Algorithms for Straight Line Planar Graph Drawing

Ross Evans

August 14 2020

Abstract

This paper discusses straight line drawings of planar graphs on grids. The main topic of summary is the paper *How to Draw a Planar Graph on a Grid* by Fraysseix, Pach, and Pollack. The paper summarizes their argument, while providing extra background information, concrete examples, and motivation. Mention is also given to Tutte's *How to Draw a Graph*, and some other recent developments since Fraysseix, Pach, and Pollack's paper.

1 Introduction

Drawing a graph on the plane has been a fundamental method of visualizing a graph's structure since the beginning of the study of graph theory. In Euler's seminal paper, *Solutio problematis ad geometriam situs pertinentis*, Euler analyzed whether an individual could cross each of the seven bridges of Königsberg exactly once [8]. He notably included the first ever diagram of a graph in this paper, not only laying the foundation for graph theory, but also demonstrating the usefulness of drawings as a method to easily visualize and conceptualize graphs. Since then, many achievements have been made relating to planar graphs, one of the most notable of them being Kuratowski's characterization of planar graphs as those not containing a subgraph that is a subdivision of K_5 or $K_{3,3}$. The development of graph drawing algorithms did not occur until the 1960s, when Tutte published his paper *How to Draw a Graph*. This paper aims to provide a summary on a few straight-line graph drawing algorithms. First, it will be proved that every planar graph has a drawing using exclusively straight-line edges. Then, there will be a brief discussion on Tutte's method, and some of the issues associated with the algorithm derived from his proof. The bulk of the paper will consist of summarizing the work done by Fraysseix, Pach, and Pollack in creating an algorithm to draw any planar graph on a $2n-4$ by $n-2$ grid (this grid area is asymptotically optimal). Finally, there will be brief mentions on improvements made to this algorithm in the intervening years with regards to reducing the running time of the algorithm, as well as reducing the constant factor of the grid area.

2 Preliminaries

A graph $G = (V, E)$ will be defined in the usual way as an ordered pair of sets, with the set V representing edges, and the set E containing unordered pairs of vertices. This paper will focus exclusively on simple undirected graphs, and although this definition excludes the possibility of repeated edges, self-loops, and directionality, it will nevertheless suffice for the algorithms presented here.

Defining a planar embedding properly can be a somewhat tricky task; it requires a degree of subtlety that is easily overlooked. For our purposes, a planar embedding of a graph $G = (V, E)$ will be an ordered pair (V', E'), where V' is a finite subset of $\mathbb{R}^2$, and each edge in E' is a curve whose endpoints are in V', with the expected stipulations that each vertex in V corresponds to exactly one point in V', and that edges in E correspond to curves in E' connecting the expected points. We also require that the interiors of the edges in E' are disjoint from each other and the vertices; this yields the expected behaviour that an edge does not pass through multiple vertices on its path and that edges are non-crossing. To be fully rigorous, in this context, a curve is a subset of $\mathbb{R}^2$ of the form $f([0, 1])$ where $f : [0, 1] \to \mathbb{R}^2$ is a continuous injective function. By enforcing that the endpoints (($f(0)$ and $f(1)$) of each edge correspond to points in V', the behaviour of the embedding is as expected. These definitions of a planar embedding are presented for rigour, but often, the arguments discussed in this paper will not go into full topological detail. For example, it could be demonstrated that the faces of a graph as typically defined would, in this definition, be the connected topological spaces of $\mathbb{R}^2 - V' - E'$. It then becomes arduous to prove even simple results such as Euler's Formula, the fact that the embedding of a cycle has an inside and an outside, or that each edge borders at most two faces. Hence, we will instead opt to discuss planar graphs informally, while recognizing that the arguments presented should be carefully inspected for topological soundness with the proper rigour of something like the Jordan Curve Theorem.

Although this won't be justified rigorously, it should be reasonably clear that any planar embedding using curves could also instead use a union of finitely many line segments. This is intuitively because a curve will be better approximated as more line segments are used, and therefore using enough line segments will yield a new embedding which also has no edge-crossings. A question one could ask after realizing this result, is whether all graphs with a planar embedding also have a planar embedding in which all the edges are a single line segment between its vertex endpoints. This question motivates the first definition and the first theorem of the paper.

3 Fáry Embeddings

Definition 3.1 (Fáry embedding)**.** A Fáry embedding is a planar embedding of a graph such that all edges are a single line segment between its vertex endpoints.

