Information Processing Letters 79 (2001) 73–79

Information Processing Letters

www.elsevier.com/locate/ipl

The symmetry number problem for trees

Kien-Weh Chin, Hsu-Chun Yen *

Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan

Received 4 July 2000
Communicated by T. Lengauer

Abstract

For trees, we define the notion of the so-called *symmetry number* to measure the size of the maximum subtree that exhibits an *axial symmetry* in graph drawing. For unrooted unordered trees, we are able to demonstrate a polynomial time algorithm for computing the symmetry number. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Design of algorithms; Graph drawing

1. Introduction

Graphs are known to be useful for modeling various scientific/engineering problems in the real world. Because of the popularity of graphs, *graph drawing* has emerged as a research topic of great importance in graph theory. In many cases, a 'pretty drawing' often offers more insights into the nature of a graph. A natural question arises: How to define 'pretty drawings?' Aesthetic guidelines suggested in the literature (see, e.g., [1,8]) for drawing pretty graphs include minimizing the number of edge crossings, minimizing the variance of edge lengths, minimizing the number of bends, as well as drawing edges orthogonally or using straight-line segments. Such criteria are by no means comprehensive in spite of the fact that they are widely recognized as the most frequently used guidelines in graph drawing in general. From the viewpoint of computational complexity, many of the graph drawing related problems are intractable.

Recently, another aesthetic criterion, namely *symmetry*, has received increasing attention in the graph drawing community [2,4,6,7]. In particular, in [2] several types of symmetries (including *reflectional* and *rotational* symmetries) have been characterized in a unified way using geometric automorphism groups. As a symmetric graph can be 'decomposed' into a number of isomorphic subgraphs, only a portion of the graph, together with the symmetric information, is sufficient to define the original graph. In this way, symmetric graphs can often be represented in a more succinct fashion than their asymmetric counterparts. Moreover, to draw a graph nicely, a good starting point might be to draw its symmetric subgraph as large as possible first, and then add the remaining nodes and edges to the drawing. Unfortunately, like many of the graph drawing problems, deciding whether a graph has an axial (reflectional) or rotational symmetry is computationally intractable [7].

In this paper, we define a new quantitative measure of symmetry (called *symmetry number*) for trees. More precisely, the *symmetry number* of a tree is the size (number of nodes) of the maximum subtree which

* Corresponding author.
E-mail address: yen@cc.ee.ntu.edu.tw (H.-C. Yen).

0020-0190/01/$ – see front matter
PII: S0020-0190(00)00174-5

exhibits axial symmetry. (A tree is said to have an *axial symmetry* if we can draw (in the fashion of upward drawing) the tree together with an axis such that each node (and edge) has a unique corresponding image on the opposite side of axis.) The *symmetry number problem* is that of, given a tree G and a number k, deciding whether the symmetry number of G is greater than or equal to k. For unrooted unordered trees, we are able to come up with a polynomial time algorithm to solve the symmetry number problem.

2. Graphs, graph drawing and symmetry

An *unordered unrooted tree* is a connected, undirected, and acyclic graph without a specific root. A *subtree* of an unordered unrooted tree G is simply a connected subgraph of G. The main concern in this research is to decide, given an unordered unrooted tree G, G's maximum subtree that exhibits an *axial symmetric* in the fashion of straight-line upward drawing [1].

A *drawing* of a graph G on the plane is a mapping D from the nodes of G to $\mathbb{R}^2$, where $\mathbb{R}$ is the set of real numbers. That is, each node v is placed at point $D(v)$ on the plane, and each edge (u, v) is displayed as a line segment connecting $D(u)$ and $D(v)$. We require that the drawings of two distinct line segments do not intersect at more than one point. Figs. 1(a) and (b) display the symmetric drawings of two of the subtrees of an unordered unrooted tree. The top level of a drawing can be either a node or an edge as Fig. 1 indicates. In our subsequent discussion, *rooted trees* refer to trees whose top level (either a node or an edge) is fixed, and the top level is called the *root*. A rooted tree is said to have an *axial symmetry* if we can draw the tree (in the fashion of upward drawing) together with a straight line (called the *symmetry axis*) such that each node (and edge) is either on the symmetry axis or has a unique corresponding image on the opposite side of the axis. (That is, the drawing is symmetric with respect to the axis.) Unless stated otherwise, we simply use 'symmetry' to denote 'axial symmetry' and trees are assumed to be unordered throughout the rest of this paper. (The reader is referred to [2,7] for more about symmetry in graph drawing and other types of symmetries such as *rotational symmetry*.)

