Graph Crossing Number and Isomorphism
SPUR Final Paper, Summer 2012

Mark Velednitsky
Mentor Adam Bouland
Problem suggested by Adam Bouland, Jacob Fox
MIT

Abstract

The polynomial-time tractability of graph isomorphism is a longstanding open problem. In this paper, we examine the fixed-parameter tractability of graph isomorphism with respect to crossing number and related parameters. We show that graph isomorphism if fixed-parameter tractable with respect to convex crossing number. We conjecture that graph isomorphism is FPT in the regular crossing number, and explain why it is difficult to prove this conjecture.

Introduction

In general, the problem of determining if two graphs are isomorphic is believed to be NP-intermediate. It is natural to ask what makes graph isomorphism so difficult and if knowledge about the graph's fundamental structure can made it tractable. It is known that planar graph isomorphism can be solved in polynomial time (Hopcroft, Wong 1974), so it is natural to ask if graph isomorphism is also easy on graphs which are nearly planar, i.e. have low crossing number. The framework of fixed-parameter tractability is used to capture the feasibility of problems when controlling graph structure. In this paper, we explore if Graph Isomorphism is fixed-parameter tractable with respect to crossing number.

Only a few fixed-parameter tractability results for Graph Isomorphism are known. It is known that Graph Isomorphism is fixed-parameter tractable with respect to tree-depth, largest multiplicity of an eigenvalue of the adjacency matrix (Evdokimov, Ponomarenko 1999), size of the largest class color (Arvind, et. al. 2009), and a few other parameters (Bouland, Dawar, Kopczynski 2012), but its tractability is unknown with respect to crossing number. Unlike Colorability, Clique, or Hamiltonian Cycle, Graph Isomorphism studies the relation between two graphs, not just the properties of one. This oddity of Graph Isomorphism makes it a very difficult problem to reduce to, so there are no hardness results known for the complexity of Graph Isomorphism with respect to any parameters discussed in this paper.

On the other hand, quite a lot is known about crossing number. It is known that in sufficiently dense graphs, the crossing number is linear in the number of vertices (Fox, Toth 2008). It is also known that computing crossing number is fixed-parameter tractable with respect to crossing number (Grohe 2002). Grohe's proof relies on expressing crossing number in monadic second-order logic and applying Courcelle's Theorem, which states that properties that can be expressed in monadic second-order logic are decidable in linear time on graphs of bounded tree-width. In fact, Grohe shows that it is possible to explicitly construct a minimal embedding of the input graph. Unfortunately, though, the construction relies on a tree-width decomposition, which can not necessarily be chosen in an isomorphism-invariant way. This is what makes it difficult to apply in the present setting.

Of the several potential approaches to showing Graph Isomorphism is fixed-parameter tractable, many of theme have a common theme. It would be sufficient to show that one can find a crossing edge of a graph in an isomorphism invariant way in $f(k)$ steps (where k is the crossing number). One could then remove the edge and proceed by induction on the crossing number. The challenge with these algorithms is that the removal of a single edge could completely change the set of drawings with minimal crossing number, thus obviating induction. That is, certain graphs are very unstable and small perturbations of their minimal drawings require them to be completely redrawn. Another approach would be to somehow limit the number of embeddings of the graph to $f(k)$. In this case, the challenge is in understanding graph embeddings. In general, the behavior of graph embeddings of low crossing number is poorly understood.

Following the definitions, this paper shows the fixed-parameter tractability of graph isomorphism with respect to convex crossing number. We then explain why it is difficult to extend this result to regular crossing number.

Definitions

Parameterized Complexity

While traditional complexity theory evaluates the running times of algorithms in terms of their input size, parameterized complexity theory evaluates the running time of algorithms in terms of their input size and a second parameter.

Definition 1. A parameter, k, is a map from problem instances Q to the natural numbers $\mathbb{N}$.

Definition 2. A parameterized problem, with parameter k, is said to be in XP with respect to k if there is an algorithm that solves it in time $n^{f(k)}$ for some function f and some constant c.

Definition 3. A parameterized problem, with parameter k, is said to be fixed-parameter tractable with respect to k if there is an algorithm that solves it in time $f(k)n^c$ for some function f and some constant c (Flum, Grohe 2006).

Example 1. It is easy to prove that determining the crossing number of a graph is at least in XP with respect to crossing number. Given a graph, G, simply remove all subsets of k edges and test for planarity. If at least one of the resulting graphs is planar, then G has crossing number at most k. If no resulting graph is planar, then G has crossing number larger than k. The running time of the algorithm is $\binom{n}{k} = O(n^k)$.

