A Quality Metric for Symmetric Graph Drawings

Amrya Meidiana†
University of Sydney
Seok-Hee Hong‡
University of Sydney
Peter Eades§
University of Sydney
Daniel Keim¶
University of Konstanz

ABSTRACT
Symmetry is an important aesthetic criteria in graph drawing and network visualisation. Symmetric graph drawings aim to faithfully represent automorphisms of graphs as geometric symmetries in a drawing.

In this paper, we design and implement a framework for quality metrics that measure symmetry, that is, how faithfully a drawing of a graph displays automorphisms as geometric symmetries. The quality metrics are based on geometry (i.e., Euclidean distance) as well as mathematical group theory (i.e., orbits of automorphisms).

More specifically, we define two varieties of symmetry quality metrics: (1) for displaying a single automorphism as a symmetry (axial or rotational) and (2) for displaying a group of automorphisms (cyclic or dihedral). We also present algorithms to compute the symmetric quality metrics in $O(n \log n)$ time for rotational symmetry and axial symmetry.

We validate our symmetry quality metrics using deformation experiments. We then use the metrics to evaluate a number of established graph drawing layouts to compare how faithfully they display automorphisms of a graph as geometric symmetries.

Index Terms: Human-centered computing [Visualization]: Visual-ization design and evaluation methods

1 INTRODUCTION
Graph drawing aims to construct a visually-informative drawing of an abstract graph in the plane. Symmetry is one of the most important aesthetic criteria that represent the structure and properties of a graph visually [3]. Symmetric graph drawings aim to faithfully represent automorphisms of graphs as geometric symmetries in the drawing. Also, a symmetric drawing of a graph enables an understanding of the entire graph to be built up from that of a smaller subgraph.

Symmetric drawings of a graph $G$ are clearly related to the automorphisms of $G$, and algorithms for constructing symmetric drawings have two steps:

Step 1. Find the “appropriate” automorphisms, and
Step 2. Draw the graph displaying these automorphisms as symmetries.

The problem of determining whether a graph has a nontrivial automorphism is automorphism complete [19]. However, the problem of determining whether a graph has a nontrivial geometric automorphism is NP-complete [23]. Linear time algorithms to construct symmetric drawings for restricted classes of graphs exists (e.g., trees [21], outerplanar graphs [22], series-parallel digraphs [15], and planar graphs [12][14][16]). For general graphs, heuristics [7][19] and exact algorithms are available [15].

It is important to ensure that a symmetric graph drawing accurately represent the automorphisms of the underlying graph to the greatest extent possible. However, existing symmetry detection and quality metrics for graph drawings do not focus on this comparison between detected geometric symmetry in a graph drawing and the automorphisms of the graph.

In this paper, we design and implement a framework for quality metrics for graph drawings that measure symmetry, that is, how faithfully a drawing of a graph displays automorphisms as geometric symmetries. The quality metrics are based on geometry (i.e., Euclidean distance) as well as mathematical group theory (i.e., orbits of automorphisms).

More specifically, we present the following contributions:

1. We design and implement a framework for a quality metric for symmetric graph drawing, which measures the symmetry quality of a drawing of a graph based on the comparison between the symmetry of the drawing and the automorphism of the graph. We define two varieties of the symmetry quality metrics, one to measure how well a graph drawing displays a single automorphism as a symmetry (rotational or axial) and one to measure how well a graph drawing displays a group of automorphism simultaneously (cyclic or dihedral groups).

2. We present algorithms to compute the symmetry quality metrics in $O(n \log n)$ time for rotational and axial symmetry and $O(kn \log n)$ for automorphism groups, where $n$ is the number of vertices in the graph and $k$ is the size of the automorphism group.

3. We validate the single automorphism detection version of the symmetry quality metrics through deformation experiments of graph drawings, showing that the scores computed by our metric decrease when the drawings are distorted further from exact symmetry.

4. We validate the automorphism group detection version of the symmetry quality metrics through comparing drawings displaying different groups of automorphisms, showing that our metric effectively captures the difference in symmetry quality between drawings that display different numbers of automorphisms as symmetries.

5. We use our metric to compare a number of established graph drawing layouts to compare how faithfully they display automorphisms of a graph as geometric symmetries. We confirm the effectiveness of the concentric circles layout in displaying a graph’s automorphisms as symmetries.

