The Galois Complexity of Graph Drawing: Why Numerical Solutions Are Ubiquitous for Force-Directed, Spectral, and Circle Packing Drawings*

Michael J. Bannister, William E. Devanny,
David Eppstein, and Michael T. Goodrich

Department of Computer Science, University of California, Irvine

Abstract. Many well-known graph drawing techniques, including force directed drawings, spectral graph layouts, multidimensional scaling, and circle packings, have algebraic formulations. However, practical methods for producing such drawings ubiquitously use iterative numerical approximations rather than constructing and then solving algebraic expressions representing their exact solutions. To explain this phenomenon, we use Galois theory to show that many variants of these problems have solutions that cannot be expressed by nested radicals or nested roots of low-degree polynomials. Hence, such solutions cannot be computed exactly even in extended computational models that include such operations.

1 Introduction

One of the most powerful paradigms for drawing a graph is to construct an algebraic formulation for a suitably-defined optimal drawing of the graph and then solve this formulation to produce a drawing. Examples of this *algebraic graph drawing* approach include the force-directed, spectral, multidimensional scaling, and circle packing drawing techniques (which we review in the full version of the paper for readers unfamiliar with them).

Even though this paradigm starts from an algebraic formulation, the ubiquitous method for solving such formulations is to approximately optimize them numerically in an iterative fashion. That is, with a few exceptions for linear systems [1–3], approximate numerical solutions for algebraic graph drawing are overwhelmingly preferred over exact symbolic solutions. It is therefore natural to ask if this preference for numerical solutions over symbolic solutions is inherent in algebraic graph drawing or due to some other phenomena, such as laziness or lack of mathematical sophistication on the part of those who are producing the algebraic formulations.

In this paper, we introduce a framework for deciding whether certain algebraic graph drawing formulations have symbolic solutions, and we show that exact symbolic solutions are, in fact, impossible in several algebraic computation models, for some simple examples of common algebraic graph drawing formulations, including force-directed

* This research was supported in part by ONR MURI grant N00014-08-1-1015 and NSF grants 1217322, 1011840, and 1228639.

C. Duncan and A. Symvonis (Eds.): GD 2014, LNCS 8871, pp. 149–161, 2014.

graph drawings (in both the Fruchterman–Reingold [4] and Kamada–Kawai [5] approaches), spectral graph drawings [6], classical multidimensional scaling [7], and circle packings [8]. Note that these impossibility results go beyond saying that such symbolic solutions are computationally infeasible or undecidable to find—instead, we show that such solutions do not exist.

To prove our results, we use Galois theory, a connection between the theories of algebraic numbers and abstract groups. Two classical applications of Galois theory use it to prove the impossibility of the classical Greek problem of doubling the cube using compass and straightedge, and of solving fifth-degree polynomials by nested radicals. In our terms, these results concern quadratic computation trees and radical computation trees, respectively. Our proofs build on this theory by applying Galois theory to the algebraic numbers given by the vertex positions in different types of graph drawings. For force-directed and spectral drawing, we find small graphs (in one case as small as a length-three path) whose drawings directly generate unsolvable Galois groups. For circle packing, an additional argument involving the compass and straightedge constructibility of Möbius transformations allows us to transform arbitrary circle packings into a canonical form with two concentric circles, whose construction is equivalent to the calculation of certain algebraic numbers. Because of this mathematical foundation, we refer to this topic as the *Galois complexity* of graph drawing.

Related Work. The problems for which Galois theory has been used to prove unsolvability in simple algebraic computational models include shortest paths around polyhedral obstacles [9], shortest paths through weighted regions of the plane [10], the geometric median of planar points [11], computing structure from motion in computer vision [12], and finding polygons of maximal area with specified edge lengths [13]. In each of these cases, the non-existence of a nested radical formula for the solution is established by finding a Galois group containing a symmetric group of constant degree at least five. In our terminology, this shows that these problems cannot be solved by a radical computation tree. We are not aware of any previous non-constant lower bounds on the degree of the polynomial roots needed to solve a problem, comparable to our new bounds using the root computation tree model. Brightwell and Scheinerman [14] show that some circle packing graph representations cannot be constructed by compass and straightedge (what we call the quadratic computation tree model).

2 Preliminaries

Models of Computation. We define models of computation based on the *algebraic computation tree* [15, 16], in which each node computes a value or makes a decision using standard arithmetic functions of previously computed values. Specifically, we define the following variant models:

- A *quadratic computation tree* is an algebraic computation tree in which the set of allowable functions for each computation node is augmented with square roots and complex conjugation. These trees capture the geometric constructions that can be performed by compass and unmarked straightedge.