Graph Isomorphism Problem

The Graph Isomorphism problem has numerous applications to Physics, Chemistry, Artificial Intelligence, and many other fields. Its classical complexity has been well studied and, in practice, good algorithms exist to determine isomorphisms between small graphs.

Definition 4. Two graphs are isomorphic if there exists a bijection between their vertices which preserves edges and vertex labels.

Definition 5. A canonical labeling algorithm assigns to each graph G a bijection $f_G : V_G \to \{1, 2, \ldots, |V_G|\}$ such that if H is isomorphic to G then $f_H^{-1}(f_G(v))$ is an isomorphism.

Example 2. Note that any canonical labeling algorithm can be turned into a graph isomorphism algorithm. Given two graphs, each with a canonical labeling, we consider the bijection which maps 1 to 1, 2 to 2, . . ., and n to n. The two graphs are isomorphic if and only if this bijection is an isomorphism. Thus, the problem of determining a canonical labeling is at least as hard as the graph isomorphism problem.

Crossing Number

The crossing number of a graph is, intuitively, the minimum number of edge crossings required to draw it in the plane. In a sense, it captures how far a certain graph is from being planar. Crossing number is an upper bound on the edge deletion distance from planarity. Furthermore, the crossing number is always at least as large as the genus of the graph. A graph with crossing number k can always be drawn on a genus k surface by looping edges through holes to avoid crossings. This means that if graph isomorphism is fixed-parameter tractable with respect to genus, it is immediately fixed-parameter tractable with respect to crossing number. Thus, it is easier to prove fixed-parameter tractability with respect to crossing number than with respect to genus.

Definition 6. A graph is said to be simple if it contains no loops and any two vertices are connected by at most one edge. Equivalently, it has no cycles of length 2. In this paper, we will assume all the graphs in question are simple.

Definition 7. The crossing number of a graph, denoted Cr(G) is defined as the minimal number of crossings needed to draw it in the plane. The crossing number of a planar graph is 0 while the crossing number of $K_{3,3}$ is 1.

Definition 8. The convex crossing number of a graph, denoted Con-Cr(G) is defined as the minimal number of straight edges necessary to draw the graph in the plane such that its vertices lie on the unit circle (thus, all of its edges are chords).

Definition 9. An embedding of a graph G is a drawing of G in the plane with a minimal number of crossings.

Definition 10. A graph is said to be k-connected if there are at least k vertex-disjoint paths between any two vertices.

Example 3. It is easy to see that $\mathrm{Cr}(G) \leq \text{Con-Cr}(G)$. Given a drawing of G with convex-crossing number k, it is also a drawing with crossing number k.

Planar Graph Isomorphism

Since planar graph isomorphism can be solved in polynomial time, it is natural to examine if the planar graph isomorphism algorithm can be generalized to graphs of low crossing number. At the heart of planar graph isomorphism are three very important results.

Lemma 1. Every graph can be decomposed into a rooted tree of depth three such that each node is a subgraph of its parent. The root of the tree is the original graph. Furthermore, the nodes at depth two are at least 2-connected and the leaves are at least 3-connected. This decomposition can be computed in linear time. (Hopcroft, Tarjan 1974).

Lemma 2. There are only two embeddings of a planar, 3-connected graph. (Whitney 1933).

Lemma 3. Tree Isomorphism can be solved in linear time. (Lindell 1992).

Theorem 1. Planar graph isomorphism can be solved in polynomial time (Hopcroft, Tarjan 1974).

Let T_1 be the decomposition of G_1 and T_2 the decomposition of G_2, as described in Lemma 1. We check isomorphism of T_1 and T_2, where a node of T_1 is equivalent to a node of T_2 if and only if the corresponding graphs are isomorphic. By Lemma 2, we can check if two leaves are equivalent in linear time. By Lindell's algorithm for tree isomorphism, this suffices to test for isomorphism of T_1 and T_2. By Lemma 1, T_1 is isomorphic to T_2 if and only if G_1 is isomorphic to G_2. This decides isomorphism with running time $O(n)$. □

Convex Crossing Number

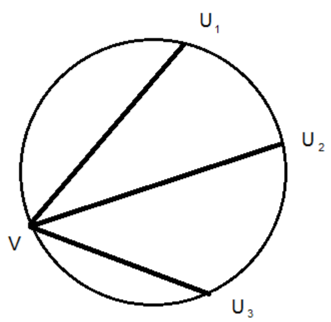
We begin by showing that graph isomorphism is fixed-parameter tractable with respect to convex crossing number. We can extend the techniques of the planar result to convex crossing number.