Given a rooted tree T and a vertex v, we write T_v to denote the rooted tree (with root v) whose vertices are all descendants of v, and C_v to represent the set of v's children ($C_v = \emptyset$ if v is a leaf node). T_v is called a *subtree* of T. An *r-subtree* of T is a rooted tree

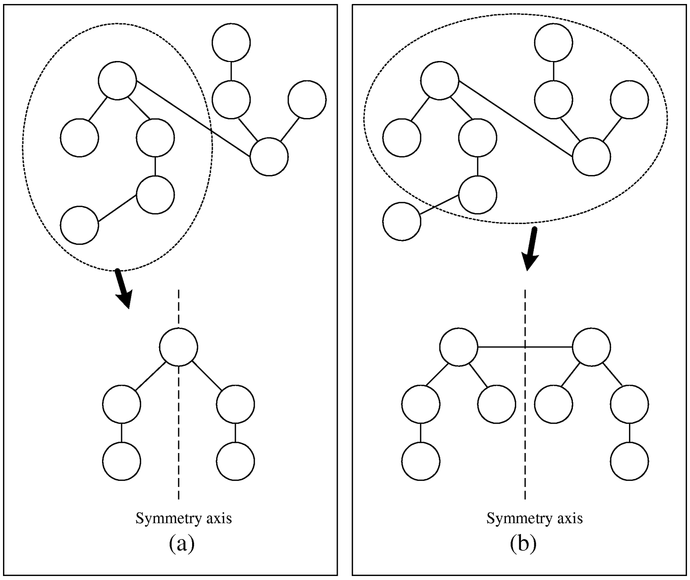


Fig. 1. Subtrees of an unrooted tree and their symmetric drawings.

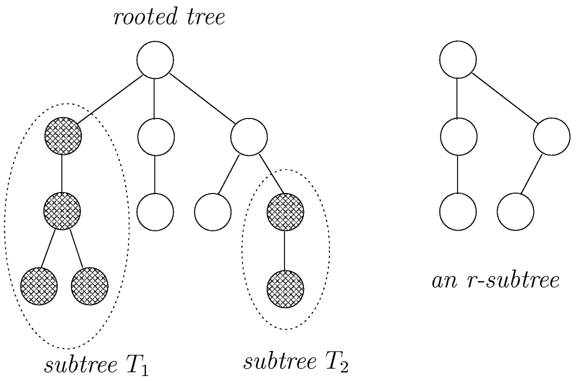


Fig. 2. Subtree and r-subtree of a rooted tree.

generated from T by cutting off some of T's subtrees. (See Fig. 2 for an example of an *r-subtree* by cutting off subtrees T_1 and T_2.) Notice that T and any of its nonempty *r-subtrees* share the same root.

The *symmetry number* of an unrooted tree G is defined to be the maximum number of *nodes* among G's subtrees that have an axial symmetry. (Recall that a subtree of an unrooted tree is simply a connected subgraph.) The *symmetry number problem* is the problem of, given an unordered unrooted tree G and an integer k, determining whether the symmetry number of G is greater than or equal to k.

Even though in this paper we mainly focus on trees, we feel that the notion of 'drawing the maximum symmetric subgraph' of a general graph is likely to play an interesting role in graph drawing. To draw a graph nicely, a good starting point might be to draw its symmetric subgraph as large as possible first, and then add the remaining nodes and edges to the drawing. Following a result in [7], for general graphs the symmetry number problem is NP-complete.

