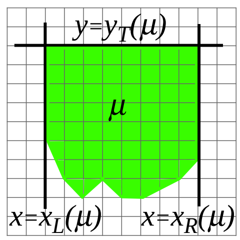


Figure 19: Shape of a cluster in the algorithm to construct strictly upward order-preserving straight-line NC-drawings of c-connected clustered trees.

The above inductive hypothesis is easily verified in the base case. Namely, if $G(r)$ has only one vertex v, draw it on a grid point. The clusters containing v are drawn as squares enclosing each other.

Now, suppose that $G(r)$ has more than one node. Inductively assume to have an NC-drawing Γ_i of each C_i. For each i such that $1 \leq i \leq k$, consider the set V_i of vertices of $G(r_i)$ and the set S_i of clusters belonging to T_i that do not contain r. Let $x_L(\Gamma_i) = \min_{v \in V_i, \mu \in S_i}\{x(v), x_L(\mu)\}$, $x_R(\Gamma_i) = \max_{v \in V_i, \mu \in S_i}\{x(v), x_R(\mu)\}$, and $y_T(\Gamma_i) = \max_{\mu \in S_i}\{y(r_i), y_T(\mu)\}$. For each i such that $1 \leq i \leq k$, remove the part of Γ_i that is inside one of the three half-planes $x < x_L(\Gamma_i)$, $x > x_R(\Gamma_i)$, and $y > y_T(\Gamma_i)$. This gives us partial drawings Γ'_i of all the C_i's, where the notations $x_L(\Gamma)$, $x_R(\Gamma)$, and $y_T(\Gamma)$ are extended to $x_L(\Gamma')$, $x_R(\Gamma')$, and $y_T(\Gamma')$, respectively, in the obvious way.

Place the Γ'_i's one beside the other, with $x_L(\Gamma'_{i+1}) = x_R(\Gamma'_i) + 1$, and so that all the r_i's lie on the same horizontal line h. Place r $2n^2$ units above and on the same vertical line of r_1. Draw straight-line edges between r and its children. Consider the clusters $\mu_1, \mu_2, \ldots, \mu_l$ containing r ordered so that μ_j is a sub-cluster of μ_{j+1}, for $1 \leq j < l$. In the following we show how to draw each cluster μ_j:

- Draw $T(\mu_j)$ as an horizontal segment between points $(x_L(\Gamma'_1) - j, y(r) + j)$ and $(x_R(\Gamma'_k) + j, y(r) + j)$.

- Draw $L(\mu_j)$ as a vertical segment between points $(x_L(\Gamma'_1) - j, y(r) + j)$ and $(x_L(\Gamma'_1) - j, y_T(\Gamma'_1) + l - j + 1)$.

- Draw $R(\mu_j)$ as a vertical segment between endpoints $(x_R(\Gamma'_k)+j, y(r)+j)$ and $(x_R(\Gamma'_k)+j, y_T(\Gamma'_k) + l - j + 1)$.

- We show how to draw $B(\mu_j)$. For each Γ'_i and each μ_j such that T_i does not contain μ_j, with $1 \leq i \leq k$ and $1 \leq j \leq l$, draw an horizontal segment between points $(x_L(\Gamma'_i), y_T(\Gamma'_i) + l - j + 1)$ and $(x_R(\Gamma'_i), y_T(\Gamma'_i) + l - j + 1)$. Notice that now for each Γ'_i and each μ_j the part $B(\Gamma'_i, \mu_j)$ of $B(\mu_j)$ between x-coordinates $x_L(\Gamma'_i)$ and $x_R(\Gamma'_i)$ has been drawn. For each pair $(\Gamma'_i, \Gamma'_{i+1})$ and each μ_j, with $1 \leq i < k$ and $1 \leq j \leq l$, connect $B(\Gamma'_i, \mu_j)$ and $B(\Gamma'_{i+1}, \mu_j)$ by a segment between the rightmost point of $B(\Gamma'_i, \mu_j)$ and the leftmost point of $B(\Gamma'_{i+1}, \mu_j)$. Polygonal line $B(\mu_j)$ is completed by a segment connecting $(x_L(\Gamma'_1) - j, y_T(\Gamma'_1) + l - j + 1)$ and the leftmost point of $B(\Gamma'_1, \mu_j)$ and a segment connecting $(x_R(\Gamma'_k) + j, y_T(\Gamma'_k) + l - j + 1)$ and the rightmost point of $B(\Gamma'_k, \mu_j)$.