2 RELATED WORKS
2.1 Symmetries and geometric automorphisms
An automorphism of a graph is a permutation of the vertices that preserves the adjacency of each vertex. Suppose that a permutation group $A$ acts on a set $V$. We say that $\phi \in A$ fixes $v \in V$ if $\phi(v) = v$; if $\phi$ fixes $v$ for every $\phi \in A$ then $A$ fixes $v$. If $V' \subseteq V$ and $\phi(v') \in V'$ for all $v' \in V'$ then $\phi$ fixes $V'$ (setwise). A subset $V'$ of $V$ is an orbit of $A$ if, for each $u, v \in V'$, there is $\phi \in A$ such that $\phi(u) = v$, and $V'$ does not contain a nonempty subset with this property.
It is important to use a rigorous model, introduced by Eades and Lin [9], for the intuitive concept of symmetry display. The symmetries of a set of points in the plane (such as a two-dimensional graph drawing) form a group called the symmetry group of the set. A symmetry \( \sigma \) of a drawing \( D \) of a graph \( G \) induces an automorphism of \( G \), in that the restriction of \( \sigma \) to the points representing vertices of \( G \) is an automorphism of \( G \). A drawing \( D \) of a graph \( G \) displays an automorphism \( \phi \) of \( G \) if there is symmetry \( \sigma \) of \( D \) which induces \( \phi \). The symmetry group of a graph drawing induces an automorphism group of the graph. An automorphism group \( A \) of a graph \( G \) is a geometric automorphism group if there is a drawing of \( G \) which displays every element of \( A \).

A non-trivial symmetry of a finite set of points in the plane is either a rotation about a point or a reflection about a line. A geometric automorphism is a rotational automorphism (respectively reflectional/axial automorphism) if it is induced by a rotation (respectively reflection). Eades and Lin [9] showed that that a nontrivial geometric automorphism group is one of three kinds:

1. a group of size 2 generated by an axial automorphism;
2. a cyclic group of size \( k \) generated by a rotational automorphism;
3. a dihedral group of size \( 2k \) generated by a rotational automorphism of order \( k \) and an axial automorphism.

### 2.2 Symmetric Graph Drawing

In general, determining whether a graph can be drawn symmetrically in two dimensions is NP-complete [19]. Exact algorithms based on Branch and Cut [5] and group theory [1] are available.

Linear-time algorithms are available for symmetric drawings of limited classes of graphs, such as trees [21], outerplanar graphs [22], and series-parallel digraphs [15]. Linear-time algorithms have also been presented for maximally symmetric drawings of triconnected [15], biconnected [12], oneconnected [13], and disconnected planar graphs [14]. For a survey on symmetric drawings of graphs in two dimensions, see [8].

### 2.3 Graph Drawing Symmetry Quality Metrics

Purchase [24] defined a metric measuring the symmetry of a graph drawing by checking, for each pair of vertices, whether there is a symmetric subgraph around the pair, calculating a weighted symmetry value of the symmetric subgraph if it exists, and adding the weighted symmetry value of all symmetric subgraphs. However, this metric has a best runtime of \( O(n^2) \) and only considers axial symmetry.

Klauwak et al. [17] defined a metric which detects rotational, axial, and translational symmetry in a node-link graph drawing, using methods adapted from computer vision. Taking as input an image of a drawing of a graph, this method detects symmetries by computing a geometric transformation, which is then used to compute the symmetry quality for each pair of vertices, giving the runtime as \( O(n^2) \) time in the number of edges, with the worst case of \( O(n^2) \) time in the number of vertices for dense graphs.

### 2.4 Geometric Symmetry Detection

Optimal algorithms for exact symmetry detection for two-dimensional point sets and polygons run in \( O(n \log n) \) time for detecting rotational or axial symmetry in two-dimensional point sets [21,23]. These algorithms work by sorting the points by their angle around a center point and then finding a palindromic sequence of angles and distances (for axial symmetry) or a subsequence that is repeated within the full sequence (for rotational symmetry). However, these algorithms only give a binary answer to whether a point set displays a symmetry and are not suitable to quantifying the approximate symmetry of a point set.

Zabrodsky et al. [27] introduced the symmetry distance method, providing another method to quantify the approximate symmetry of a geometric object. This method takes as input a point set \( P \) and a symmetry \( \sigma \) to be checked and computes a symmetric point set \( P' \) realizing the input symmetry. This is done through a “folding” transform method that “folds” points in the same orbit, averages their positions, and “unfolds” them into a symmetric configuration. This method minimizes the Euclidean distance between each point in \( P \) and its image in \( P' \), and the symmetry distance is given as the weighted average of these Euclidean distances for every point in \( P \).

### 3 Symmetry Quality Metric Framework

We propose a new quality metric for graph visualization, the symmetry quality metric, for measuring how well the drawing of a graph displays selected geometric automorphisms of the underlying graph. Our metric is a faithfulness metric comparing the geometric symmetry detected in a drawing of a graph with the graph’s automorphisms, the ground truth information, unlike existing metrics which only attempt to detect symmetry from the drawing. Figure 1 summarizes the framework used for our proposed metric.

![Figure 1: Framework for the symmetry quality metric.](image)

Let \( G \) be a graph and \( \phi \) be a automorphism of \( G \). It is important that \( \phi \) be a geometric automorphism of \( G \), as otherwise it is impossible to display it as a symmetry of a drawing of \( G \) [9]. The framework computes the symmetry quality score using the following steps:

**Framework: Symmetry Quality Metric**

**Step 1:** Apply a layout algorithm to \( G \) to obtain a graph drawing \( D \), which provides geometric positions for each vertex in \( G \).
**Step 2:** Detect a geometric symmetry of \( D \), obtaining symmetry \( \sigma \).
**Step 3:** Compare \( \sigma \) to \( \phi \) to compute the symmetry quality metric.