3. Deciding the symmetry number for trees

In this section, we design a polynomial time algorithm to calculate the symmetry number for unrooted unordered trees. Our algorithm utilizes the solution of the *weighted matching problem* which is defined as follows.

A *matching* M on a graph $G = (V, E)$ is a subset of E (i.e., $M \subseteq E$) such that any two edges in M have no common vertex. The *weighted matching problem*

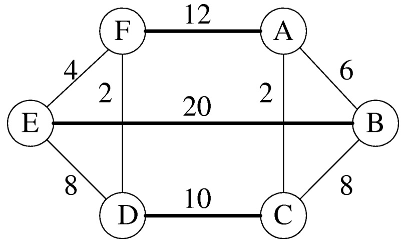


Fig. 3. An instance of the weighted matching problem.

is that of given a graph $G = (V, E)$ with a *weight function* $w : E \to N$, finding a matching M such that $\sum_{e\in M} w(e)$ is maximum. Take Fig. 3 for example. It is reasonably easy to see that edges (A, F), (B, E) and (C, D) constitute a maximum matching whose total weight is 42. It is known that the weighted matching problem for graphs is solvable in $O(n^3)$ time, where n is the number of nodes (see [5]). For n-node m-edge bipartite graphs with integral edge weights, the problem can be solved in $O(\sqrt{n} * m * \log(nW))$ time, where W is the maximum weight (see [3]).

We are now in a position to describe our algorithm for finding the symmetry number for unrooted trees. It should be noted that our algorithm is not responsible for doing the actual drawing of the maximum symmetric subtree, although displaying such a symmetric drawing is interesting and deserves further investigation.

Theorem 3.1. *The symmetry number problem for unrooted unordered trees is solvable in polynomial time.*

Proof. First consider an n-node rooted tree $T = (V, E)$, and without loss of generality, we let $V = \{v_1, v_2, \ldots, v_n\}$. We define two functions $A(v_i)$ and $B(v_i, v_j)$, where $v_i, v_j \in V$, as follows.

- Function $A(v_i)$ returns the number of nodes in the maximum symmetric r-subtree of T_{v_i} subject to the constraint that v_i being on the symmetry axis. (For the rooted tree displayed in Fig. 4(a), $A(3) = 5$. See Fig. 4(b).)
- Function $B(v_i, v_j)$ returns the number of nodes in the maximum r-subtree of T_{v_i} that is isomorphic to an r-subtree of T_{v_j}. In words, $B(v_i, v_j)$ is the size of the maximum common r-subtree of T_{v_i} and T_{v_j}. (Notice that $B(v_i, v_j) = B(v_i, v_j)$.) As Fig. 4(c) indicates, for the tree in Fig. 4(a), $B(3, 4) = 4$.