- A *radical computation tree* is an algebraic computation tree in which the set of allowable functions is augmented with the k^{th} root operation, where k is an integer parameter to the operation, and with complex conjugation. These trees capture the calculations whose results can be expressed as nested radicals.
- A *root computation tree* is an algebraic computation tree in which the allowable functions include the ability to find complex roots of polynomials whose coefficients are integers or previously computed values, and to compute complex conjugates of previously computed values. For instance, this model can compute any algebraic number. As a measure of complexity in this model, we define the *degree* of a root computation tree as the maximum degree of any of its polynomials. A *bounded-degree root computation tree* has its degree bounded by some constant unrelated to the size of its input. Thus, a quadratic computation tree is exactly a bounded-degree root computation tree (of degree two).

Our impossibility results and degree lower bounds for these models imply the same results for algorithms in more realistic models of computation that use as a black box the corresponding primitives for constructing and representing algebraic numbers in symbolic computation systems. Because our results are lower bounds, they also apply *a fortiori* to weaker primitives, such as systems limited to *real* algebraic numbers, which don't include complex conjugation.

It is important to note that each of the above models can generate algebraic numbers of unbounded degree. For instance, even the quadratic computation tree (compass and straightedge model) can construct regular 2^k-gons, whose coordinates are algebraic numbers with degrees that are high powers of two. Thus, to prove lower bounds and impossibility results in these models, it is not sufficient to prove that a problem is described by a high-degree polynomial; additional structure is needed.

Algebraic Graph Theory. In algebraic graph theory, the properties of a graph are examined via the spectra of several matrices associated with the graph. The *adjacency matrix* $A = \mathrm{adj}(G)$ of a graph G is the $n \times n$ matrix with $A_{i,j}$ equal to 1 if there is an edge between i and j and 0 otherwise. The *degree matrix* $D = \deg(G)$ of G is the $n \times n$ matrix with $D_{i,i} = \deg(v_i)$. From these two matrices we define the *Laplacian matrix*, $L = \mathrm{lap}(G) = D - A$, and the *transition matrix*, $T = \mathrm{tran}(G) = D^{-1}A$.

Lemma 1. *For a regular graph G, $\mathrm{adj}(G)$, $\mathrm{lap}(G)$, and $\mathrm{tran}(G)$ have the same set of eigenvectors.*

Lemma 2. *For the cycle on n vertices, the eigenvalues of $\mathrm{adj}(G)$ are $2\cos(2\pi k/n)$, for $0 \leq k < n$.*

Möbius Transformations. We may represent each point p in the plane by a complex number, z, whose real part represents p's x coordinate and whose imaginary part represents p's y coordinate. A *Möbius transformation* is a fractional linear transformation, $z \mapsto (az + b)/(cz + d)$, defined by a 4-tuple (a, b, c, d) of complex numbers, or the complex conjugate of such a transformation. We prove the following in the full version of the paper.

Lemma 3. *Given any two disjoint circles, a Möbius transformation mapping them to two concentric circles can be constructed using a quadratic computation tree.*

Number Theory. The *Euler totient function*, $\phi(n)$, counts the number of integers in the interval $[1, n-1]$ that are relatively prime to n. It can be calculated from the prime factorization $n = \prod p_i^{r_i}$ by the formula

$$\phi(n) = \prod p_i^{r_i - 1}(p_i - 1).$$

A *Sophie Germain prime* is a prime number p such that $2p + 1$ is also prime [17]. It has been conjectured that there are infinitely many of them, but the conjecture remains unsolved. The significance of these primes for us is that, when p is a Sophie Germain prime, $\phi(2p + 1)$ has the large prime factor p. An easy construction gives a number n for which $\phi(n)$ has a prime factor of size $\Omega(\sqrt{n})$: simply let $n = p^2$ for a prime p, with $\phi(n) = p(p - 1)$. Baker and Harman [18] proved the following stronger bound.

