

California State University, San Bernardino

CSUSB ScholarWorks

Theses Digitization Project

John M. Pfau Library

2003

The embedding of complete bipartite graphs onto grids with a minimum grid cutwidth

Mário Rocha

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/etd-project>

 Part of the [Mathematics Commons](#)

Recommended Citation

Rocha, Mário, "The embedding of complete bipartite graphs onto grids with a minimum grid cutwidth" (2003). *Theses Digitization Project*. 2311.
<https://scholarworks.lib.csusb.edu/etd-project/2311>

This Thesis is brought to you for free and open access by the John M. Pfau Library at CSUSB ScholarWorks. It has been accepted for inclusion in Theses Digitization Project by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.

THE EMBEDDING OF COMPLETE BIPARTITE GRAPHS
ONTO GRIDS WITH A MINIMUM GRID CUTWIDTH

A Thesis

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Arts

in

Mathematics

by

Mario Rocha

June 2003

THE EMBEDDING OF COMPLETE BIPARTITE GRAPHS
ONTO GRIDS WITH A MINIMUM GRID CUTWIDTH

A Thesis

Presented to the
Faculty of
California State University,
San Bernardino

by

Mario Rocha


June 2003

Approved by:



Joseph Chavez, Committee Chair


Date



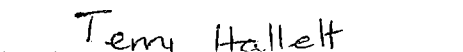
Roland Trapp, Committee Member



Joel Stein, Committee Member



Peter Williams, Chair
Department of Mathematics



Terry Hallett,
Graduate Coordinator
Department of
Mathematics

ABSTRACT

Algorithms will be demonstrated for how to embed complete bipartite graphs onto $2 \times n$ type grids, where the minimum grid cutwidth is attained. The algorithms that will be created will be utilizing a vertex formula found in Matt Johnson's paper together with some newly developed techniques. Johnson's vertex formula distributes the vertices of a graph evenly on a linear host, which will be modified to work on $2 \times n$ type grids. Specifically, algorithms will be demonstrated and proven for how to embed the $K_{1,n}$ and $K_{2,n}$ graphs into $2 \times n$ type grids, with the minimum grid cutwidth. In addition, we will show some embeddings for the $K_{3,n}$ graph. In general, we will utilize the algorithm for the $K_{2,n}$ graph to generalize an algorithm for the $K_{m,n}$ graph, for m even.

ACKNOWLEDGMENTS

I would like to acknowledge Dr. Joseph Chavez for being a great person and who has contributed not only his time, but also his great wisdom throughout my studies and in every project I have been involved in. His patience and guidance have made him a true mentor and advisor. I can truly say that Dr. Chavez has not only been a person who has inspired many students such as myself, but has been a role model to look up to and follow in his footsteps. Together with Dr. Chavez, I would also like to thank Dr. Rolland Trapp for being also a very strong supporter in my projects, and for serving as a mentor in the projects I have been involved with. As well, I would also like to thank Dr. Javier Torner for being such a great supporter of my education and for giving me the motivation to achieve the many goals I have strived to accomplish. Finally, I would like to give thanks to the rest of my committee for their time and support, as well as the entire Math Department, from faculty to staff to students, that I have had the pleasure of working with throughout these many years.

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGMENTS	iv
LIST OF TABLES	vi
LIST OF FIGURES	vii
CHAPTER ONE: INTRODUCTION	
Purpose of this Project	1
Technical Terms and Concepts	2
CHAPTER TWO: THE LEMMAS	
Lemma 1	12
Lemma 2	13
CHAPTER THREE: THE PROPOSITIONS	
Proposition 1	14
Proposition 2	22
Embedding the $K_{3,n}$ Graph	31
CHAPTER FOUR: THE THEOREMS	
Theorem 1	37
Theorem 2	50
CHAPTER FIVE: CONCLUSIONS AND RECOMMENDATIONS	
Conclusions	58
Recommendations	59
APPENDIX: TECHNICAL TERMINOLOGY	60
REFERENCES	63

LIST OF TABLES

Table 1. X-Values for $m=2$ and $n=3$	10
Table 2. X-Values for $m=1$ and Values of n	15
Table 3. X-Values for $m=2$ and Values of n	32

LIST OF FIGURES

Figure 1. General Graphs	2
Figure 2. Bipartite Graphs	3
Figure 3. Embeddings of $K_{2,4}$ onto Host 2×3 Grids ...	4
Figure 4. $K_{1,4}$ Embedded onto a 2×3 Grid	5
Figure 5. 2×3 Embeddings of $K_{2,4}$ Graph	6
Figure 6. Grid Notation	8
Figure 7. Adjacent Vertices	8
Figure 8. Positions of X	11
Figure 9. Linear Embedding	11
Figure 10. Connecting Adjacent Vertices	12
Figure 11. $K_{2,4}$ and $K_{4,2}$ Graphs	13
Figure 12. Partial Embedding of $K_{1,9}$ Graph	17
Figure 13. $K_{1,1}$ through $K_{1,13}$ Graphs	18
Figure 14. Case 1's Partial Embedding of $K_{1,n}$	20
Figure 15. Case 2's Partial Embedding of $K_{1,n}$	21
Figure 16. $K_{2,1}$, through $K_{2,8}$ Graphs	25
Figure 17. Case 1's Embedding of $K_{2,n}$, for n Even ..	28
Figure 18. Case 2's Embedding of $K_{2,n}$, for n Even ..	30
Figure 19. $K_{3,1}$, through $K_{3,7}$ Graphs	34
Figure 20. Case 1's Vertex Layout of $K_{m,n}$	39
Figure 21. Case 2's Vertex Layout of $K_{m,n}$	43
Figure 22. Case 3's Vertex Layout of $K_{m,n}$	46

Figure 23. Case 1's Vertex Layouts for n Odd	52
Figure 24. Case 2's Vertex Layouts for n Odd	54

CHAPTER ONE

INTRODUCTION

Purpose of this Project

During the spring and summer of 1988, I began a project with Dr. Chavez, Dr. Trapp, and Sara Hernandez, with the support of the McNairs Scholar's Program, at California State University, San Bernardino. The project was titled, "Graph Theory: The Embedding of Complete M-ary Trees into Grids as Means of Finding the Minimum Cutwidth" [8]. After successfully completing this project, I felt that research in this field of mathematics is an area that I had developed a great interest for. As a consequence, for this project I decided to pursue similar ideas using complete bipartite graphs. For this project we had decided to pursue a solution for the problem that through an algorithm the complete bipartite graph can be embedded in a grid with the minimum grid cutwidth. In this project we develop areas of the problem that will eventually point out the way to the solution. Many people such as Matt Johnson [6], Alvin Sacdalan [9], and Annie Wang [11], under the supervision and guidance of Dr. Joseph Chavez and Dr. Rolland Trapp, have pursued similar ideas as the ones of

this project. The reference page at the end will provide the necessary information if one would like to locate these papers. It is recommended but not a necessity that one reads these papers, as they will facilitate in understanding of the concepts and ideas that will be discussed in this project. However, this paper has been written so that any individual with some math background would be able to read and understand it. In addition, this paper alone could be read without the need of any background from other papers, as most ideas and terms have been defined and discussed.

Technical Terms and Concepts

A graph, $G = (V,E)$, consists of a finite set of vertices, V , and a finite set of edges, E , joining pairs of distinct vertices. For example, Figures 1a through 1c below represent graphs where vertices and edges are represented by points and lines.