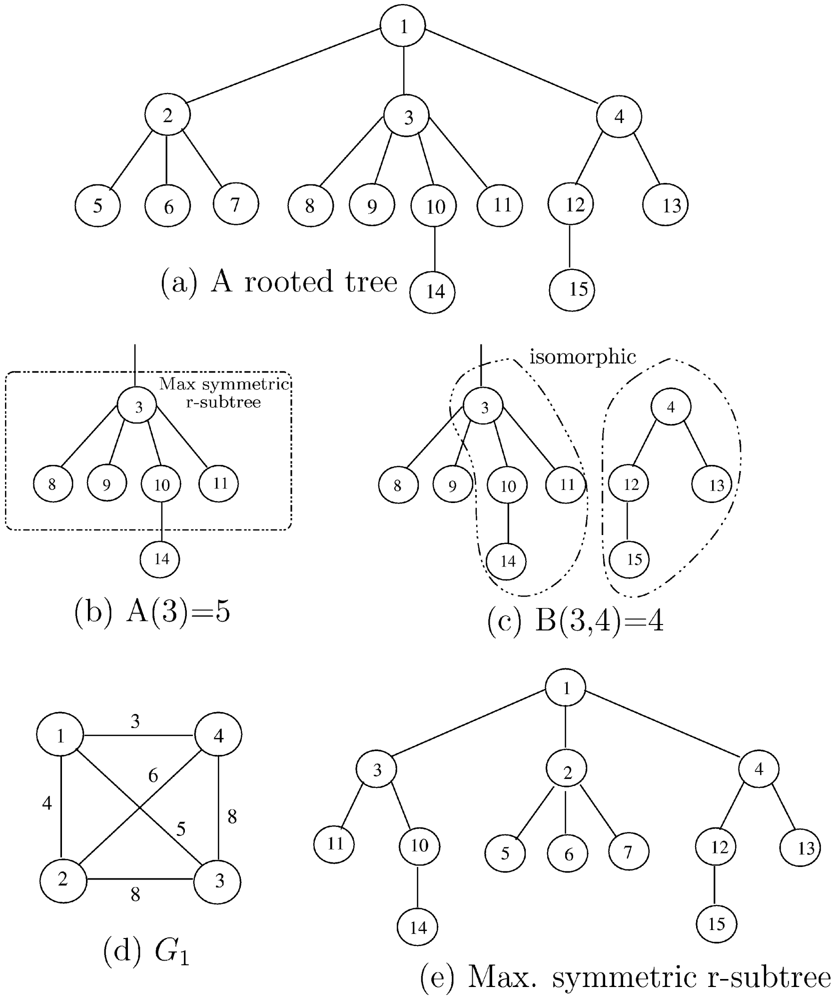


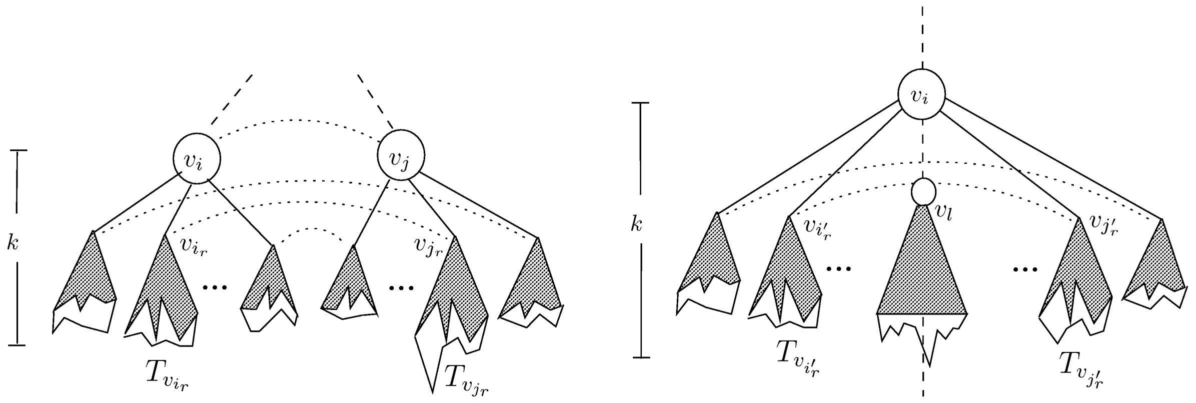
Fig. 4. The use of maximum weighted matching to find the maximum symmetric r-subtree.

(Intuitively, $B(v_i, v_j)$ is to capture the following idea: should v_i be mapped to v_j in a symmetric drawing, $B(v_i, v_j)$ is the maximum amount that each of v_i and v_j (with their r-subtrees) can contribute to the symmetric drawing.)

$A(v_i)$ and $B(v_i, v_j)$ are computed recursively by procedures $\mathcal{A}(v_i)$ and $\mathcal{B}(v_i, v_j)$ as follows.

