Deciding monotonicity of good drawings of the complete graph

Oswin Aichholzer*[1], Thomas Hackl†[1], Alexander Pilz‡[1], Gelasio Salazar§[2], and Birgit Vogtenhuber¶[1]

[1]Institute for Software Technology, Graz University of Technology, Austria.
[2]Instituto de Fisica, Universidad Autonoma de San Luis Potosi, Mexico.

Abstract

We describe an $O(n^5)$ time algorithm for deciding whether a good drawing of the complete graph K_n, given in terms of its rotation system, can be re-drawn using only x-monotone arcs.

Introduction

In this paper we investigate *x-monotone drawings* of complete graphs. We recall that a drawing of a graph is *x-monotone* if each vertical line intersects each edge at most once. We are interested in *good* drawings of the complete graphs K_n. We recall that in a *good drawing* of a graph no two edges share more than one point (either a common end vertex or a crossing) and no edge crosses itself (this last condition is obviously satisfied in every x-monotone drawing). An important motivation to focus the attention on good drawings is that every crossing-minimal (in the usual definition of crossing number) drawing of a graph is good.

Besides their natural aesthetic appeal, x-monotone drawings provide a nice generalization of rectilinear and pseudolinear (see Section 3) drawings. Very little seems to be known about this natural class of drawings. Pach and Tóth proved in [4] two Hanani-Tutte type theorems for (arbitrary, not necessarily good) x-monotone drawings; their results were later strengthened by Fulek et al. in [3]. Pach and Tóth also showed that, in sharp contrast with the behavior of the rectilinear and the pseudolinear crossing numbers, which cannot be bounded from above by any function of the usual crossing number, the x-monotone crossing number of a graph is at most twice the square of its usual crossing number [5].

It is natural to ask what is the complexity of verifying if a given drawing of a graph is equivalent to an x-monotone drawing (that is, if there is a self-homeomorphism of the sphere and a stereographic projection that takes the original drawing into an x-monotone drawing). We call such a drawing *monotone*. For monotone drawings of K_n, tight lower bounds on the crossing number are known [1, 2]. The only algorithmic question related to monotone drawings that we are aware of was settled in [3], where an $O(n^2)$ algorithm is given that tests whether a graph with given x-coordinates assigned to the vertices has an x-monotone embedding (respecting the given x-coordinates).

Here we adopt the broader point of view that we are given the rotation system of the drawing (as opposed to, say, its cell structure). We recall that in a given drawing, the *rotation* at a vertex is the clockwise ordering of edges at that vertex, and that the *rotation system* is the collection of rotations at its vertices. We investigate the question of whether, given a rotation system that corresponds to a good drawing of K_n, there exists a monotone drawing of K_n with the same rotation system (i.e., the drawings are *weakly isomorphic*). We show (see Theorem 5) that there is a polynomial-time algorithm that settles this decision problem.

A major tool in our algorithm is a variation of a characterization of monotonicity by Balko, Fulek, and Kynčl [2, Lemma 4.8] (see Theorem 2). The algorithm can be easily modified to test whether a given good drawing is weakly isomorphic to a 2-page book drawing (see Corollary 7). Finally, we briefly discuss in Section 3 a related result on characterizing pseudolinear drawings of complete graphs.

1 A different characterization of monotonicity

For a drawing D of K_n let $S = (v_1, \ldots, v_s)$ be a sequence of its vertices. For $1 \leq i < j \leq s$, we denote with $D(S, i, j)$ the drawing obtained from D by removing $v_1, \ldots, v_{i-1}, v_{j+1}, \ldots, v_s$. If D is seen as a subset of the plane, then a *cell* of D is a connected component of $\mathbb{R}^2 \setminus D$.

*Email: oaich@ist.tugraz.at. Supported by the ESF EUROCORES programme EuroGIGA–ComPoSe, Austrian Science Fund (FWF): I 648-N18.

†Email: thackl@ist.tugraz.at. Supported by the Austrian Science Fund (FWF): P23629-N18 ‘Combinatorial Problems on Geometric Graphs’.

