Fast detection and display of symmetry in outerplanar graphs

Joseph Manning* and Mikhail J. Atallah

Department of Computer Sciences, Purdue University, West Lafayette, IN 47907, USA

Received 27 January 1989
Revised 2 April 1990

Abstract


The automatic construction of good drawings of abstract graphs is a problem of practical importance. Symmetry appears as one of the main criteria for achieving goodness. Algorithms are developed for enumerating all planar axial and rotational symmetries of a biconnected outerplanar graph, and it is shown how to construct a drawing which simultaneously displays all these symmetries. These results are then expanded to obtain algorithms for the detection and display of outerplanar axial and rotational symmetry in the entire class of outerplanar graphs. All algorithms run in both time and space which are linear in the size of the graph, and hence are optimal to within a constant factor.

1. Drawing graphs

Mathematically, an abstract graph simply consists of two sets, \( V \) and \( E \subset V \times V \), and as such is completely specified by an enumeration of these sets or by adjacency lists or an adjacency matrix [1]. However, such representations convey little structural information about the graph, and so graphs are instead often presented by means of drawings. Every given abstract graph may be drawn in various different ways, all equally “correct”, but some certainly “better” than others in the sense that they clearly display important structural properties of the graph. For example, both drawings in Fig. 1 represent the same abstract graph (the Herschel graph), but key properties such as planarity, biconnectivity, symmetry, diameter, and even bipartition are readily apparent from the second drawing but not from the first.

As discussed in [12], there are many different and sometimes conflicting criteria which may be used to guide the construction of good drawings of a graph. Amongst

* Current address: Department of Computer Science, Vassar College, Poughkeepsie, NY 12601, USA.
these, the best overall general criteria appear to be the display of axial symmetry and, to a somewhat lesser extent, the display of rotational symmetry. It should be emphasized here that both axial and rotational symmetry are inherent properties of the abstract graph, independent of any particular drawing. In fact these properties can be formally defined in a purely algebraic manner in terms of automorphisms of the graph. However, for the present study it is more convenient to work with the equivalent geometric concept: an abstract graph is defined to have an axial (rotational) symmetry if there is some drawing of the graph which displays that axial (rotational) symmetry in the geometric sense.

Here, a (straight-edge) drawing of a graph consists of a set of distinct disks in the plane, one disk for each vertex, with straight-line segments between disks representing adjacent vertices. To avoid ambiguous representations, a line segment may not intersect a disk unless the corresponding edge and vertex are incident. Note that for simplicity, all drawings considered here use straight-line segments to draw edges; this restriction ensures that a drawing of a graph is fully specified merely by the positions of its vertices, and also does not conflict with the display of either symmetry or planarity [7].

Despite their significance in the construction of visually informative drawings, the basic problems of detecting any axial or rotational symmetry in general graphs appear to be computationally intractable, with the associated counting problems being correspondingly difficult, as shown by the following results:

**Theorem 1.1** [11]. The problems of determining if an arbitrary graph has any axial symmetry or any rotational symmetry are both NP-complete.

**Theorem 1.2** [11]. The problems of counting the number of axial symmetries or rotational symmetries in an arbitrary graph are both #P-complete.

Planar graphs constitute an important subclass of general graphs, and frequently admit efficient solutions to problems which are generally NP-complete. Moreover,
the graphs used in the proofs of the above theorems are nonplanar, so the present research has accordingly focussed on symmetry in planar graphs. Linear algorithms have already been obtained in [12] for detection, enumeration, and display of axial and rotational symmetry in trees; this paper now extends those results by establishing corresponding algorithms for outerplanar graphs.

A planar symmetry of a graph is a symmetry which can be displayed in a planar drawing of the graph. Clearly, only planar graphs can have planar symmetries, although they may also have nonplanar symmetries.

Since planarity is a desirable general property in drawings of outerplanar graphs, only planar drawings, and so only planar symmetries, are considered in this paper. In addition, only undirected simple graphs are considered; however, all the results have natural extensions to directed graphs and to graphs with loops and parallel edges. The adjacency list representation of graphs is assumed in all the algorithms. For the basic definitions and concepts of graph theory not explicitly presented here, consult any standard text such as [3, 4, 8].

2. Outerplanar graphs and biconnectivity

The existing definitions, theorems and algorithms concerning those aspects of outerplanarity and biconnectivity relevant to the problem of symmetry detection are collected together and stated below:

**Definition.** A graph is **outerplanar** if it has a planar embedding in which every vertex lies on the outer face.

**Algorithm 2.1** [13]. An $O(n)$-time algorithm to test if an $n$-vertex graph is outerplanar.

**Definition.** A **cutvertex** of a connected graph is any vertex whose removal would disconnect the graph. A graph is **biconnected** if it is connected and has no cutvertex. The **blocks** of a graph are its maximal biconnected subgraphs. (Cutvertices and blocks are also called **articulation points** and **biconnected components**, respectively.)

The blocks of a connected graph form a special tree-like structure:

**Definition.** The **block-cutvertex tree** of a connected graph is the tree which has a vertex for every block and every cutvertex of the graph, and an edge between the image of every cutvertex and the images of all blocks containing it.

This block-cutvertex tree can be constructed efficiently by using:

**Algorithm 2.2** [1]. An $O(m)$-time algorithm to decompose a connected graph with $in$ edges into its blocks.
An important link between outerplanarity and biconnectivity is established by the following theorem:

**Theorem 2.3** [8]. *A graph is outerplanar iff each of its blocks is outerplanar.*

The significance of biconnectivity to the development of symmetry algorithms for outerplanar graphs stems from:

**Theorem 2.4** [6]. *Every biconnected outerplanar graph with at least three vertices has a unique Hamilton cycle.*

Furthermore, this Hamilton cycle can be obtained efficiently from:

**Algorithm 2.5** [13]. An $O(n)$-time algorithm to find the Hamilton cycle of a biconnected outerplanar graph with $n \geq 3$ vertices.

### 3. Detecting symmetry in biconnected outerplanar graphs

Let $G$ be any biconnected outerplanar graph. The essence of the following symmetry detection algorithms is to encode $G$ into a string $S$, in such a way that planar symmetries of $G$ correspond to certain string symmetries of $S$. Efficient pattern-matching techniques are then used to enumerate these string symmetries, from which the original graph symmetries are easily recovered. (This method was applied in [2,9,15] to detect symmetries of geometric figures; the present application, by contrast, occurs in a purely combinatorial setting.)