### 3.1 Symmetry Quality Metric for a Single Automorphism

Given a drawing of a graph with an exact symmetry, a geometric symmetry detection algorithm can be used to detect a symmetry of the drawing, which induces an automorphism of the graph, and this result can be compared to the input automorphism. However, in practice, automatic graph layout algorithms may produce drawings that are not perfectly symmetric. We therefore define a refinement that uses approximate symmetry detection to quantify how far the drawing is from an exact symmetry that displays an input automorphism.

The approximate symmetry of a point set \( P \) can be defined as a Euclidean distance between it and a symmetric point set \( P' \). In
this refinement of our metric, for each orbit of φ, we compute the Euclidean distance needed to transform the position of its points in D to perfect symmetry, and use this distance to compute the symmetry quality.

We define two different formulas to compute the symmetry quality metric, $SQ$, given the number of symmetric orbits and the Euclidean distance to exact symmetry for the asymmetric orbits. The first formula gives equal weight to the proportion of symmetric orbits and the average Euclidean distance to perfect symmetry of asymmetric orbits:

$$SQ_1 = \frac{1}{2} \left( \frac{|O_{sym}|}{|O|} + \frac{1}{|O_{asym}|} \sum_{k=1}^{|O|} sd(o_k) \right)$$  \hspace{1cm} (1)

where $O$, $O_{sym}$, and $O_{asym}$ are the sets of all orbits of φ, orbits that are displayed symmetrically in $D$, and orbits that are displayed asymmetrically in $D$ respectively; $o_k$, $k = 1, 2, ..., |O|$ are the orbits of $φ$; and $sd(o_k)$ denotes the Euclidean distance from the positions of points belonging to orbit $o_k$ in $D$ to exact symmetry. The second formula defines possible ranges of values for a drawing based on the number of orbits that are drawn as symmetric and is computed as:

$$SQ_2 = \frac{1}{1 + |O_{asym}|} - \left( 1 - \frac{1}{|O_{sym}|} \right) \sum_{o \in O_{asym}} sd(o)$$  \hspace{1cm} (2)

The fraction $\frac{1}{1 + |O_{asym}|}$ limits the range of metric values based on the number of orbits displayed as symmetries in the drawing, while the sum $\left( 1 - \frac{1}{|O_{sym}|} \right) \sum_{o \in O_{asym}} sd(o)$ penalizes the results based on the Euclidean distance from exact symmetry.

### 3.2 Symmetry Quality Metric for Automorphism Groups

Another refinement of the metric is to take as input an automorphism group, to detect the extent to which multiple automorphisms are simultaneously displayed in a drawing of a graph. Our metric is able to take as input dihedral groups, which contain both rotational and reflectional automorphisms.

To compute the group symmetry quality $SQG$, we consider a weighted sum based on the maximum orbit size of each orbit set. Given $A$ as the automorphism group to be checked, we define weight $w$ as:

$$w = \sum_{\phi \in A} K(\phi)$$  \hspace{1cm} (3)

where $K(\phi)$ is the size of the largest orbit of an automorphism $\phi \in A$. For reflectional automorphisms, the value of $K(\phi)$ is always 2. While for rotational automorphisms, the value of $K(\phi)$ is the order of the rotation. Given this weight $w$, the group symmetry quality score $SQG$ is computed using the formula:

$$SQG = \begin{cases} \frac{1}{w} \sum_{\phi \in A} K(\phi) \times SQ(\phi) \times \frac{1}{2} & \text{if } |A_{sym}| = 0 \\ \left( 1 + \frac{1}{w} \sum_{\phi \in A} K(\phi) \times SQ(\phi) \right) \times \frac{1}{2} & \text{otherwise} \end{cases}$$  \hspace{1cm} (4)

where $A_{sym}$ is the subset of $A$ containing automorphisms that are displayed as exact symmetries in the drawing $D$ and $SQ(\phi)$ is the score computed by the $SQ$ metric with $\phi$ as the input automorphism. $SQ(\phi)$ can be computed with either $SQ_1$ (Equation 1) or $SQ_2$ (Equation 2).

The usage of the weight and the multiplier $K(\phi)$ for each orbit is meant to give preference to drawings that display higher orders of symmetry, i.e. possessing more symmetries. Thus, a drawing that is symmetric but only of a lower order of symmetry will score lower than a symmetric drawing displaying a higher order of symmetry. Furthermore, adding 1 to the sum when at least an automorphism is realized as a symmetry ensures that every symmetric drawing will obtain a higher score than every asymmetric drawing.

### 4 Algorithms for the Symmetry Quality Metric

We present algorithms used to compute our metrics. Section 4.1 and 4.2 provide algorithms for detecting exact and approximate symmetries respectively within the framework of our metric. Section 4.3 presents the algorithm to compute the $SQ$ version of our metric, and Section 4.4 presents the algorithm to compute the $SQG$ version of our metric.