Theorem 3.1 (Fáry's Theorem)**.** *All simple planar graphs have a Fáry embedding. [5]*

The idea of a Fáry embedding will come to motivate the main algorithm in this paper, but before getting to that point, it will prove instructive both to consider some examples, and to prove Fáry's theorem outright. A good example to demonstrate the difficulty of the problem is a lattice graph with one extra vertex, whose edges connect to the outer vertices of the lattice. An example of this is shown in Figure 1. Clearly, any lattice graph is planar. It is also easy to find a planar embedding for the lattice graph with the extra vertex, so long

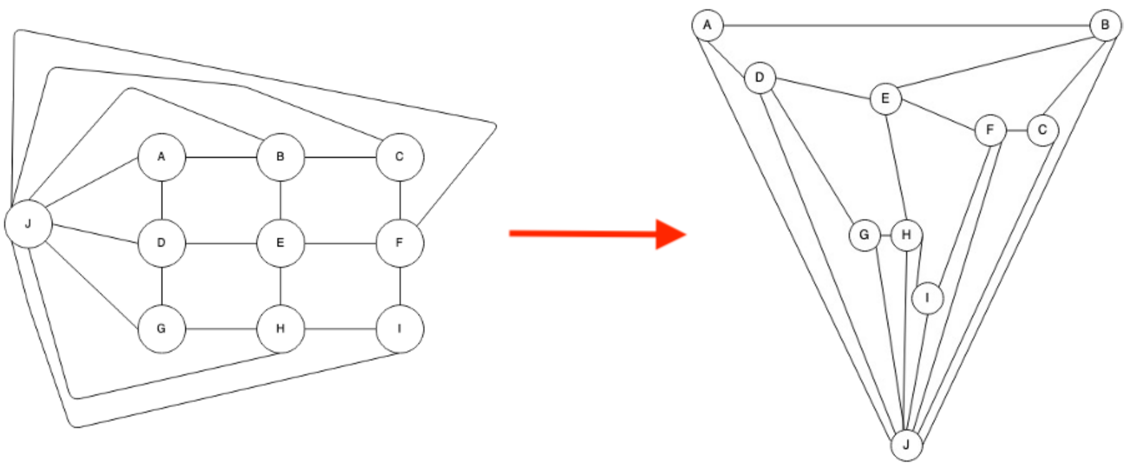

Figure 1: Left: A planar embedding of a graph. Right: A Fáry embedding of the same graph

as curved edges are allowed. Despite this, a procedure for transforming a planar embedding into a Fáry embedding is non-obvious, as evidenced by Figure 1. This example is only for a three by three lattice graph, one can imagine that for larger and larger lattice graphs, finding a valid Fáry embedding could become exceedingly complicated to perform, perhaps at some point even becoming impossible. This is where the proof of Fáry's theorem becomes helpful; as a result, when given a planar graph, the algorithm can focus exclusively on outputting the Fáry embedding, without needing to consider whether the embedding outright exists. We first state some elementary facts about planar graphs before proving Fáry's theorem.

Lemma 3.2. *A maximal planar graph with $n \geq 3$ has $3n - 6$ edges.*

Proof. This follows from Euler's formula. ■

Lemma 3.3. *If G is a maximal planar graph with $n \geq 4$, then every vertex in G has a degree of at least 3.*

Proof. Consider some vertex v in G. $G - v$ is also planar, and $n - 1 \geq 3$, so Lemma 3.2 holds for $G - v$. Hence $m - \deg(v) \leq 3(n-1) - 6$. Since G is maximal planar, $m = 3n - 6$. $3n - 6 - \deg(v) \leq 3n - 9 \implies \deg(v) \geq 3$ [2] ■

Lemma 3.4. *A maximal planar graph with at least 4 vertices has at least 4 vertices with degree less than or equal to 5.*

Proof. Consider some maximal planar graph G with $n \geq 4$. Let the maximum degree of a vertex in G be denoted as $\deg(G)$. Since G is maximal planar, $m = 3n - 6$, which implies that $2m = 6n - 12$. Let n_i be the number of vertices with degree i, and let d be the maximum degree of G. From the handshaking lemma, $\sum_{v \in V} \deg(v) = 2m$, this can alternatively be written as $\sum_{i=1}^{d} i \cdot n_i$. Finally, since $n = \sum_{i=1}^{d} n_i$, we have that $\sum_{i=1}^{d} i \cdot n_i = 6 \sum_{i=1}^{d} n_i - 12$. Rearranging, we see that $\sum_{i=1}^{d} (6 - i) n_i = 12$. Notice that vertices with degree greater than or equal to 6 do not contribute positively to this sum. And, by Lemma 3.3, each vertex has a degree of at least 3. As a result, there must be at least 4 vertices with degree less than or equal to 5. [2] ■

The proof of Fáry's theorem now follows, by induction on the number of vertices in the graph. We will prove the statement for maximal planar graphs, noting that for non-maximal planar graphs we can simply add edges until it is maximal, find the embedding, and then delete the extra edges to get a Fáry embedding of the non-maximal graph.