Let $G$ have $n$ vertices and $m$ edges; assume $n \geq 3$, since symmetry detection is trivial for $n < 3$, and recall that outerplanarity implies $m \leq 2n - 3$. The actual encoding of $G$ resembles that used in [6] for an outerplanar isomorphism test. Using Algorithm 2.5, the unique Hamilton cycle $H$ of $G$ is first obtained, and one of its two possible orientations is arbitrarily assigned to $H$. For each vertex $v$, each incident edge $(v, w)$ is encoded as an ordered pair of integers, denoting the distances from $v$ to $w$ along $H$ in the chosen orientation and reverse orientation, respectively. Vertex $v$ is then encoded by the string $c(v)$ consisting of the concatenation of these ordered pairs, sorted in ascending order by first component. Figure 2 shows a graph, and the encoding of vertices based on a clockwise orientation of its Hamilton cycle.

A string $S$ which encodes $G$ is then constructed by starting from any vertex and traversing $H$ along the chosen orientation, concatenating the encodings of the vertices as they are encountered. Since $S$ depends on the selected starting vertex, it is subscripted with the name of that vertex. For example, in Fig. 2,

$$S_A = 15245115511542511524511551154251.$$  

Let $v - i$ and $v + i$, for $0 \leq i \leq n - 1$, denote respectively the $i$th predecessor and $i$th successor of vertex $v$ along $H$ in the chosen orientation. Let $\gets{(s, k)}$ denote the result
of removing the leftmost $k$ elements of any string $s$ and appending them, in order, to the right end of $s$; this operation is known as a left cyclic shift.

The following properties then hold for the string $S_u$, for each vertex $u$:

- Although the $n$ possible choices for $u$ generate $n$ corresponding encodings $S_u$, these strings differ from one another only by cyclic shifts; for any $0 \leq i \leq n - 1$,

$$S_{u+i} = \left( S_u, \sum_{j=0}^{i-1} |c(u+j)| \right).$$

- The structure of $G$ can be fully recovered from $S_u$, regardless of the choice of $u$ or the orientation of $H$. In decoding $S_u$, observe that the element 1 occurs precisely at the boundaries of each $c(v)$, corresponding to the Hamilton cycle edges at $u$.

- Each edge of $G$ generates two elements of $S_u$ for each of its two endpoints; thus $|S_u| = 4m \leq 8n - 12$, and so $|S_u| \in O(n)$.

- $S_u$ can be constructed in time $O(n + m) = O(n)$ using the following algorithm. Each traversal of $H$ starts at $u$ and follows the chosen orientation; note that all vertex encodings $c(u)$ are computed simultaneously rather than sequentially:

**Step 1.** Initialize $t$ to 1.

Traverse $H$, and at each vertex $v$:
- set $\#v = t$ and increment $t$ (tag vertices along $H$ with $1, 2, \ldots, n$);
- initialize $c(v)$ to the empty string.

**Step 2.** Traverse $H$, and at each vertex $v$:
- for each edge $(w, v)$ with $\#v < \#w$, append $\#w - \#v$ and $n - (\#w - \#v)$ to $c(v)$.

**Step 3.** Traverse $H$, and at each vertex $v$:
- for each edge $(w, v)$ with $\#v > \#w$, append $n - (\#v - \#w)$ and $\#v - \#w$ to $c(v)$.

**Step 4.** Initialize $S_u$ to the empty string.

Traverse $H$, and at each vertex $v$:
- concatenate $c(v)$ to the end of $S_u$. 

---

Fig. 2. Encoding of vertices, under clockwise orientation of the Hamilton cycle.
To establish the correctness of this simple algorithm, let \( v \) be an arbitrary vertex and let \( c(v) = a_1 a_2 \ldots a_p a_{p+1} \ldots a_q \) where, recalling the definitions of \( c \) and \( \# \),

\[
a_i = \text{encoding of edge } (v, w_i) \text{ where } \begin{cases} \#v < \#w_i \leq n, & \text{for } i \leq p, \\ \#v > \#w_i \geq 1, & \text{for } i > p. \end{cases}
\]

Step 2 appends \( a_1 \ldots a_p \) to the initially empty string \( c(v) \), while Step 3 then appends \( a_{p+1} \ldots a_q \) to the result. (An alternate linear-time algorithm, based on bucket sorting all edges by distances spanned around \( H \), is used in [6]; however, the above algorithm appears simpler and has superior space efficiency.)

**Theorem 3.1.** Let \( G \) be a biconnected outerplanar graph with at least three vertices, and let \( u \) be any vertex of \( G \). Then there is a one-to-one correspondence between

- planar axial symmetries of \( G \), and
- values of \( k \), for \( 0 \leq k < \frac{1}{2} |S_u| \), such that \( \leftarrow (S_u, k) \) is a palindrome.

**Proof.** Consider any planar axial symmetry of \( G \). In the corresponding drawing, one of the three configurations shown in Fig. 3 must occur; here, the solid and broken lines represent the Hamilton cycle \( H \), while the dotted line indicates the axis of symmetry. These three configurations correspond to the axis cutting \( H \) at two edges, one edge and one vertex, and two vertices, respectively, and may be denoted by \( \langle (A, D), (B, C) \rangle \), \( \langle (A, C), B \rangle \), and \( \langle A, B \rangle \).

Without loss of generality, assume that \( H \) was assigned a clockwise orientation. Then, letting \( s^R \) denote the reverse of a string \( s \), the axial symmetry in Fig. 3(a) implies that \( c(A) = c(D)^R \), \( c(A + 1) = c(D - 1)^R \), \( \ldots \), \( c(B) = c(C)^R \), and since \( S_A = c(A)c(A + 1)\ldots c(B)c(C)\ldots c(D - 1)c(D) \), it follows that \( S_A \) is a palindrome. Similarly, each configuration gives rise to a pair of palindromes, as shown:

- Fig. 3(a): \( S_A \) and \( S_C \),
- Fig. 3(b): \( S_A \) and \( \leftarrow (S_B, \frac{1}{2} |c(B)|) \),
- Fig. 3(c): \( \leftarrow (S_A, \frac{1}{2} |c(A)|) \) and \( \leftarrow (S_B, \frac{1}{2} |c(B)|) \).