Lemma 4 (Baker and Harman [18]). *For infinitely many prime numbers p, the largest prime factor of $\phi(p)$ is at least $p^{0.677}$.*

Field Theory. A *field* is a system of values and arithmetic operations over them (addition, subtraction, multiplication, and division) obeying similar axioms to those of rational arithmetic, real number arithmetic, and complex number arithmetic: addition and multiplication are commutative and associative, multiplication distributes over addition, subtraction is inverse to addition, and division is inverse to multiplication by any value except zero. A field K is an *extension* of a field F, and F is a *subfield* of K (the *base field*), if the elements of F are a subset of those of K and the two fields' operations coincide for those values. K can be viewed as a vector space over F (values in K can be added to each other and multiplied by values in F) and the *degree* $[K : F]$ of the extension is its dimension as a vector space. For an element α of K the notation $F(\alpha)$ represents the set of values that can be obtained from rational functions (ratios of univariate polynomials) with coefficients in F by plugging in α as the value of the variable. $F(\alpha)$ is itself a field, intermediate between F and K. In particular, we will frequently consider field extensions $\mathbf{Q}(\alpha)$ where $\mathbf{Q}$ is the field of rational numbers and α is an *algebraic number*, the complex root of a polynomial with rational coefficients.

Lemma 5. *If α can be computed by a root computation tree of degree $f(n)$, then $[\mathbf{Q}(\alpha) : \mathbf{Q}]$ is $f(n)$-smooth, i.e., it has no prime factor $> f(n)$. In particular, if α can be computed by a quadratic computation tree, then $[\mathbf{Q}(\alpha) : \mathbf{Q}]$ is a power of two.*

Proof. See the full version of the paper. □

A *primitive root of unity* ζ_n is a root of $x^n - 1$ whose powers give all other roots of the same polynomial. As a complex number we can take $\zeta_n = \exp(2i\pi/n)$.

Lemma 6 (Corollary 9.1.10 of [19], p. 235). $[\mathbf{Q}(\zeta_n) : \mathbf{Q}] = \phi(n)$.

Galois Theory. A *group* is a system of values and a single operation (written as multiplication) that is associative and in which every element has an inverse. The set of permutations of the set $[n] = \{1, 2, \ldots, n\}$, multiplied by function composition, is a standard example of a group and is denoted by S_n. A *permutation group* is a subgroup of S_n; i.e., it is a set of permutations that is closed under the group operation.

A *field automorphism* of the field F is a bijection $\sigma : F \to F$ that respects the field operations, i.e., $\sigma(xy) = \sigma(x)\sigma(y)$ and $\sigma(x + y) = \sigma(x) + \sigma(y)$. The set of all field automorphism of a field F forms a group denoted by $\mathrm{Aut}(F)$. Given a field extension K of F, the subset of $\mathrm{Aut}(K)$ that leaves F unchanged is itself a group, called the *Galois group* of the extension, and is denoted

$$\mathrm{Gal}(K/F) = \{\sigma \in \mathrm{Aut}(K) \mid \sigma(x) = x \text{ for all } x \in F\}.$$

The *splitting field* of a polynomial, p, with rational coefficients, denoted $\mathrm{split}(p)$ is the smallest subfield of the complex numbers that contains all the roots of the polynomial. Each automorphism in $\mathrm{Gal}(\mathrm{split}(p)/\mathbf{Q})$ permutes the roots of the polynomial, no two automorphisms permute the roots in the same way, and these permutations form a group, so $\mathrm{Gal}(\mathrm{split}(p)/\mathbf{Q})$ can be thought of as a permutation group.

Lemma 7. *If α can be computed by a radical computation tree and K is the splitting field of an irreducible polynomial with α as one of its roots, then $\mathrm{Gal}(K/\mathbf{Q})$ does not contain S_n as a subgroup for any $n \geq 5$.*

Proof. If α is computable by a radical computation tree, it can be written as an expression using nested radicals. If K is the splitting field of an irreducible polynomial with such an expression as a root, $\mathrm{Gal}(K/\mathbf{Q})$ is a solvable group (Def. 8.1.1 of [19], p. 191 and Theorem 8.3.3 of [19], p. 204). But S_n is not solvable for $n \geq 5$ (Theorem 8.4.5 of [19], p. 213), and every subgroup of a solvable group is solvable (Proposition 8.1.3 of [19], p. 192). Thus, $\mathrm{Gal}(K/\mathbf{Q})$ cannot contain S_n ($n \geq 5$) as a subgroup. □

The next lemma allows us to infer properties of a Galois group from the coefficients of a *monic* polynomial, that is, a polynomial with integer coefficients whose first coefficient is one. The *discriminant* of a monic polynomial is (up to sign) the product of the squared differences of all pairs of its roots; it can also be computed as a polynomial function of the coefficients. The lemma is due to Dedekind and proven in [19].