#### 4.1 Exact Symmetry Detection

While exact symmetry detection for two-dimensional point sets and polygons can be run in $O(n \log n)$ time, node-link drawings of graphs are often not simple polygons, which necessitates a modification to the approach. With symmetric graph drawings, it is possible to detect an automorphism of the graph by detecting the geometric symmetry of the vertex point set of the drawing and checking the adjacency of the vertices. The following describes the algorithm:

**Algorithm 1: Algo-ExactSym**

**Input:** Graph $G$, drawing $D$ of $G$

**Output:** true or false

1. Sort the vertex point set of $D$ according to their angle around and distance from the centroid.
2. Detect the geometric symmetry of the point set. If no symmetry is detected, return false, otherwise obtain the geometric symmetry $\sigma$.
3. For each orbit $o_k$ in $\sigma$, compare the adjacency of the vertices included in $o_k$. If all of the vertices do not share the same pattern of adjacency, return false.
4. Return true if for every orbit, all the vertices share the same adjacency pattern.

**Algo-ExactSym** runs in $O(n \log n + m)$ time. Exact symmetry detection algorithms (both rotational and axial) for two-dimensional point sets, run in worst case $O(n \log n)$ time \cite{25}. As each vertex only appears in one orbit, the total running time of the symmetry detection step is of $O(n \log n)$ time in the number of vertices. Step 3 takes linear time in the degree of the vertices, i.e. linear in the number of edges, putting the time complexity of the algorithm as $O(n \log n + m)$ time, with $n$ as the number of vertices and $m$ as the number of edges.

Recently, De Luca et al \cite{20} presented a method to detect axial, rotational, and translational symmetries in graph drawings using a machine learning approach. However, this approach is a classification approach which only returns whether a drawing contains certain symmetries and with an 8% misclassification rate.

#### 4.2 Approximate Symmetry Detection

The previous approach is limited by the nature of the exact symmetry detection algorithm, which either returns a positive answer for an exact symmetry or a negative answer otherwise and does not quantify how far a negative answer is to exact symmetry. There is therefore a need for a method that is able to compare the approximate symmetry of a drawing to the automorphism of the underlying graph.

Given an algorithm that computes the approximate symmetry of a point set, we can use it to determine whether each input orbit of automorphism is displayed as a geometric symmetry in the drawing of the graph. For our implementation, we use the symmetry distance approach of Zabrodsky et al \cite{27} and we compute $sd$ as mentioned in the formulas of $SQ_1$ and $SQ_2$ as below:

**Algorithm 2: Algo-ApproxSym**

**Input:** Graph $G$, drawing $D$ of $G$, automorphism $\phi$ of $G$

**Output:** $sd$
1. Normalize the points in the vertex point set $P$ of drawing $D$
   such that all of the points lie within the unit circle centered on the
centroid of the point set.

2. For each orbit $o_k$ in an automorphism $\phi$:
   (a) Take the subset $P_k$ of $P$, containing points corresponding
to vertices contained in $o_k$.
   (b) Compute the symmetric configuration $P_0$ closest to $P_k$.
   (c) For every vertex $v$ included in $o_k$, compute the Euclidean
distance between its position in $D$ and its image in $P_0$,
average the values, and divide by 2 to obtain the average
distance $d$.
   (d) Subtract $d$ from 1 to obtain the value $sd$.

The distance computed by the symmetry distance approach is
dependent on the area taken by the drawing $D$. To normalize this
distance, we scale the point set such that the whole drawing fits in
a unit circle, which limits the maximum possible distance to 2. We
then divide the average distance $d$ by 2 and subtract the result from
1 in order to get a score where 1 corresponds to exact symmetry and
lower values correspond to drawings that are further from exact
symmetry.

In theory, only orbits where the average distance $d$ is 0 should be
considered symmetric. However, computers work with floating point
precision numbers rather than real numbers, leading to unavoidable
round-off errors. To account for this, we define a threshold $\varepsilon$
such that orbits with $d$ less than $\varepsilon$ are considered to be symmetric.

Algorithm 3: Algo-SQ

Input: Graph $G$, drawing $D$ of $G$, group of automorphisms $A$ of $G$
Output: $SQ$ metric score

1. For each orbit $o_k$ in $\phi$:
   (a) Compute the symmetry distance-based score $sd$ using
       Algo-ApproxSym
   (b) Add $o_k$ to the set $O_{sym}$ if the value of $sd$ is less than or
equal to a threshold $\varepsilon$, or add it to the set $O_{asym}$ otherwise
2. Compute the symmetry quality using either $SQ_1$ or $SQ_2$.

Using Algo-ApproxSym, given a center or axis of symmetry, the
$sd$ values can be computed in $O(n\log n)$ time in the number of
vertices. When this information is not given, we select the center or
axis of symmetry in the following way:

- For rotational symmetries, we compute the centroids for each
  orbit (the point itself in the case of fixed points) and select one
  that is the closest to all other centres (i.e. geometric median).