An example of application of the algorithm is shown in Fig. 20. We obtain the following:

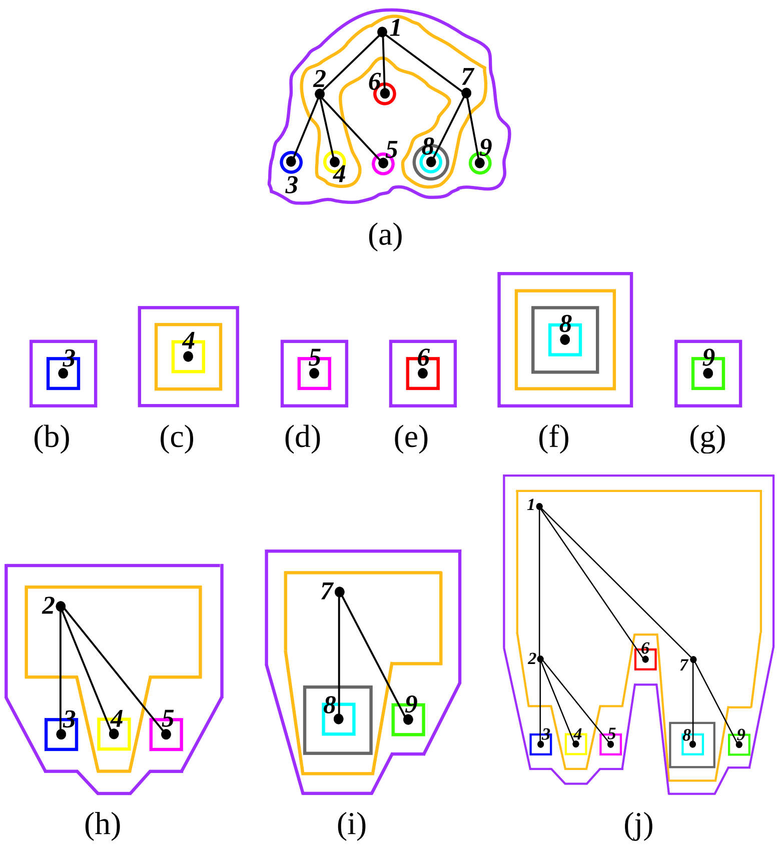


Figure 20: (a) An ordered c-connected clusterd tree $C = (G, T)$. (b)–(j) Application of the algorithm in Section 5 to obtain an NC-drawing of C.

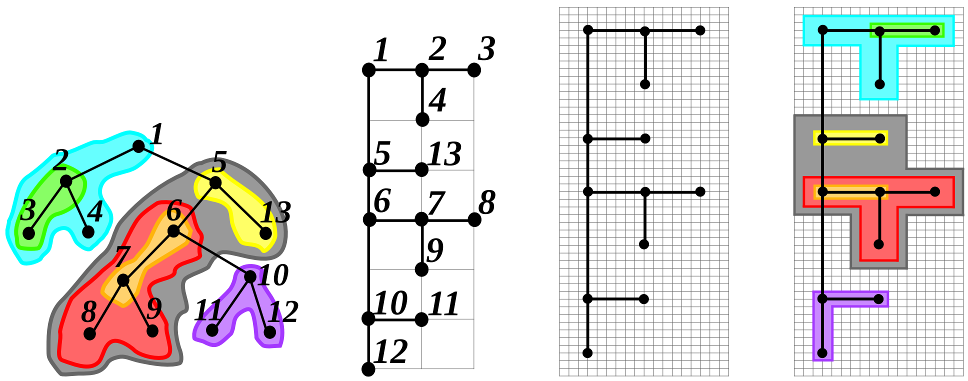
Theorem 9 *For every c-connected clustered tree there exists a strictly upward order-preserving straight-line NC-drawing in $O(n^4)$ area.*

Proof sketch: Let $C = (G, T)$ be a c-connected clustered tree. Apply the algorithm described in this secion with C as an input. It's easy to see that the obtained drawing Γ is strictly upward, order-preserving, planar and straight-line. An easy inductive argument can be used to prove the c-planarity of Γ. In particular, the absence of edge-region crossings is guranteed by the high value of the slopes of the edges of G.

Concerning the area bound, it's easy to see that the height of the drawing increases by $O(n^2)$ at each inductive step; since there are $O(n)$ steps the height of the drawing is $O(n^3)$. Concerning the width, the observation that for each vertical line there is either a vertex or one of the two lateral sides enclosing a cluster leads to a $O(n)$ width. □

We conclude the section with the following theorem.

Theorem 10 *For every c-connected binary clustered tree $C = (G, T)$ there exists a straight-line orthogonal upward NC-drawing with $O(n^3 \log n)$ area.*



(a) (b) (c) (d)

Figure 21: (a) A c-connected binary clustered tree $C = (G, T)$. Notice that $h(T) = 4$. (b) An hv-drawing Γ of G [3] with $O(n)$ height and $O(\log n)$ width. (c) Augmenting the grid of Γ. Notice that $2(h(T) - 1) = 6$. (d) The NC-drawing of C constructed on Γ.