In each pair, the two strings differ by a left cyclic shift of exactly half their length.
Then since all encodings of $G$ are related by left cyclic shifts, for the given vertex $u$ there exists in each of the configurations a unique integer $k$, with $0 \leq k < \frac{1}{2} |S_u|$, such that $\nabla (S_u, k)$ equals precisely one of these palindromes. This establishes the correspondence from planar axial symmetries to palindromes.

Conversely, suppose there exists an integer $k$, with $0 \leq k < \frac{1}{2} |S_u|$, such that $\nabla (S_u, k)$ is a palindrome. Recalling that the element 1 occurs in the string $S_u$ precisely at the boundaries of each $c(u)$, there are two cases which can arise:

If $\nabla (S_u, k)$ starts with a 1, then there exists a unique integer $i$ which satisfies $\sum_{j=0}^{i-1} |c(u+j)| = k$; let $v = u + i$. Thus

$$S_v = \nabla (S_u, k)$$

and so $G$ must have a planar axial symmetry, as follows:

- for $n$ even: $(v, v-1), (v+n-1, v+n)$ as in Fig. 3(a), with $v = A$,
- for $n$ odd: $(v, v-1) + \frac{1}{2} (n-1)$ as in Fig. 3(b), with $v = A$.

If $\nabla (S_u, k)$ does not start with a 1, then there exists a unique integer $i$ which satisfies $\sum_{j=0}^{i-1} |c(u+j)| + \frac{1}{2} |c(u+i)| = k$; again, let $v = u + i$. Thus

$$S_v \cdash (\nabla (S_u, k)) = \nabla (S_u, k)$$

and so $G$ must have a planar axial symmetry, as follows:

- for $n$ even: $(v, v+1) + \frac{1}{2} n$ as in Fig. 3(c), with $v = A$,
- for $n$ odd: $(v, v+\frac{1}{2} (n-1), v+\frac{1}{2} (n+1))$ as in Fig. 3(b), with $v = B$.

This establishes the correspondence from palindromes to planar axial symmetries.

Since the second correspondence is the inverse of the first, it follows that both are one-to-one. Note that the above proof is constructive, and that in particular, for each vertex $u$ it shows how to recover axial symmetries of $G$ from values of $k$.

Example. In Fig. 2, where $\frac{1}{2} |S_u| = 16$ for each $u$,

$$\nabla (S_e, 10) = \nabla (S_f, 6) = \nabla (S_a, 0) = 15245115511542511524511551154251,$$

$$\nabla (S_b, 10) = \nabla (S_c, 6) = \nabla (S_d, 0) = 15245115511542511524511551154251,$$

which are palindromes, corresponding to the vertical axis of symmetry, while

$$\nabla (S_f, 14) = \nabla (S_a, 8) = \nabla (S_b, 2) = 51154251152451155115425115245115,$$

$$\nabla (S_c, 14) = \nabla (S_d, 8) = \nabla (S_f, 2) = 51154251152451155115425115245115,$$

which are palindromes, corresponding to the horizontal axis of symmetry.
Fig. 4. Nonplanar axial symmetry of a biconnected outerplanar graph.

Example. The restriction in Theorem 3.1 to planar axial symmetries is needed. The graph in Fig. 4 has a nonplanar axial symmetry, indicated by the dotted line. Choosing the orientation ABCDEF for the Hamilton cycle gives

\[ S_A = 152433511551154251152433511551154251, \]

containing two 4335 sequences, so \( (S_A, k) \) cannot be a palindrome for any \( k \).

An \( O(|S_u|) \)-time algorithm to enumerate those values of \( k \), for \( 0 \leq k < \frac{1}{3} |S_u| \), such that \( (S_u, k) \) is a palindrome, is based on the following simple result:

**Lemma 3.2.** For any string \( s \) and integer \( k \), with \( 0 \leq k < \frac{1}{3} |s| \),

\[ (s, k) \text{ is a palindrome } \iff \text{ } s \text{ occurs at position } |s| - 2k + 1 \text{ of } s^R s^R. \]

**Proof.** Elementary, upon decomposing \( s = xyz \) with \( |x| = |y| = k \).

Recall that the Knuth–Morris–Pratt (KMP) pattern-matching algorithm [10] can enumerate all positions in which a string \( s_1 \) occurs as a substring of a string \( s_2 \) in \( O(|s_1| + |s_2|) \) time and space. Combined with Lemma 3.2, this leads to:

**Algorithm 3.3.**

**Input:** A biconnected outerplanar graph \( G \) with \( n \) vertices.

**Output:** A list of all planar axial symmetries of \( G \).

**Time:** \( O(n) \)

**Space:** \( O(n) \)

**Step 1.** If \( n < 3 \) then output the axial symmetries directly, and halt. Otherwise, find the unique Hamilton cycle using Algorithm 2.5, assign it an orientation, select any vertex \( u \), and compute \( S_u \) as previously described.

**Step 2.** Using Lemma 3.2 and the KMP algorithm, enumerate those values of \( k \), for \( 0 \leq k < \frac{1}{3} |S_u| \), such that \( (S_u, k) \) is a palindrome.

**Step 3.** Recover the planar axial symmetries of \( G \) from the values of \( k \), as described in the second part of the proof of Theorem 3.1.

Note that the linear time and space complexities of Algorithm 3.3 are both optimal.
Theorem 3.4. Let \( G \) be a biconnected outerplanar graph with at least three vertices, and let \( u \) be any vertex of \( G \). Then there is a one-to-one correspondence between
- planar rotational symmetries of \( G \), and
- values of \( k \), for \( 0 \leq k < |S_u| \), such that \( (S_u, k) = S_u \).

Proof. Consider any planar rotational symmetry of \( G \). In the corresponding drawing, the Hamilton cycle \( H \) must define a closed nonself-intersecting curve; moreover, since \( H \) is unique this curve must rotate onto itself under the symmetry. Thus for some unique integer \( i \), with \( 0 \leq i < n \), the symmetry maps each vertex \( v \) to \( v + i \) and so \( c(v) = c(v + i) \). Letting \( k = \sum_{j=0}^{i-1} |c(u+j)| \), it follows that
\[
(S_u, k) = c(u+i)c(u+i+1)\ldots c(u+i-1)
= c(u+i)c(u+i+1)\ldots c(u-1+i)
= c(u)c(u+1)\ldots c(u-1)
= S_u.
\]
This establishes the correspondence from planar rotational symmetries to left cyclic shifts preserving \( S_u \).

