*Proc. 2nd Saudi Sci. Conf.,Fac. Sci., KAU, 15-17 March 2004, **Part IV,*** pp. 19-29 (2005 A.D./1425 A.H.)

A New Algorithm for Compact Box-Drawing of Trees

AHMED A. A. RADWAN[1] and ESAM H. HOUSSIN[2]
Department of Computer Science, Faculty of Science, Minia University, Egypt
[1]E-mail: aaaradwaneg@yahoo.co.uk
[2]E-mail : esamhh@yahoo.com

ABSTRACT. Several researchers have discussed the Box-drawing of trees. In this paper the more recent algorithms are investigated and a new one have been introduced, in which each cell is drawn by a square box, no two boxes overlap each other, all boxes corresponding to siblings cells of the tree have the same x-coordinate at their left and right sides, and a parent cell is drawn nearly at the middle of its children. A Box-drawing of a tree is compact if it attains the minimum possible rectangle area enclosing the drawing. Our new algorithm is a linear time algorithm for finding a compact Box-drawing of a tree with area equal $(2l-1)(2h+1)$ where l is the number of leaves and h is the height of tree T.

AMS Subj. Classification: 68R10

Keywords: *Computational geometry, Graph drawing, Trees and Box drawing.*

1. Introduction

It is well known that human beings are very good in processing visual information, as is succinctly and effectively captured in the adage "a picture is worth a thousand words". Hence, a graph is an abstract structure that is used to model information. Graphs may be used to represent any information that can be modeled as objects and connections between those objects. Thus graph drawing addresses the problem of constructing geometric representations of conceptual structures that are modeled by graphs. For example graphs are used in computer networking to describe hierarchies and interconnections of components in computer network; each component is represented as a vertex in a graph and the connection between a component x and a component y is represented by an edge from x to y.

Other than computer networks and circuits schematics, automatic graph drawings have important applications that involve algorithm animation, software engineering, human interaction, visual display of information can be very useful, databases, VLSI circuit design, project planning, scientific data analysis, and graphics design. Further applications can be found in other science and engineering disciplines, such as medical science (concept lattices), chemistry (molecular drawings), civil engineering (floor plan maps) and cartography (map schematics). In fact, with the ever-growing capability of the visual display technology, the list of such applications is growing every day, and so is the

complexity of visual information that is needed to be displayed see, e.g., (Di Battista *et al.*, 1994; Tamassia, 1994; Trevisan, 1996; Garg & Tamassia, 1994; Tamassia & Tollis, 1995; Garg, 1996).

Graph layout problems in such a way as to satisfy given conditions have been frequently studied; among them is the VLSI layout. The "tidy drawing problem" which we treat here is a type of graph layout problem and is the problem of drawings tree-structures is a way, which satisfies given aesthetic conditions. If attributes are removed from the tree-structured diagram it becomes a tree (Miyadera *et al.*, 1998).

Tidy drawing problems of trees have been studied extensively since around 1970. First, aesthetic conditions for binary trees and several approximate layout algorithms in linear time, which correspond to the aesthetic conditions, were presented see e.g., (Wetherell & Shannon, 1979; Reingold & Tilford, 1981).

The rest of the paper is organized as follows. In section 2, some definitions and notations are used through this paper are given. In section 3, we outline the previous work on Box-drawing of trees. In section 4, the formulation of new aesthetic conditions for the layout is explained then we introduce the new algorithm. Finally in section 5, we give our conclusion.

2. Preliminaries and notations

In this section we give some definitions and present some notations also, in this paper we consider a tree in which "cells" with bounded sizes are located on integral lattice also, a tree T means the so-called ordered rooted tree in which there is an ordering of the children of v for each cell v in T.

Through this paper some terminology and notations are used; *A Grid drawing* is such that the cells and bends along the edges have integer coordinates, *i.e.,* grid points. *Planar drawings* where edges do not intersect are especially important because they improve the readability of the drawing (see, e.g., Trevisan, 1996; Reingold & Tilford, 1981).