Proof. First, note that if $n \leq 3$, then clearly there exists a Fáry embedding, the most complicated graph that could arise with only 3 vertices is K_3, which is just a triangle. This establishes the base case for our inductive proof.

Inductive Hypothesis: Consider some maximal planar graph with n vertices and an associated planar embedding of the graph with vertices a, b, and c on the unbounded face of this embedding. Assume for all graphs of this type that there exists a Fáry embedding of the graph which preserves the same adjacency between faces as the original embedding, and also has a, b and c on the unbounded face.

Inductive Step: Consider some maximal planar graph with $n + 1$ vertices and vertices u, v, and w on the unbounded face. By Lemma 3.4, there exists a vertex x not equal to u, v, or w, such that the degree of x is between 3 and 5. Consider a planar embedding (V', E') of G. Removing x from G gives us a face F whose vertices are the neighbours of x in G. By the inductive hypothesis, we can construct a Fáry embedding of $G - x$. The goal then is simply to add x inside F, and connect it with straight lines to its neighbours. If F is convex, we can add x anywhere and connect using straight lines. If F is not convex, then it's surrounding face is either a quadrilateral or a pentagon. From here, the argument boils down to some geometric arguments, which will not be of much relevance to the following algorithms. These arguments are therefore omitted, more detailed explanations can be found in the book *Graphs & Digraphs* [2] ■

What happened here? Essentially, once the previous lemmas have been stated, the proof boils down to a simple inductive argument. We wish to pick a good vertex to induct on, this is provided by Lemma 3.4. If we did not have a vertex of bounded degree, then the polygon associated with the face of $G - x$ could have had arbitrarily many edges. This would complicate the geometric argument. Furthermore, if we could not ensure that there were at least 4 vertices with a degree of at least 5, then we could have run into the situation where u, v, or w is our vertex with low degree. Removing this vertex would have prevented us using our inductive hypothesis, since we would no longer be able to ensure an isomorphism exists between the faces of our two drawings. Using the inductive hypothesis essentially takes our planar embedding and "straightens out" the edges, while preserving the overall structure of the drawing. This allows the reinsertion of x in an appropriate place so as to construct a Fáry embedding for G.

This inductive style of constructing embeddings is one of the main ideas behind the algorithm of Fraysseix, Pach, and Pollack. Of course, the proof requires that we first have a planar embedding of the graph (possibly using curves), and finding this planar embedding seems in the general case to be as difficult as finding a Fáry embedding. Instead, the algorithm will consider some cleverly selected sequence of subgraphs $G_4, G_5, ..., G_n$ with $|V(G_k)| = k$, such that the process of placing the extra vertex to get from G_k to G_{k+1} is relatively simple.

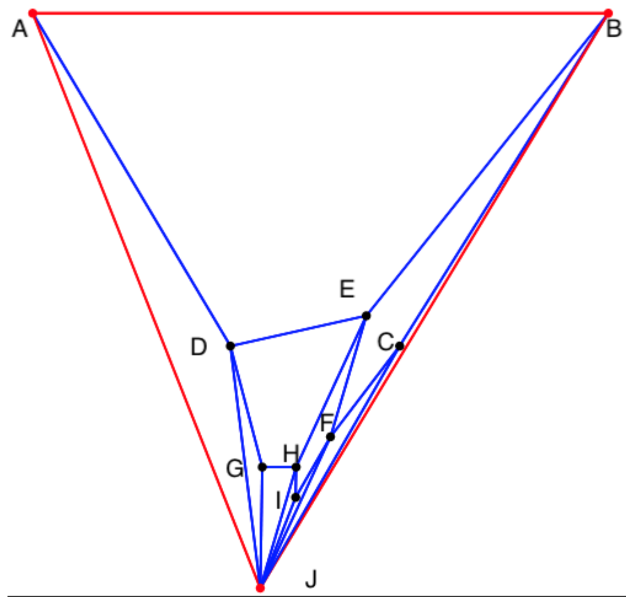

Figure 2: A Tutte Embedding of the graph seen in Figure 1

4 Tutte's Method

In *How to Draw a Graph*, Tutte proved a slightly stronger statement than Fáry, this time about simple planar 3-connected graphs, instead of simple planar graphs.