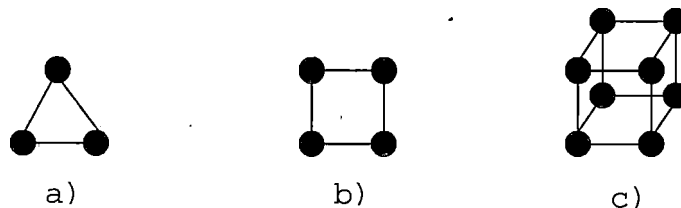


Figure 1. General Graphs

A bipartite graph consists of two disjoint sets of vertices, A and B , where $|A| = m$ and $|B| = n$, such that all edges connecting to vertices in set A connect to vertices in set B . In addition, no two vertices in the same set are connected by an edge. A complete graph, K_n , is a graph where each vertex is joined by an edge from all the other vertices. A complete bipartite graph, $K_{m,n}$, is a bipartite graph that is complete. Figures 2a through 2c below are examples of complete bipartite graphs.

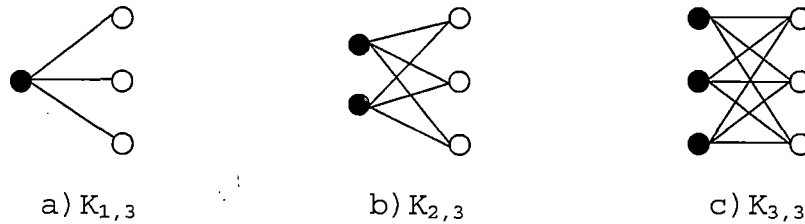


Figure 2. Bipartite Graphs

Embedding is the process of rearranging a graph's known form onto a host graph. For this project the only host graph we are interested in is a grid. Figure 3 demonstrates two ways that the complete bipartite graph, $K_{2,4}$, can be embedded onto a 2×3 grid.

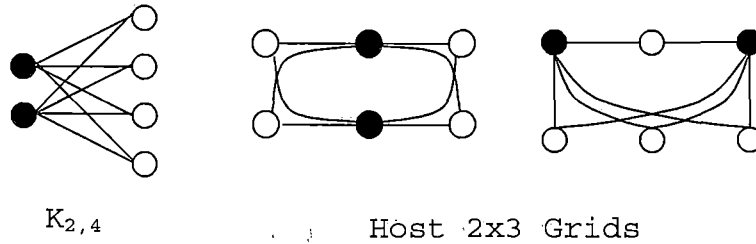


Figure 3. Embeddings of $K_{2,4}$ onto Host 2x3 Grids

Notice that when we embed our graphs onto a grid host, such as in Figure 3 above, we curve edges and avoid connecting to a vertex. Technically what we mean by such a description is that our edges are running vertically or horizontally but not connecting with the vertex being crossed over. Moreover, whenever a vh-edge is mentioned in this paper we mean a curved edge that runs vertically first and horizontally second, and similarly an hv-edge is one that runs horizontally first then vertically second.

When embedding graphs onto grids you will notice that the number of vertices of some graphs do not correspond to the number of vertices in the grid. For example, notice that if we embed a $K_{1,4}$ graph onto a 2x3 grid there will be an unused vertex. Whenever such an occurrence happens we will use a gray vertex. Thus, it should be understood that a gray vertex does not belong to the graph being embedded, but is a needed vertex to keep the grid's rectangular

shape. See the Figure 4 below for an example of an embedding of a $K_{1,4}$ graph onto a 2×3 grid.



Figure 4. $K_{1,4}$ Embedded onto a 2×3 Grid

Several parameters have been studied when embedding graphs onto host graphs, for instance, the bandwidth and the cutwidth. The bandwidth is the parameter where one tries to minimize the length of the longest edge in a host graph. In this project the parameter that we are interested in, while embedding onto host graphs, is the process of achieving a minimum cutwidth from all the different possible embeddings. To understand the cutwidth, it is defined as the maximum of all the cuts, where the cut is the number of edges running between adjacent vertices. Since our host graphs will only consists of grids, we will denote the host's cutwidth by the grid cutwidth. Moreover, in a grid, the cut can be split up as either being a vertical cut, if the two adjacent vertices lie on a vertical alignment, or a horizontal cut, if two adjacent vertices lie in a horizontal alignment. We will denote a

vertical cut by vcut and a horizontal cut by hcut to keep names short. Figures 5a through 5c below demonstrate different embeddings of the $K_{2,4}$ graph onto 2×3 grids, with different grid cutwidths. Notice that Figure 5a has only hcuts of 2 and vcuts of 2, 0, and 2, thus the grid cutwidth for this embedding is 2. Figure 4b has again hcuts of only 2 and vcuts 0, 4, 0, thus the grid cutwidth for this embedding is 4. Finally, Figure 5c has hcuts of 3, 1, 3, and 1 and vcuts 1, 2, and 1, thus the grid cutwidth for this embedding is 3. Notice from all three of the embeddings that Figure 5a the $K_{2,4}$ graph, has the minimum grid cutwidth. Thus the choice from among the three embeddings would be the one in Figure 5a.

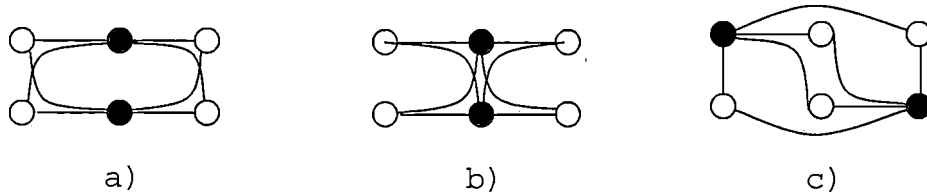


Figure 5. 2×3 Embeddings of $K_{2,4}$ Graph

In an embedding of a complete bipartite graph onto a host $2 \times n$ grid, the grid cutwidth of a graph can be measured using a counting technique, which we will call the cutwidth counting technique. This cutwidth counting technique only

requires the placement of the m vertices of set A of the $K_{m,n}$ graph on the grid without having to run any edges. The technique goes as follows: between any two columns of vertices count how many edges will cross through by looking at how vertices of set A connect to vertices of set B , from the left side of the grid to the right, and vice versa. Then divide the total by two, since there is only two ways to travel horizontally through the given column. This specific technique will give you all the expected hcuts throughout the grid. Using this same technique we can find all the expected vcuts. Thus, taking the maximum of the expected hcuts and vcuts one can determine the grid cutwidth of an embedding without having to run any edges. This cutwidth counting technique is especially important when we are determining the lower bounds for the embeddings of a particular graph.

The location of any vertex in a grid can be described by the coordinate (m,n) , where m is the row and n is the column of the vertex. See Figure 6 below for the locations of two specific vertices, $(1,1)$ and $(2,4)$ on a grid.

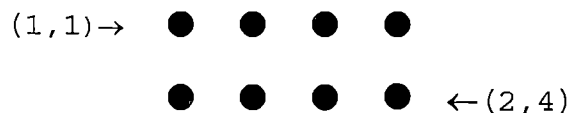


Figure 6. Grid Notation

Two vertices are adjacent if the entries of their coordinates have one entry the same and the other entry differs by one. For example, vertex $(1,1)$ is adjacent to vertices $(1,2)$ and $(2,1)$. See the Figure 7 below.