A tree-structure is $T=(V, E, r, width, height)$, where (V, E) is an order tree (a tree denotes an order), V is a set of cells (nodes or vertices), $n=|V|$, and E is a set of edges, The root cell is $r \in V$. The map *height*: $V \rightarrow Z$ is *height* function of the cells. The horizontal length is represented by *height* (v), which is called the *height* of cell v. the map *width*: $V \rightarrow Z$ is the *width* function of the cell v. The vertical length is represented by *width* (v), which is called the *width* of cell v (in our algorithm we consider the vertical length of the cell equal the horizontal length of the cell equal one unit and this for all vertices of tree).

A placement of a tree-structure $T=(V, E, r, width, height)$ is defined by the following function $\pi : \pi \rightarrow Z^2$.

Symbols $\pi_x(v)$ and $\pi_y(v)$ denote the x-coordinate and y-coordinate of v, respectively, and $\pi(v) = (\pi_x(v), \pi_y(v))$.

Hereafter, (Miyadera *et al.* 1998) assumes that the x-coordinate directs from left to right and the y-coordinate directs downward and they assume that the top left point of a cell assigned by its coordinate.

A tree is a connected acyclic graph. *A leaf* of a free tree is a cell that has degree one

in the tree. The number of leaves in T is denoted by l, n is the number of cells, h is the height of tree T and n_i is the number of cells in level i.

The length of a path P is the number of edges of P. The *height* of the tree T, denoted by $h(T)$, is the length of the longest path from the root of T to a leaf. The number of edges between cell v and the root cell define the level of a cell v as follows:

$$Level(v) = \begin{cases} 0, \text{ if } v \text{ is the root cell,} \\ m, \text{ if there exists a path from root to cell } v \\ \quad \text{such as } root = v_0, v_1, ..., v_m = v (m \geq 1). \end{cases}$$

For convenience, if $Level(v_i) = i$ $(1 \leq i \leq m)$ then a cell v_{i-1} is called the *parent* cell of v_i, and v_i is called the *child* of v_{i-1}. For a cell v, the order of cell v is defined by the function *index* as follows:

$$Index(v) = \begin{cases} 0, \text{ if } v \text{ is root cell,} \\ i, \text{ if } v \text{ is the } i^{th} \text{ child of the parent of } v \end{cases}$$

3. The More Recent Algorithms for Compact Box-Drawings of Trees

In this section we introduce the previous work on Box-drawing of trees for the seek of giving the reader a short review about the latest results for solving this problem. It is known that the complexity of algorithms for finding a drawing of a tree significantly depends on the aesthetic conditions considered. Whereas for a different set of aesthetic conditions an $O(n\sqrt{n \log n})$ time algorithm is known for finding a drawing of a binary tree on an integer grid of minimum area (Eades *el al.*, 1992).

Reingold & Tilford, (1981) gave a linear-time algorithm for tidier drawing of trees with a certain set of aesthetic conditions where the cells are drawn as points and the drawing does not give optimum compactness (Reingold & Tilford, 1981; Wetherell & Shannon, 1979; Hasan *et al.*, 2002).

The size of a cell is described by the width and height of a box representing the cell (Hasan *et al.*, 2002), in which ordered trees brothers (cells) are numbered in downward order. Figure 1(a) shows a tree with prescribed width and height for each cell; the integer attached to each cell represents the cell number. Two Box-drawings of the tree in Fig. 1(a) are shown in Fig. 1(b) and Fig. 1(c), where the y-coordinate of the top side of a parent box is one less than that of its first child if the parent has two or more children, otherwise the top sides of a parent and the child have the same y-coordinate. Moreover, the ordering of the children of each cell in Fig. 1(a) is preserved in both of Fig. 1(b) and Fig. 1(c). Box-drawings of trees has practical applications in VLSI layout where a placement is to be found for modules with prescribed width and height see, e.g., (Lengauer, 1990; Sherwani, 1995).