Definition 4.1. A graph G is k-connected if $|V(G)| > k$ and there is not a set $X \subseteq V(G)$ with $|X| < k$ such that $G - X$ is disconnected.

Definition 4.2 (Tutte Embedding)**.** A Tutte Embedding of a graph is a Fáry embedding such that the vertices on the outer face form a convex polygon, and each interior vertex is located at the average position of its neighbours.

Theorem 4.1 (Tutte's Spring Theorem)**.** *Given a 3-connected graph G, fix the position of some vertices such that they form a convex polygon. From this starting embedding, there always exists a unique Tutte Embedding, where the interior faces of the graph are always convex. [10]*

The proof of this theorem is fairly complex; we will not discuss it. However, note that a very natural algorithm arises from the theorem's description. To produce a smooth animation, one could repeatedly move vertices to the neighbours of their positions. Alternatively, since each vertex needs to be moved to the average position of its neighbours, one could rewrite the positions of the vertices as a system of linear equations which would be solvable using some process such as LU factorization. An LU factorization takes the same running time as matrix multiplication, so the current best running time for Tutte's approach would be approximately $O(n^{2.37})$, and this method could never run faster than quadratic time, which leaves much to be desired. Furthermore, the algorithm arising from Tutte's approach requires high precision arithmetic, and vertices appear very close together. If the constraint is added that each vertex has to appear on a grid, then the grid size becomes exponential [9]. An example of this can be seen in Figure 2. Beyond requiring high precision arithmetic to store, this example demonstrates that although the Tutte Embeddings can often appear pleasing if the graph has some level of symmetry, graphs without symmetry often end up lopsided. In summary, although Tutte's result is very elegant, as an algorithm it fails to scale well as graph sizes increase. Additionally it begs the question as to whether bounds can be improved with regards to the grid size required as well as the running time of the algorithm. The answer to both of these questions turns out to be yes, as discovered by Fraysseix, Pach, and Pollack.

5 The Main Algorithm

5.1 Canonical Orderings

This portion of the paper is where we describe the algorithm developed by Fraysseix, Pach, and Pollack in their paper *How to Draw a Planar Graph on a Grid*. The algorithm's function depends on the existence of a canonical ordering, which is an ordering of the vertices that allows the embedding to be constructed incrementally with minimal adjustment to the previous embedding at each step.

Definition 5.1 (Canonical Ordering)**.** Let G be a maximal planar graph, with some Fáry embedding G' such that the vertices on the unbounded face are u, v, and w. A canonical ordering is a labelling of the vertices $v_1 = u$, $v_2 = v$, v_3,, ..., $v_n = w$ such that for every $4 \leq k \leq n$, the subgraph G_{k-1} induced by $v_1, v_2, ..., v_{k-1}$ is 2-connected, the vertices comprising the unbounded face form a cycle C_{k-1} containing the edge uv, and the vertex v_k can be placed in the unbounded face and connected to adjacent vertices in C_{k-1}

This definition provides a good starting point for which to develop an algorithm. The general procedure is to start with an embedding for G_3, which is a simple triangle, and then progressively add vertices to the outside of the embedding, connecting them to adjacent vertices on the outer cycle. Note by the fact that the vertices are adjacent on the cycle C_{k-1}, we have to worry less about being able to reach all neighbours with straight lines, as opposed to if they were on opposite sides of the cycle. Nevertheless, there is some care necessary in the design and approach. First, we need to ensure that all maximal planar graphs have a canonical ordering. Then, we need to ensure that the next vertex v_k can inserted in a convenient position on the grid such that it can be connected to all of its neighbours, or if this is not possible, that some parts of the embedding can be adjusted to achieve this. Finally, we'll need to analyze time and grid size requirements to ensure that this algorithm does actually outperform previous approaches such as Tutte's method. Thanks to the work done in *How to Draw a Planar Graph on a Grid*, we have proof that all of these conditions are satisfiable. We start with the crux of the algorithm: the existence of a canonical ordering.

Theorem 5.1. *Every maximal planar graph has a canonical ordering. [6]*

Proof. This is done by induction, choosing v_n as the base case, and using an inductive hypothesis to label vertices up until v_3. To begin, by the nature of u, v, and w being on the unbounded face of the graph, we can simply let $v_n = w$, and choose to let G_{n-1} be the subgraph of $G - w$. By the fact that u and v were on the unbounded face in G, after removing v, we have that they are contained in a cycle on the unbounded face in G_{n-1}.