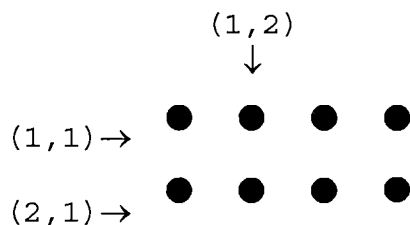


Figure 7. Adjacent Vertices

While embedding graphs on grids, we will utilize an algorithm from Matt Johnson's paper [2], for how to evenly distribute and embed vertices of a complete bipartite graph on a linear host graph. In addition, the algorithm also minimizes the cutwidth of the linear host graph, which is a particular area we are very interested in. Johnson's algorithm contains a vertex distribution formula, which lays vertices on a linear host graph evenly. The formula is described by $\lfloor (xm/m+n) + 1/2 \rfloor$, where given the position x

on a linear host graph, the formula tells you how many vertices of set A should be placed to the left of x . Figure 8 below shows what is meant by the positions x on a linear arrangement. Once the m vertices of set A and n vertices of set B have been placed on a linear host graph, we run edges from each of the m vertices of A to each of the n vertices of B, allowing us to have the minimum cutwidth in the linear host graph. We will call Johnson's vertex distribution formula by the vertex formula just to keep the wording short. Now let's look at an example that illustrates how the formula works. Keep in mind that a linear graph can be thought of as a $1 \times n$ grid. In addition, whenever we embed a complete bipartite graph we will always use black vertices to identify the vertices of set A and white vertices to identify the vertices of set B. Now let's embed a $K_{2,3}$ graph on a 1×5 grid. Table 1 below shows the calculations for the different positions x , when $m=2$ and $n=3$. Looking at the table, notice that at $x=1$ the formula yields 0, which means no black vertices will be placed on the vertices located on the left of $x=1$. When $x=2$ the formula yields 1, which means that one black vertex must be on the left of $x=2$. Therefore, we can assume that it must be the vertex in position $(1,2)$, since we had

already concluded that the vertex (1,1) could not be a black vertex. Moreover, at $x=3$ the formula yields 1, which means that a black vertex will not be placed in the position (1,3) of 1×5 grid. At $x=4$ the formula yields a 2, which means the vertex in position (1,4) will be the position of the next black vertex. Finally, at $x=5$ the formula yields 2, which again means that two black filled vertices must be placed to the left of the position $x=5$, which are actually located on the positions (1,2) and (1,4). Figure 9 shows the placement and the embedding of the $m=2$ and $n=3$ vertices of set A and B of the $K_{2,3}$ graph when using vertex formula. Note that this vertex formula could also be made to apply to grids with multiple rows by simply applying the formula to each individual row. The only thing that the algorithm does not give is how to run edges between the different rows or how many vertices of set A to include on each row. Therefore this paper will concentrate mainly in describing how to achieve such a task while achieving the minimum grid cutwidth.

Table 1. X-Values for $m=2$ and $n=3$

	X=1	X=2	X=3	X=4	X=5
$m=2, n=3$	0	1	1	2	2

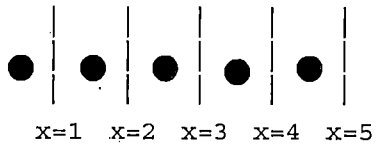


Figure 8. Positions of X

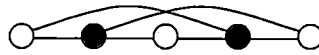


Figure 9. Linear Embedding

CHAPTER TWO

THE LEMMAS

Lemma 1

For any graph embedded in a host graph, the contribution of the cutwidth, from all vertices connecting to adjacent vertices is at most one.

Proof:

Since no overlapping will ever be achieved by running edges from two adjacent vertices, then cutwidth will only be increased by 1. See Figure 10 for an example.

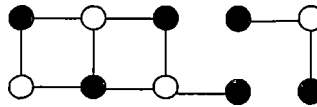


Figure 10. Connecting Adjacent Vertices

In any embedding onto a grid the first set of edges being run are always to adjacent vertices. As Lemma 1 will point out the cutwidth will only increase by 1, and for this reason it is the first and most crucial step to begin with while embedding. \square

Lemma 2

The $K_{m,n}$ graph is isomorphic to a $K_{n,m}$ graph.

Proof:

It is obvious that if we interchange m and n then the graphs $K_{m,n}$ and $K_{n,m}$ are isomorphic, simply because the m vertices of set A are connecting to the n vertices of set B in the same manner that n vertices of B connect to the m vertices of A . \square

Since the embeddings of isomorphic graphs are simply the rearrangements, then it is not very hard to see that the embeddings must also be isomorphic. See Figure 11 below for an example of a $K_{2,4}$ and $K_{4,2}$ and notice that no matter which graph you have, two vertices are connected to four vertices, despite the fact that the graphs have color differences.

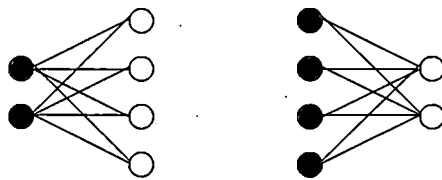


Figure 11. $K_{2,4}$ and $K_{4,2}$ Graphs

CHAPTER THREE

THE PROPOSITIONS

Proposition 1

The complete bipartite graph, $K_{1,n}$, can be embedded in a:

$2x[(n+2)/2]$ grid, for n even, and

$2x[(n+1)/2]$ grid, for n odd,

with a minimum grid cutwidth of $\lceil n/3 \rceil$.

Proof:

We will show that this proposition holds true by describing an algorithm that demonstrates how to embed the $K_{1,n}$ graph onto the specified grid sizes described in our proposition. To begin, there will be a few things to keep in mind as we demonstrate this algorithm. First, we will use the Vertex Formula to place our single black vertex in the second row only. Since the grid has symmetry, it would be the same to place the black vertex on either the first row or the second. The calculations using the Vertex Formula have been provided in Table 2 below to assist in the placement of the black vertex on the second row, for the different embeddings.

Table 2. X-Values for $m=1$ and Values of n

	$n=1$	$n=2$	$n=3$	$n=4$
$x=1$	1	0	0	0
$x=2$	1	1	1	0
$x=3$		1	1	1
$x=4$			1	1
$x=5$				1

Second, if in any embedding there is a vertex that will be left unused, illustrated by the gray vertex, then we will always place it at one end of the second row. If n is not a multiple of four then we will place the gray vertex in position $(2,1)$, otherwise it is placed in the other corner of the second row. What one may notice is that this gray vertex will alternate between the corners of the second row for the consecutive embeddings that contain this gray vertex.

Now let's show that $\lceil n/3 \rceil$ will generate the minimum grid cutwidth, for the embeddings of the $K_{1,n}$ graph. For $n \geq 4$ the Vertex Formula will not place the black vertex on a corner, but in a location closest to the center of the second row. So it is easy to see that the black vertex will only have three paths from which to run edges to the n white vertices. Since the black vertex has three ways to run n edges onto n white vertices, then evenly distributing

the n edges among the three possible directions will yield a lower bound of $n/3$. Now unless n is a multiple of 3, $n/3$ will not equal a whole number. Thus, setting the lower bound to $\lceil n/3 \rceil$ will secure that one side might have one more or less of an edge over the other two sides, since the n edges will not evenly distribute among the three ways. Moreover, for $n \leq 3$, the black vertex lies on a corner, but fortunately enough the lower bound still applies. Looking at the cutwidths of Figures 13a through 13c one can see that the lower bound does verify for $n \leq 3$.

Let's begin by demonstrating our algorithm by embedding the $K_{1,9}$ graph onto a 2×5 host grid. Since the second row is our choice of location for the black vertex, then Table 2 above tells us that it will be placed in position $(2,3)$. Using Lemma 1 we run the first set of edges, which run from the black vertex to vertices $(2,2)$, $(1,3)$, and $(2,4)$. Then we run edges along the vertices of the second row. So far, the max hcut of 2 is found between the vertices adjacent to the black vertex on the second row. See the Figure 12 below.

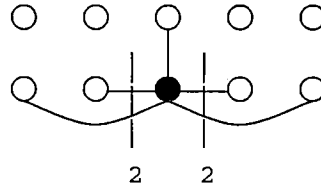


Figure 12. Partial Embedding of $K_{1,9}$ Graph

Now from this point we will concentrate on running edges up to the first row utilizing the fact that we know the lower bounds for the embedding of this cutwidth to be 3. Now let's start by running edges onto the vertices of the first row, located to the right side of the black vertex. We will only run one hv-edge, since the cut between the black vertex and vertex (2,4) will have the required lower bound of 3, as a consequence. The rest of the edges going to the right hand side of the first row will be vh-edges. Utilizing the same strategy on the left hand side of the first row we can achieve similar results. As a consequence, this strategy will always result in the vcut, between the black vertex and the vertex above, to be less than or equal to the hcuts next to the black vertex. See Figure 13i for the complete embedding of this example. Figure 13 below shows a complete list of embedding from $K_{1,1}$ to $K_{1,13}$ so that one could verify this algorithm for a few cases.