Hasan *et al.,* (2002) showed that the Box-drawing is a *Compact Box-drawing* if the smallest rectangle enclosing the drawing has the minimum possible area. The Box-drawing in Fig. 1(c) is a Compact Box-drawing whereas the Box-drawing of Fig. 1(b) is not a Compact Box-drawing. In their paper they give a linear-time algorithm for finding a Compact Box-drawing of a tree. The algorithm in (Miyadera *et al.*, 1998) finds a Box-

drawing of a tree in time $O(n^2)$ if the tree has n cells, but the obtained drawing is not always a compact drawing.

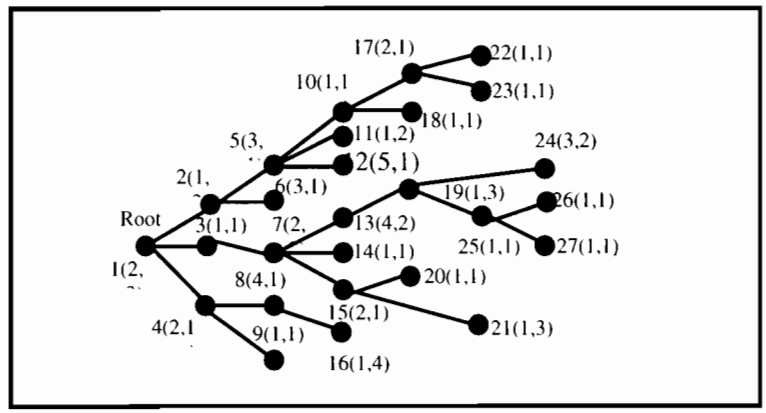

Fig. 1(a). A tree T with 27 nodes, 14 leaves and height 6.

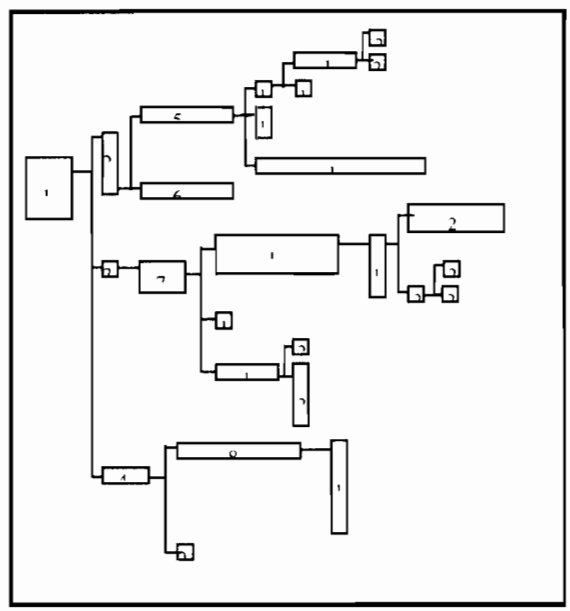

Fig. 1(b). An initial box drawing of T.

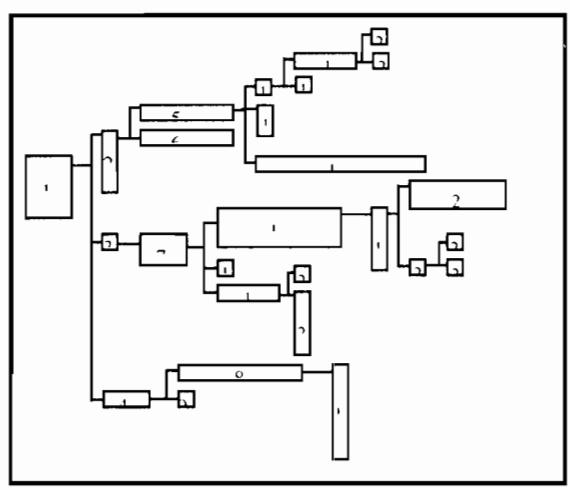

Fig. 1(c). A compact box drawing of T.

Walker (1990) gave a linear-time algorithm with a set of aesthetic conditions similar to that of Reingold and Tilford where the sizes of all the cells are variable in only one direction (*i.e.*, either in horizontal or in vertical direction) and fixed in the other direction.

