Drawability of Complete Graphs Using a Minimal Slope Set

GREG A. WADE AND JIANG-HSING CHU*

* *Department of Computer Science, Southern Illinois University, Carbondale, IL 62901, USA*

In this paper we will study the problem of drawing graphs with a minimum number of slopes. We will show that the minimum number of slopes needed to draw a complete graph of *n* vertices is *n*. We will also prove that for a complete graph of *n* vertices to be drawn using only *n* slopes its vertices must form a convex polygon. Finally, we will present an algorithm which checks whether a complete graph of *n* vertices can be drawn using only slopes from a given set of *n* slopes.

Received July 2, 1993, revised November 9, 1993

1. INTRODUCTION

Pictures are often more useful than words in illustrating ideas. It is said that 'A picture is worth 1000 words.' Whether pictures are drawn neatly or not has a great impact on overall quality of documents.

More and more people are preparing documents with the help of document preparation systems. However, difficulties in handling pictures constitute a serious drawback in most, if not all, of the available document preparation systems, e.g. (Knuth, 1984) and TROFF (Ossanna, 1979). There are many means to incorporate pictures in documents. One common approach to typesetting pictures within a document involves using a special graphic font. This font contains tiny line segments as its character set. Pictures are drawn by putting those tiny line segments in appropriate positions. LaTeX (Lamport, 1986), a TeX macro package which is one of the most popular typesetting systems, only allows lines with slopes a/b, with $1 \leqslant |a|, |b| \leqslant 6$ and, of course, horizontal lines and vertical lines. Thus when a typesetting system similar to LaTeX is used, one can easily see the zigzag pattern in some lines. Lines with different slopes have different visual appeal. Some lines look prettier than the others. Only lines with certain slopes appear smooth. This is illustrated in Figure 1.

To make pictures attractive, it is often desirable to draw them with only 'pretty' lines. Most pictures which are included in documents are actually graphs which represent some abstract ideas. A graph is a set of vertices (dots) together with a set of edges (line segments whose endpoints are vertices). Graphs are usually used to convey some abstract ideas, therefore where we place the vertices is often irrelevant. Thus, we can freely move the vertices around. By doing so, we may be able to draw pictures with only 'pretty' lines.

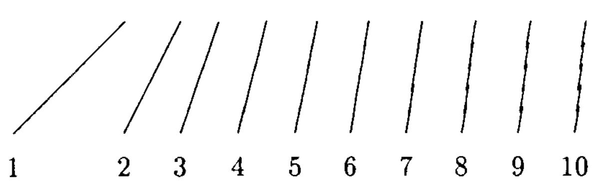

FIGURE 1. Lines with slopes 1 to 10 (LaTeX).

* Correspondence to J.-H. Chu.

There have been some works (e.g. Batini *et al.*, 1986; Rowe and Davis, 1987; Tamassia *et al.*, 1988) which concentrated on how to draw trees and graphs with a nice structural outlook. There were no attempts made to ensure that only 'pretty' lines were used. There were also some results discussing the slope problem in certain types of graphs (Czyzowicz, 1989, 1991; Battista *et al.*, 1990; Czyzowicz *et al.*, 1990a,b).

In this paper, first we investigate how many *different* slopes are needed to draw a complete graph. We define the *slope number* of a graph G, denoted by $/G/$, as the minimum number of slopes required to draw G. We assume we want to draw graphs which consist of a set of vertices and a set of edges which connect the vertices. We will also assume that the edges are straight line segments. Edges are allowed to *intersect*, but not to *overlap*. Later in this paper, we address another interesting question, which is to decide whether it is possible to draw a complete graph using only slopes in a given slope set or not.

2. SLOPE NUMBER OF COMPLETE GRAPHS

A graph is *complete* if there exists an edge between every pair of vertices. We will denote a complete graph of n vertices by K_n.

We would like to answer the question whether a K_n can be drawn using only slopes from a given set S or not. We first have to know whether S has enough slopes or not, i.e. we must find $/K_n/$ first. Let us present a few useful lemmas before we try to find $/K_n/$.