Lemma 8 (Dedekind's theorem). *Let $f(x)$ be an irreducible monic polynomial in $\mathbf{Z}[x]$ and p a prime not dividing the discriminant of f. If $f(x)$ factors into a product of irreducibles of degrees $d_0, d_1, \ldots d_r$ over $\mathbf{Z}/p\mathbf{Z}$, then $\mathrm{Gal}(\mathrm{split}(f)/\mathbf{Q})$ contains a permutation that is the composition of disjoint cycles of lengths $d_0, d_1, \ldots, d_r$.*

A permutation group is *transitive* if, for every two elements x and y of the elements being permuted, the group includes a permutation that maps x to y. If K is the splitting field of an irreducible polynomial of degree n, then $\mathrm{Gal}(K/\mathbf{Q})$ (viewed as a permutation group on the roots) is necessarily transitive. The next lemma allows us to use Dedekind's theorem to prove that $\mathrm{Gal}(K/\mathbf{Q})$ equals S_n. It is a standard exercise in abstract algebra (e.g., [20], Exercise 3, p. 305).

Lemma 9. *If a transitive subgroup G of S_n contains a transposition and an $(n-1)$-cycle, then $G = S_n$.*

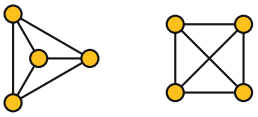

Fig. 1. Two stable drawings of K_4.

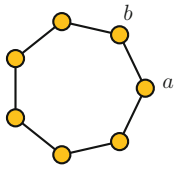

Fig. 2. A drawing whose coordinates cannot be computed by a quadratic computation tree.

3 Impossibility Results for Force Directed Graph Drawing

In the Fruchterman and Reingold [4] force-directed model, each vertex is pulled toward its neighbors with an attractive force, $f_a(d) = d^2/k$, and pushed away from all vertices with a repulsive force, $f_r(d) = k^2/d$. The parameter k is a constant that sets the scale of the drawing, and d is the distance between vertices. We say that a drawing is a Fruchterman and Reingold equilibrium when the total force at each vertex is zero.

In the Kamada and Kawai [5] force-directed model, every two vertices are connected by a spring with rest length and spring constant determined by the structure of the graph. The total energy of the graph is defined to be

$$E = \sum_i \sum_{j>i} \frac{1}{2} k_{ij} \big(\operatorname{dist}(p_i, p_j) - \ell_{ij} \big)^2,$$

where p_i = position of vertex v_i, d_{ij} = graph theoretic distance between v_i and v_j, L = a scaling constant, $\ell_{ij} = L d_{ij}$, K = a scaling constant, and $k_{ij} = K/d_{i,j}^2$. We say that a drawing is a Kamada–Kawai equilibrium if E is at a local minimum. The necessary conditions for such a local minimum are as follows:

$$\frac{\partial E}{\partial x_j} = \sum_{i \neq j} k_{ji}(x_j - x_i) \left(1 - \frac{\ell_{ji}}{\operatorname{dist}(p_j, p_i)} \right) = 0 \qquad 1 \leq j \leq n$$

$$\frac{\partial E}{\partial y_j} = \sum_{i \neq j} k_{ji}(y_j - y_i) \left(1 - \frac{\ell_{ji}}{\operatorname{dist}(p_j, p_i)} \right) = 0 \qquad 1 \leq j \leq n.$$

For either of these approaches to force-directed graph drawing, a graph can have multiple equilibria (Figure 1). In such cases, typically, one equilibrium is the "expected" drawing of the graph and others represent undesired drawings that are not likely to be found by the drawing algorithm. To make the positions of the vertices in this drawing concrete, we assume that the constants k (Fruchterman–Reingold), L, and K (Kamada–Kawai) are all equal to 1. As we will demonstrate, there exist graphs whose expected drawings cannot be constructed in our models of computation. Interestingly, the graphs we use for these results are not complicated configurations unlikely to arise in practice, but are instead graphs so simple that they might at first be dismissed as insufficiently challenging even to be used for debugging purposes.

Root Computation Trees. Consider the cycle C_n with n vertices. When drawn with force directed algorithms, either Fruchterman and Reingold or Kamada and Kawai, the embedding typically places all vertices equally spaced on a circle, such that neighbors are placed next to each other, as shown in Figure 2. As an easy warm-up to our main results, we observe that this is not always possible using a quadratic computation tree.