As an inductive hypothesis, assume that we have determined the canonical ordering from v_n up until some vertex v_{k+1}. Consider the unbounded face in the subgraph G_k which is given by $V(G) \setminus \{v_{k+1}, ..., v_n\}$. This unbounded face forms a cycle C_k. There exists some vertex on this cycle y such that $y \neq u$, $y \neq v$, and y is not an endpoint of any chord of C_k (a chord is defined to be two vertices c_i and c_j on C_k such that the edge c_ic_j is present, and c_ic_j is not an edge of C_k). This can be proved by a simple minimality argument. If there are no chords in C_k, then this is simple. On the other hand, if there are, we can choose a chord that "bridges" the shortest distance, i.e, if we label the vertices of C_k as $c_1, ..., c_k$, then we choose the chord defined by the vertices c_i and c_j such that $j - i$ is minimal. Note that we must have $j > i + 1$, otherwise the vertices are just adjacent. But then c_{i+1} is not an endpoint of any chord, otherwise we would contradict minimality. We can choose c_{i+1} to be v_k in our canonical ordering, and we satisfy the desired properties. It is not equal to u or v since those are the first and last vertices in our cycle, and we explicitly chose it to be in between two other vertices. By the nature of it not being adjacent to any chords, we have that $V(G) \setminus \{v_k, v_{k+1}, ..., v_n\}$ still has a cycle comprising the unbounded face, which contains the edge uv. Hence, this satisfies the properties necessarily, and we can induct until v_3 as necessary. [6] ■

Essentially, through some simple arguments using chords, we can guarantee this canonical ordering exists as necessary. I believe it is important to mention the implicit assumption that was made in this proof that a Fáry embedding exists for us to induct on. If Theorem 3.1 had not been proved, then we would not have a starting point to induct on, and so this proof that a canonical ordering exists would have a large initial assumption.

5.2 The Shift Method

Having established the theory behind Fáry embeddings and the idea of a canonical ordering, we can now finally present the main theorem given by Fraysseix, Pach, and Pollack in *How to Draw a Planar Graph on a Grid*.

Theorem 5.2. *Any planar graph with n vertices has a Fáry embedding on the $2n - 4$ by $n - 2$ grid. [6]*

Proof. Assume that G is a maximal planar graph with n vertices. It suffices to prove this theorem for maximal planar graphs since one could simply add dummy edges in order to make it maximal planar, run the algorithm, and then remove those edges again in some post-processing.

Choose some arbitrary face u, v, w to comprise G's unbounded face, and let $v_1 = u$, $v_2 = v, v_3, ..., v_n = w$ be the canonical ordering of its vertices. Assume that at some step k of the algorithm, we have the following invariants preserved:

1. v_1 is placed at $(0, 0)$ and v_2 is placed at $(2k - 4, 0)$

2. If $c_1 = v_1, c_2, ..., c_m = v_n$ denote the vertices on the outer cycle C_k, and $x(c_i)$ denotes the x-coordinate of c_i, then $x(c_1) < x(c_2) < ... < x(c_m)$;

3. The edges $c_i c_{i+1}$ between adjacent vertices of the outer cycle all have slopes +1 or −1 (apart from $c_1 c_m = v_1 v_n$, which has a slope of 0 as necessitated by invariant 1). Note that a corollary of this is that all vertices on the outer cycle have an even Manhattan ($|x_1 - x_2| + |y_1 - y_2|$) distance from one another.

These invariants are powerful enough for us to prove the theorem as desired. We first note our starting conditions and our ending conditions. We start at the step $k = 3$. In this case, we choose a Fáry embedding we place v_1 at $(0, 0)$, v_2 at $(2, 0)$ and v_3 at $(1, 1)$, and see that the necessary invariants are satisfied. When $k = n$, note that we have u, v, w comprising the unbounded face. By invariant 1, we have that $u = v_1$ is at $(0, 0)$, and $v = v_1$ is at $(2n - 4, 0)$, and by invariant 3, we have that $w = v_n$ is at $(n - 2, n - 2)$, since then we have that the slope of $v_1 v_n$ is +1, and the slope of $v_n v_2$ is −1. Therefore, we know that if we can preserve these invariants while also preserving the embedding as a Fáry embedding at each step, then we can prove that every planar graph has an embedding within an enclosed area of $2n - 4$ by $n - 2$. It is important to note that nothing about these invariants actually necessitates that all the vertices appear on the grid itself, which is to some extent one of the most important features of this algorithm, since we already had the ability to Fáry embed a graph within any arbitrary area via Tutte's method, with the crucial stipulation that vertex positions are not integral. The fact that every vertex position is integral will arise from the process of going from step k to step $k + 1$.