Hasan *et al.* (2002) showed that the drawing of a tree a Box-drawing if each cell of the tree is drawn as an axis-parallel rectangular box and the drawing satisfies the following three aesthetic conditions:

(C1) No two boxes overlap each other;

(C2) All boxes corresponding to siblings have the same x-coordinate at their left sides, and the x- coordinate is equal to that of their parent plus the width of the parent; and

(C3) A parent cell is drawn at a given distance vertically apart from its first child: the y-coordinate of the topside of a parent is smaller than that of its first child by a given integer.

Similarly as in Miyadera *et al.* (1998), a box is not drawn in the exact scale of the prescribed size, but the width and the height of a drawn box are less than the prescribed size by a half of one coordinate unit. This amount of gap is used to show the interconnection among a parent and its children. A box is placed in the grid in such a manner that the top-left corner has an integral coordinate value but the bottom-right corner has a half- integral coordinate value. The interconnection among a parent and its children is represented by horizontal and vertical line segments as illustrated in Fig. 1(b) and Fig. 1(c). A horizontal line starts from the point, which is on the right side of a parent box and is one fourth of one coordinate unit lower from the top-right corner of the box, and ends on a vertical line at the middle of the gap.

Miyadera *et al.* (1998) gave an $O(n^2)$ time algorithm for finding a Box-drawing of a tree (Miyadera *et al.*, 1998). Their algorithm works in two phases. In the first phase it finds an initial Box-drawing which is not always compact, but satisfies an additional constraint that the drawing areas of any two subtrees whose roots are siblings do not overlap with each other. In the second phase it tries to compact the Box-drawing vertically by relaxing the additional constraint, but it does not always find a Compact Box-drawing. The first phase takes time $O(n)$, and the second phase takes time $O(n^2)$. The drawing in Fig. 1(b) is an initial Box-drawing found by the algorithm. Clearly it is not compact. On the other hand, the Box-drawing in Fig. 1(c) found by Hasan *et al.* (2002) algorithm is compact, and does not satisfies the additional constraint: for example, the drawing areas (*i.e.*, minimum rectangles) of the two subtrees rooted at cells 2 and 3 overlap with each other.

4. The New Algorithm

Before explaining the idea of the new algorithm we redefine some concepts and notations to fit its implementation.

Hereafter, we assume that the x-coordinate directs from left to right and the y-coordinate directs upward and they assume that the bottom left point of a cell is assigned by its coordinates.

We consider a box-drawing of a tree where each cell is drawn by a square box with one unit side, no two boxes overlap each other, all boxes corresponding to siblings of the tree have the same x-coordinate at their left and right sides, and a parent cell is drawn nearly at the middle of the vertical line between the (Lower y-coordinate) y_L-coordinate

of the first child and (Upper y-coordinate) y_U-coordinate of the last child and the leaves are drown in one level such that every two consecutive leaves are separated by one of coordinate unit on the y-coordinate. Also, we consider a type of tree called a "tree-structured" consisting of cells called "cells", which have bounded size, linked to each other in a tree-structure shape and placed on a grid

The tree is ordered and the sibling cells are numbered in upward order. We assume that the downmost leftmost point of the grid has coordinates (0,0), and that the first and the second coordinates increase rightwards and upwards, respectively. Finally, we want our drawing to take up as little space as possible. To measure space, we require the cells of each box corner to have integer coordinates (grid point). We define the *width* of the drawing as the maximum value of the first coordinate of the points of the drawings and the *height* of the drawing as the maximum value of the second coordinate of the points of the drawings. The *area* of a drawing is the product of its width times its height (Trevisan, 1996).