‡Email: apilz@ist.tugraz.at. Supported by the ESF EUROCORES programme EuroGIGA–ComPoSe, Austrian Science Fund (FWF): I 648-N18.

§Email: gsalazar@ifisica.uaslp.mx. Supported by CONACYT grant 222667.

¶Email: bvogt@ist.tugraz.at.

Definition 1 (Ábrego et al. [1]) *A drawing D of K_n is* s-shellable *if there exists a sequence $S = (v_1, \ldots, v_s)$ of vertices and a cell C of D with the following property: for all $1 \leq i < j \leq s$, the vertices v_i and v_j are on the boundary of the region of $D(S, i, j)$ that contains C. The sequence S is an* s-shelling *of D* witnessed *by C.*

Definition 2 *Let D be a drawing of K_n. We say that a sequence of vertices $(v_1, \ldots, v_s)$ is a* partial shelling sequence from v_1 to v_s *if there is a cell C of D such that for all $1 < i < s$, v_i is incident to the cell of $D(S, i, s)$ that contains C. Similarly, we say that $(v_1, \ldots, v_s)$ is a* partial shelling sequence from v_s to v_1 *if there is a cell C of D such that for all $1 < i < s$, v_i is incident to the cell of $D(S, 1, i)$ that contains C. In either case, C* witnesses *the partial shelling sequence.*

The following lemma is similar to [2, Observation 4.5], but uses a slightly more general formulation.

Lemma 1 *A sequence $S = (v_1, \ldots, v_s)$ of vertices is an s-shelling of a drawing D of K_n witnessed by a cell C of D if and only if (i) v_1 and v_s are incident with C, (ii) $(v_1, \ldots, v_s)$ is a partial shelling sequence from v_1 to v_s witnessed by C, and (iii) $(v_1, \ldots, v_s)$ is a partial shelling sequence from v_s to v_1 witnessed by C.*

The following theorem is the key ingredient to our algorithm. It connects monotonicity (the existence of a monotone good drawing) of a rotation system with three simple properties. For the proof of this theorem we need the notion of the *unbounded cell* of a good drawing D. This is the cell of D of which every point is connected to infinity.

Theorem 2 *A good drawing D of K_n is monotone if and only if there exists a permutation $\pi = (v_1, \ldots, v_n)$ of its vertices such that the following three properties hold.*

1. *For every v_i, the rotation of v_i contains a consecutive subsequence that contains exactly the elements $v_{i+1}, \ldots, v_n$.*
2. *For every v_i, the edge $v_i v_{i+1}$ does not cross any edge $v_a v_b$, for all $a, b > i$.*
3. *No 3-cycle in D separates v_1 from v_n.*

Proof. The three properties clearly hold if D is monotone w.r.t. π. A drawing is monotone if and only if it contains a permutation $\pi = (v_1, \ldots, v_n)$ of its vertices such that π is an n-shelling of D and the path defined by π is plane [2, Lemma 4.8]. Planarity of the path $(v_1, \ldots, v_n)$ in D directly follows from Property 2. By [2, Lemma 4.7], Property 3 is equivalent to v_1 and v_n sharing a cell. It remains to prove that π is an n-shelling of D (witnessed by a cell C). By Lemma 1 this is the case if D has a partial shelling sequence from v_1 to v_n and one from v_n to v_1, witnessed by C.

We first show that D has a partial shelling sequence from v_1 to v_n. The vertices v_1 and v_n share at least one cell C, the unbounded cell. By Property 2, $v_1 v_2$ is not crossed. Thus, v_1 and v_2 share two cells, which merge with C when removing v_1. Continuing this argument for any i from 2 to n, we can derive from Properties 2 and 3 that v_i and v_n share a cell (the unbounded cell) that contains C in $D(\pi, i, n)$. Hence, v_i and v_n share a cell (the unbounded cell) in $D(\pi, i, n)$ and D has a partial shelling sequence from v_1 to v_n witnessed by C.