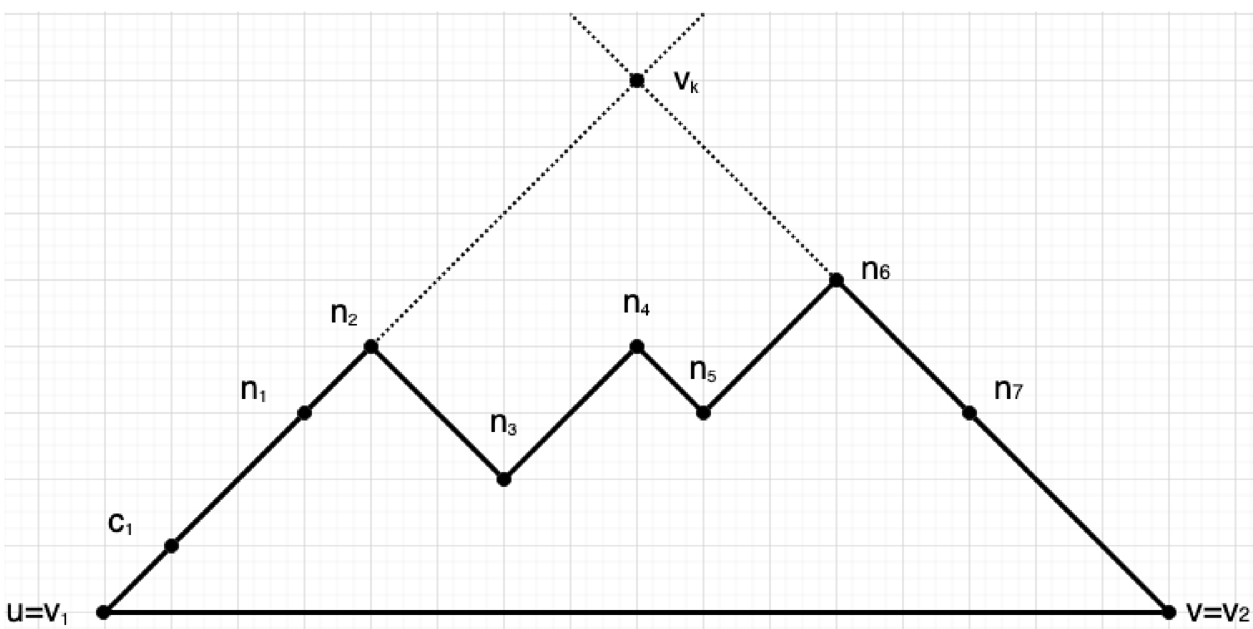

Figure 3: The newly placed vertex v_k unable to see n_1 or n_7 (figure adapted from [6])

To go from step k to step $k + 1$ we have to place to vertex v_{k+1}, and connect it with its neighbours via straight-line edges. As previously mentioned, the canonical ordering eases this task slightly for us. We know all of v_{k+1}'s neighbours lie on the outer cycle C_k, and we know that all of its neighbours are adjacent to one another. Number the neighbours of v_{k+1} as $n_1, ..., n_q$, with the stipulation as in invariant two that the as the index of the neighbour increases, so does its x position on the current embedding. We then know that if we were to place the v_{k+1} and connect it to its neighbours, the new outer cycle would consist of v_1, whatever vertices are between v_1 and n_1, v_k, n_q, and whatever vertices are between n_q and v_2. Therefore, in order to preserve invariant three, we require that the edge $n_1 v_k$ has a slope of 1, and $v_k n_q$ has a slope of -1. It therefore seems intuitive to extend lines out from n_1 and n_q with slopes of 1 and -1, respectively, find the intersection point, and place v_k there. By invariant 3, n_1 and n_q have an even Manhattan distance from one another, and so the lines extended from those vertices will intersect at a point on the grid. There is one small obstacle with this approach however, it may fail to see some of the first and last neighbours (see Figure 3 for an example). An easy way to fix this is to deform the graph slightly, so that the slope of $n_1 n_2$ is slightly less than 1, and the slope of $n_{q-1} n_q$ is slightly greater than -1. This can be done by moving $n_2, n_3, ..., n_{q-1}$ one unit to the right, and moving $n_q, ..., v_m$ (all vertices on the cycle C_k to the right of n_q) two units to the right. However, we must be cautious of the effect that this adjustment has on the embedding of G_k, by deforming the previous embedding as such we may have introduced new crossings. To recover from this, we will introduce another component to our algorithm, which is at each step keeping track of which internal vertices need to be shifted if a vertex c_i on the outer cycle is shifted.