Theorem 2. Graph Isomorphism is fixed-parameter tractable with respect to convex crossing number.

We rely heavily on the following lemma, which effectively "replaces" Lemma 2.

Lemma 4. Any 3-connected graph with $n \geq 4$ vertices has convex crossing number $k \geq \frac{n}{4}$.

Let G be a 3-connected graph and fix it in one of its minimal embeddings. Consider a vertex $v \in G$. The degree of v is at least 3. Consider the edges which leave v. These edges are chords in the unit circle. At least one of them will divide the circle into two distinct regions such that each region contains another edge. We can think of this edge as the "middle" edge, call it (v, u_2). By construction, there exist edges (v, u_1) and (v, u_3) such that u_1 and u_3 are in separate regions of the circle.



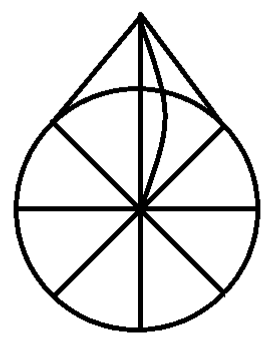
By assumption, there must be at least three vertex disjoint paths between u_1 and u_3. All three paths must pass through (v, u_2), but at most one can pass through v and at most one can pass through u_2. Thus, there is at least one path with an edge that crosses the edge (v, u_2), resulting in a crossing. Map v to this crossing. Notice that any v will always map to a crossing on an edge that it is adjacent to, so at most four vertices will map to the same crossing. This gives us $k \geq \frac{n}{4}$.

Now, any graph can be decomposed into an isomorphism-invariant tree with 3-connected components as leaves in time $O(n)$ (Lemma 1). Graph ismorphism can be solved on these leaves in time $(4k)!$ or constant time on graphs with $n < 4$. As in the planar case, we can propagate upwards in linear time (Lemma 3). Thus, isomorphism of the original graph is checked in time $O((4k)!n)$, giving the desired fixed-parameter tractability result. □

Extending Planar Graph Isomorphism

Ideally, we would like to extend the algorithm for planar graph isomorphism to graphs of low crossing number. This would require us to generalize Lemma 2 and show that graphs of low crossing number have few embeddings. Unfortunately, the number of embeddings of a 3-connected non-planar graph is not bounded by 2. In fact, it's not even bounded purely by a function of the crossing number. As a simple counter-example, consider the following construction.

Example 4. Start by taking a wheel graph on $n - 1$ vertices (an $n - 2$ cycle with all of its vertices connected to another point outside the cycle). Add another vertex to the graph and connect it to three adjacent vertices along the wheel as well as to the vertex of degree $n - 2$.



This graph has at least n embeddings. In fact, the number of embeddings of a 1-connected graph can be exponential in n.

Example 5. Given an embedding of a graph graph G with crossing number k, consider subdividing any edge involved in a crossing. This increases the number of vertices by 1, but doubles the number of possible embeddings. Thus, there are $O(n)$ vertices and $O(2^n)$ embeddings.

While this example makes it clear that there is no hope for bounding the number of embeddings in a general graph, subdivisions are not a valid operation in 3-connected graphs because they result in a vertex of degree 2. Thus, it may still be possible that any 3-connected graph with n vertices and k crossings has a fixed-parameter tractable number of embeddings. That is, a number of embeddings that is exponential in the number of crossings but only polynomial in the number of vertices. We state this as a conjecture.

Conjecture 1. The number of embeddings of a 3-connected graph with n vertices and crossing number k is bounded by $k^k n^2$.

In the planar case, this is immediately true. Unfortunately, we do not have any evidence supporting the conjecture other than being unable to find a counter-example.

Edge Classes

Given the failure of Lemma 2 in the generalization of the planar isomorphism algorithm, we are interested in exploring another avenue. One general method for proving problems are fixed-parameter tractable is the method of bounded search trees (Flum, Grohe 2006). The method of bounded search trees essentially shows that a recursive algorithm is sufficient to solve the problem in fixed-parameter tractable time. Given a problem instance p with parameter k, we would like to show that the problem can be reduced to at most $f(k)$ instances with parameter $k-1$. If we think of these instances as children of the original instance, then we have a tree of instances with height k and fan-out at most $f(k)$. If we know that the problem in question can be solved in polynomial time on the leaves, i.e. when $k=0$, then an algorithm which recurses up the tree will solve the original instance p in fixed-parameter tractable time.