LEMMA 1. In a drawing of a K_n, all edges that share the same vertex must have different slopes.

Proof. Suppose $\overline{v_i v_j}$ and $\overline{v_i v_k}$ have the same slope. Since we are dealing with a complete graph, the edge $\overline{v_j v_k}$ exists, and has the same slope as $\overline{v_i v_j}$ and $\overline{v_i v_k}$. It is

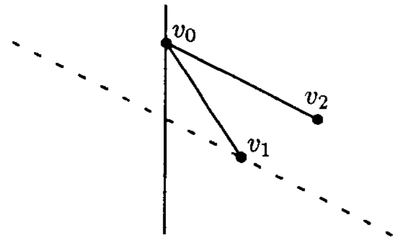

FIGURE 2. Slope of edge $\overline{v_0v_2}$ cannot be used by v_1.

obvious that we cannot draw these three parallel edges without overlapping. □

This lemma can also be restated as 'In a drawing of a K_n, each slope can be used by any vertex at most once'.

LEMMA 2. $/K_n/ \geqslant n$, for $n \geqslant 3$.

Proof. In Figure 2, the vertices are chosen based on the following criteria:

1. Vertex v_0 is the leftmost vertex (the lowest one if there are ties).
2. The (signed) angle between the y-axis and edge $\overline{v_0v_1}$ is the smallest among all the edges incident to v_0.
3. The (signed) angle between the y-axis and edge $\overline{v_0v_2}$ is the second smallest among all the edges incident to v_0.

If the slope of edge $\overline{v_0v_2}$ is also used by v_1, say by edge $\overline{v_1v_i}$, then v_i must fall in the dashed line. However, as v_i must be connected to v_0, no matter which part of the dashed line vertex v_i falls in, it violates one of our selection criteria for vertices v_0, v_1 and v_2. Therefore, we conclude that the slope of edge $\overline{v_0v_2}$ cannot be used by v_1. And from Lemma 1 we know that each vertex uses $n-1$ different slopes. Hence at least n different slopes are needed to draw a K_n. □

We now present a canonical way to draw a K_n. A K_n can be drawn by placing its vertices evenly spaced on the unit circle such that the point $(0, -1)$ is always a mid-point of an edge. The vertices are labeled $v_0, v_1, \ldots, v_{n-1}$ in a clockwise order. When n is odd, v_0 is the top most vertex. When n is even, v_0 is the right one of the two top vertices. A K_n drawn this way is called a *canonical drawing* of the K_n. Two examples of canonical drawing are shown in Figure 3.

LEMMA 3. $/K_n/ \leqslant n$.

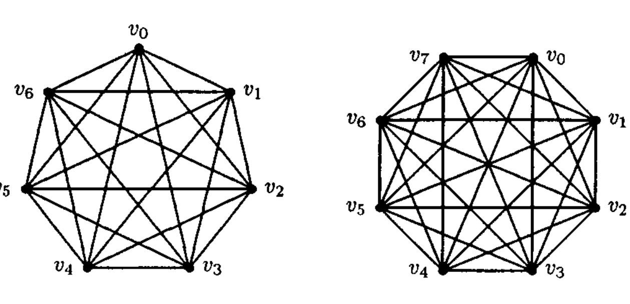

FIGURE 3. Canonical drawings of K_7 and K_8.

Proof. We observe (and can show it algebraicly) that only n distinct slopes are used in the canonical drawing of a K_n, no matter whether n is odd or even. Thus $/K_n/ \leqslant n$. □

THEOREM 1

$$/K_n/ = \begin{cases} 0 & n = 1 \\ 1 & n = 2 \\ n & n \geqslant 3 \end{cases}$$

Proof. The cases of $n = 1$ and $n = 2$ can be trivially seen. From Lemma 2 and Lemma 3, we conclude that $/K_n/ = n$. □

3. DRAWABLE COMPLETE GRAPHS ARE CONVEX

Whether a K_n can be drawn using a set of n slopes can be decided by assigning every edge a slope and checking whether the drawing is drawable. We say a slope assignment is *drawable* if the corresponding drawing exists. We now face the following two problems.