Procedure $\mathcal{A}(v_i)$
  if v_i is a leaf node **then** $\mathcal{A}(v_i) = 1$
  else
    construct a weighted complete graph $G_{v_i} = (V', E')$
    with weight function w, such that
      $V' = \{v_i\} \cup C_{v_i}$ (C_{v_i} = the set of v_i's children)
      $w(v_i, v_l) = \mathcal{A}(v_l),\ \forall v_l \in C_{v_i}$
      $w(v_p, v_q) = 2\mathcal{B}(v_p, v_q),\ \forall v_p, v_q \in C_{v_i}$
    return $1 +$ (weight of maximum matching of G_{v_i})

Procedure $\mathcal{B}(v_i, v_j)$
  if v_i or v_j is a leaf node **then** $\mathcal{B}(v_i, v_j) = 1$
  else
    construct a weighted complete
    bipartite graph $G_{v_i, v_j} = (V_i \cup V_j, E')$
    with weight function w, such that
      $V_i = C_{v_i}$ and $V_j = C_{v_j}$
      $w(v_p, v_q) = \mathcal{B}(v_p, v_q),\ \forall v_p \in V_i, v_q \in V_j$
    return $1 +$ (weight of maximum matching of G_{v_i, v_j})



(a) (b)

Fig. 5. Symmetric drawing vs. maximum matching.

To give the reader a better feeling for the above argument, consider the tree T (with root 1) depicted in Fig. 4(a). Clearly, the size of the maximum symmetric r-subtree rooted at node 2 is 4; $A(2)$ is 4. $B(2,3)=4$ since the size of the maximum isomorphic r-subtrees rooted at nodes 2 and 3 is 4. Similarly, it is easy to see that $A(3)=5$, $A(4)=3$, $B(2,4)=3$, and $B(3,4)=4$. Fig. 4(d) displays the complete graph G_1 during the computation of $A(1)$ with respect to the tree in Fig. 4(a). The maximum weighted matching with respect to G_1 contains the edges $(1,2)$ and $(3,4)$ with total weight 12. Thus, the number of nodes in the maximum symmetric r-subtree of T is 13. A drawing of the corresponding maximum symmetric r-subtree of T is shown in Fig. 4(e). (The display is merely for the illustrating purpose; our algorithm does not produce such a drawing.) Notice that node 2 is on the symmetry axis, for edge $(1,2)$ is included in the maximum matching of G_1.

We now prove the correctness of procedures $\mathcal{A}(v_i)$ and $\mathcal{B}(v_i,v_j)$. First consider procedure $\mathcal{B}(v_i,v_j)$. The proof of the correctness of procedure $\mathcal{B}(v_i,v_j)$ (i.e., showing $\mathcal{B}(v_i,v_j)=B(v_i,v_j)$) is done by induction on the height k in the shorter of T_{v_i} and T_{v_j}. (Recall that the *height* of a rooted tree is the length of a longest path from the root to a leaf.) The case $k=0$ is trivial. Assuming that the assertion holds for $0\leqslant k\leqslant l$, we consider $k=l+1$. To prove $\mathcal{B}(v_i,v_j)=B(v_i,v_j)$, we proceed by showing both $\mathcal{B}(v_i,v_j)\leqslant B(v_i,v_j)$ and $B(v_i,v_j)\leqslant \mathcal{B}(v_i,v_j)$. Let f be a mapping (between r-subtrees of T_{v_i} and T_{v_j}) which witnesses $B(v_i,v_j)$. Notice that $f(v_i)=v_j$ and suppose $f(v_{i_r})=v_{j_r}$ ($\forall r,\ 1\leqslant r\leqslant m$, for some m), where $\{v_{i_1},\dots,v_{i_m}\}\subseteq C_{v_i}$ and $\{v_{j_1},\dots,v_{j_m}\}\subseteq C_{v_j}$. See Fig. 5(a). Then according to the induction hypothesis,