Next we show that D has a partial shelling sequence from v_n to v_1 witnessed by C. For the sake of contradiction assume that there is no such partial shelling sequence. Recall that, by Property 3, v_1 and v_n share the unbounded cell C in D. Therefore, there is an edge $v_i v_{i+1}$ ($i \leq n-1$) that is crossed by an edge $v_j v_k$ of the sub-drawing induced by $v_1, \ldots, v_{i+1}$. Both j and k have to be less than i. W.l.o.g., let $j < k$. Note that $k = j + 1$ is not possible due to Property 2. Thus, $1 \leq j \leq i - 3$ and $j + 1 < k < i$. Consider the drawing induced by v_j, v_k, v_i, v_{i+1}, plus v_{k-1} (see Figure 1). The vertices v_i and v_{i+1} partition the rotation around v_k into two parts. By Property 1, v_{k-1} and v_j have to be in the same partition. However, by Property 2, the path $(v_j, \ldots, v_k)$ must not cross any of the edges of the 3-cycle $v_k v_i v_{i+1}$, a contradiction.

Thus, by Lemma 1, π is an n-shelling of D and since the path $(v_1, \ldots, v_n)$ is plane, it follows that D is monotone w.r.t. π. □

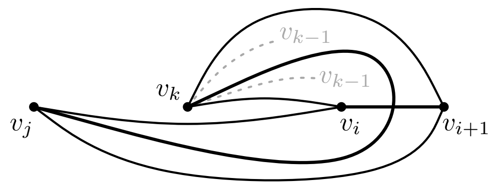

Figure 1: There cannot be an edge $v_j v_k$ crossing an edge $v_i v_{i+1}$.

Note that the second part of the proof shows that our properties imply the reverse formulation of Property 2 (i.e., where $a, b < i + 1$).

2 The algorithm

In this section, we provide an $O(n^5)$ time and $O(n^4)$ space algorithm for deciding monotonicity, based on Theorem 2. The algorithm is separated into two phases, one where the checks for the properties are prepared, and one where we pick the rightmost vertex v_n and test the properties with respect to this choice. Both phases rely on dynamic programming.

Throughout this section, an *interval* (w, k) in the rotation of a vertex v is the consecutive subsequence of its rotation that starts with w and has length k.

Preprocessing. In the preprocessing phase we prepare two data structures to be able to check Properties 1 and 2 of Theorem 2 (see Lemmata 3 and 4, respectively) in constant time per problem instance. For Property 3 we need no preprocessing as it can be checked directly in the permutation search algorithm (see the proof of Theorem 5).

Lemma 3 *Let D be a good drawing. Given two vertices v and v', as well as an interval (w, k), $2 \leq k \leq n-1$, in the rotation around v that contains v'. Let $D(v, w, k)$ be a sub-drawing induced by v and the vertices in the interval (w, k). In $O(n^5)$ time we can construct a data structure of size $O(n^4)$ to answer the following query. Is there a consecutive subsequence of $k-1$ vertices in the rotation of v' that contains exactly the vertices of $D(v, w, k)$ without v and v'? In addition, the data structure provides the first element of the subsequence, if it exists.*

Proof. Let σ be the linearly ordered sequence obtained from the rotation around v' when removing v (i.e., σ starts with the element after v). Add the first $k-1$ vertices of σ to a queue Q and let c be the number of vertices in Q that are in the interval (w, k) in the rotation around v (c can be obtained in linear time). If $c = k-1$, Q is the required subsequence and we are done. (As an answer we only store the first element of Q in our data structure.) Otherwise, add the next element v_l of σ that has not been added to Q to it, and remove the first element v_f of Q from it. (The invariant that $|Q| = k-1$ is maintained.) If v_f is part of the interval at v, there cannot exist a subsequence as required at v' and we are done. If v_l is part of the interval at v, we increase c by one. If at some point $c = k-1$, we are done and Q defines the requested subsequence. The first element of that subsequence (i.e., the first element of Q) can obviously be reported in constant time in that case. Otherwise, if we added all vertices of σ to Q without c becoming $k-1$, we know that the required subsequence does not exists. We repeat this process for any combination of v, v', w, and k, resulting in overall $O(n^5)$ time and $O(n^4)$ space. ☐