Theorem 1. *There exist a graph with seven vertices such that it is not possible in a quadratic computation tree to compute the coordinates of every possible Fruchterman and Reingold equilibrium or every possible Kamada and Kawai equilibrium.*

Proof. Let G be the cycle C_7 on seven vertices. Both algorithms have the embedding shown in Figure 2 (suitably scaled) as an equilibrium. In this embedding let a and b be two neighboring vertices and α and β their corresponding complex coordinates. Then α/β is equal to $\pm\zeta_7$ the seventh root of unity. By Lemma 6

$$[\mathbf{Q}(\zeta_7) : \mathbf{Q}] = \phi(7) = 6.$$

Since 6 is not a power of two, Lemma 5 implies that ζ_7 cannot be constructed by a quadratic computation tree. Therefore, neither can this embedding. □

Theorem 2. *For arbitrarily large values of n, there are graphs on n vertices such that constructing the coordinates of all Fruchterman and Reingold equilibria on a root computation tree requires degree $\Omega(n^{0.677})$. If there exists infinitely many Sophie Germain primes, then there are graphs for which computing the coordinates of any Fruchterman and Reingold equilibria requires degree $\Omega(n)$. The same results with the same graphs hold for Kamada and Kawai equilibria.*

Proof. As in the previous theorem we consider embedding cycles with their canonical embedding, which is an equilibrium for both algorithms. The same argument used in the previous theorem shows we can construct ζ_n from the coordinates of the canonical embedding of the cycle on n vertices.

We consider cycles with p vertices where p is a prime number for which $\phi(p) = p-1$ has a large prime factor q. If arbitrarily large Sophie Germain primes exist we let q be such a prime and let $p = 2q + 1$. Otherwise, by Lemma 4 we choose p in such a way that its largest prime factor q is at least $p^{0.677}$. Now, by Lemma 6 we have:

$$[\mathbf{Q}(\zeta_p) : \mathbf{Q}] = \phi(p) = p - 1.$$

This extension is not D-smooth for any D smaller than q, and therefore every construction of it on a root computation tree requires degree at least q. □

Thus, such drawings are not possible on a bounded-degree root computation tree.

Radical Computation Trees. To show that the coordinates of a Fruchterman and Reingold equilibrium are in general not computable with a radical computation tree we consider embedding the path with three edges, shown in Figure 3. We assume that all of the vertices are embedded colinearly and without edge or vertex overlaps. These assumptions correspond to the equilibrium that is typically produced by the Fruchterman and Reingold algorithm.

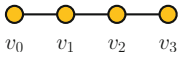

Fig. 3. A graph whose Fruchterman–Reingold coordinates cannot be computed by a radical computation tree.

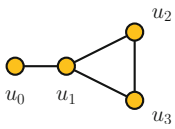

Fig. 4. A graph whose Kamada–Kawai coordinates cannot be computed by a radical computation tree.

Let $a > 0$ be the distance from v_0 to v_1 (equal by symmetry to the distance from v_2 to v_3) and let $b > 0$ be the distance from v_1 to v_2. We can then express the sum of all the forces at vertex v_0 by the equation

$$F_0 = a^2 - \frac{1}{a} - \frac{1}{a+b} - \frac{1}{2a+b} = \frac{2a^5 + 3a^4b + a^3b^2 - 5a^2 - 5ab - b^2}{2a^3 + 3a^2b + ab^2},$$

and the sum of all the forces at vertex v_1 by the equation

$$F_1 = -a^2 + \frac{1}{a} + b^2 - \frac{1}{b} - \frac{1}{a+b} = \frac{-a^4b - a^3b^2 + a^2b^3 - a^2 + ab^4 - ab + b^2}{a^2b + ab^2}.$$

In an equilibrium state we have $F_1 = F_2 = 0$. Equivalently, the numerator p of F_1 and the numerator q of F_2 are both zero, where

$$p(a,b) = 2a^5 + 3a^4b + a^3b^2 - 5a^2 - 5ab - b^2 = 0$$
$$q(a,b) = -a^4b - a^3b^2 + a^2b^3 - a^2 + ab^4 - ab + b^2 = 0.$$

To solve this system of two equations and two unknowns we can eliminate variable a and produce the following polynomial, shown as a product of irreducible polynomials, whose roots give the values of b that lead to a solution.