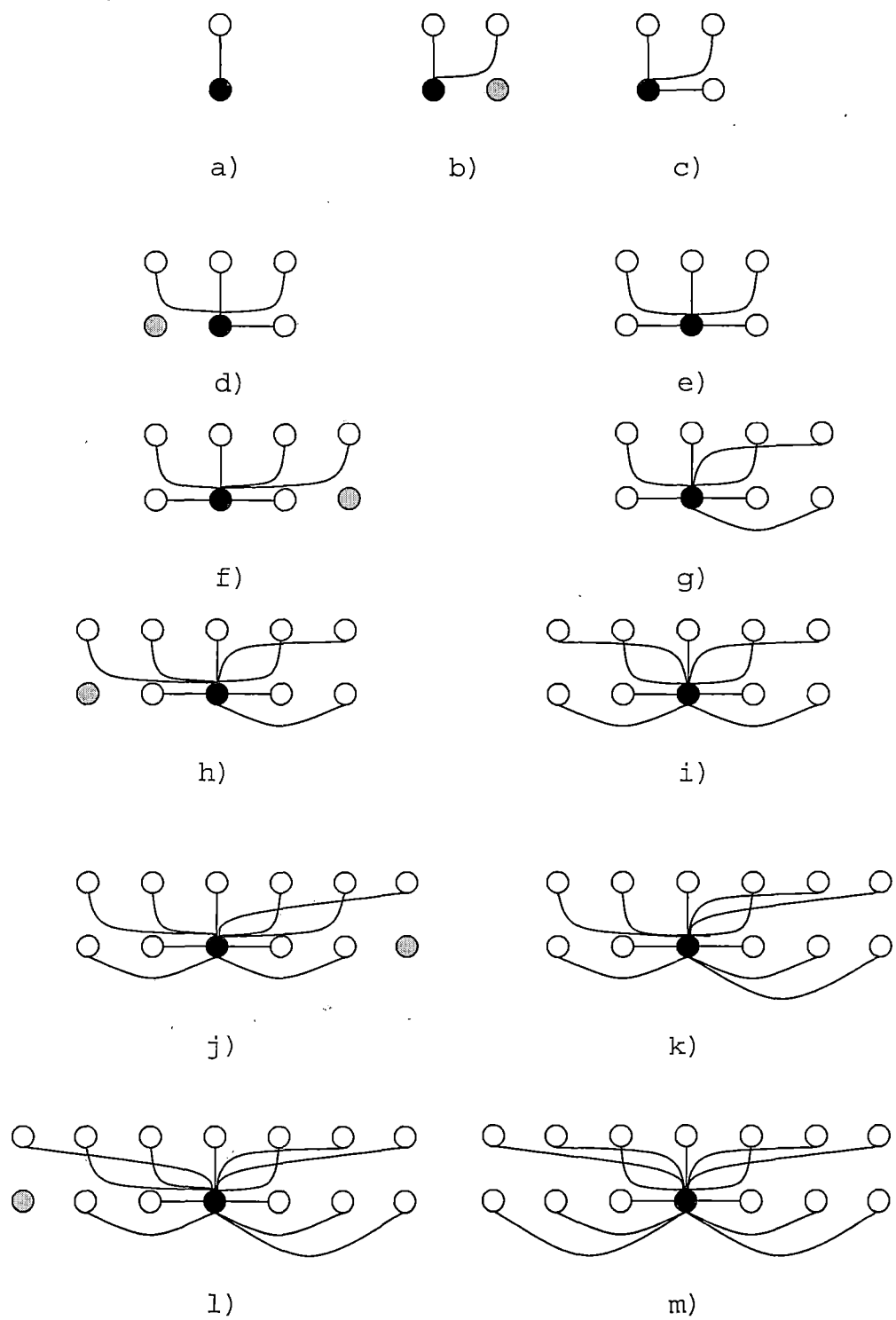


Figure 13. $K_{1,1}$ through $K_{1,13}$ Graphs

Now let's prove that this algorithm will work in general. We will split the proof into four cases but will only prove two in full, since the other two cases follow by the same idea. In each case we will verify that the vcut and the hcuts next to the black vertex will achieve a minimum grid cutwidth of $\lceil n/3 \rceil$. To keep things short when performing any calculations we will let the lower bound $\lceil n/3 \rceil = L$.

Case1: When $n = 4x+1$, for $x = 1, 2, 3, \dots$

The $K_{1,n}$ Graph will be embedded in a $2 \times \lceil (n+1)/2 \rceil$ grid.

Given the conditions on n and the particular grid size, the Vertex Formula will place the black vertex in the most center position of the second row. The second row will contain $(n-1)/4$ white vertices on the left and right sides of the black vertex. Running edges along the first row will yield max hcuts of $(n-1)/4$ next to the black vertex. See the Figure 14 below.

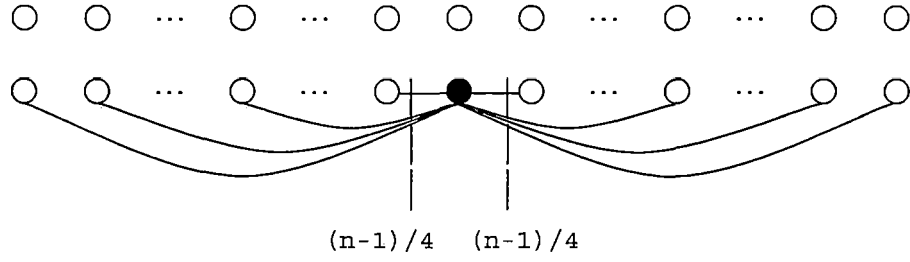


Figure 14. Case 1's Partial Embedding of $K_{1,n}$

Now in order to max the hcuts, up to the lower bound L , we simply run two sets of $(4L-n+1)/4$ hv-edges up to the left and right side of the first row from the black vertex. This means that there will be a total of $(4L-n+1)/2$ vertices occupied by the hv-edges on the first row and $n-2L$ vertices left in the first row to run vh-edges. Thus, running vh-edges will yield a vcut of $n-2L$. If we do a quick check, we can see that the vcut is less than the hcuts:

$$\begin{aligned}
 3\lceil n/3 \rceil &\geq n \\
 \Rightarrow 3L &\geq n \\
 \Rightarrow 2L + L &\geq n \\
 \Rightarrow L &\geq n - 2L.
 \end{aligned}$$

This implies that the minimum grid cutwidth is $L = \lceil n/3 \rceil$.

Case 2: When $n = 4x-1$, for $x = 1, 2, 3, \dots$

Using the same strategy as above, we use the Vertex Formula and place the black vertex on the most center position on the second row. Notice that in this case the black vertex have $(n+1)/4$ white vertices on the right and $(n-3)/4$ white vertices on the left. See Figures 13c, 13g, and 13k for specific examples for which this case will be proving. Now running edges along the second row yields max hcuts of $(n+1)/4$ and $(n-3)/4$ on the right and left. See Figure 15 below for a description.