We denote by $S(v)$ to the box associated with vertex v. The geometry of $S(v)$ for Box-drawing is specified by its Leftmost x-coordinates (x_L) and Rightmost x-coordinates (x_R), and its Lower and Upper y-coordinates:

$$S(v) = ([x_L, x_R],[y_L, y_U]). \tag{1}$$

We denote the first child of the cell v_i by v_i' and last child of the cell v_i by v_i'' also, we refer to $y_L(v_i')$ as the lower y-coordinate of the first child of the cell v_i, $y_U(v_i'')$ as the upper y-coordinate of the last child of the cell v_i.

Hence $y_L(v_i) = \lfloor (\pi_{y_L}(v_i') + \pi_{y_U}(v_i''))/2 \rfloor$ and $y_U(v_i) = y_L(v_i) + 1$.

In Subsection 4.1, the aesthetic conditions for the tree layout using box-drawing are formulated and a new different version to the algorithm given in Hasan *et al.* (2002) to draw a tree with n cells into a grid of area $(2l-1)(2h+1)$ is introduced. Miyadera *et al.* (1998) and Hasan *et al.* (2002) have chosen variable size for the cells and we can not find in these papers any explanation to show the reader the advantage of this choice. Also, from our point of view we can not see any advantage for the free size. Using a fixed size (square) for the cells will be more aesthetic and will reduce the drawing area in the most of the cases. Also, we show that the drawing of a tree is a Box-drawing if each cell of the tree is drawn as an axis-parallel square box have the same size and the drawing satisfies aesthetic conditions given in the following subsection:

4.1 Aesthetic Conditions for Tree-Structure

Now we introduce constraints concerning a placement of T as follows.

$C0$: Let $p_1, p_2, ..., p_n$ $(n \geq 1)$ be child cells of p and $q_1, q_2, ..., q_m$ $(m \geq 1)$ be child cells of q;

If $\pi_{x_L}(p) = \pi_{x_L}(q)$ and $\pi_{y_D}(p) \prec \pi_{y_D}(q)$ then $\pi_{y_D}(q_1) \succ \pi_{y_D}(p_n) + width(p_n)$. $(C0-1)$

For any cell p and q, $area(p,\pi) \bigcap area(q,\pi) = \phi$ for all $p, q \in V\ (p \neq q)$, $(C0-2)$.

Where, $area\ (p,q) \equiv \{(x,y) \mid \pi_{x_L}(p) \leq x \leq \pi_{x_L}(p) + height\ (p), \pi_{y_D}(p) \leq y \leq \pi_{y_D}(p) + width\ (p)\}$.

$(C0-1)$: means that none of the edges cross any other edges.

$(C0-2)$: means that none of the cells overlap any other cells.

($C1$): Let $p_1, p_2, ..., p_l$ be the leaves of T then, put $\pi_{y_L}(p_1) = 0$, $\pi_{y_U}(p_1) = 1$, and then,

$$\left.\begin{aligned} \pi_{y_L}(p_i) &= \pi_{y_U}(p_{i-1}) + 1, \\ \pi_{y_U}(p_i) &= \pi_{y_L}(p_i) + 1 \end{aligned}\right\} i = 2,3,...,l.$$

Condition ($C1$); calculate the y_L-coordinate and y_U-coordinate of the leaves.

($C2$): Let $p_1, p_2, ..., p_l$ be the leaves of T then, $\pi_{x_L}(p_1) = \pi_{x_L}(p_2) = ... = \pi_{x_L}(p_l) = 2h$ and $\pi_{x_R}(p_1) = \pi_{x_R}(p_2) = ... = \pi_{x_R}(p_l) = 2h+1$.

Condition ($C2$); means that the leaf cells have the same x_L-coordinate and the same x_R-coordinate.

Hasan *et al.* (2002) have drawn the parent cell at a given distance vertically from its first child: the y-coordinate of the topside of a parent is smaller than that of its first child by a given integer. But in this drawing the y-coordinate of the parent is chosen to be in the middle of the difference between the y-coordinate of the first child and the y-coordinate of the last child nearly, that is:

($C3$): If a cell p has k children $p_1, p_2, ..., p_i (1 \le i \le k)$, then $\pi_{y_L}(p) = \lfloor (\pi_{y_L}(p_1) + \pi_{y_U}(p_k))/2 \rfloor$, and $\pi_{y_U}(p) = \pi_{y_L}(p) + 1$.