$$\frac{1}{3}b^2(3b^{15} - 48b^{12} + 336b^9 - 1196b^6 + 1440b^3 + 144).$$

The factor b^2 corresponds to degenerate drawings and may safely be eliminated. Let f be the degree-fifteen factor; then $f(x) = g(x^3)$ for a quintic polynomial g. A radical computation tree can compute the roots of f from the roots of g, so we need only show that the roots of g cannot be computed in a radical computation tree. To do this, we convert g to a monic polynomial h with the same splitting field, via the transformation

$$h(x) = \frac{x^5}{144}g(6/x) = x^5 + 60x^4 - 299x^3 + 504x^2 - 432x + 162.$$

The polynomial h can be shown to be irreducible by manually verifying that it has no linear or quadratic factors. Its discriminant is $-2^6 \cdot 3^9 \cdot 2341^2 \cdot 2749$, and h factors modulo primes 5 and 7 (which do not divide the discriminant) into irreducibles:

$$h(x) \equiv (x+1)(x^4 + 3x^3 + 6x^2 + x + 1) \pmod 7$$
$$h(x) \equiv (x^2 + 3x + 4)(x^3 + 2x^2 + x + 3) \pmod 5.$$

By Dedekind's theorem, the factorization modulo 7 implies the existence of a 4-cycle in $\mathrm{Gal}(\mathrm{split}(h)/\mathbf{Q})$, and the factorization modulo 5 implies the existence of a permutation that is the composition of a transposition and a 3-cycle. Raising the second permutation to the power 3 yields a transposition. By Lemma 9, $\mathrm{Gal}(\mathrm{split}(h)/\mathbf{Q}) = S_5$. So by Lemma 7 the value of b cannot be computed by a radical computation tree. Thus, we cannot compute the equilibrium coordinates of the path with three edges under the assumptions that the vertices are collinear and there are no vertex or edge overlaps.

Theorem 3. *There exists a graph on four vertices such that it is not possible on a radical computation tree to construct the coordinates of every possible Fruchterman and Reingold equilibrium.*

To show that the coordinates of a Kamada and Kawai equilibrium are in general not computable with a radical computation tree we consider the graph depicted in Figure 4.

Theorem 4. *There exists a graph on four vertices such that it is not possible on a radical computation tree to construct the coordinates of every possible Kamada and Kawai equilibrium.*

Proof. See the full version of the paper. □

4 Impossibility Results for Spectral Graph Drawing

Root computation trees. We begin with the following result for root computation trees.

Theorem 5. *For arbitrarily large values of n, there are graphs on n vertices such that constructing spectral graph drawings based on the adjacency, Laplacian, relaxed Laplacian, or transition matrix requires a root computation tree of degree $\Omega(n^{0.677})$. If there exist infinitely many Sophie Germain primes, then there are graphs for which computing these drawings requires degree $\Omega(n)$.*

Proof. Since all of the referenced matrices have rational entries, it suffices to consider the computability of their eigenvalues. Further, if we restrict our attention to regular graphs it suffices to consider the eigenvalues of just the adjacency matrix, $M = \mathrm{adj}(G)$, by Lemma 1. Let p be a prime and G the cycle on p vertices. By Lemma 2 the eigenvalues of $A = \mathrm{adj}(G)$ are given by $2\cos(2\pi k/p)$ for $0 \leq k \leq p-1$. In a root computation tree of degree at least 2 the primitive root of unity $\zeta_p = \exp(2i\pi/p)$ can be computed from $2\cos(2\pi k/p)$ for all $k \neq 0$. Therefore, from the proof of Theorem 2, for arbitrarily large n, there are graphs on n vertices such that M has one rational eigenvector (for $k = 0$) and the computation of any other eigenvector on a root computation tree requires degree $\Omega(n^{0.667})$. If infinitely many Sophie Germain primes exist, there are graphs for which computing these eigenvectors requires degree $\Omega(n)$. □

Thus, such drawings are not possible on a bounded-degree root computation tree.

Radical Computation Trees. To show that in general the eigenvectors associated with a graph are not constructible with a radical tree we consider the graph, Y, on nine vertices in Figure 5 for the Laplacian and relaxed Laplacian matrices, and in the

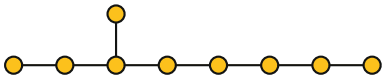

Fig. 5. A graph Y whose Laplacian eigenvectors are uncomputable by a radical tree

full version of the paper we consider another graph for the adjacency and transition matrices.