Conversely, suppose there exists an integer \( k \), with \( 0 \leq k < |S_u| \), such that \( (S_u, k) = S_u \). Since the element 1 occurs in the string \( S_u \) precisely at the boundaries of each \( c(v) \), there exists a unique integer \( i \), with \( 0 \leq i < n \), such that \( \sum_{j=0}^{i-1} |c(u+j)| = k \). Thus \( c(v+i) = c(v) \), for each vertex \( v \). Consider now any one of the many planar drawings of \( G \) in which the curve defined by \( H \) maps onto itself under a rotation through \( i \) vertices in the direction of its orientation. Recalling that edges are represented by straight line segments, whose images under any rotation are completely determined by those of their endpoints, it follows that the entire drawing must similarly map onto itself, yielding a planar rotational symmetry of \( G \) which maps each vertex \( v \) to \( v + i \). This establishes the correspondence from left cyclic shifts preserving \( S_u \) to planar rotational symmetries.

Since the second correspondence is the inverse of the first, it follows that both are one-to-one. Again, the above proof is constructive; in particular, for each vertex \( u \) it shows how to recover rotational symmetries of \( G \) from values of \( k \). \( \square \)

Example. In Fig. 2, where \( |S_u| = 32 \) for each \( u \),
- \( (S_u, 0) = S_u \), for each \( u \), corresponding to the identity rotation;
- \( (S_u, 16) = S_u \), for each \( u \), corresponding to a (clockwise) rotation through \( 180^\circ \).

Example. The restriction in Theorem 3.4 to planar rotational symmetries is needed. The graph in Fig. 5 has a nonplanar rotational symmetry through \( 180^\circ \), as shown. Choosing the orientation ABCDEFGH for the Hamilton cycle gives
\[
S_A = 17267117711762711771177117267117711771176271,
\]
containing only \( c(F) = c(A) \); but \( c(F-1) = c(E) \neq c(H) = c(A-1) \), so \( (S_A, k) \) cannot equal \( S_A \) for any \( 0 < k < |S_u| \).
An $O(|S_u|)$-time algorithm to enumerate those values of $k$, for $0 \leq k < |S_u|$, such that $\leftarrow (S_u, k) = S_u$, is based on the following simple result:

**Lemma 3.5.** For any string $s$ and integer $k$, with $0 \leq k < |s|$, 

$$\leftarrow (s, k) = s \quad \text{occurs at position } k + 1 \text{ of } ss.$$

**Proof.** Elementary, upon decomposing $s = xy$ with $|x| = k$. $\Box$

Combining the KMP pattern-matching algorithm with Lemma 3.5 leads to:

**Algorithm 3.6.**

*Input:* A biconnected outerplanar graph $G$ with $n$ vertices.

*Output:* A list of all planar rotational symmetries of $G$.

*Time:* $O(n)$

*Space:* $O(n)$

*Step 1.* If $n < 3$ then output the rotational symmetries directly, and halt. Otherwise, find the unique Hamilton cycle using Algorithm 2.5, assign it an orientation, select any vertex $u$, and compute $S_u$ as previously described.

*Step 2.* Using Lemma 3.5 and the KMP algorithm, enumerate those values of $k$, for $0 \leq k < |S_u|$, such that $\leftarrow (S_u, k) = S_u$.

*Step 3.* Recover the planar rotational symmetries of $G$ from the values of $k$, as described in the second part of the proof of Theorem 3.4.

Note that the linear time and space complexities of Algorithm 3.6 are both optimal.

4. **Displaying symmetry in biconnected outerplanar graphs**

Recall that a polygon is *regular* if all its sides have the same length and the interior angle between every two adjacent sides is the same. A *chord* of a polygon is a straight-line segment joining two of its nonadjacent corners. Note that every regular polygon is convex; thus, no side intersects either another side or a chord except possibly at its endpoints, and every chord, being straight, lies in the interior.
**Definition.** Let $G$ be a biconnected outerplanar graph with $n \geq 3$ vertices. Then a *regular polygon drawing* of $G$ is one in which its unique Hamilton cycle is drawn as an $n$-sided regular polygon.

Since only *straight-edge* drawings are considered, the vertices of $G$ must appear precisely at the corners of this polygon, and all edges not on the Hamilton cycle must be drawn as chords. Moreover, all regular polygon drawings of $G$ are identical up to translation, reflection, rotation, and scaling. These operations preserve symmetry and may thus be ignored in the present context; hence any drawing from this class will be termed the regular polygon drawing of $G$.

**Lemma 4.1.** Let $G$ be a biconnected outerplanar graph with at least three vertices. Then the regular polygon drawing of $G$ is a planar drawing of $G$.

**Proof.** As noted above, segments representing edges on the Hamilton cycle $H$ do not cross any other segments. Suppose then that the drawing has a pair of chords, representing edges $(A, C)$ and $(B, D)$ say, which cross. Let $G'$ be the subgraph of $G$ obtained by removing all other edges not on $H$. Then $G'$ is depicted in Fig. 6(a) with the broken lines representing $H$. But $G'$ is a subdivision of the forbidden subgraph $K_4$, as shown in Fig. 6(b), thereby contradicting the outerplanarity of $G$. □

**Corollary 4.2.** Let $G$ be a biconnected outerplanar graph with at least three vertices. Then the regular polygon drawing of $G$ is an outerplanar drawing of $G$.

**Proof.** The Hamilton cycle, which contains all vertices of $G$, defines the boundary of the outer face, since all other edges are drawn as chords. □

(In fact, to within topology-preserving deformations and at most a single reflection, the regular polygon drawing of such a graph $G$ is its *only* outerplanar drawing.)

Observe from Lemma 4.1 that every symmetry of the regular polygon drawing represents a planar symmetry of $G$. More significantly, the converse is also true:

**Theorem 4.3.** Let $G$ be a biconnected outerplanar graph with at least three vertices. Then the regular polygon drawing of $G$ simultaneously displays all its planar axial symmetries and all its planar rotational symmetries.