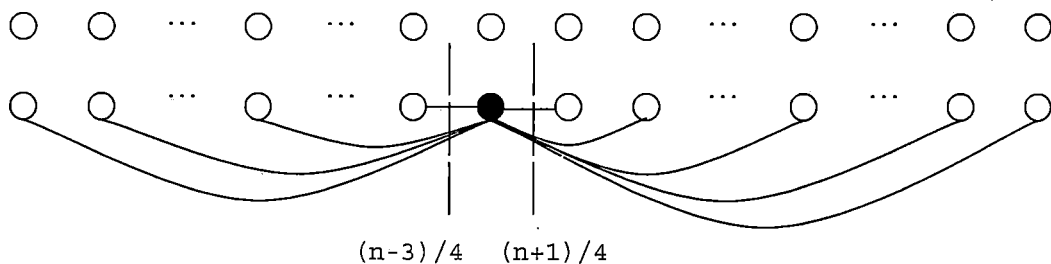


Figure 15. Case 2's Partial Embedding of $K_{1,n}$

Now if we run $(4L-n-1)/4$ hv-edges onto the left side and $(4L-n+3)/4$ hv-edges onto the right side of the first row, from the black vertex, we will max the hcuts up to the lower bound. Since there will be $(4L-n-1)/4 + (4L-n+3)/4$ vertices occupied on the first row, doing the proper subtraction from the $(n+1)/2$ total vertices available, yields once more $n-2L$ vertices are left to run vh-edges.

So our vcut once again comes out to be $n-2L$, and we could just use the same reasoning as Case 1 to show that the hcuts are greater than or equal to the vcut. Thus again our lower bound $L = \lceil n/3 \rceil$ is our minimum grid cutwidth.

Using the same technique one can verify the other two cases, when $n = 4m$ and when $n = 4m-2$, where gray vertices would be present. Therefore, in general we can see that this algorithm yields the minimum grid cutwidth of $\lceil n/3 \rceil$, concluding this proof. \square

Proposition 2

The complete bipartite graph, $K_{2,n}$, can be embedded in a:

$2 \times \lceil (n+2)/2 \rceil$ grid, for n even,

$2 \times \lceil (n+3)/2 \rceil$ grid, for n odd,

with minimum grid cutwidths of:

$n/2$, for n even,

$(n+1)/2$, for n odd.

Proof:

We will demonstrate an algorithm which shows how to embed the $K_{2,n}$ graph onto the specified grid dimensions, with a the minimum grid cutwidths. Let's demonstrate how this algorithm breaks down.

First, notice that the $K_{2,n}$ graph has two black vertices. What we will do is place one black vertex in the first row and the other in the second row. This way the vertices can evenly distribute edges onto the other n white vertices of the grid keeping the grid cutwidth to a minimum. Using the Vertex Formula we place one black vertex on the first row, and the other on the second row. However, when placing a black vertex on the first row we will reverse the orientation of the Vertex Formula, by starting the x -values from the right hand side opposed to the left. With such a technique one will notice that the black vertices would either laid on top of or slightly offset from each other. Again, Table 2 from Proposition 1 can be used to assist in the placement of the black vertices, for the different embeddings.

Second, our embedding process will go as follows: we will only run hv -edges in our embedding when having to connect edges from the black vertex in one row to white vertices in the opposite row. This process makes sense since we will be running the same number of edges from the first row to the second row and vice versa, so there is no need to run vh -edges at all. In addition, it allows us to easily determine the cuts and more importantly the minimum

grid cutwidth of the general embeddings. For some embeddings we will actually run a vertical edge from the black vertex to an adjacent vertex on the opposite row. This edge will not contribute to the overall grid cutwidth of the embeddings, since we will determine later that the minimum grid cutwidth is an hcut within the rows.

Third, we will only create embedded host graphs for the $K_{2,n}$ graphs, for n even, and utilize those embeddings to create the graphs, for n odd. The process will simply consist in the removal of edges connecting to the vertex in position $(1,1)$, from the n even cases, which yields the embeddings for the n odd cases. Thus the gray vertex will always be found in position $(1,1)$, for n odd cases.

Now let's embed a specific graph and show how the algorithm works. Let's embed the $K_{2,8}$ graph onto a 2×5 grid. Using the Vertex Formula and its calculations in Table 2, we see that the black vertices will lay on top of each other and in positions $(1,3)$ and $(2,3)$. If we run edges along the individual rows, from the black vertices, the max hcut so far will be 2, since two white vertices are positioned to the left and right of the black vertices. Finally, let's run hv-edges between the rows, and the result can be seen in Figure 16h. Notice that the grid

cutwidth for this embedding is 4, which checks with the minimum grid cutwidth claimed by this proposition. Now let's show how to derive the embedding of the $K_{2,7}$ graph from the embedding of the $K_{2,8}$ graph. If we eliminate all the edges connecting onto the white vertex (1,1) in the embedding of the $K_{2,8}$ graph we will achieve our desired result. See Figure 14g for the result of the embedding of the $K_{2,7}$ graph. Notice that this embedding also has a minimum grid cutwidth of 4, which again checks with the claim of this proposition. In actuality, this will always be the result with such a technique in this proposition. Figures 14a through 14f have been provided as verifications of this algorithm for the cases $n=1$ through $n=6$.

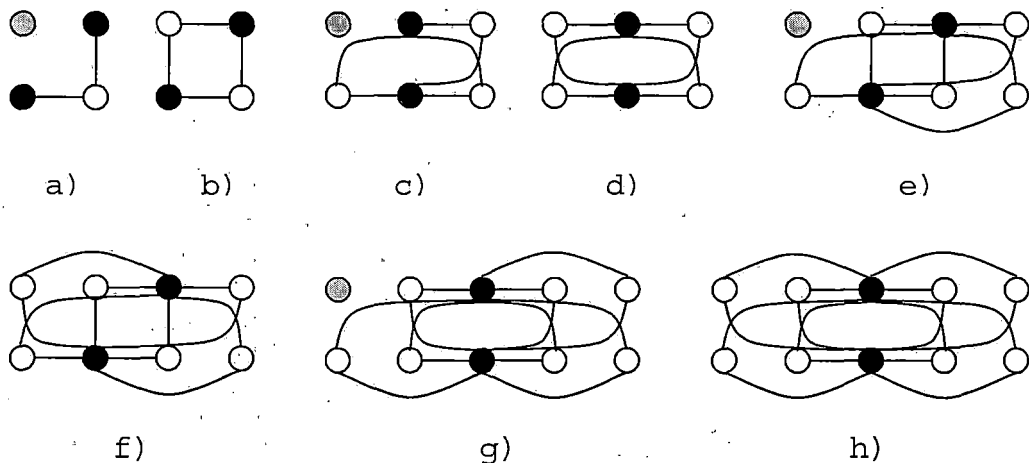


Figure 16. $K_{2,1}$, through $K_{2,8}$ Graphs

Let's demonstrate how this algorithm works in general, and verify that our embedding process achieves the minimum grid cutwidth. We will split the proof into two cases. Case 1 will prove the case when the black vertices lay on top of each other, just like Figure 16h. Case 2 will prove the case when the black vertices lay slightly offset from each other, just like Figure 16f. So we have determined that when $n = 4x$, for $x = 1, 2, 3, \dots$ then $K_{2,n}$ graphs fall under Case 1, and when $n = 4x-2$, for $x = 1, 2, 3, \dots$ then $K_{2,n}$ graphs fall under Case 2.