If we are to have any hope of finding a recursive algorithm and invoking the method of bounded search trees, we will need to define a new parameter which does not depend on fixing an embedding before recursing downwards. As was mentioned in the introduction, computing crossing number is fixed-parameter tractable with respect to crossing number (Grohe 2002). This allows us to compute the following set in fixed-parameter tractable time:

$$F_G = \{e \in G | \mathrm{Cr}(G-e) < \mathrm{Cr}(G)\}$$

In other words, F_G is the set of edges that participate in a crossing in at least one embedding and thus decrease the crossing number when they are removed. In a sense, F_G captures the behavior of an edge across *all* embeddings. When we were working with crossing number, an edge may or may not have been crossed depending on the embedding. F_G has the advantage that every edge either is or is not in F_G. We would like to show that graph isomorphism is fixed-parameter tractable with respect to $|F|$. Note that $\mathrm{Cr}(G) < |F_G|$ for all graphs G, so a fixed-parameter tractable algorithm for Graph Isomorphism with respect to $\mathrm{Cr}(G)$ would imply one for $|F_G|$. In that sense, $|F_G|$ is a strictly easier parameter to work with.

Unfortunately, F_G is not as well-behaved under edge deletions as it first appears. Consider the following (flawed) attempt to prove that F_G is closed under edge deletions.

Given a graph G, consider an edge $e_1 \in F_G$. Say that $\mathrm{Cr}(G) = k$ and $\mathrm{Cr}(G - e_1) = k'$. We would like to show

$$e_2 \in F_{G-e_1} \implies e_2 \in F_G$$

which is equivalent to

$$e_2 \notin F_G \implies e_2 \notin F_{G-e_1}$$

We will examine what happens when we remove e_2 followed by e_1 and then we will examine what happens when we remove e_1 followed by e_2. This will allow us to study if $e_2 \in F_{G-e_1}$. Let $\mathrm{Cr}(G-e_1-e_2) = k''$. At first, this seems promising. We know that $\mathrm{Cr}(G-e_2) = k$. It would seem that removing e_1 from $G-e_2$ would result in a crossing number k', thus showing $k' = k''$, but that is deceptive. It is possible that deleting e_2 gives $G - e_2$ sufficient flexibility to be drawn in a way such that it has crossing number $k + k'''$, for some k''', but now with e_1 having more than $k''' + (k - k')$ crossings. The other direction is equally hopeless. If we delete e_1 first, it is possible that the resulting graph can be completely redrawn such that there is *some* minimal embedding containing e_2, in which case deleting e_2 after e_1 further decreases the crossing number. Note that if the graph were to be completely redrawn, it would have to be redrawn such that restoring e_1 to the drawing would result in a crossing number significantly larger than k. In short, there is simply no good way to characterize minimal drawings of $G - e_1$ in terms of minimal drawings of G and so nothing useful can be said about $|F_G|$.

In summary, we believe the following conjecture is false. If it were true, it would show that Graph Isomorphism is fixed-parameter tractable with respect to $|F_G|$ by the method of bounded search trees. That is, deleting an edge of G would always decrease $|F_G|$, so we could construct a bounded search tree with depth and fan-out both equal to $|F_G|$.

Conjecture 2. If F_G is the set of edges which decrease the crossing number of G, then $F_{G-e} \subset F_G$ for all $e \in F_G$.

Ordering Embeddings

Yet another approach would be to order the embedding of a graph to define a small set of minimal embeddings and check isomorphism on them in polynomial time. That is, we would like to define an ordering or partial ordering on embeddings of G that satisfies the following two criteria.

1. The set of minimal embeddings of G has fixed-parameter tractable size.
2. The set of minimal embeddings of G can be found in fixed-parameter tractable time.

The planar graph isomorphism algorithm is actually a canonical labeling algorithm (Hopcroft, Wong 1974). This canonical labeling lends itself naturally to an ordering on planar graph embeddings. That is, we begin by canonically labeling the graph and defining the tuple $(e_{1,2}, e_{1,3}, \ldots, e_{n-1,n})$, where $e_{a,b}$ is 1 if there exists an edge from a to b and 0 otherwise. The ordering of graphs follows immediately from the ordering on vectors (ex. lexicographic ordering). We say that this ordering is induced by the canonical labeling algorithm.

Given this ordering, it is natural to wonder if a similar ordering can be imposed on embeddings of low crossing number. Even if the ordering is only partial, having at most $f(k)n^c$

embeddings on each level of the partial ordering would be sufficient to imply our result. All we would need to do is efficiently find the minimal embeddings, thereby restricting ourselves to at most $f(k)n^c$ embeddings to check. Unfortunately, the existing ordering for planar graphs relies on a canonical labeling algorithm, so it does not immediately generalize. This leads us to propose two definitions for ordering, with the goal of defining the ordering such that finding the minimal embeddings is easy.