We now consider three more invariants related to keeping track of vertices that must be considered when shifting occurs. Assume that at some step k of the algorithm for each vertex c_i on the outer cycle C_k we have $M_{k,c_i} \subseteq V(G)$ so that

(a) $c_j \in M_{k,c_i}$ if and only if $j \geq i$

(b) $M_{k,c_m} \subset M_{k,c_{m-1}} \subset ... \subset M_{k,c_1}$

(c) For any non-negative numbers $\alpha_1, \alpha_2, ..., \alpha_m$, if we sequentially translate all vertices in M_{k,c_i} a distance α_i to the right, then the embedding of G_k remains a Fáry embedding.

Remember that our starting configuration at $k = 3$ is that v_1 is at $(0, 0)$, v_2 is at $(2, 0)$, and v_3 is at $(1, 1)$. Note that by our notation of denoting the vertices of the cycle C_k as increasing by x-position, we have $c_1 = v_1$, $c_2 = v_3$, and $c_3 = v_2$. We define $M_{3,c_3} = \{c_3\}$, $M_{3,c_2} = \{c_2, c_3\}$, and $M_{3,c_1} = \{c_1, c_2, c_3\}$. This meets invariants a, b, and c, and so this works as a starting place for our induction. Then, we choose the α quantities associated with n_2 and n_q to be 1, and all other α quantities to be 0. Performing the translation of all vertices as necessitated by M_{k,n_2} and M_{k,n_q} yields a new Fáry embedding of G_k, which allows us to place v_{k+1} at the intersection of the lines extended from n_1 and n_q with slope 1 and -1 as previously described. This ensures that the invariants 1, 2, and 3 are preserved from before. Now, if we can simply demonstrate a way of constructing the M_{k+1,c_i} for the new outer cycle C_{k+1} such that the invariants a, b, and c are preserved, then our algorithm will be complete.

For each c_i on the outer cycle of C_{k+1} we define

$$M_{k+1,c_i} = \begin{cases} M_{k,c_i} \cup \{v_{k+1}\} & x(c_i) < x(v_{k+1}) \\ M_{k,n_2} \cup \{v_{k+1}\} & c_i = v_{k+1} \\ M_{k,c_i} & x(c_i) > x(v_{k+1}) \end{cases}$$

By construction, the invariants a and b hold. It is our final task to prove that the invariant c holds as well. By induction, we have that the motion of the vertices contained inside the outer cycle of G_k continue to produce a Fáry embedding if all shifted to the right as described in invariant 3. And, the portion of G_{k+1} for which this doesn't apply, (i.e, v_{k+1} and it's neighbours) are all defined to move the same amount by the definition of M_{k+1,c_1} (they all move a distance of $\alpha(c_1) + \alpha(c_2) + ... + \alpha(n_1) + \alpha(v_{k+1})$). As a result, we can be confident that any shifting which may be necessary in adding v_{k+2} will be possible, since all of our invariants are preserved [6]. ■

The proof of this theorem gives an algorithm to Fáry embed a maximal planar graph on a $2n - 4$ by $n - 2$ grid. There are some other details to consider here for practical implementation. For example, can we quickly test that a graph is planar so that we can reject bad inputs? We also glossed over the task of finding the canonical ordering of the graph. Neither of these questions turn out to pose major obstacles. There exist many planarity-testing algorithms which run in linear time. The process of finding the canonical ordering can be treated as a traversal problem, and since planarity enforces a linear number of edges with respect to the number of vertices, this can also be made to run in linear time [6]. As a final piece of analysis for the algorithm we described, we can find a simple upper bound for the running time as $O(n^2)$, since we have n steps and in each step we may have to iterate over all vertices that have been placed in the previous iterations. In the worst case, we are on step n, and iterating over n vertices, giving us the $O(n^2)$ running time.

6 Further Developments

One improvement that was made in *How To Draw a Graph* which was not described in this paper is the improvement of the running time to $O(n \log n)$. This is achieved by refraining from performing each embedding, instead only keeping necessary information in the form of inductively defined permutations. These permutations allow a simpler way of querying for the elements in the M_{k,c_i} subsets, which allows finding the coordinates of the next vertex to be inserted in logarithmic time instead of linear time [6].