First let's prove that the lower bound for the n even cases is $n/2$. Now each of the black vertices will be located in the most central location of the grid, where only two directions are available from which to send edges up to white vertices. Now each black vertex has a total of $n/2$ white vertices to connect to, on both the left and right hand side. So there are two directions from which edges can travel from each black vertex to white vertices, either through top or bottom. See the calculation below, which yields the lower bound for this case.

$$LB = [1(n/2) + 1(n/2)]/2$$

$$\Rightarrow LB = [n/2 + n/2]/2$$

$$\Rightarrow LB = n/2$$

In the same way, we can determine the lower bound for n odd cases. Notice that when n is odd the grid dimensions will include an extra vertex to keep the grid's rectangular shape. Remember that we use gray vertex to identify this specific vertex, which will be located somewhere in the grid. Let's place the gray vertex on the left hand side of the grid, and calculate the lower bound for such cases. Since the gray vertex is located on the left side then the max hcut will be found on the immediate right of the black vertices. Each black vertex will connect to $(n+1)/2$ white vertices on its left in only two ways. Performing a similar calculation as was done previously the lower bound will come out to $(n+1)/2$. See the calculation below.

$$LB = [2(n+1)/2]/2$$

$$\Rightarrow LB = [2(n+1)]/4$$

$$\Rightarrow LB = (n+1)/2$$

Case 1: When $n = 4x$, for $x = 1, 2, 3, \dots$

Let's embed the $K_{1,n}$ graph into a $2x[(n+2)/2]$ grid. Keep in mind that in this case the black vertices are in the most

central position. Using the Vertex Formula we find that the positions of the black vertices will be $(1, (n+4)/4)$ and $(2, (n+4)/4)$. On the left and right hand sides of both rows, we will find $n/4$ white vertices. Since our embedding process will be the same when running edges from one row onto the other we will keep things short by focusing on the second row only. But keep in mind that whatever is done to the second row will have to be done exactly to the first row. If we run edges along the row the max hcut will be $n/4$ at this point, on the immediate hcuts of black vertex. Now running hv-edges from the black vertex onto the opposite row will add an extra $n/4$ edges onto the immediate hcuts of the black vertex. Thus this brings the to immediate hcuts of the black vertex to a max of $n/2$, which checks with this proposition's grid cutwidth. See Figure 17 below for an illustration of this general embedding.

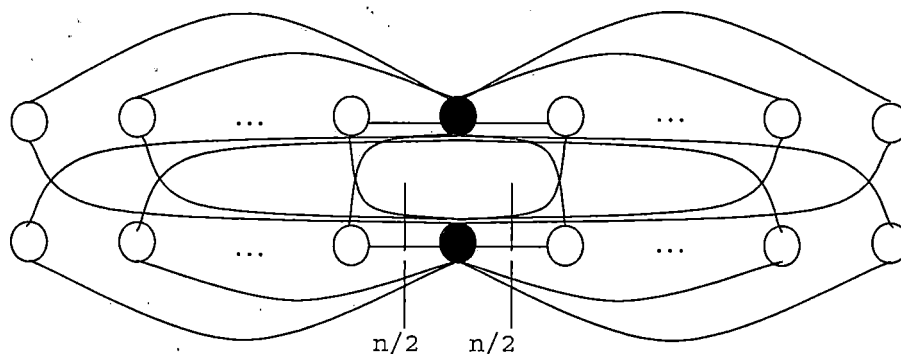


Figure 17. Case 1's Embedding of $K_{2,n}$, for n Even

Now using the result of this general embedding above we can obtain the $K_{2,n-1}$ graph by removing the edges connecting to vertex $(1,1)$. Since the removal of the edges from vertex $(1,1)$ does not affect the right side of the general embedding above, then the embedding $K_{2,n-1}$ graph will have the same max hcut as the embedding of the $K_{2,n}$ graph.

Case 2: When $n = 4x-2$, for $x = 1,2,3,\dots$

This case is very similar to Case 1, although the only difference is in the positioning of the black vertices by the Vertex Formula. For this case the black vertices are in positions $(1, (n+6)/4)$ and $(2, (n+2)/4)$ in the first and second row. We will find $(n+2)/4$ on the left and $(n-2)/4$ white vertices on the left and right hand sides of the first row. The same will happen for the black vertex on the second row, but in the reverse order. Again, by the symmetry of the grid, we will just focus on the second row. If we run edges along the row the hcuts at this point will be $(n-2)/4$, on the immediate hcut on the left and $(n+2)/4$ on the immediate hcut on the right of the black vertex. Running hv-edges between the rows we will have added an extra $(n+2)/4$ edges on the immediate left and $(n-2)/4$ on the immediate right side of the black vertex. Thus this

brings the hcut to a max of $n/2$ on the right of the black vertex. See the Figure 18 below for a description of this general embedding.

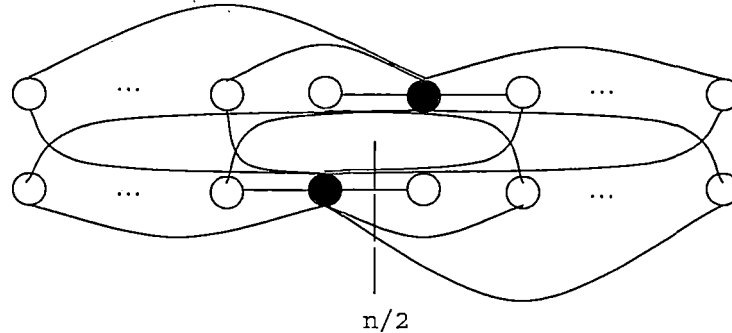


Figure 18. Case 2's Embedding of $K_{2,n}$, for n Even

Now using this general embedding above we can obtain the prior embedding, the $K_{2,n-1}$ graph, by removing the edges connecting to vertex $(1,1)$. Again, this embedding will also have the same max hcut as the above general embedding.

Finally, let's show that the max vcut is less than or equal to the max hcut. Now the max vcut will be 2 located between any two vertically aligned white vertices, and the max hcut is $n/2$. Let's do a quick check.

$$2 \leq n/2$$

$$4 \leq n$$

As we can see $n/2$ is greater than or equal to 2 when $n \geq 4$.

For $n < 4$, see the specific embeddings in Figure 16a

through 16c as a verification that the max vcut is less than or equal to the max hcut in each embedding.

Thus, we have shown in Case 1, Case 2, and the verification above that minimum grid cutwidth is $n/2$ for the Embedding of the $K_{2,n}$ graph onto the specified grid sizes. \square

Embedding the $K_{3,n}$ Graph

Figures 19a through 19g below show the embeddings for the graphs $K_{3,1}$, through $K_{3,7}$. Now the $K_{3,1}$ graph in Figure 19a was derived using Lemma 2 from the embedding of the $K_{1,3}$ graph in Proposition 1. The embeddings for the $K_{3,n}$ graph, where n is odd, were created using an algorithm. The embeddings for the $K_{3,n}$ graph, where n is even, were derived by removing the edges of a vertex that was located in a different location on the first row for different embeddings. For the embeddings of $K_{3,4}$ through $K_{3,7}$ graphs in Figures 19d through 19g, the embeddings were split up into three individual parts so as to facilitate visualization of the embedding algorithm. Moreover, one may notice that the gray vertex for some embeddings can be found in the interior part of the grid, in comparison to Propositions 1 and 2, where it was mainly located on

corners. Now let's describe the algorithm we utilized to embed the $K_{3,n}$ graphs for n odd and $n \leq 15$. First, we will always place two black vertices on the second row and one black vertex on the first row positioning them using the Vertex Formula. Table 3 below and Table 1 from Proposition 1 will provide all the calculations needed to assist in the layout of these black vertices.

Table 3. X-Values for $m=2$ and Values of n

	$n=1$	$n=2$	$n=3$	$n=4$
$x=1$	1	1	0	0
$x=2$	1	1	1	1
$x=3$	2	2	1	1
$x=4$		2	2	1
$x=5$			2	2
$x=6$				2

Second, when running edges from the single black vertex onto the second row we will only run hv-edges. In addition, when running edges from the two black vertices on the second row onto white vertices on the first row we will alternate using hv-edges and vh-edges. The alternation can be described as follows: first, the black vertex farthest to the left on the second row will run hv-edges onto the white vertices located on its left and right on the closest pair. The next farther pair of white vertices will be

connected using vh-edges and at this point is where alternation takes place. The other black vertex farthest to the right on the second row will do just the opposite, where vh-edges are run first, then hv-edges. Now to get the cases where m is odd we would remove the edges from a particular white vertex located on the first row. One can determine the location by comparing to consecutive embeddings where n is odd, and noticing which column was removed on the larger embedding to derive the smaller embedding. This column points out the location on the first row where one should place the gray vertex. See Figures 19g and 19e below, and notice that the first column in Figure 19g removed yields 19e. Also it can be seen that the gray vertex for Figure 19f is located on the first vertex of the first row. Moreover, we have verified this algorithm ourselves up to $n \leq 15$, except when $n=5$. In particular, one can verify the algorithm provided in Figure 19 for the embeddings of the $K_{3,2}$ through $K_{3,7}$ graphs, with the exception of the $K_{3,5}$ graph.