$\forall r,\ 1\leqslant r\leqslant m,$

$\mathcal{B}(v_{i_r},v_{j_r})=B(v_{i_r},v_{j_r}).$

Hence,

$$B(v_i,v_j)=1+\sum_{r=1}^{m}B(v_{i_r},v_{j_r})=1+\sum_{r=1}^{m}\mathcal{B}(v_{i_r},v_{j_r}),$$

which is less than or equal to one plus the maximum matching of G_{v_i,v_j} (because $\bigcup_{r=1}^{m}\{(v_{i_r},v_{j_r})\}$ forms a matching on G_{v_i,v_j}). Hence, $B(v_i,v_j)\leqslant\mathcal{B}(v_i,v_j)$. Conversely, suppose $\bigcup_{r=1}^{m'}\{(v_{i'_r},v_{j'_r})\}$ (for some m') is a matching on graph G_{v_i,v_j}. Since for every r, $v_{i'_r}$ (respectively, $v_{j'_r}$) is a child of v_i (respectively, v_j), by the induction hypothesis $\mathcal{B}(v_{i'_r},v_{j'_r})=B(v_{i'_r},v_{j'_r})$. A common r-subtree between T_{v_i} and T_{v_j} can be found by mapping v_i to v_j, and the r-subtrees of $T_{v_{i'_r}}$ and $T_{v_{j'_r}}$ witnessing $\mathcal{B}(v_{i'_r},v_{j'_r})$ to each other, $1\leqslant r\leqslant m'$. By doing so, we immediately have that $\mathcal{B}(v_i,v_j)\leqslant B(v_i,v_j)$. In view of the above, we conclude that $\mathcal{B}(v_i,v_j)=B(v_i,v_j)$, which completes the proof of the induction step for procedure $\mathcal{B}(v_i,v_j)$.

The proof of the correctness of procedure $\mathcal{A}(v_i)$ is carried out by induction on the height k of T_{v_i}, in conjunction with the correctness of procedure $\mathcal{B}$ proven above. Again the case $k = 0$ is trivial. Assuming that the assertion (i.e., procedure $\mathcal{A}(v_i)$ correctly returns $A(v_i)$) holds for $0 \leqslant k \leqslant l$, we consider the case when $k = l + 1$. In what follows, we show both $\mathcal{A}(v_i) \leqslant A(v_i)$ and $A(v_i) \leqslant \mathcal{A}(v_i)$.

Suppose M is a matching on G_{v_i}. Consider two cases:

(1) *v_i is involved in M*. That is,

$$M = \{(v_i, v_l)\} \cup \left(\bigcup_{r=1}^{m'} \{(v_{i'_r}, v_{j'_r})\}\right),$$

for some l and m'. By the induction hypothesis, a symmetric r-subtree of T_{v_l} with $\mathcal{A}(v_l)$ nodes can be found. By placing the isomorphic r-subtrees (of size $\mathcal{B}(v_{i'_r}, v_{j'_r})$, guaranteed by the correctness of procedure $\mathcal{B}$) of $T_{v_{i'_r}}$ and $T_{v_{j'_r}}$ $(1 \leqslant r \leqslant m')$ on the two sides of the symmetry axis, a symmetric drawing of $1 + \mathcal{A}(v_l) + \sum_{i=1}^{m'} 2\mathcal{B}(v_{i'_r}, v_{j'_r})$ nodes can be found. See Fig. 5(b).

(2) *v_i is not involved in M*. That is,

$$M = \bigcup_{r=1}^{m'} \{(v_{i'_r}, v_{j'_r})\},$$

for some m'. By placing the isomorphic r-subtrees (of size $\mathcal{B}(v_{i'_r}, v_{j'_r})$, guaranteed by the correctness of procedure $\mathcal{B}$) of $T_{v_{i'_r}}$ and $T_{v_{j'_r}}$ $(1 \leqslant r \leqslant m')$ on the two sides of the symmetry axis, a symmetric drawing of $1 + \sum_{r=1}^{m'} 2\mathcal{B}(v_{i'_r}, v_{j'_r})$ nodes can be found.

Either (1) or (2) above indicates that an r-subtree of $T(v_i)$ with at least $\mathcal{A}(v_i)$ nodes can be drawn symmetrically. Hence, $\mathcal{A}(v_i) \leqslant A(v_i)$.