Definition 11. Given a graph G with n vertices and crossing number k, draw it in the plane with exactly k crossings. Study each crossing and remove one of the two edges that crosses there. When at most k edges have been removed, the remaining graph is planar. Label this graph canonically and then restore the edges that were removed. As in the planar case, construct a tuple representing the edge set with respect to this canonical labeling. This tuple, associated with the particular embedding, would be the basis for an ordering of embeddings.

The only flaw in this ordering is that there is no isomorphism invariant way to choose which edges will be involved in crossings. For example, in K_5, any pair of edges be the pair that cross in a minimal drawing. This means that finding the minimal embedding could take n^k time. In terms of the criteria above, this definition satisfies criterion 1, but fails criterion 2. We consider the next definition, which attempts to approach the ordering problem directly without appealing to the planar case.

Definition 12. It is not difficult to show that any graph can be drawn such that its vertices lie on a sufficiently large lattice. To see this, imagine creating an arbitrary lattice and perturbing each vertex a distance ϵ such that it lies at rational coordinates. The lattice can then be subdivided to match the graph. Given a lattice, it is easy to define an ordering on points in the lattice and thus assign a value to each vertex. To compare two graph embeddings, consider the value of their smallest vertex. If they differ in the value of their smallest vertex, consider the second-smallest vertex. Repeat until a difference is found. If two graphs have the same vertex set, then compare the set of edges between the vertices as in the previous definition.

The flaw in this ordering is, as mentioned in the introduction, that small perturbations in the graph will dramatically affect its minimal embedding. This leaves little hope for approaching the problem of finding minimal embeddings as it immediately obviations recursion, leaving only brute force as a viable option. Thus, again, we have satisfied criterion 1, but not criterion 2.

It is interesting that both of the orderings above had the same flaw. Although one ordering was based on an extension of planar graph ordering and the other was more ad hoc, both resulted in a small number of minimal embeddings which were hard to find. In the case of the first algorithm, the computational time was well-defined but too large. In the case of the second algorithm, the minimal drawings were poorly understood due to their instability under edge deletion. While neither definition works, their flaws lend some insight into what makes this problem so difficult.

Conclusion

Using the planar graph isomorphism algorithm, we have shown that Graph Isomorphism is fixed-parameter tractable with respect to convex crossing number. Unfortunately, extending this result to traditional crossing number is quite difficult. The number of minimal embeddings can be large in n and drawings can be highly unstable under edge deletion. We conjecture that Graph Isomorphism *is* fixed-parameter tractable with respect to crossing number, but a proof would require more advanced techniques.

Acknowledgements

I would like to thank the SPUR program for making this project possible, Professor Jacob Fox for suggesting the project, and especially my mentor Adam Bouland for the many hours he spent teaching and working with me.

References

[1] Arvind, V., Dan B., Johannes, K., Toda, S., *Colored Hypergraph Isomorphism is Fixed Paramter Tractable*. Electronic Colloquium on Computation Complexity, 93 (2009).

[2] Bouland, A., Dawar, A., Kopczyński, E., *On Tractable Parameterizations of Graph Isomorphism*. IPEC 2012, LNCS 7535, 218-230 (2012).

[3] Evdokimov, S., Ponomarenko, I., *Isomorphism of Colored Graphs with Slowly Increasing Multiplicity of Jordan Blocks*. Combinatorica, 19(3) (1999).

[4] Flum, J., Grohe, M., *Parameterized Complexity* Springer Texts in Theoretical Computer Science, (2006).

[5] Fox, J., Toth, C., *On the Decay of Crossing Numbers*. Journal of Combinatorial Theory (B), 33-42 (2008).

[6] Grohe, M., *Computing Crossing Numbers in Quadratic Time*. Proceedins of the 32nd ACM Syposium on Theory of Computing, 231-236 (2002).

[7] Hopcroft, J. E., Wong, J. K., *Linear Time Algorithm for Isomorphism of Planar Graphs*. Proceedings of the sixth annual ACM symposium on Theory of computing, 172-184 (1974).

[8] Hopcroft, J., Tarjan, R., *Dividing a Graph into Triconnected Components*. SIAM Journal of Computation, 135-158 (1974).

[9] Lindell, S., *A Logspace Algorithm for Tree Canonization*. Proceedings of the 24th Annual ACM Symposium on Theory of Computing, 400-404 (1992).