In the case of ties, we compute the $sd$ of each orbit and select
the one with the largest $sd$ score. After initially sorting the
points in $O(n\log n)$ time, computing the geometric median can
be done with two $O(n)$-time sweeps along the x- and y-axes
and computing the $sd$ scores when needed takes $O(n)$ time,
keeping the runtime at $O(n\log n)$ time.

- For axial symmetries, we compute the best line of symmetry using
  Singular Value Decomposition (SVD) as a pre-processing
  step, as proposed by Zabrodsky et al. [26]. While SVD runs
  in $O(n^3)$ in the size of the input matrices, in our case the matrix
  is of a fixed size $2 \times 2$ so we only consider drawings in
  2 dimensions, keeping the runtime complexity of the metric
  computation at $O(n\log n)$ time in the number of vertices.

Therefore, Algo-SQ runs in $O(n\log n)$ time.

4.4 Algorithm for Computing $SQG$

The $SQG$ metric takes as input multiple automorphisms of a
diagram that can be displayed simultaneously in the same drawing
and returns a symmetry quality metric score that measures how
faithfully the drawing displays all the automorphisms at once. Given
a drawing $D$ of a graph $G$ and a group of automorphisms $A$ of $G$,
the metric is computed using the following steps:

Algorithm 4: Algo-SQG

Input: Graph $G$, drawing $D$ of $G$, group of automorphisms $A$ of $G$
Output: $SQG$ metric score

1. For each automorphism $\phi$ in $A$, compute $SQ(\phi)$ using Algo –
   $SQ$.
2. Compute the weighted sum of $SQ(\phi)$ for all automorphisms in
   $A$, where the scores are weighted by the maximum orbit size
   of each automorphism.
3. Compute the $SQG$ using Equation $4$.

The computation of each $SQ$ score for each automorphism runs
in $O(n\log n)$ time. With $k$ as the number of automorphisms in the
input group, the $SQG$ computation as a whole runs in $O(kn\log n)$
time.

5 EXPERIMENT 1: $SQ$ METRIC VALIDATION EXPERIMENTS

5.1 Experiment Design

To validate the $SQ$ version of our metric, we perform experiments
where we take a symmetric drawing of a graph and deform it in a
way that breaks the symmetry.

We expect that not only will our metric effectively capture the
distortion from perfect symmetry induced by the deformations, but
also that it will perform better than existing approximate symmetry
detection approaches which only rely on Euclidean distance, here
represented as $sd$. We formulate the following hypotheses:

- Hypothesis 1: The symmetry quality scores $SQ_1$ and $SQ_2$ will
  both decrease as the drawing $D$ of $G$ is further deformed.
- Hypothesis 2: The symmetry quality scores $SQ_1$ and $SQ_2$ will
  reflect the extent of distortion from perfect symmetry more
  effectively than $sd$.

We performed five sets of experiments each to validate our metric
on rotational and axial symmetry. For each experiment, we perform
the following steps:

1. For a symmetric graph $G$, select $\phi$, one of its geometric auto-
   morphisms.
2. Create a drawing $D$ of $G$ that displays $\phi$ as an exact symmetry.
3. Select a subset of vertices of $G$ and perturb their position by taking them further from their initial position. Repeat until a desired number of perturbation steps is reached.

We used a mix of well-known symmetric graphs from graph theory literature and newly-generated symmetric graphs. The graphs we generated for validation experiments with rotational symmetry (titled in the format c[order]x[ # of orbits]) were generated as follows. First, we decide on the order of rotational symmetry $k$ and the number of orbits $m$, then define a graph $G$ with $k \times m$ vertices and initially zero edges. We then connect each of the $m$ sets of $k$ vertices into $m$ distinct cycles of length $k$, and, except for the “innermost” cycle, we connect them to previous cycles with a number of edges between $k$ and $2k$ such that the graph possesses the order of symmetry desired.

With graphs generated for axial symmetry, we first determine the orbits, including fixed points, create edges between vertices not in the same orbit, then “mirror” the edges by adding edges between the points which share orbits with the endpoint of the created edges.

5.2 Rotational Symmetry Experiments

![Diagram of rotational symmetry experiments](image)

Figure 2: Validation experiments detecting rotational symmetry (order 7) on Coxeter graph. Each subsequent perturbation destroys a new orbit or further perturbs an already destroyed one.

5.3 Axial Symmetry Experiments

![Diagram of axial symmetry experiments](image)

Figure 3: Symmetry quality metrics for the experiment in Figure 2. Both $SQ_1$ and $SQ_2$ captures the effect of the perturbations better than $sd$ which is only based on Euclidean distance, with $SQ_2$ capturing the effects of further perturbations better.

Figure 4: Validation experiment detecting axial symmetry on Heawood graph.
We formulate the following hypotheses:

• Hypothesis 1: The symmetry quality scores $SQG_1$ and $SQG_2$ will be higher for a drawing $D_1$ that displays as symmetries a larger subset of the automorphism group or an automorphism with a larger order than a drawing $D_2$ displaying as symmetries a smaller subset of the automorphism group or an automorphism with a smaller order.