Condition ($C3$): means that, "a parent cell is placed nearly at the middle of the vertical distance between its first and last child".

($C4$): If a cell p has a child cells $p_1, p_2, ..., p_k$ then, the left x-coordinate will be the same for all siblings children and its equal $\pi_{x_L}(p_i) = \pi_{x_L}(p) + 2$ and the right x-coordinate will be the same for all siblings children and its equal $\pi_{x_R}(p_i) = \pi_{x_R}(p) + 2$, for $i = 1,2,..,k$. Except if some siblings children are leaves their x_L-coordinate and x_R-coordinate will be determined as in *Condition* ($C2$).

4.2 Drawing Algorithm

In this section, we explain the implementation of the new method for obtaining the placement under the aesthetic conditions introduced in Section 4.1.

The procedure of the placement is easily constructed as follows:

Step1: Drawing the Cells

1-1 The children of each cell of T are arbitrarily ordered and the leaves are labeled as $v_0, ..., v_{l-1}$ from down to up.

1-2 Draw leaves of the tree as a box ;

FOR $i = 0$ *TO* $l - 1$ *DO*

$S(v_i) = ([2h, 2h+1], [2i, 2i+1])$;

1-3 Draw internal cells of the tree as a box;

FOR $i = h - 1$ *TO* 1 *DO*

FOR $j = 0$ *TO* $n_i - 1$ *DO*

$S(v_j) = ([2i, 2i+1], [\pi_{y_L}(v_j), \pi_{y_U}(v_j)])$;

Step2: Drawing the Edges

2-1 After each parent draw a vertical line starts from y_L *(first child)+ half of one coordinate unit* to y_U *(last child)- half of one coordinate unit* and its x-coordinate is fixed and equal x_R *(parent)+half of one coordinate unit.*

2-2 A horizontal line starts from the point, which is on the middle of the right side of a parent box to the vertical line with *half of one coordinate unit* length.

2-3 With respect to connecting the children to their parents we use horizontal lines orthogonal to the vertical line between the parent and it's children, there are two main cases :

Case 2-3-1: If the parent has one child then, the child is connected to its parent by a horizontal line segment with *half of one coordinate unit* starts from the point which, is on the middle of the left side of the child cells to the vertical line between the parent and the child.

Case 2-3-2: If the parent has more than one child we have three cases:

Case 2-3-2-1: For the down most child the vertical line is connected by a horizontal line segment starts from the point which, is on the middle of the left side of the child to the vertical line between the parent and the child.

Case 2-3-2-2: For the up most child the vertical line is connected by a horizontal line segment starts from the point which, is on the middle of the left side of the child to the vertical line between the parent and the child.

Case 3-2-3: For other children (if any), the vertical line is connected by horizontal line starts from the point which, is on the middle of the left side of each child to the vertical line between the parent and the child.

Theorem1

Let T be a tree with n cells, l leaves and height h. The area required by a Box-drawing of T is $\Omega(l.h)$. Also, a Box-drawing of T with $O(l.h)$ area can be constructed in $O(n)$ time.

Proof

In this implementation we draw the leaves in one column such that every two consecutive leaves are separated by a single row, we start from the first row so the number of used rows is equals $2l-1$ which is the width of the drawing area i.e. $width = 2l-1$ also, we draw all the vertices in the same level in one column and every two consecutive levels are separated by a single column, this for all the levels; hence, the height of the drawing equals; $height = 2h+1$. The area of the Box-drawing is at least $(2l-1)(2h+1)$ *i.e.,* the lower bound of the area is $\Omega(l.h)$.