1. How to check the drawability of slope assignments?
2. How to enumerate all possible slope assignments?

Note when we make a slope assignment, we actually construct a linear system of the vertices and the slopes. Whether the slope assignment is drawable or not depends on whether the associated linear system has a non-trivial solution or not. For example, suppose we want to draw a K_3 with $S = \{s_0, s_1, s_2\}$ and we make the following slope assignment: slope s_0 is assigned to edge $\overline{v_1v_2}$, slope s_1 is assigned to edge $\overline{v_0v_2}$, and slope s_2 is assigned to edge $\overline{v_0v_1}$.

The question is whether we can place v_0, v_1, and v_2 properly in order to satisfy the assignment. Let (x_i, y_i) be the coordinates of v_i. We have to satisfy the following equations:

$$0x_0 - s_0x_1 + s_0x_2 + 0y_0 + y_1 - y_2 = 0$$
$$s_1x_0 + 0x_1 - s_1x_2 - y_0 + 0y_1 + y_2 = 0$$
$$s_2x_0 - s_2x_1 + 0x_2 - y_0 + y_1 + 0y_2 = 0$$

Note there always exists a trivial solution, i.e. $x_0 = x_1 = x_2$ and $y_0 = y_1 = y_2$. If the linear system gives a non-trivial solution, then the slope assignment is drawable. Of course, some exception handling is necessary for the vertical slopes.

A *parallel-set* consists of the edges which are assigned the same slope. Two drawings of a graph are said to be *parallel-set equivalent* if we can relabel one of the drawings so that both drawings have identical parallel-sets. We will show that if a slope assignment is drawable, then the drawing produced by the slope assignment is parallel-set equivalent to the canonical drawing of the complete graph. Our argument is mainly based on the number of the edges that can be in a convex hull.

Given a set of points, the *convex hull* of that set of points is defined as the smallest convex polygon which encloses all points in that set.

Here are some facts about the convex hull. If an edge $\overline{v_i v_j}$ is part of the convex hull, then all other vertices lie on the same side of the line which contains the edge (Lee and Preparata, 1984). Note in such a case, the points v_i and v_j are corners of the convex hull. The convex hull of a set of n points consists of at most n edges. If there are n edges in the convex hull, all points appear as corners of the convex hull.

THEOREM 2. If a graph K_n can be drawn by using n slopes, then all the vertices must appear as corners of the convex hull. The produced drawing is parallel-set equivalent to the canonical drawing of the K_n.

Proof.

Case 1—n is odd

A K_n has $n(n-1)/2$ edges. Since only n different slopes are available, each slope is used by $(n-1)/2$ edges on the average. When n is odd, each slope can be used by at most $(n-1)/2$ edges. Thus each slope is used by exactly $(n-1)/2$ edges. Since each slope contributes one edge to the convex hull, there are n edges in the convex hull. Therefore all vertices are part of the convex hull. Figure 4 shows how a slope is used. From Figure 4, we can easily see the produced drawing is parallel-set equivalent to the canonical drawing.

Case 2—n is even

Recall that a K_n has $n(n-1)/2$ edges. Since only n different slopes are available, each slope is used by $(n-1)/2$ edges on the average. When n is even, at least $n/2$ slopes are used $n/2$ times in order to make up the average. Each such slope contributes two edges to the convex hull. Since we can have at most n edges in the convex hull, there are *exactly* $n/2$ slopes which are used exactly $n/2$ times. This implies that all vertices are part of the convex hull. To make up the average, each of the rest of the $n/2$ slopes must be used exactly $n/2-1$ times and *cannot* contribute any edge to the convex hull. The way the slopes are used is shown in Figure 5. From the figure, we can easily see the produced drawing is parallel-set equivalent to the canonical drawing.