Conversely, consider an r-subtree D of T_{v_i} that exhibits a symmetric drawing. Depending on whether a node in C_{v_i} lies on the symmetry axis or not, we have the following two cases:

(i) *A v_l is on the axis*. In this case, the size of T_{v_l}'s symmetric r-subtree in D is bounded by $A(v_l)$ $(= \mathcal{A}(v_l)$, by the induction hypothesis). (Recall that $A(v_l)$ defines the maximum size of symmetric r-subtrees of T_{v_l}.) This, in conjunction with the correctness of procedure $\mathcal{B}$, suggests that the size of D is bounded by

$$1 + \mathcal{A}(v_l) + \sum_{r=1}^{m} 2\mathcal{B}(v_{i_r}, v_{j_r}),$$

where $\{v_{i_1}, \dots, v_{i_m}, v_{j_1}, \dots, v_{j_m}\}$ is the set of v_i's children participated in D, and the corresponding image of v_{i_r} in the symmetric drawing is v_{j_r}.

(ii) *None of v_i's children is on the axis*. By the correctness of procedure $\mathcal{B}$, the size of the symmetric drawing D is bounded by

$$1 + \sum_{r=1}^{m} 2\mathcal{B}(v_{i_r}, v_{j_r}),$$

where $\{v_{i_1}, \dots, v_{i_m}, v_{j_1}, \dots, v_{j_m}\}$ is the set of v_i's children participated in D, and the corresponding image of v_{i_r} in the symmetric drawing is v_{j_r}.

By the definition of procedure $\mathcal{A}$, we immediately have that the size of D is bounded by the maximum matching on G_{v_i}. Hence, $A(v_i) \leqslant \mathcal{A}(v_i)$.

For rooted tree T of n nodes, let *time*$(\mathcal{A}(v_i))$ and *time*$(\mathcal{B}(v_i, v_j))$ be the times needed for procedures $\mathcal{A}(v_i)$ and $\mathcal{B}(v_i, v_j)$, respectively. Let $n_i = |C_{v_i}|$ and $n_j = |C_{v_j}|$, i.e., the numbers of children of v_i and v_j, respectively. It is easy to observe that for each pair of nodes v_i and v_j, $\mathcal{B}(v_i, v_j)$ is computed at most once, since there is exactly one path from the root to any node in T. In procedure $\mathcal{B}(v_i, v_j)$, the time needed to construct the bipartite graph G_{v_i,v_j} (which has $\mathrm{O}(n_i * n_j)$ edges) is bounded by $\mathrm{O}(n_i * n_j)$, given that $\forall v_p \in C_{v_i}, \forall v_q \in C_{v_j}, \mathcal{B}(v_p, v_q)$ are already computed. Recall that for n-node m-edge bipartite graphs with integral edge weights, the maximum matching problem can be solved in $\mathrm{O}(\sqrt{n} * m * \log(nW))$ time, where W is the maximum weight [3]. The complexity of computing $\mathcal{B}(v_i, v_j)$ is therefore bounded by

$$\begin{aligned} &\mathrm{O}\big(n_i n_j + \sqrt{n_i + n_j}\, n_i n_j \log((n_i + n_j) * n)\big) \\ &\quad = \mathrm{O}\big(n_i n_j (1 + \sqrt{n_i + n_j}\,) \log n\big) \\ &\quad \leqslant \mathrm{O}\big(n_i n_j \sqrt{n} \log n\big). \end{aligned}$$

Hence,

$$\begin{aligned} &\sum_{v_i, v_j \in V} \mathit{time}\big(\mathcal{B}(v_i, v_j)\big) \\ &\quad = \sum_{i=1}^{n} \sum_{j=1}^{n} \mathrm{O}\big(n_i n_j \sqrt{n} \log n\big) \\ &\quad \leqslant \mathrm{O}\left(\left(\sum_{i=1}^{n} n_i\right)\left(\sum_{j=1}^{n} n_j\right)\sqrt{n} \log n\right) \\ &\quad = \mathrm{O}\big(n * n * \sqrt{n} \log n\big) = \mathrm{O}\big(n^{2.5} \log n\big). \end{aligned}$$