![Fig. 6. Crossing chords in the regular polygon drawing.](image-url)
Proof. Consider any planar symmetry of \( G \). In the regular polygon drawing of \( G \), the outer polygon itself displays this symmetry restricted to the vertices and edges of the Hamilton cycle \( H \), since it displays all planar symmetries of \( H \). But then the chords must also display this same symmetry, since, being straight, their images under the symmetry are completely determined by the images of their endpoints.

\[ \square \]

Corollary 4.4. All planar symmetries of a biconnected outerplanar graph can be simultaneously displayed in a single drawing.

Using simple counterexamples, it can be shown that the latter result no longer holds if the requirements of either biconnectivity or outerplanarity are removed.

An immediate consequence of Theorem 4.3 is that a maximally-symmetric planar drawing of any biconnected outerplanar graph may in fact be constructed directly, \textit{without} actually calculating its symmetries. Nevertheless, an explicit enumeration of these symmetries, as provided by Algorithms 3.3 and 3.6, is still important since:

- the appearance of a drawing is generally enhanced if it is aligned so that one of its axes of symmetry is vertical;
- a clear visual indication of the symmetries in a drawing, such as with dotted lines or curved arrows as in previous examples, is often desirable;
- apart from their application in guiding the construction of drawings, geometric symmetries of graphs may also be of independent theoretical interest.

5. Partitioning outerplanar graphs into isomorphism classes

In extending the preceding results to include outerplanar graphs which are not biconnected or even connected, a key step is the partitioning of a set of outerplanar graphs into equivalence classes of isomorphic graphs, known as \textit{isomorphism classes}. To help accomplish this, the concept of a \textit{signature} is first introduced:

**Definition.** A \textit{signature} is a function from a class of graphs to strings, such that two graphs in the class have the same signature value iff they are isomorphic.

Compact and computationally efficient signatures for the class of outer-planar graphs are presented in [6,14]. The signature developed below is essentially that of [6], but the details are included here in order both to adapt them to the present framework and to extend the scheme to the class of \textit{rooted} outerplanar graphs.

Recall the earlier definition of the block-cutvertex tree of a connected graph. Vertices representing blocks and cutvertices will be termed \textit{B-nodes} and \textit{C-nodes}, respectively. Figures 7(a) and 7(b) show a graph and its block-cutvertex tree; in the latter, B-nodes are drawn as solid disks while C-nodes are drawn as empty circles. Now every block-cutvertex tree has a single center, since all its leaves are B-nodes, successive levels in from the leaves consist alternately of all C-nodes and all B-nodes,
and no pair of B-nodes or C-nodes is adjacent. Thus every block-cutvertex tree may be rooted as its center, as illustrated in Fig. 7(c).

In deriving a signature $\Sigma$ for an outerplanar graph $G$, an auxiliary string-valued function $\sigma$ is first defined at each vertex in each of its rooted block-cutvertex trees; for completeness, the encoding $S_u$ must now be extended to the blocks $K_1$ and $K_2$ as the empty string and 1111 respectively, with $c(x) = 11$ for each vertex $x$ of $K_2$. The definition of $\sigma$ is by induction, proceeding upwards from leaves to roots:

For each B-node $v$: $\sigma(v) = S_v$, defined as follows: Let $b$ be the block of $G$ represented by $v$. For each vertex $x$ of $b$, first define $\hat{c}(x)$: if $x$ is a cutvertex represented by a child $w$ of $v$ (with $\sigma(w)$ known by induction), let $\hat{c}(x) = c(x)\sigma(w)$; otherwise, let $\hat{c}(x) = c(x)$. Traverse the Hamilton cycle of $b$, starting at any vertex $u$, and concatenate all these $\hat{c}$ values to form a string $S_u$ (just as $S_v$ was formed from $c$ values). Let $z$ be the cutvertex of $b$ represented by the parent of $v$, and orient the Hamilton cycle of $b$ to give the smaller lexicographic value of $S_z$; if $v$ is a root and so has no parent, let $z$ be any vertex of $b$ giving the smallest lexicographic value of $S_z$ under either orientation of the Hamilton cycle.

For each C-node $v$: $\sigma(v) = [\sigma(w_1)] [\sigma(w_2)] ... [\sigma(w_k)]$ where $w_1, ..., w_k$ are the children of $v$, arranged in increasing lexicographic order according to their $\sigma$ values, already known by induction.

Figure 7 illustrates the calculation of $\sigma$ at certain selected vertices in the rooted block-cutvertex tree of a connected outerplanar graph.

---

**Fig. 7.** Calculation of the function $\sigma$. (a) A connected outerplanar graph. (b) Its block-cutvertex tree. (c) Block-cutvertex tree rooted at center.

$\sigma(F) = \sigma(G) = [111][11212211221];$

$\sigma(B) = \sigma(C) = [133113223111331][1111][122112211221][132231];$

$\sigma(A) = 132231$

$[133113223111331][1111][122112211221][132231]$

$1331$

$132231$

$[133113223111331][1111][122112211221][132231]$

$1331.$
Letting $\omega$ denote the number of connected components in $G$, its signature $\Sigma(G)$ is now defined as:

$$\Sigma(G) = [\sigma(r_1)][\sigma(r_2)]\ldots[\sigma(r_\omega)]$$

where $r_1, \ldots, r_\omega$ are the roots of the rooted block-cutvertex trees of $G$, arranged in increasing lexicographic order according to their $\sigma$ values.

Since $\Sigma(G)$ is uniquely determined from the structure of $G$, isomorphic graphs always produce the same value of $\Sigma$. Conversely, since the structure of $G$ can be fully recovered from $\Sigma(G)$, nonisomorphic graphs always produce different values of $\Sigma$. Thus $\Sigma$ is indeed a signature for the class of outerplanar graphs.

Suppose $G$ has $n$ vertices, $m$ edges, and $\beta$ blocks. The encoding of each block contains four elements for each of its edges; a pair of square brackets then enclose the encoding of each noncenter block, and each of the $\omega$ connected components. Thus $|\Sigma(G)| \leq 4m + 2\beta + 2\omega \in O(m + n) = O(n)$, since $m \in O(n)$.