Further papers have also been published reducing the running time and grid size required. Within a couple years after *How To Draw a Graph* was published Chrobak and Payne published an improved algorithm which modifies the shifting strategy slightly and uses a system to anchoring vertices to a specific parent, which allows for linear running time [4]. Since then, many papers have been published reducing the size of the grid requirements. Chrobak and Nakano later proved that each dimension of the grid must be at least $\left\lfloor \frac{2(n-1)}{3} \right\rfloor$. They also provided an algorithm to embed a graph on a $\left\lfloor \frac{2(n-1)}{3} \right\rfloor$ by $4\left\lfloor \frac{2(n-1)}{3} \right\rfloor - 1$ grid [3]. The most recent improvement that I was able to find was by Brandenburg, who improved the required area to just $\frac{8}{9}n^2$ [1]. It is still an open problem whether any planar graph can be embedded using just $\frac{4}{9}n^2$ area. One other interesting open problem is that of Harborth's conjecture, which states that every planar graph has a Fáry embedding in which all edges have integer lengths [7]. If this conjecture does eventually get proved it will be interesting to see whether the proof naturally describes an algorithmic approach, or if further work will be done to yield a useful algorithm, as was the case with the proof of Fáry embeddings to the development of the shift method.

7 Conclusion

This paper has discussed straight-line planar drawing algorithms in depth, proving the existence of a Fáry embedding for all planar graphs, describing Tutte's method which can find Fáry embeddings but only with exponential grid size, and finally covering Fraysseix, Pach, and Pollack's work to draw any planar graph on a $2n - 4$ by $n - 2$ grid. I believe that the problem of straight-line planar graph drawing is one which demonstrates an interesting contrast to those that we studied during this term's offering of CS 466. For example, in CS 466, a large portion of the course focused on the study of randomized algorithms. These algorithms often required little insight into the structural features of the problem itself. For example, Karger's algorithm picked and contracted edges completely randomly, little understanding was needed of the problem to create the algorithm. Another example is the "fix-up" algorithm for solving the k-SAT problem; trying random configurations of variables is not one that requires any insight into the way which variables are related. The difficulty here arises from the analysis of the performance of the algorithms, for example, Moser's proof using random bit compression is highly unique and non-trivial. This contrasts the work demonstrated in this paper. To explain and prove the algorithms in this paper, we required lots of theory about Fáry embeddings and canonical orderings to be established before a discussion about algorithms could arise. In the randomized algorithm case, a very simple approach can prove effective so long as it's backed up by a rigorous analysis, whereas in the deterministic case we require rigorous analysis of the problem before an algorithm can be written. Of course, it would be asinine to assert that all randomized algorithms have simple inspiration; I found the multiplicative weights update method to be a highly insightful and non-obvious approach, for example. But, it nevertheless forges some sort of dichotomy, where in order for deterministic algorithms to still be relevant in modern algorithm design and analysis, more and more clever insight will be required into the structure of a given problem in order to make progress. This is as opposed to the approaches we saw in CS 466, which are all relatively modern and so may not have been as exhaustively attempted as the simpler approaches taught in CS 341.

References

[1] Franz J. Brandenburg. Drawing planar graphs on $\frac{8}{9}n^2$ area. *Electronic Notes in Discrete Mathematics*, 31:37–40, 2008.

[2] Gary Chartrand, Linda Lesniak, and Ping Zhang, *Graphs & Digraphs Fifth Edition*. Chapman and Hall/CRC, Baton Rouge, 2011.

[3] Marek Chrobak and Shinichi Nakano. Minimum-width grid drawings of plane graphs. *Comput. Geom. Theory Appl.*, 11:29–54, 1998.

[4] Marek Chrobak and Thomas Payne. A linear-time algorithm for drawing planar graphs. *Inform. Process. Lett.*, 54:241–246, 1995.

[5] István Fáry. On straight lines representation of planar graphs. *Acta Univ. Szeged. Sect. Sci. Math.*, 11:229–233, 1948.

[6] Hubert de Fraysseix, János Pach, and Richard Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990.

[7] Arnfried Kemnitz and Heiko Harborth. Plane integral drawings of planar graphs. *Discrete Mathematics*, Graph theory), 236 (1–3): 191–195, 2001.

[8] Teo Paoletti/MAA, 2011: Leonard Euler's Solution to the Konigsberg Bridge Problem. Accessed 8 August 2020, https://www.maa.org/press/periodicals/convergence/leonard-eulers-solution-to-the-konigsberg-bridge-problem

[9] Roberto Tamassia. *Handbook of Graph Drawing and Visualization*. Chapman & Hall/CRC, Baton Rouge, 2016.

[10] William T. Tutte. How to draw a graph. *Proceedings London Mathematical Society*, 13(52):743–768, 1963.