The running time of the algorithm is the sum of running times for each statement executed; in *step* 1-1 of the algorithm, ordering each vertex takes c_0 (constant time) hence, the arbitrarily ordering contributes $c_0 n$ to the total running time; labeling the leaves from down to up contributes $c_0 l$ to the total running time. Drawing a box $S(v_i)$ related to vertex v_i takes time c_1. In *step* 1-2, drawing the leaves contributes $c_1 l$ to the total running time. In *step* 1-3, drawing the internal vertices and the root of T contributes $c_1(n-l)$ to the total

running time. Also, in *step 2,* drawing the vertical lines takes time $c_2(n-l)$ and drawing the horizontal lines takes time $c_2(n-1)$ so, *step 2* contributes $c_2(2n-l-1)$ to the total running time. Finally, the total running time $T(n)$ of the algorithm can be obtained as follow:

$$T(n)=c_0n+c_0l+c_1l+c_1(n-l)+c_2(2n-l-1)=c_0n+c_0l+c_1n-c_1l+2c_2n-c_2l-c_2;$$

Since, $l<n$

$$T(n)\approx c_0n+c_0n+2c_2n-c_2n-c_2\approx(2c_0+c_2)n\approx Cn;$$

This leads to $T(n)=O(n)$.

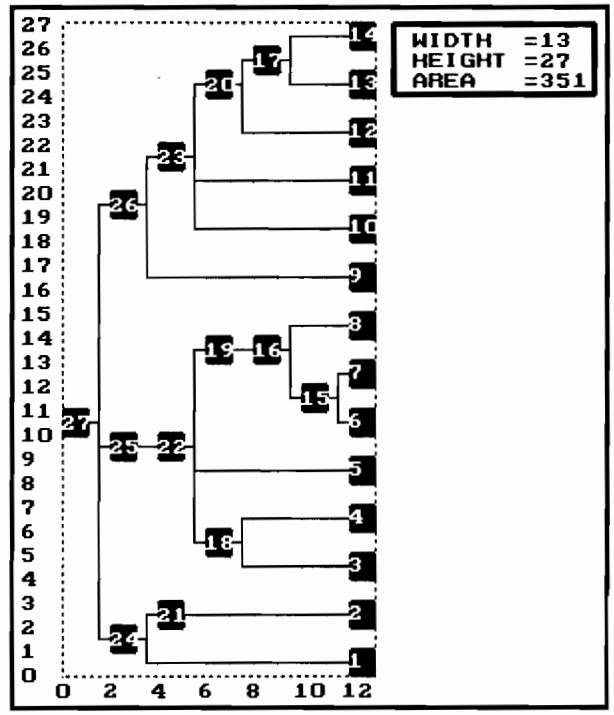

Fig. 2. The output of the input tree given in Fig. 1(a).

5. Conclusion

In this paper we give a survey about the best-known previous algorithms for Box-drawing of trees. Hence we introduce a modified combination of them using some of their aesthetic conditions (Miyadera *et al.,* 1998; Hasan *et al.,* 2002) and our aesthetic condition. The resultant algorithm was a new linear-time algorithm; hence we give a theorem, which proves that the area of the drawing has lower bound $\Omega(l.h)$.

The drawing satisfies the some important aesthetic criteria, which are symmetry, large angles between edges incident on the same cell, small total edge length, small area, uniform distribution of vertices, visibility representation and placing the most important cell in the center of the drawing, etc.

We have implemented a Pascal program using Borland Pascal 7.0, which can be used as procedure in graphics software package. The input for this program should be a rooted tree with n vertices, l leaves and height h and the output produces the Box-drawing of trees with *width* $2l-1$, *height* $2h+1$ and area $(2l-1)(2h+1)$.

References

Bloesch, A. (1993), "Aesthetic layout of generalized trees", *Software-Practice and Experience,* **23,** pp. 817-827.

Di Battista, G., Eades, P., Tamassia, R. and **Tollis, G.** (1994), "Algorithms for drawing graphs: an annotated bibliography", *Comput. Geom., Theory Appl.,* **4:** 235-282.

Eades, P., Lin, T. and **Lin, X.** (1992), "Minimum Size h-v Drawings", Advanced Visual Interfaces (Proceedings of AVI 92), *World Series in Comput. Sci.* **Vol. 36,** PP. 386-394.