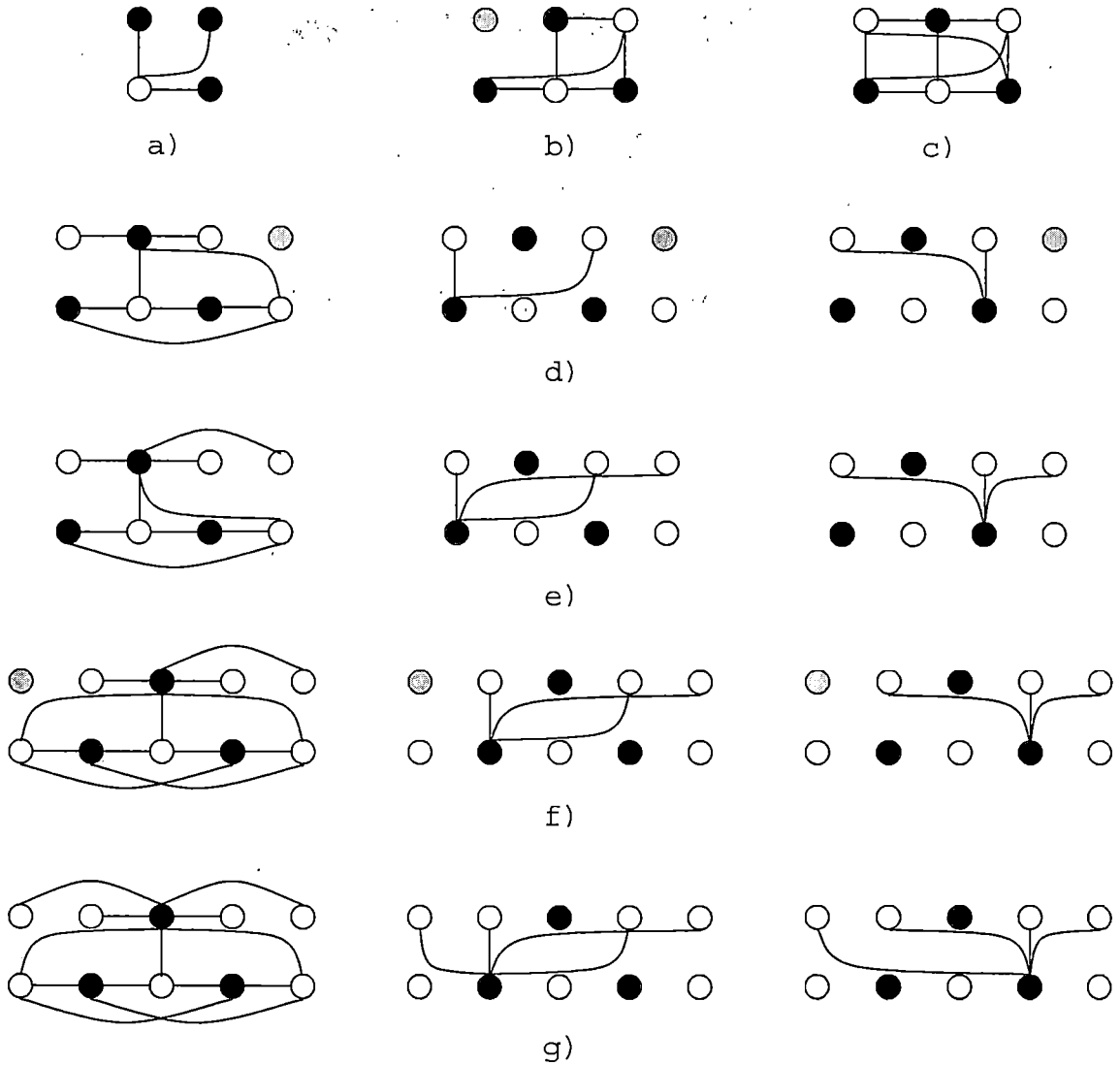


Figure 19. $K_{3,1}$, through $K_{3,7}$ Graphs

This algorithm seems like it will go farther than $n=15$ and might actually be the algorithm that does the embedding job, but describing the grid cutwidth, as we found, could be very tricky. The only embedding that did not fit the criteria of this algorithm was the embedding for the $K_{3,5}$

graph displayed in Figure 19e above, where we modified the algorithm for one particular edge. Notice that the single black vertex on the first row actually runs a vh-edge, contrary to the algorithm, which said that only hv-edges would be utilized when connecting to white vertices on the second row from this black vertex.

Let's show that the algorithm we demonstrated above actually yields the minimum grid cutwidth for the embeddings of Figure 19. In particular, let's show that the embedding of the $K_{3,7}$ graph in Figure 19g does in fact have the minimum grid cutwidth. Keep in mind that the same argument we will use for the embedding of the $K_{3,7}$ graph will work for the rest of the embeddings. Now the embedding of the $K_{3,7}$ graph has a grid cutwidth of 5, if we check all the cuts in Figure 19g. Let's show that the lower bound is in fact 5. Now there are 2 central locations to start from, where we could place a vertical line and use the cutwidth counting technique to find the lower bound. So let's choose the one farthest to the left, which cuts the embedding into 1 black and 3 white on the left side and 2 black and 4 white on the right side of the vertical line. Calculating hcut will yield:

$$\text{hcut} = [1(4)+2(3)]/2$$

$$\Rightarrow \text{hcut} = 5.$$

Now suppose we interchange black and white vertices about the vertical so that white vertices from the left interchange with black vertices from the right. The outcome can be categorized into two cases. One case will be when 2 black and 2 white vertices are on the left and 1 black and 5 white vertices are on the right side. The second case will be when 3 black and 1 white vertices are on the left and 0 black and 6 white vertices are on the right of the vertical line. The calculations below will determine the hcuts for each case at the vertical line.

$$\text{hcut} = [2(5)+1(2)]/2$$

$$\Rightarrow \text{hcut} = 6$$

and

$$\text{hcut} = [3(6)+0(1)]/2$$

$$\Rightarrow \text{hcut} = 9$$

As we can see, it is clear that the lower bound must be 5, which matches with the grid cutwidth of the embedding of the $K_{3,7}$ graph.

CHAPTER FOUR

THE THEOREMS

Theorem 1

For m and n even, the Complete Bipartite Graph, $K_{m,n}$, can be embedded in a $2 \times [(n+m)/2]$ grid, with the minimum grid cutwidth of $mn/4$.

Proof:

One may notice that this theorem partially generalizes Propositions 2 of Chapter 2 for the m and n even cases. In actuality, Proposition 2 was meant to give a better understanding once we arrive to this generalization. So the same algorithm will be applied to this general theorem as a consequence. Let's recall the main parts of the algorithm that apply to this generalization. First, we will only use the algorithm to embed the cases where $m \leq n$. Recall that the embeddings for $K_{m,n}$ when $n < m$ are derived using Lemma 2 from the embeddings of $K_{n,m}$ where $n < m$. Second, the m black vertices of $K_{m,n}$ will be split in half, such that $m/2$ vertices will be distributed evenly along on each row, using the Vertex Formula. Remember that we reverse the orientation of the Vertex Formula for the first row by starting from the right hand side opposed to

starting from the left for the second row. Lastly, if edges will travel between the rows then only hv-edges will be used.