• Hypothesis 2: The symmetry quality scores $SQG_1$ and $SQG_2$ will decrease as a drawing $D$ is further deformed.

For these experiments, we start by selecting a symmetric graph $G$ and a geometric automorphism group $A$. We then create drawings of $G$ displaying different subsets of $A$ as symmetries. In addition, we also perform deformation experiments similar to that described in Section 5.1 where we perform steps of deformation that gradually destroys more orbits and/or brings perturbed vertices further from their initial positions. We perform six sets of experiments with drawings displaying the input automorphism groups as exact symmetries and four sets of experiments with drawing deformations.

### 6.2 Cyclic Group Experiments

Figure 6 shows an example of the validation experiment for the automorphism group testing version of the metric, using a graph we created called $c12x3$ with a rotational automorphism of order 12 and 3 orbits. In this experiment, the input automorphisms are cyclic groups of order 12, 6, 4, 3, 2. In accordance, we created a set of drawings where the largest order of rotational symmetry displayed in each drawing correspond to one of the cyclic group orders in the input.

Figure 9 (a) shows the $SQG$ metrics computed for the experiment displayed in Figure 6. The largest order of rotational symmetry displayed by each drawing is lower than the previous drawing - the first drawing, $D_1$, displays rotational symmetry of order 12, while the last drawing, $D_5$, only displays rotational symmetry of order 2. It can be seen that both $SQG_1$ and $SQG_2$ computes lower scores for drawing where the largest order of symmetry displayed is smaller, in line with Hypothesis 3.

### 6.3 Dihedral Group Experiments

Figure 7 displays two more sets of validation experiments, where the input automorphisms are dihedral groups of order 10, 5, and 2 on the dodecahedral graph and dihedral groups of order 6, 3, and 2 on the cuboctahedral graph. In both examples, we start with a drawing displaying the highest order of symmetry by drawing each orbit on concentric circles. For subsequent drawings, we rotate a number of vertices in each orbit such that the drawing only shows a lower order of symmetry.

Figures (b) and (c) shows the $SQG$ scores computed for the validation experiments in Figure 7. It can be seen that for both sets of experiments, the scores computed by both $SQG_1$ and $SQG_2$ again decrease with drawings displaying lower orders of symmetry while still staying above 0.5, supporting Hypothesis 3.

### 6.4 Automorphism Group with Perturbations Experiments

Figure 8 shows an example of one automorphism group detection validation experiment with perturbations of the drawing. We start with drawings shown in Figure 6. In this case taking $D_1$ and $D_2$, then perform deformation steps by selecting an orbit, selecting a vertex from the orbit, and randomly perturbing its position such that the orbit is no longer displayed as symmetric in the drawing. Here, we show three steps of perturbation for each drawing.

Figure 9 (d) shows the $SQG$ metrics computed on the drawings. It can be seen that for each set, the computed values gradually decrease as more orbits are perturbed, supporting Hypothesis 4.
After validating the usage of the SQG metric in computing the effectiveness of the SQG metric in capturing the difference in quality of drawings displaying different subsets of an automorphism group of a graph, as well as in capturing the difference in quality when such drawings are perturbed to produce asymmetric drawings.

### 7 EXPERIMENT 3: LAYOUT COMPARISON EXPERIMENTS

#### 7.1 Experiment Design

After validating the usage of the SQG metric to compute a quantitative score measuring how well a drawing displays a group of automorphisms of a graph, we conduct experiments comparing a number of different graph layout algorithms. We select the following automatic layouts to be compared: Fruchterman-Reingold (FR) [10], Stress Majorization [11], Pivot MDS [4], spectral, and Tutte. We also produce drawings using the Concentric Circles layout [1], where we create concentric circles according to the number of orbits of a selected automorphism, assign each orbit to a circle, and place the vertices belonging to the orbit in a regular convex polygon position around the circle.

We perform the experiments using the following steps:

1. We select a symmetric graph G and draw it using all of the selected layout algorithms.
2. We define a group of automorphisms A of G as the input.
3. We compute the SQG metrics for the drawings produced by each layout algorithm with A as the input automorphism group.

In Step[1], we generate one drawing each using each layout algorithm, except for FR, where we generate five drawings per graph, each time starting from a random initial layout. This is due to the non-deterministic nature of FR, compared to the other selected layout algorithms. We then average the SQG metrics computed for all the FR-generated drawings to obtain a value that can be compared to those computed for other layout algorithms.

Based on how the Concentric Circles layout places vertices in regular convex polygon positions along concentric circles, we expect that this layout will perform well in displaying automorphisms as geometric symmetry. We also expect that the Tutte layout, due to being designed to display planar graphs, will perform well in displaying planar automorphisms, which are automorphisms that can be displayed in planar graph drawings. We formulate the following hypothesis for this experiment:

- Hypothesis 5: The Concentric Circle layout will always attain SQG metrics of 1 and Tutte will always display planar automorphisms when the graph has planar automorphisms.