Garg, A. (1996), "Where to draw the line", *Ph.D. Thesis,* Department of Computer Science, Brown University.

Garg, A. and **Tamassia, R.** (1994), "Advances in graph drawing", *In:* Algorithms and Complexity (Proc. CIAC' 94), volume 778 of Lecture Notes in Computer Science, pages 12-21. Springer-Verlag.

Hasan, M., Rahman, S. and **Nishizeki, T.** (2002), "A linear Algorithm for Compact Box-Drawings of Trees", *Proceedings of the 14th Canadian Conference on Computational Geometry, University of Lethbridge, Alberta, Canada, August12-14,* www.informatik.uni-trier.de/~ley/db/conf/cccg/cccg2002.html.

Lengauer, T. (1990), *"Combinatorial Algorithms for Integrated Circuit Layout",* John Wiley & Sons, Chichester and B.G. Teubner, Stuttgart.

Miyadera, Y., Anzai, K., Unno, H. and **Yaku, T.** (1998), "Depth first layout algorithm for trees", *Information Processing Letters,* **66,** pp. 187-194.

Reingold, M. and **Tilford, S.** (1981), "Tidier drawing of trees", *IEEE Transaction on Software Engineering,* pp. 223-228.

Sherwani, N. (1995), *"Algorithm for VLSI Physical Design Automation",* Kluwer Academic Publishers, Boston.

Supowit, J. and **Reingold, M.** (1983), "The complexity of drawing trees nicely", *Acta-Inform.,* **18,** pp. 377-392.

Tamassia, R. (1994), *"Advances in the Theory and Practice of Graph Drawing".* http://www.cs.brown.edu/people/rt/papers/ordal96/ordal96.html web version appears to be a conversion of: "Advances in Graph Drawing", Algorithms and Complexity (Proc. CIAC 94), Lecture Notes in Computer Science vol 778, Springer Verlag, pp. 12-21.

Trevisan (1996), "A note on minimum-area upward drawing of complete and fibonacci trees", *Inform. Process. Lett.,* **57:** 231-236.

Tamassia, R. and **Tollis, G.** (1995), Eds. "Graph Drawing (Proc. GD '94)", Vol. 894, *Lecture Notes in Computer Science,* Springer-Verlag.

Walker, Q. (1990), "A node-positioning algorithm for general trees", *Software-Practice and Experience,* **20,** pp. 685-705.

Wetherell, C. and **Shannon, A.** (1979), "Tidy Drawings of Trees", *IEEE Trans. Software Engrg.* **SE-5,** 514-520.

خوارزمية جديدة لرسم الأشجار في مساحة محكمة

أحمد على أحمد رضوان[١] ، و عصام حليم حسين[٢]

قسم علوم الحاسب ، كلية العلوم ، جامعة المنيا ، جمهورية مصر العربية

[١] بريد الإلكتروني : aaaradwaneg@yahoo.co.uk

[٢] esamhh@yahoo.com

المستخلص. العديد من المؤلفين ناقشوا تمثيل خلايا (عقد) الأشجار بمستطيلات ذات أبعاد مختلفة. وفى هذا البحث نسرد أحدث الخوارزميات التي تمثل الخلايا بمستطيلات، وأيضاً نقدم خوارزمية جديدة لتمثيل الخلايا بمستطيلات ذات أبعاد متساوية (مربعات) بشرط أن لا يوجد تداخل بين أي خليتين وكل الخلايا الموجودة في مستوى واحد يكون لها نفس الإحداثي x لكلا ضلعي المربع (الضلع الأيمن والأيسر)، والخلية الأب (parent) تمثل تقريباً في منتصف خلايا الأبناء (children).

الرسم بطريقة (المستطيلات والمربعات) يكون محكم إذا كانت مساحة الرسم أقل ما يمكن. والخوارزمية الجديدة خطية والرسم الناتج عنها يشغل مساحة $(2l-1)(2h+1)$ حيث إن l هي عدد الخلايا الطرفية و h هو ارتفاع الشجرة.