The characteristic polynomial, $p(x) = \det(M - xI)$, for the Laplacian matrix for Y, can be computed to be

$$\begin{aligned} p(x) &= \mathrm{char}(\mathrm{lap}(Y)) \\ &= x(x^8 - 16x^7 + 104x^6 - 354x^5 + 678x^4 - 730x^3 + 417x^2 - 110x + 9). \end{aligned}$$

Lemma 10 (Stäckel [21]). *If $f(x)$ is a polynomial of degree n with integer coefficients and $|f(k)|$ is prime for $2n + 1$ values of k, then $f(x)$ is irreducible.*

Let $q = p(x)/x$. The polynomial q is irreducible by Lemma 10, as it produces a prime number for 17 integer inputs from 0 to 90. The discriminant of q is $2^8 \cdot 9931583$ and we have the following factorizations of q modulo the primes 31 and 41.

$$\begin{aligned} p_1(x) &\equiv (x + 27)(x^7 + 19x^6 + 25x^5 + 25x^4 + 3x^3 + 26x^2 + 25x + 21) \pmod{31} \\ p_1(x) &\equiv (x + 1)(x^2 + 15x + 39)(x^5 + 9x^4 + 29x^3 + 10x^2 + 36x + 16) \pmod{41}. \end{aligned}$$

By Dedekind's theorem, the factorization modulo 31 implies the existence of a 7-cycle, and the factorization modulo 41 implies the existence of a permutation that is the composition of a transposition and a 5-cycle. The second permutation raised to the fifth power produces a transposition. Thus, Lemma 9 implies $\mathrm{Gal}(\mathrm{split}(p_1)/\mathbf{Q}) = S_8$. So by Lemma 7 the only eigenvalue of $\mathrm{lap}(Y)$ computable in a radical computation tree is 0. For the relaxed Laplacian we consider the two variable polynomial $f(x, \rho) = \mathrm{char}(\mathrm{lap}_\rho(Y))$. Since setting ρ equal to 1 produces a polynomial with Galois group S_8, Hilbert's irreducibility theorem tells us that the set of ρ for which the Galois group of $f(x, \rho)$ is S_8 is dense in $\mathbf{Q}$.

Theorem 6. *There exists a graph on nine vertices such that it is not possible to construct a spectral graph drawing based on the Laplacian matrix in a radical computation tree. For this graph there exists a dense subset A of $\mathbf{Q}$ such that it is not possible to construct a spectral graph drawing based on the relaxed Laplacian with $\rho \in A$ in a radical computation tree.*

In the full version of the paper we similarly prove that spectral drawings based on the adjacency matrix and the transition matrix cannot be constructed by a radical computation tree. In the full version of the paper we similarly prove that drawings produced by classical multidimensional scaling cannot be constructed by a radical computation tree.

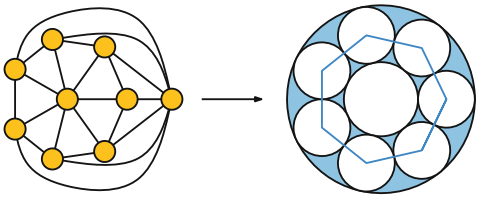

Fig. 6. The graph $\mathrm{Bipyramid}(7)$ and its associated concentric circle packing

5 Impossibility Results for Circle Packings

Root Computation Trees. A given graph may be represented by infinitely many circle packings, related to each other by Möbius transformations. But as we now show, if one particular packing cannot be constructed in our model, then there is no other packing for the same graph that the model can construct.

Lemma 11. *Suppose that a circle packing P contains two concentric circles. Suppose also that at least one radius of a circle or distance between two circle centers, at least one center of a circle, and the slope of at least one line connecting two centers of circles in P can all be constructed by one of our computation models, but that P itself cannot be constructed. Then the same model cannot construct any circle packing that represents the same underlying graph as P.*

Proof. Suppose for a contradiction that the model could construct a circle packing Q representing the same graph as P. By Lemma 3 it could transform Q to make the two circles concentric, giving a packing that is similar either to P or to the inversion of P through the center of the concentric circles. By one more transformation it can be made similar to P. The model could then rotate the packing so the slope of the line connecting two centers matches the corresponding slope in P, scale it so the radius of one of its circles matches the corresponding radius in P, and translate the center of one of its circles to the corresponding center in P, resulting in P itself. This gives a construction of P, contradicting the assumption. □

We define $\mathrm{Bipyramid}(k)$ to be the graph formed by the vertices and edges of a $(k+2)$-vertex bipyramid (a polyhedron formed from two pyramids over a k-gon by gluing them together on their bases). In graph-theoretic terms, it consists of a k-cycle and two additional vertices, with both of these vertices connected by edges to every vertex of the k-cycle. The example of $\mathrm{Bipyramid}(7)$ can be seen in Figure 6, left.