Lemma 4 *Let D be a good drawing. Given two vertices v and v', as well as an interval (w, k), $2 \leq k \leq n-1$, in the rotation around v that contains v'. Let $D(v, w, k)$ be a sub-drawing induced by v and the vertices in the interval (w, k). In $O(n^5)$ time we can construct a data structure of size $O(n^4)$ to answer the following query. Is there an edge in $D(v, w, k)$ that crosses vv'?*

Proof. We use dynamic programming. Clearly, for $k = 2$, the edge vv' is uncrossed. Suppose it is also uncrossed for a subproblem given by the tuple $(v, v', (w, k))$. If there is an edge crossing vv' in the subproblem $(v, v', (w, k+1))$, this crossing must be with an edge that is incident to the additional vertex. (Note that given the rotation system, we can determine in constant time whether two edges of a good drawing of K_n cross.) There is only a linear number of choices for the other end vertex for an edge crossing vv'. Thus, one step can be performed in linear time and the $O(n^5)$ time and $O(n^4)$ space bounds follow. ☐

Permutation search. Using the data structures from the preprocessing we can now prove our claim about checking monotonicity. The algorithm is implicitly given in the proof of the following theorem.

Theorem 5 *Given the rotation system of a good drawing D of K_n, we can test in $O(n^5)$ time and $O(n^4)$ space whether there exists a permutation π of its vertices such that D is monotone w.r.t. π.*

Proof. Using dynamic programming we build a permutation $\pi = (v_1, \ldots, v_n)$ of the vertices of D and show that it fulfills the properties of Theorem 2, or that such a permutation does not exist. In a first phase, we apply the preprocessing provided by Lemmata 3 and 4. For the second phase, we guess a vertex v_n, the last element of π. We then apply Theorem 2 as follows. For any vertex v_i and an interval (w, k) in its rotation containing v_n, let $D(v_i, w, k)$ be the sub-drawing induced by v_i and the vertices in the interval for $k \leq n-1$. We check whether there is a permutation of the vertices of $D(v_i, w, k)$ starting with v_i and ending with v_n such that Properties 1 and 2 are fulfilled. At the end of the second phase for each v_n, we check whether there exists a vertex and an interval around it, such that this vertex can be v_1 (in combination with the guessed v_n), fulfilling Property 3.

In the following, we describe the second phase in more detail. For each choise of v_n we consider problem instances comprising a vertex v_i and an interval (w, k) around v_i. The base cases $(v_i, (w, 3))$ can be easily decided. Using dynamic programming, we can assume that all problem instances up to $k-1$ are decided.

For each vertex v_i and an interval (w, k) around v_i that is containing v_n, we guess a successor v_{i+1} from that interval. We check whether the rotation of v_{i+1} has a consecutive subsequence that matches the chosen interval at v_i; let $(w', k-1)$ be this interval at v_{i+1}. Then, we check whether the edge $v_i v_{i+1}$ crosses any edge in the sub-drawing $D(v_i, w, k)$. Both checks can be done in constant time (see Lemmata 3 and 4) after the preprocessing phase. As soon as one of these checks fails, we know that this problem instance is not part of the required permutation and continue with the next instance. If both checks are positive, we obtain w' and there is a permutation of the vertices of $D(v_i, w, k)$ starting at v_i and ending at v_n fulfilling Properties 1 and 2 if there exists one such permutation for v_{i+1} and the interval $(w', k-1)$.

After all problem instances $(v_i, (w, n-1))$ have been decided for one fixed v_n, it remains to find a valid v_1. Note that Properties 1 and 2 are fulfilled for one interval $(w, n-1)$ at v_1 if and only if they are fulfilled for all intervals of length $n-1$ around v_1. We check Property 3 for each of the $O(n)$ possibilities for v_1 in a brute-force way ($O(n^3)$ triangle side tests) and are done for the fixed v_n.

The first phase takes $O(n^5)$ time and $O(n^4)$ space. There are linearly many possibilities to chose v_n, there is a cubic number of subproblems $(v_i, (w, k))$, and for each subproblem, we check a linear number of successor vertices. Checking all possible choices for v_1 takes $O(n^4)$ time for each v_n. Hence, the overall algorithm to decide monotonicity needs $O(n^5)$ time and $O(n^4)$ space. □

Note that in case such a permutation π exists, we can easily retrieve it with the algorithm by simply storing valid successor vertices. For a valid vertex v_1 it is then standard to retrieve π in linear time.