Proof sketch: Let $C = (G, T)$ be a c-connected binary clustered tree (see Fig 21.a). Construct an hv-drawing Γ of G [3] with $O(n)$ height and $O(\log n)$ width, by the algorithm in [3] (see Fig 21.b). Augment the grid by inserting, for each column of Γ (for each row of Γ), $2(h(T) - 1)$ vertical grid lines (resp. $2(h(T) - 1)$ horizontal grid lines), where $h(T)$ is the number of edges in the longest path from the root to a leaf in T (see Fig 21.c). Such lines are used to draw each cluster μ as an orthogonal non-convex polygon $P(\mu)$. This can be easily done by proceeding bottom-up on the inclusion tree; each cluster surrounds the already drawn clusters and the part of the tree that it contains (see Fig 21.d). □

6 Conclusions

In this paper we dealt with the problem of obtaining minimum area c-planar drawings of clustered graphs whose underlying graph is supposed to be a tree. Tables 1 and 2 summarize area bounds proved for the different drawing standards considered. Tables 1 and 2 are also a reference point for classifying open problems. They correspond to question marks, to cells where upper and lower bounds do not match, and to cells where a drawing is in general not feasible. Such latter cells open the problem of recognizing the clustered trees that have one. We would like to explicitly mention three of such open problems that seem especially interesting.

1. Concerning c-connected clustered trees, we have shown that polynomial area is sufficient for obtaining c-planar drawings in most of the drawing's styles, sharply contrasting with the result presented in [6, 8], where it is shown that c-planar straight-line drawings of c-connected clustered graphs generally require exponential area. However, determining the area requirement of straight-line (strictly upward) order-preserving C-drawings of c-connected trees remains, in our opinion, an interesting open problem. In fact, an $O(n^2)$ area bound on such a problem would imply most of our positive results on c-connected trees.

2. Concerning non-c-connected clustered trees, we have shown that straight-line drawings could require as much area as that required by clustered graphs, hence nothing is earned by supposing the underlying graph to be a tree. Even if the mentioned lower bound is matched by an upper bound in the case of C-drawings, whether straight-line R-drawings of non-c-clustered trees always exist is still an open question. Such a problem is, as far as we know, open also for c-connected clustered graphs.

3. Concerning NC-drawings, we believe that polynomial-area bounds can be achieved in all drawing standards for the underlying tree, even for non-c-connected clustered trees. However, finding exact area bounds in each standard requires some more research efforts.

References

[1] T. M. Chan. A near-linear area bound for drawing binary trees. *Algorithmica*, 34(1):1–13, 2002.

[2] T. M. Chan, M. T. Goodrich, S. Rao Kosaraju, and R. Tamassia. Optimizing area and aspect ratio in straight-line orthogonal tree drawings. *Computational Geometry*, 23(2):153–162, 2002.

[3] P. Crescenzi, G. Di Battista, and A. Piperno. A note on optimal area algorithms for upward drawings of binary trees. *Computational Geometry*, 2:187–200, 1992.

[4] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing*. Prentice Hall, 1999.

[5] G. Di Battista, R. Tamassia, and I. G. Tollis. Area requirement and symmetry display of planar upward drawings. *Discrete & Computational Geometry*, 7:381–401, 1992.

[6] P. Eades, Q. Feng, X. Lin, and H. Nagamochi. Straight-line drawing algorithms for hierarchical graphs and clustered graphs. *Algorithmica*, 44(1):1–32, 2006.

[7] P. Eades, Q. Feng, and H. Nagamochi. Drawing clustered graphs on an orthogonal grid. *Journal of Graph Algorithms and Applications*, 3(4):3–29, 1999.

[8] Q. Feng. *Algorithms for Drawing Clustered Graphs*. PhD thesis, The University of Newcastle, Australia, 1997.

[9] Q. Feng, R. F. Cohen, and P. Eades. How to draw a planar clustered graph. In D. Du and M. Li, editors, *1st Annual International Computing and Combinatorics Conference (COCOON)*, pages 21–30, 1995.

[10] Q. Feng, R. F. Cohen, and P. Eades. Planarity for clustered graphs. In P. G. Spirakis, editor, *Third Annual European Symposium on Algorithms (ESA)*, pages 213–226, 1995.

[11] A. Garg, M. T. Goodrich, and R. Tamassia. Planar upward tree drawings with optimal area. *International Journal of Computational Geometry and Applications*, 6(3):333–356, 1996.

[12] A. Garg and A. Rusu. Straight-line drawings of binary trees with linear area and arbitrary aspect ratio. *Journal of Graph Algorithms and Applications*, 8(2):135–160, 2004.

[13] L. Trevisan. A note on minimum-area upward drawing of complete and Fibonacci trees. *Information Processing Letters*, 57(5):231–236, 1996.

[14] L. G. Valiant. Universality considerations in VLSI circuits. *IEEE Transactions on Computers*, 30(2):135–140, 1981.