Thus the theorem is proved. □

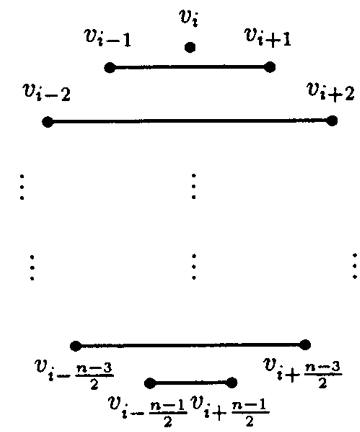

FIGURE 4. The way slopes are used for odd n.

4. DRAWABILITY OF COMPLETE GRAPHS

In a complete graph of n vertices, there are $n(n-1)/2$ edges. If each slope can be assigned to every edge, there are $n^{n(n-1)/2}$ possible slope assignments. It would not be feasible to check all possible slope assignments. Fortunately, many of those slope assignments can be rejected immediately without solving their associated linear systems.

From the discussion in the previous section, for a slope assignment to be drawable, those edges which are in the same parallel-set must be assigned the same slope. Since they are n parallel-sets, the number of possible slope assignments is now down to $n!$, which is still a big number.

Since we know the drawing produced by a drawable slope assignment has all the vertices appear in the convex hull, we can label the vertices $v_0, v_1, \ldots, v_{n-1}$ in the clockwise order. We also know the drawing produced by a drawable slope assignment is parallel-set equivalent

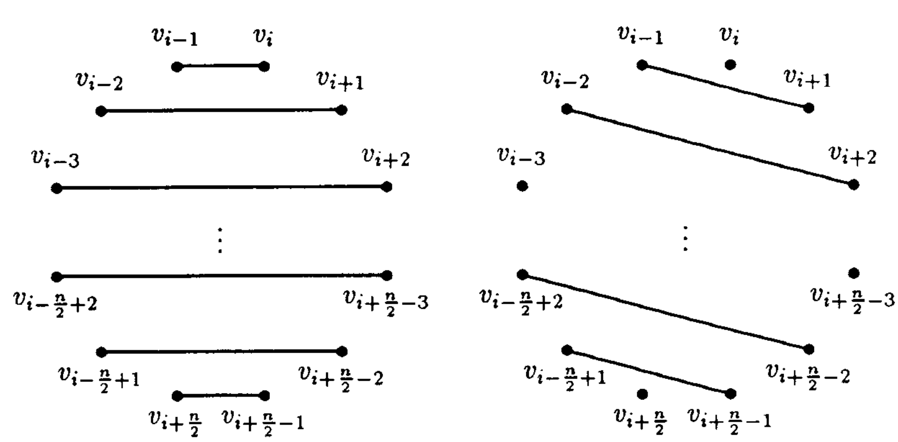

FIGURE 5. The way slopes are used for even n.

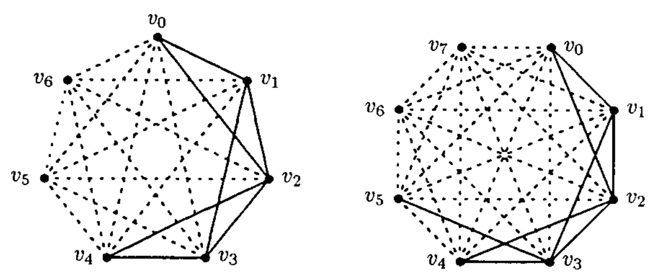

FIGURE 6. Representatives of parallel-sets in K_7 and K_8.

to the canonical drawing. We can select a representative from each parallel-set. We will then assign different slopes to the representatives. One possible selection of the representatives consists of the edges $\overline{v_0v_1}$, $\overline{v_0v_2}$, $\overline{v_1v_2}$, $\overline{v_1v_3}$, $\overline{v_2v_3}$, ... $\overline{v_{n/2-2}v_{n/2-1}}$, $\overline{v_{n/2-2}v_{n/2}}$, and $\overline{v_{n/2-1}v_{n/2}}$, plus $\overline{v_{n/2-1}v_{n/2+1}}$ if n is even, since they are from different parallel sets. Note the slopes of those edges are in a *cyclic decreasing order*, i.e. a cyclic shift of the decreasing order (see Figure 6).