#### 7.2 Layout Comparison Results

Figure[1] shows an example of the layout comparison experiment using the dodecahedral graph. In this experiment, we use the SQG...
metric with the input dihedral groups of order 10, 5, and 2. Figure 12 displays the symmetry quality scores computed for this experiment. Concentric Circles obtains a metric score of 1, supporting Hypothesis 5. Stress Majorization also obtains a metric score of 1, and behind the two, Tutte obtains the third highest metric score.

A similar result is shown in the examples using the tesseract graph in Figures 13 and 14 where we take as input dihedral groups of order 8, 4, and 2. Concentric Circles and Stress Majorization again obtain $SQG$ values of 1, with Tutte the third highest.

Figure 15 shows another example of the layout comparison experiment, taking as input dihedral group of order 5 on the Petersen graph. Figure 16 shows the $SQG_1$ and $SQG_2$ computed for this experiment. Although the automorphism is not planar, Tutte obtains $SQG$ of one, the same as Concentric Circles, while all other layouts obtain $SQG$ of lower than 0.3.

### 7.3 Discussion and Summary

The results for layout comparisons, as summarized in the averages over nine graphs shown in Figure 10, show that the Concentric Circles, Tutte, and Stress Majorization layouts obtain the top 3 values for the $SQG$ metric. The results for Concentric Circles supports Hypothesis 5.

The results for the Tutte layout may arise from the nature of the layout algorithm, where given a set of vertices forming the outer face fixed as a regular convex polygon, all other vertices are placed at the barycenter of its neighbors. This results in a unique embedding that minimizes a linear system of equations, which could lead to its ability to capture automorphisms of the graph as symmetries. However, as the layout is designed for planar graphs, the results favors planar automorphisms. This causes it to score lower on examples such as the dodecahedral graph in Figure 11 where it displays a dihedral automorphism of order 5, compared to Stress Majorization which displays the (non-planar) dihedral automorphism of order 10.

With Stress Majorization, it has been shown that it is possible to obtain symmetric layouts with low stress starting from the Concentric Circles layout [9]. Unlike force-directed layouts such as FR, the set of minima for stress-based layouts is more limited, increasing the chances of producing a symmetric layout, which can explain its superior performance compared to FR.

In summary, our experiments have supported hypothesis 5 by showing that the concentric circles layout always obtains a score of 1 on $SQG$ and that Tutte layout is always able to display the highest order of planar automorphism when a graph possesses planar automorphisms. We also observe that Stress Majorization performs better at displaying graph automorphisms as symmetries compared to the remaining tested layouts: FR, Pivot MDS, and Spectral.

### 8 Conclusion

We have introduced a new quality metric to measure how faithfully a graph drawing visualizes the graph’s automorphisms as symmetries, based on both Euclidean distance and mathematical group theory. The metrics, $SQ$ and $SQG$, are suited to detect symmetries corresponding to rotational, reflectional, and dihedral automorphisms.

We defined algorithms to compute the metrics, running in $O(n \log n)$ time for the $SQ$ version taking as input a single automorphism and in $O(kn \log n)$ time for the $SQG$ version taking as input an automorphism group, with $n$ as the number of vertices and $k$ as the size of the automorphism group.

Experiments have validated the effectiveness of both $SQ$ and $SQG$ in capturing the extent to which the drawing’s symmetries reflect the automorphisms of the graph. We demonstrated the effectiveness of $SQ$ in reflecting the extent of distortion of a drawing from exact symmetry, and the effectiveness of $SQG$ in reflecting the difference in quality between drawings displaying different automorphism groups of a graph.

We have also compared layout algorithms using our metric, and in the process, confirm the effectiveness of the concentric circles layout in displaying a graph’s automorphisms as symmetries and similarly for the Tutte layout for planar automorphisms.

Future work may include extending the metric to graph drawings in three dimensions.
Figure 10: Average SQG metrics computed for all nine layout comparison graphs. Concentric Circles, Tutte, and Stress Majorization obtain the three highest scores.

Figure 11: Layout comparison experiment with the dodecahedral graph, detecting dihedral symmetries of order 10, 5, and 2. Among automatic layouts, Stress Majorization realizes all of the input automorphisms, while Tutte displays the largest planar automorphism as dihedral symmetry of order 5.

Figure 12: SQG metrics computed for the layout comparison experiment with the dodecahedral graph (Figure 11). Concentric Circles and Stress Majorization obtain scores of 1, while Tutte, which displays a planar automorphism of a lower order, obtains lower scores than the two.

Figure 13: Layout comparison experiment with the tesseract graph, detecting dihedral symmetries of order 8, 4, and 2. Similar to the dodecahedral graph in Figure 11, Stress Majorization displays the highest order of automorphism (order 8) while Tutte realizes the highest order of planar automorphism with dihedral symmetry of order 4.
Figure 14: SQG metrics computed for the layout comparison experiment with the tesseract graph (Figure 13). Concentric Circles and Stress Majorization, displaying the highest order of automorphism, obtains scores of 1.