Once all the $\mathcal{B}(v_i, v_j)$, $1 \leqslant i, j \leqslant n$, are calculated, computing $\mathcal{A}(v_i)$ requires first building a weighted graph G_{v_i} of $n_i + 1$ nodes and then solving the weighted maximum matching problem for G_{v_i}. The former is doable in $\mathrm{O}((n_i + 1)^2)$ time, and the latter can be done in $\mathrm{O}((n_i + 1)^3)$ [5]. Hence,

$$\begin{aligned}
&\sum_{v_i \in V} \bigl(\mathit{time}(\mathcal{A}(v_i))\bigr) \\
&\quad = \sum_{i=1}^{n} \bigl(\mathrm{O}((n_i + 1)^2) + \mathrm{O}((n_i + 1)^3)\bigr) \\
&\quad = \sum_{i=1}^{n} \mathrm{O}\bigl((n_i)^3\bigr) \leqslant \mathrm{O}\left(\left(\sum_{i=1}^{n} n_i\right)^3\right) \\
&\quad = \mathrm{O}\bigl(n^3\bigr)
\end{aligned}$$

time.

The above derivation is under the assumption that the root of a tree is given. Now for an unrooted tree $G(= (V, E))$, the symmetry number equals

$$\max\Bigl\{\max_{v_i \in V}\{\mathcal{A}(v_i)\}, \max_{(v_i, v_j) \in E}\{\mathcal{B}(v_i, v_j)\}\Bigr\},$$

whose computation time amounts to $\mathrm{O}(n^4) + \mathrm{O}(n^{3.5} \log n) = \mathrm{O}(n^4)$. (The first (respectively, second) term corresponds to the case when the top level of the maximum symmetric subtree is node v_i (respectively, edge (v_i, v_j)). Also notice that T_{v_i} and T_{v_j} depend on the root of tree T; hence, $\mathcal{A}(v_i)$ and $\mathcal{B}(v_i, v_j)$ have to be re-computed when the root of the tree changes.) This completes the proof of the theorem. $\square$

Acknowledgements

The authors thank the anonymous referees for comments which improved the presentation as well as the correctness of the paper.

References

[1] G. Battista, P. Eades, R. Tamassia, I. Tollis, Graph Drawing: Algorithms for the Visualization of Graphs, Prentice-Hall, Englewood Cliffs, NJ, 1999.

[2] P. Eades, X. Lin, Spring algorithms and symmetry, in: Computing and Combinatorics (COCOON'97), Lecture Notes in Comput. Sci., Vol. 1276, Springer, Berlin, 1997, pp. 202–211.

[3] H. Gabow, R. Tarjan, Faster scaling algorithms for network problems, SIAM J. Comput. 18 (5) (1989) 1013–1036.

[4] S. Hong, P. Eades, A. Quigley, S. Lee, Drawing algorithms for series-parallel digraphs in two and three dimensions, in: Graph Drawing'98, Lecture Notes in Comput. Sci., Vol. 1547, Springer, Berlin, 1998, pp. 198–209.

[5] E. Lawer, Combinatorial Optimization: Networks and Matroids, Holt, Rinehart & Winston, New York, 1976.

[6] J. Manning, M. Atallah, Fast detection and display of symmetry in trees, Congr. Numer. 64 (1988) 159–169.

[7] J. Manning, Geometric symmetry in graphs, Ph.D. Dissertation, Dept. of Computer Science, Purdue University, West Lafayette, IN, 1990.

[8] R. Tamassia, Graph drawing, in: J.E. Goodman, J. O'Rourke (Eds.), CRC Handbook of Discrete and Computational Geometry, CRC Press, Rockville, MD, 1997.