Therefore we assign the slope in the decreasing order. We can sort the slopes first. The slopes are then assigned (in the decreasing order) to edges $\overline{v_0v_1}$, edge $\overline{v_0v_2}$, edge $\overline{v_1v_2}$, edge $\overline{v_1v_3}$, edge $\overline{v_2v_3}$, etc.

An algorithm to check whether a complete graph K_n can be drawn by using a set of n slopes or not is described below.

1. Sort the slopes in decreasing order.
2. Assign a slope (in the decreasing order) to the representative edges $\overline{v_0v_1}$, $\overline{v_0v_2}$, $\overline{v_1v_2}$, $\overline{v_1v_3}$, $\overline{v_2v_3}$, etc.
3. Assign the remaining edges a slope which is the same as the slope assigned to the representative of its parallel-set.
4. Solve the linear system.
5. If there is a nontrivial system then K_n can be drawn and the positions of the vertices are known. Otherwise, K_n cannot be drawn with the slope set.

The step 4 is the slowest part of the algorithm. In general, it takes $O(nm^2)$ time to solve an n by m matrix. The matrices we are solving here are $n(n-1)/2$ by $2n$. Thus the complexity of the algorithm is $O(n^4)$.

5. SUMMARY

We have shown the slope number of K_n is n. We have also proved that, in order for a K_n to be drawn with n different slopes, its vertices must be placed as vertices of a convex polygon. We have developed an algorithm which takes a complete graph with n vertices and a set of n slopes as its input and decides whether the graph can be drawn using only slopes in the set of slopes.

Several questions are yet to be answered. How to find the slope number of any graph? How to decide whether a graph with n vertices can be drawn by using only slopes in a given set of more than n slopes? Sometimes we have to preserve the topological relations among vertices, e.g. a certain vertex has to be on top of another vertex. How do we draw graphs with 'pretty' lines if topological constraints exist?

REFERENCES

Batini, C., Nardelli, E and Tamassia, R. (1986) A layout algorithm for data flow diagrams. *IEEE Trans. Software Eng.*, **12**, 538–546.

Battista, G., Liu, W. and Rival, I. (1990) Bipartite graphs, upward drawings, and planarity. *Information Processing Lett.*, **36**, 317–322.

Czyzowicz, J. (1989) Planar lattices and the slope problem. *ARS Combinatoria*, **27**, 101–112.

Czyzowicz, J. (1991) Lattice diagrams with few slopes. *J. Combin. Theory, Ser. A*, **56**, 96–108.

Czyzowicz, J., Pelc, A. and Rival, I. (1990a) Drawing orders with few slopes. *Discrete Math*, **82**, 233–250.

Czyzowicz, J., Pelc, A., Rival, I. and Urrutia, J. (1990b) Crooked diagrams with few slopes. *Order*, **7**, 133–143.

Knuth, D. (1984) The TeXbook. Addison-Wesley, Reading, MA.

Lamport, L. (1986) LaTeX: A Document Preparation System. Addison-Wesley, Reading, MA.

Lee, D. and Preparata, F. (1984) Computational geometry—a survey. *IEEE Trans. Comp.*, **33**, 1072–1101.

Ossanna, J. F. (1979) *Nroff/Troff User's Manual.* Publisher, Location?

Rowe, L. A. and Davis, M. (1987) A browser for directed graphs. *Software—Practice and Experience*, **17**, 00–00.

Tamassia, R., Battista, G. and Batini, C. (1988) Automatic graph drawing and readability of diagrams. *IEEE Trans. Syst. Man Cybernet.*, **18**, 61–79.