Lexicographic sorting arises in computing $\sigma$ at C-nodes with multiple children and in computing $\Sigma$ for disconnected graphs. Now any finite set of strings may be sorted lexicographically in optimal time which is linear in the sum of their lengths [1, §3.2]. Nevertheless, computing $\sigma$ directly from its definition can lead to an $O(n \log n)$- or even $\Omega(n^2)$-time algorithm for $\Sigma$, since the strings being sorted are constantly growing as the computation of $\sigma$ progresses from the leaves of the block-cutvertex tree to its root. To restrict this growth in string length and attain an $O(n)$-time algorithm, an auxiliary compact encoding $\sigma'$ is defined as follows. Let $v$ be any vertex in the tree and let $w$ be a general child of $v$. If $v$ is a B-node, then $\sigma'(v) = \sigma(v)$ except that $\sigma'(w)$ rather than $\sigma(w)$ is used in its computation. If $v$ is a C-node, then the $\sigma'$ values for the entire level of B-nodes immediately beneath $v$ are first sorted, and the rank of $\sigma'(w)$ in the resulting list, rather than the value of $\sigma(w)$, is used in forming $\sigma'(v)$. The key to achieving efficiency is that, although the values of $\sigma$ are still computed and still increase in length as they propagate up the tree, only their corresponding $\sigma'$ values actually take part in the sorting process. This compact encoding scheme is similar to that used in [1] to attain a linear-time tree isomorphism test.

Finally, computing $\sigma$ at a B-node which is also a tree root involves choosing $z$ so as to minimize $\tilde{S}_z$ lexicographically. This may be accomplished efficiently by computing $\tilde{S}_u$ for any vertex $u$ in the same block, and then using the linear-time algorithm of [5] to find a left cyclic shift of $\tilde{S}_u$ yielding a minimum result.

Careful implementation of the above algorithm, in particular the use of pointer relinking to achieve constant-time string concatenation, leads to:

**Theorem 5.1.** Every $n$-vertex outerplanar graph has a signature of length $O(n)$, which can be computed in $O(n)$ time and space.

Once a signature has been obtained, a set of graphs is then easily and efficiently partitioned into isomorphism classes by means of a simple lexicographic sort on the signature values, followed by equality tests on adjacent sorted values. Thus:
Theorem 5.2. Every set of outerplanar graphs, having a total of \( n \) vertices, can be partitioned into isomorphism classes in \( O(n) \) time and space.

The preceding concepts and results apply in general to all outerplanar graphs; in the present context, it is useful to also introduce the following variation:

**Definition.** A *rooted graph* is a connected graph with a single distinguished vertex called its *root*. A *rooted isomorphism* is an isomorphism between two rooted graphs which maps one root onto the other. A *rooted signature* is a signature for a class of rooted graphs, abstracting rooted isomorphism.

Consider now a rooted outerplanar graph \( G \) with root \( r \). Since \( G \) is connected, it has a single block-cutvertex tree \( T \). A rooted signature for \( G \) may be obtained by modifying the signature \( \Sigma \) of the corresponding unrooted graph as follows:

If \( r \) is a cutvertex of \( G \): then root \( T \) at the \( C \)-node representing \( r \), instead of at its center.

If \( r \) is not a cutvertex of \( G \): then root \( T \) at the \( B \)-node \( v \) representing the block containing \( r \), instead of at its center, and define \( \sigma(v) = \tilde{S}_r \) instead of \( \tilde{S}_z \), for the previously-defined vertex \( z \).

This indeed produces a rooted signature, since the resulting string fully determines, and is determined by, the structure of \( G \) together with its root \( r \). Hence:

**Theorem 5.3.** Every \( n \)-vertex rooted outerplanar graph has a rooted signature of length \( O(n) \), which can be computed in \( O(n) \) time and space.

**Theorem 5.4.** Every set of rooted outerplanar graphs, having a total of \( n \) vertices, can be partitioned into rooted isomorphism classes in \( O(n) \) time and space.

6. Detecting and displaying symmetry in outerplanar graphs

Define an *outerplanar symmetry* of a graph to be any symmetry which can be displayed in an outerplanar drawing of the graph. Every outerplanar symmetry is clearly a planar symmetry. It follows easily from Theorem 4.3 and Corollary 4.2 that for biconnected outerplanar graphs, every planar symmetry is actually outerplanar. Moreover, for all outerplanar graphs, every planar rotational symmetry is likewise outerplanar, as isomorphic subgraphs which rotate onto one another can always be redrawn in a uniform outerplanar manner without altering the original symmetry. However, in the absence of biconnectivity an outerplanar graph may have a planar axial symmetry which is not outerplanar, as illustrated in Fig. 8. Outerplanarity is clearly a desirable property in drawings of outerplanar graphs, so in the remainder of this paper only outerplanar symmetries will be considered. Linear algorithms for general planar symmetries of outerplanar graphs have also been found, but these will not be presented here.
Let $G$ be a connected outerplanar graph. Its single block-cutvertex tree may be constructed by using Algorithm 2.2. According as the center of this tree is a B-node or a C-node, the graph $G$ is said to be $B$-centered or $C$-centered and the block or cutvertex corresponding to this center is called its central block or central cutvertex, respectively. Because of a basic difference between symmetries of $B$-centered and $C$-centered graphs, as explained later, these two cases will be examined separately.

6.1. Case: $G$ is $B$-centered

Clearly, any symmetry of $G$ must map the central block onto itself, but may permute amongst themselves the remaining subgraphs, which are known as tufts:

**Definition.** A tuft of a B-centered graph is a maximal connected rooted subgraph which remains when all edges of the central block are removed; the root of a tuft is its single vertex belonging to the central block.

Note that in all outerplanar drawings of a B-centered graph, the cyclic ordering of its tufts clockwise around its central block is fixed, to within a global reversal, since the outerplanar embedding of this block is unique up to a single reflection.

Axial symmetry in $G$ will generally be increased by the presence of $A$-tufts, which can straddle potential axes of symmetry:

**Definition.** An $A$-tuft of a B-centered graph is a tuft having an axial symmetry which both maps its root onto itself and can be displayed in an outerplanar drawing where all other vertices lie on one side of some straight line passing through the root.

The second condition here is necessary to ensure that an A-tuft may be drawn symmetrically in the exterior of the central block. A linear algorithm is given later which decides if a tuft is an A-tuft, and if so, constructs a corresponding drawing.

Under any symmetry of $G$ each tuft must either map entirely onto another tuft, with root mapping onto root, or map onto itself, with root remaining fixed. In the former case, there must be a rooted isomorphism between the tufts; in the latter,
unless the symmetry is merely the identity rotation, the tuft must be an A-tuft.