Theorem 7. *There exists a graph whose circle packings cannot be constructed by a quadratic computation tree.*

Proof. Consider the circle packing of $\mathrm{Bipyramid}(7)$ in which the two hubs are represented by concentric circles, centered at the origin, with the other circle centers all on the unit circle and with one of them on the x axis. One of the centers of this packing is at the root of unity ζ_7. By Lemma 6, $[\mathbf{Q}(\zeta_7) : \mathbf{Q}] = \phi(7) = 6$. 6 is not a power of two, so by Lemma 5 ζ_7 cannot be constructed by a quadratic computation tree. By Lemma 11, neither can any other packing for the same graph. □

In the full version of the paper, we prove that certain circle packings also cannot be constructed by radical computation trees nor by bounded-degree root computation trees.

6 Conclusion

We have shown that several types of graph drawing cannot be constructed by models of computation that allow computation of arbitrary-degree radicals, nor by models that allow computation of the roots of bounded-degree polynomials. Whether the degree of these polynomials must grow linearly as a function of the input size, or only proportionally to a sublinear power, remains subject to an open number-theoretic conjecture.

It is natural to ask whether these drawings might be computable in a model of computation that allows both arbitrary-degree radicals and bounded-degree roots. We leave this as open for future research.

Acknowledgements. We used the Sage software package to perform preliminary calculations of the Galois groups of many drawings. Additionally, we thank Ricky Demer on MathOverflow for guiding us to research on large factors of $\phi(n)$.

References

[1] Chrobak, M., Goodrich, M.T., Tamassia, R.: Convex drawings of graphs in two and three dimensions. In: 12th Symp. on Computational Geometry (SoCG), pp. 319–328 (1996)
[2] Hopcroft, J.E., Kahn, P.J.: A paradigm for robust geometric algorithms. Algorithmica 7, 339–380 (1992)
[3] Tutte, W.T.: How to draw a graph. Proc. London Math. Soc. 3, 743–767 (1963)
[4] Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force-directed placement. Software: Practice and Experience 21, 1129–1164 (1991)
[5] Kamada, T., Kawai, S.: An algorithm for drawing general undirected graphs. Information Processing Letters 31, 7–15 (1989)
[6] Koren, Y.: Drawing graphs by eigenvectors: theory and practice. Computers & Mathematics with Applications 49, 1867–1888 (2005)
[7] Kruskal, J.B., Seery, J.B.: Designing network diagrams. In: Proc. First General Conf. on Social Graphics, pp. 22–50 (1980)
[8] Koebe, P.: Kontaktprobleme der Konformen Abbildung. Ber. Sächs. Akad. Wiss. Leipzig, Math.-Phys. Kl. 88, 141–164 (1936)
[9] Bajaj, C.: The algebraic complexity of shortest paths in polyhedral spaces. In: Proc. 23rd Allerton Conf. on Communication, Control and Computing, pp. 510–517 (1985)
[10] Carufel, J.L.D., Grimm, C., Maheshwari, A., Owen, M., Smid, M.: A Note on the unsolvability of the weighted region shortest path problem. In: Booklet of Abstracts of the 28th European Workshop on Computational Geometry, pp. 65–68 (2013)
[11] Bajaj, C.: The algebraic degree of geometric optimization problems. Discrete Comput. Geom. 3, 177–191 (1988)
[12] Nister, D., Hartley, R., Stewenius, H.: Using Galois theory to prove structure from motion algorithms are optimal. In: IEEE Conf. Computer Vision & Pattern Recog., pp. 1–8 (2007)
[13] Varfolomeev, V.V.: Galois groups of the Heron–Sabitov polynomials for inscribed pentagons. Mat. Sb. 195, 3–16 (2004); Translation in Sb. Math. 195, 149–162 (2004)

[14] Brightwell, G., Scheinerman, E.: Representations of planar graphs. SIAM J. Discrete Math. 6, 214–229 (1993)
[15] Ben-Or, M.: Lower bounds for algebraic computation trees. In: Proc. 15th Annu. Symp. Theory of Computing, pp. 80–86 (1983)
[16] Yao, A.C.: Lower bounds for algebraic computation trees of functions with finite domains. SIAM J. Comput. 20, 655–668 (1991)
[17] Shoup, V.: A Computational Introduction to Number Theory and Algebra. Cambridge Univ. Press (2009)
[18] Baker, R.C., Harman, G.: Shifted primes without large prime factors. Acta Arith. 83, 331–361 (1998)
[19] Cox, D.A.: Galois Theory. 2nd edn. Pure and Applied Mathematics. Wiley (2012)
[20] Jacobson, N.: Basic Algebra I, 2nd edn. Dover Books on Mathematics. Dover (2012)
[21] Stäckel, P.: Arithmetische Eigenschaften ganzer Funktionen (Fortsetzung.). J. Reine Angew. Math. 148, 101–112 (1918)