We can modify the above approach to obtain an algorithm to test whether a drawing has a 2-page book drawing. Instead of checking whether an edge vv' is crossed by edges of a sub-drawing (Lemma 4), we can check whether it is crossed at all.

Lemma 6 *The set of uncrossed edges of a good drawing D can be reported in $O(n^3)$ time.*

Proof. Since the edges are uncrossed, they define a planar graph, and therefore there are only $O(n)$ such edges. We first obtain a maximal plane sub-drawing of D consisting of a set of edges denoted by F by processing the edges in an arbitrary order. The current edge is compared against all edges already added to F (at most linearly many). If the edge crosses none of them, we add it to F. Thus, F will contain (a superset of) all uncrossed edges of D. Finally, we test each edge of F with all edges of D, reporting the uncrossed ones. □

Corollary 7 *Given a good drawing D, we can test whether D is weakly isomorphic to a 2-page book drawing in $O(n^5)$ time and $O(n^4)$ space.*

The spine of a 2-page book drawing of K_n together with the edge v_1v_n forms a plane (i.e., crossing-free) Hamiltonian cycle. It is conjectured [6] that every good drawing of K_n contains a plane Hamiltonian cycle. For our dynamic-programming approach, the interval of a vertex defining the subproblem enables us to reason about the subproblem. This interval can not be used when looking for plane Hamiltonian cycles. Is there a polynomial-time algorithm to obtain a plane Hamiltonian cycle in a good drawing of K_n, if it exists? Can the set of all uncrossed edges of a good drawing be reported in $o(n^3)$ time?

3 Pseudo-linear drawings

Consider a drawing of K_n in the projective plane $\mathbb{P}^2$. A *pseudo-line* is a bi-infinite simple curve in $\mathbb{P}^2$ that does not disconnect $\mathbb{P}^2$. A drawing in $\mathbb{P}^2$ is *pseudo-linear* if each edge can be simultaneously extended to a pseudo-line such that in the resulting set of pseudo-lines each pair of pseudo-lines intersects exactly once in a proper crossing. It is well-known that a pseudolinear drawing is monotone. Balko, Fulek, and Kynčl [2] show that a monotone drawing is pseudolinear iff it does not contain a drawing of K_4 that contains a crossing and where in this sub-drawing one vertex is in the interior of a triangle (w.r.t. the unbounded cell). We call such a drawing a *bad* K_4. We show (in the full version) that the restriction to monotone drawings in the statement of [2] is unnecessary, a result obtained independently by Arroyo, McQuillan, and Richter (personal communication).

Theorem 8 *A good drawing of K_n in $\mathbb{P}^2$ is pseudo-linear if and only if it does not contain a bad K_4.*

Acknowledgements. We want to thank Bernardo M. Ábrego, Silvia Fernández-Merchant, and Pedro Ramos for valuable discussions.

References

[1] B. M. Ábrego, O. Aichholzer, S. Fernández-Merchant, P. Ramos, and G. Salazar. Shellable drawings and the cylindrical crossing number of K_n. *Discrete Comput. Geom.*, 52(4):743–753, 2014.

[2] M. Balko, R. Fulek, and J. Kynčl. Crossing numbers and combinatorial characterization of monotone drawings of K_n. *Discrete Comput. Geom.*, 53(1):107–143, 2015.

[3] R. Fulek, M. J. Pelsmajer, M. Schaefer, and D. Štefankovič. Hanani-Tutte, monotone drawings, and level-planarity. In *Thirty essays on geometric graph theory*, pages 263–287. Springer, 2013.

[4] J. Pach and G. Tóth. Monotone drawings of planar graphs. *Journal of Graph Theory*, 46:39–47, 2004.

[5] J. Pach and G. Tóth. Monotone crossing number. *Mosc. J. Comb. Number Theory*, 2(3):18–33, 2012.

[6] N. H. Rafla. *The good drawings D_n of the complete graph K_n*. PhD thesis, McGill University, Montreal, 1988.