Symmetry detection algorithms for B-centered graphs are extensions of those for biconnected graphs. The basic approach is to take the previous encoding $S_u$ of the central block, expand it to produce a new string $\tilde{S}_u$ incorporating relevant details of the tufts, and then search for symmetries in $\tilde{S}_u$. To construct $\tilde{S}_u$, first use the algorithm of Theorem 5.4 to partition the set of tufts into rooted isomorphism classes $X_1, \ldots, X_p$. Let $v$ be any vertex in the central block, let $T$ be the tuft rooted at $v$, and let $X_i$ be the isomorphism class of $T$. Now the previous vertex encoding $c(v)$ has even length, so $c(v) = c_1(v)c_2(v)$ where $|c_1(v)| = |c_2(v)|$. Define:

$$\tilde{c}(v) = \begin{cases} c_1(v)s_i s c_2(v), & \text{if } T \text{ is an A-tuft}, \\ c_1(v)s_i s c_2(v), & \text{otherwise.} \end{cases}$$

Then for every vertex $u$ in the central block, the string $\tilde{S}_u$ is formed in the same manner as $S_u$, except that $\tilde{c}(v)$ is now used in place of $c(v)$ for each such vertex $v$. Note that $|\tilde{S}_u| = O(n)$ and that $\tilde{S}_u$ can be constructed in $O(n)$ time and space, where $n$ denotes the number of vertices in $G$.

To present the results for symmetry detection, the notion of distinct symmetries is first needed. Two symmetries of a graph are distinct unless some drawing which shows one can, by means of the same geometric symmetry, show the other simply by redrawing individual tufts. Figure 9 gives examples of nondistinct symmetries.

**Theorem 6.1.** Let $G$ be a B-centered graph and let $u$ be any vertex in its central block. If $|\tilde{S}_u|$ is odd, then $G$ has no outer-planar axial symmetry. Otherwise there is a one-to-one correspondence between

- distinct outerplanar axial symmetries of $G$, and
- values of $k$, $0 \leq k < \frac{1}{2} |\tilde{S}_u|$, such that $\leftarrow (\tilde{S}_u, k)$ is a palindrome.

**Proof.** Recall that $\tilde{S}_u$ is composed of $S_u$, which has even length, plus four elements for each A-tuft and three elements for each non-A-tuft. So if $|\tilde{S}_u|$ is odd, there

![Fig. 9. Nondistinct axial symmetries and nondistinct rotational symmetries.](image-url)
must be an odd number of non-A-tufts. In that case no outerplanar drawing of $G$
can have an axis of symmetry, since non-A-tufts must be arranged in matching pairs
on opposite sides of any axis, which requires an even number of such tufts.

Otherwise, upon observing that any axial symmetry must map the central block
onto itself, the proof of Theorem 3.1 generalizes immediately to establish the stated
correspondence between symmetries and palindromes. Note that since $|\bar{S}_u|$ is even,
the middle elements of a palindromic $\leftrightarrow (\bar{S}_u, k)$ can never be $\$i\$; a configuration
of this form would correspond to an axis passing through a non-A-tuft. \)

**Theorem 6.2.** Let $G$ be a B-centered graph and let $u$ be any vertex in its central
block. Then there is a one-to-one correspondence between

- distinct (outer)planar rotational symmetries of $G$, and
- values of $k$, $0 \leq k < |\bar{S}_u|$, such that $\leftrightarrow (\bar{S}_u, k) = \bar{S}_u$.

**Proof.** An immediate generalization of the proof of Theorem 3.4, upon observing
that any rotational symmetry must map the central block onto itself. \)

**Corollary 6.3.** All distinct outerplanar symmetries of an $n$-vertex B-centered graph
can be enumerated in $O(n)$ time and space.

**Proof.** Any such graph can possess at most $O(n)$ distinct outerplanar symmetries,
each with an $O(1)$-space description (these axial and rotational symmetries may be
specified, respectively, by the vertices and outer edges of the central block which are
cut by the axis, and by the number of vertices through which the Hamilton cycle
of the central block is rotated). As the proofs of Theorems 6.1 and 6.2 are constructive,
the above specifications of the graph symmetries can be recovered directly from
the corresponding values of $k$. These values, in turn, can be enumerated in $O(n)$ time
by using Lemmas 3.2 and 3.5 coupled with the KMP algorithm, as before. \)

By contrast, note that there may be $\Omega(2^n)$ nondistinct outerplanar symmetries,
as suggested by Fig. 9, each requiring an $\Omega(n)$-space description.

To construct a drawing which displays distinct outerplanar axial symmetries of
a B-centered graph, start with the regular polygon drawing of its central block (or
a single straight-line segment, if this block is $K_2$). Then in its exterior, draw each
A-tuft symmetrically, as described later, and draw the remaining tufts so that within
each rooted isomorphism class, consecutive tufts around the central block are
arranged in a mirror-image fashion. Figure 10(a) demonstrates this construction.
Using this method, all distinct outerplanar axial symmetries of a B-centered graph
can be simultaneously displayed in a single drawing.

Likewise, to display rotational symmetries, start with a similar drawing of the
central block. Then in its exterior, draw all the tufts so that within each rooted isomorphism class, each of the tufts is arranged as a rotated copy of its predecessor around the central block. Figure 10(b) demonstrates this construction. Again, using
Symmetry in outerplanar graphs

Fig. 10. Axial and rotational symmetries of a B-centered graph.

this method, all distinct (outer)planar rotational symmetries of a B-centered graph can be simultaneously displayed in a single drawing.

By contrast with the result for biconnected outerplanar graphs (Corollary 4.4), Fig. 10 also shows that in the case of B-centered graphs, it may not be possible to simultaneously display axial and rotational symmetries in the same drawing.

6.2. Case: G is C-centered

Clearly, any symmetry of G must map the central cutvertex onto itself, but may permute amongst themselves the remaining subgraphs, again known as tufts:

Definition. A tuft of a C-centered graph is a maximal connected rooted subgraph which remains when the central cutvertex is removed from the graph and then separately reattached to each of the resulting connected components; the root of each tuft is the central cutvertex.

Recall that in all outerplanar drawings of B-centered graphs, the cyclic ordering of tufts around the central block is essentially fixed. For C-centered graphs, however, the cyclic ordering around the central cutvertex is unconstrained. This key difference between B-centered and C-centered graphs has profound implications for symmetry, and explains why the two cases are examined separately here.