Figure 15: Layout comparison experiment with the Petersen graph, with input dihedral group of order 5. Aside from concentric circles, Tutte is the only layout that perfectly displays the automorphism group as symmetries.

Figure 16: SQG metrics computed for the layout comparison experiment with the Petersen graph (Figure 15). Concentric Circles and Tutte obtain scores of 1, while all other layouts obtain scores of less than 0.3.

**REFERENCES**

[1] D. Abelson, S.-H. Hong, and D. E. Taylor. Geometric automorphism groups of graphs. *Discrete Applied Mathematics*, 155(17):2211–2226, 2007.

[2] H. Alt, K. Mehlhorn, H. Wagener, and E. Welzl. Congruence, similarity, and symmetries of geometric objects. *Discrete & Computational Geometry*, 3(3):237–256, 1988.

[3] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph drawing: algorithms for the visualization of graphs*. Prentice Hall PTR, 1998.

[4] U. Brandes and C. Pich. Eigensolver methods for progressive multidimensional scaling of large data. In *International Symposium on Graph Drawing*, pp. 42–53. Springer, 2006.

[5] C. Buchheim and M. Jünger. Detecting symmetries by branch & cut. *Mathematical Programming*, 98(1-3):369–384, 2003.

[6] H. Coxeter. My graph. *Proceedings of the London Mathematical Society*, 3(1):117–136, 1983.

[7] H. De Fraysseix. An heuristic for graph symmetry detection. In *International Symposium on Graph Drawing*, pp. 276–285. Springer, 1999.

[8] P. Eades and S. Hong. Detection and display of symmetries. *Handbook of Graph Drawing and Visualisation*. CRC Press, Boca Raton, FL, to appear, 2013.

[9] P. Eades and X. Lin. Spring algorithms and symmetry. *Theoretical Computer Science*, 240(2):379–405, 2000.

[10] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991. doi: 10.1002/spe.4380211102

[11] E. R. Gansner, Y. Koren, and S. North. Graph drawing by stress majorization. In J. Pach, ed., *International Symposium on Graph Drawing*, pp. 239–250. Springer Berlin Heidelberg, 2004.

[12] S.-H. Hong and P. Eades. Symmetric layout of disconnected graphs. In *International Symposium on Algorithms and Computation*, pp. 405–414. Springer, 2003.

[13] S.-H. Hong and P. Eades. Drawing planar graphs symmetrically, ii: Biconnected planar graphs. *Algorithmica*, 42(2):159–197, 2005.

[14] S.-H. Hong and P. Eades. Drawing planar graphs symmetrically, iii: Oneconnected planar graphs. *Algorithmica*, 44(1):67–100, 2006.

[15] S.-H. Hong, P. Eades, and S.-H. Lee. Drawing series parallel digraphs symmetrically. *Computational Geometry*, 17(3-4):165–188, 2000.

[16] S.-H. Hong, B. McKay, and P. Eades. A linear time algorithm for constructing maximally symmetric straight line drawings of triconnected planar graphs. *Discrete & Computational Geometry*, 36(2):283–311, 2006.

[17] R. Klapaukh, S. Marshall, and D. Pearce. A symmetry metric for graphs and line diagrams. In P. Chapman, G. Stapleton, A. Moktefi, S. Perez-Kriz, and F. Bellucci, eds., *Diagrammatic Representation and Inference*, pp. 739–742. Springer International Publishing, Cham, 2018.
[18] R. J. Lipton, S. C. North, and J. S. Sandberg. A method for drawing graphs. In Proceedings of the first annual symposium on Computational geometry, pp. 153–160. ACM, 1985.

[19] A. Lubiw. Some np-complete problems similar to graph isomorphism. SIAM Journal on Computing, 10(1):11–21, 1981.

[20] F. D. Luca, M. I. Hossain, and S. G. Kobourov. Symmetry detection and classification in drawings of graphs. CoRR, abs/1907.01004, 2019.

[21] J. Manning and M. J. Atallah. Fast detection and display of symmetry in trees. Congressus Numerantium, 64:159–169, 1988.

[22] J. Manning and M. J. Atallah. Fast detection and display of symmetry in outerplanar graphs. Discrete Applied Mathematics, 39(1):13–35, 1992.

[23] J. B. Manning. Geometric Symmetry in Graphs. PhD thesis, West Lafayette, IN, USA, 1991. UMI Order No. GAX91-16430.

[24] H. C. Purchase. Metrics for graph drawing aesthetics. Journal of Visual Languages & Computing, 13(5):501–516, 2002.

[25] J. D. Wolter, T. C. Woo, and R. A. Volz. Optimal algorithms for symmetry detection in two and three dimensions. The Visual Computer, 1(1):37–48, 1985.

[26] H. Zabrodsky and D. Avnir. Continuous symmetry measures. 4. chirality. Journal of the American Chemical Society, 117(1):462–473, 1995.

[27] H. Zabrodsky, S. Peleg, and D. Avnir. Symmetry as a continuous feature. IEEE Transactions on Pattern Analysis & Machine Intelligence, (12):1154–1166, 1995.