Related to this freedom in the placement of tufts is a further difference between B-centered and C-centered graphs: it may not be possible to simultaneously display all distinct outerplanar axial symmetries in a single drawing of a C-centered graph. Such a situation is illustrated by the two drawings of the same graph in Fig. 11. Consequently, rather than attempting to find all distinct outerplanar symmetries of a C-centered graph, for the construction of symmetric drawings it is now more appropriate to determine maximum-sized sets of (necessarily distinct) outerplanar symmetries which can be simultaneously displayed in a single drawing.
Symmetries of a C-centered graph are essentially those of a single-centered tree, its block-cutvertex tree, refined to take account of the internal structure of its tufts. Results on detection and display of symmetry in trees have already been developed in [12], and are now summarized.

Given a single-centered tree with more than one vertex, its center is removed to yield a set of $c$-trees, each of which is rooted at the unique vertex previously adjacent to the center. This set is then partitioned into rooted isomorphism classes $X_1, \ldots, X_p$ containing $N_1, \ldots, N_p$ trees, respectively. A class $X_i$ is called an $A$-class if a $c$-tree from that class, together with the center of the entire tree and the edge joining the center to the root, has itself got an axial symmetry passing through both center and root; in a drawing of the tree, (the root of) a $c$-tree can lie along an axis of symmetry only if it belongs to an $A$-class. Letting $g = \gcd(N_1, \ldots, N_p)$ and $M_i = N_i/g$ (1 ≤ $i$ ≤ $p$), define the conditions (α) and (β) as follows:

(α) at most two of the $N_i$ are odd and $X_i$ is an $A$-class whenever $N_i$ is odd;
(β) at most two of the $M_i$ are odd and $X_i$ is an $A$-class whenever $M_i$ is odd.

Let $\#A$ and $\#R$ denote the maximum number of axial and rotational symmetries, respectively, which can be simultaneously displayed in single drawings of the tree; their values are then given by:

**Theorem 6.4** [12]. $\#A = 0$, if (α) fails; $g$, if (β) holds; $g/2$, otherwise.

**Theorem 6.5** [12]. $\#R = g$.

Returning now to C-centered graphs, those tufts which can straddle potential axes of symmetry are again termed $A$-tufts:

**Definition.** An $A$-tuft of a C-centered graph is a tuft having an outerplanar axial symmetry which maps its root onto itself.

This definition is somewhat simpler than that for B-centered graphs, as the second condition there is always fulfilled here. Again, a linear algorithm is given later which decides if a tuft is a $A$-tuft, and if so, constructs a corresponding drawing.

Clearly, the tufts of a C-centered graph are represented by the $c$-trees of its block-cutvertex tree, with each rooted isomorphism class of $A$-tufts represented by an
A-class of c-trees. Extending the definitions of \( p, X_i, N_i, g, M_i, (\alpha), (\beta), \#A \) and \#R from trees to C-centered graphs, it follows immediately that:

Theorems 6.4 and 6.5 also hold for C-centered graphs.

The proofs of Theorems 6.4 and 6.5 are constructive and provide linear algorithms which actually produce such maximally-symmetric drawings. For the graph shown in Fig. 12, \( p = 2, (N_1, N_2) = (4, 4), g = 4, (M_1, M_2) = (1, 1) \), so (\( \alpha \)) holds and (\( \beta \)) fails, giving \( \#A = g/2 = 2 \) (Fig. 12(a)) and \( \#R = g = 4 \) (Fig. 12(b)). Again, by contrast with the result for biconnected outerplanar graphs (Corollary 4.4), Fig. 12 also shows that in the case of C-centered graphs, it may not be possible to simultaneously display axial and rotational symmetries in the same drawing.

6.3. Recognition and drawing of A-tufts

As already seen, A-tufts play a key role in axial symmetry for both B-centered and C-centered graphs; their efficient detection and display are now examined.

**Definition.** A plume of a tuft is a maximal connected rooted subgraph which remains when the root is removed from the tuft and then separately reattached to each of the resulting connected components; the root of a plume is the root of its tuft.

Note that a tuft of a B-centered graph may consist of multiple plumes, while a tuft of a C-centered graph always consists of a single plume.

Let \( T \) be a tuft of either a B-centered or C-centered graph, and let \( r \) be the root of \( T \). Using the algorithm of Theorem 5.4, partition the plumes of \( T \) into rooted isomorphism classes, and let \( q \) denote the number of odd-sized classes. Then:

- \( q = 0 \): \( T \) is clearly an A-tuft, and may be drawn symmetrically by arranging the plumes from each class in mirror-image pairs on opposite sides of an axis through \( r \), with all plumes in the same half-plane.
- \( q = 1 \): Let \( P \) be a plume from the odd-sized class, and let \( \overline{S}_r \) be its encoding,
determined recursively. Now $T$ is an A-tuft iff $P$ is itself an A-tuft, which, by a proof similar to those of Theorems 3.1 and 6.1, is true iff $|S_{r}|$ is even and $\langle S_{r}, \frac{1}{2}|c(r)\rangle$ is a palindrome. Then $T$ may be drawn symmetrically by letting $P$ straddle an axis through $r$ and proceeding as in the case $q=0$.

- $q \geq 2$: $T$ is clearly not an A-tuft, since no corresponding symmetric outerplanar drawing is possible in this case.

This now completes the study of symmetry detection and display in connected outerplanar graphs. In the disconnected case, algorithms were also obtained to detect global symmetries and display them in a single drawing of the entire graph, although their complexities are super-linear. However, this global approach seems inappropriate for purposes of conveying structural information about the graph: if some connected components are isomorphic, it appears better to indicate this fact explicitly rather than rely on visual clues to suggest it. The algorithm of Theorem 5.2 may be used to partition the components into isomorphism classes, and a single representative from each class may be analyzed and drawn using the techniques already described. The entire graph may then be displayed either by drawing a single component for each class and indicating its multiplicity, or by replicating the drawings for each component in each class.

Acknowledgement

This author’s research was supported by the Office of Naval Research under contracts N00014-84-K-0502 and N00014-86-K-0689, the Air Force Office of Scientific Research under grant AFOSR-90-0107, the National Science Foundation under grant DCR-8451393, and the National Library of Medicine under grant R01-LM05118.

References