We will break the proof into three major cases. For each major case our proof will consist in showing that the lower bound for the grid cutwidth will correspond to the same value as the upper bound for our embedding algorithm. One will notice after going through each of the major cases that the most central location of the layout plays an important role when determining the grid cutwidth of an embedding. For this reason, for each of the figures, in each major case, a vertical line has been placed in the most central location of the layout of vertices. Keep in mind that the vertical line is not part of the embedding, but that it only serves a purpose when performing calculations.

Case 1

We will prove the cases where the layout of the black vertices on the rows appears slightly offset. This particular vertex layout occurs when both n and m are multiples of four or when both n and m are not multiples of

four. See Figures 20a and 20b below for the general descriptions of these layouts.

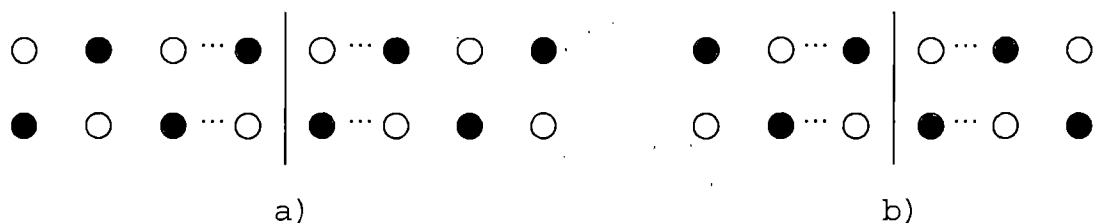


Figure 20. Case 1's Vertex Layout of $K_{m,n}$

Now for each layout in Figure 20 the vertical line splits the grid in half where there are $m/2$ black and $n/2$ white vertices on both sides. Using the cutwidth counting technique let's determine the hcut at the location where the vertical line has been placed for of the layouts above. So $m/2$ black vertices on the left will connect to $n/2$ white vertices on the right with edges, and the same will happen in the opposite direction. In addition, edges connecting black and white vertices only have two ways from which to travel through the vertical line. So the calculation performed below, using the cutwidth counting technique, will determine the hcut for the layouts at the vertical line in Figure 20 above.

$$hcut = [(m/2) * (n/2) + (m/2) * (n/2)] / 2$$

$$\Rightarrow hcut = [mn/4 + mn/4] / 2$$

$$\Rightarrow hcut = [2mn/4] / 2$$

$$\Rightarrow hcut = mn/4$$

Now let's show that the lower bound for the grid cutwidth cannot be less than $mn/4$ for each layout in Figure 20. Suppose that for each of the layouts we interchanged black vertices from the left hand side of the vertical line with white vertices from the right. Notice that it would be the same, if we did such a thing in the opposite direction. Let i represent the number of these particular switches of black and white vertices about the vertical line. Notice that with each switch there will be less black and more white vertices on the left and the opposite effect happens on the right. Although, the switching leaves the number of vertices on both sides of the vertical line unchanged. So there will be $m/2-i$ black and $n/2+i$ white vertices on the right, and $m/2+i$ black and $n/2-i$ white vertices on the left of the vertical line. Now let's determine the $hcut$ at the vertical line for each switch. Remember that the edges connecting black and white vertices travel through the vertical line in only two ways. So the

calculation below, using the cutwidth counting technique, determines the hcut at the vertical for each specific switch.

$$\text{hcut} = [(m/2-i)*(n/2-i) + (m/2+i) * (n/2+i)]/2$$

$$\Rightarrow \text{hcut} = [2mn/4 + 2i^2]/2$$

$$\Rightarrow \text{hcut} = mn/4 + i^2$$

As we can see from the result of the calculation, when $i=0$ the smallest hcut will be $mn/4$. Thus the lower bound for both of these layouts is found to be $mn/4$.

Now let's show that the upper bound for the grid cutwidth of our embedding process, for each layout in Figure 20, will be at most $mn/4$. Remember that in our embedding process only hv-edges are used when connecting vertices from one row onto the other row. So let's calculate the hcut on the second row about the vertical line. Notice it would be the same if have chosen the first row, because of the symmetry of the layout and our embedding process. Now $m/4$ black vertices on the left of the vertical line will connect to $n/2$ white vertices on the right side of the vertical line, through either a horizontal edge or a hv-edge. The same could be said with the $m/4$ black vertices located on the right side of the vertical line. So if we calculate the hcut at the vertical

line for each layout, using the cutwidth counting technique, we get that the:

$$hcut = (m/4)(n/2) + (m/4)(n/2)$$

$$\Rightarrow hcut = mn/8 + mn/8$$

$$\Rightarrow hcut = mn/4.$$

Now let's show that if we move the vertical line into other locations then the hcut is less than or equal to $mn/4$. Let i represent the number of black and j the number of white vertices that get switched from one side of the vertical to the other side as a consequence of moving the vertical line. Let's shift the vertical line towards the right and calculate the result, and make a note that the same can be done if we would shift it towards the left. Shifting the vertical line towards the right implies that there are $m/4+i$ black and $n/2+j$ white vertices on the left side, and $m/4-i$ black and $n/2-j$ white vertices on the right side of the vertical line. Now if we calculate the hcut, using the cutwidth counting technique, for each particular shift of the vertical line, we get that the hcut will be:

$$hcut = (m/4+i)*(n/2-j) + (m/4-i)*(n/2+j)$$

$$\Rightarrow hcut = 2mn/8 - 2ij$$

$$\Rightarrow hcut = mn/4 - 2ij$$

As we can see the max hcut of $mn/4$ will occur when $i=0, j=0$ or $i,j=0$. Thus the upper bound for the grid cutwidth of the layouts must be $mn/4$.

Case 2

We will prove the case where the layout of the vertices on the rows appears to be on top of each other, and the numbers of black and white vertices are both multiples of four. See Figure 21 below for the general description of the layout.

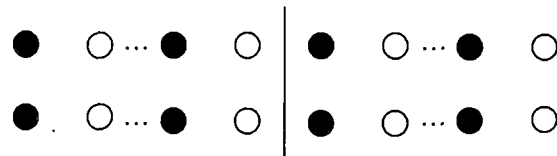


Figure 21. Case 2's Vertex Layout of $K_{m,n}$

Once more the vertical line splits the grid in half, where there are $m/2$ black and $n/2$ white vertices on both sides. If we use the cutwidth counting technique to determine the hcut at the location of the vertical line then the result will $mn/4$, which can be exactly calculated like in Case 1.

Now let's show that the lower bound for the grid cutwidth cannot be less than $mn/4$. Suppose again that we interchanged black vertices from the left side of the

vertical line with white vertices from the right. Notice once more that it would be the same, if we did such a thing in the opposite direction. Let i represent the number of these particular switches of black and white vertices about the vertical line. So there will be $m/2-i$ black and $n/2+i$ white vertices on the right, and $m/2+i$ black and $n/2-i$ white vertices on the left of the vertical line. Now let's determine the hcut at the vertical line for each switch, which is performed below.

$$\text{hcut} = [(m/2-i)*(n/2-i) + (m/2+i)*(n/2+i)]/2$$

$$\Rightarrow \text{hcut} = [2mn/4 + 2i^2]/2$$

$$\Rightarrow \text{hcut} = mn/4 + i^2$$

As we can see again from the result of the calculation, when $i=0$ the smallest hcut will be $mn/4$. Thus the lower bound for this particular layout is found to be $mn/4$.

Now let's show that the upper bound for the grid cutwidth for our embedding process on this layout will be at most $mn/4$. Calculating the hcut on one of the rows about the vertical line can be done in the same way as in Case 1, and in reality the calculations follow in the exact same way.

So let's show that if we move the vertical line into other locations then the hcut is less than or equal to $mn/4$. In the same way as Case 1, let i represent the number of black and j the number of white vertices that get switched from one side of the vertical line to the other side. Again we will shift the vertical line towards the right and calculate the result. Shifting the vertical line towards the right implies that there are $m/4+i$ black and $n/2+j$ white vertices on the left side, and $m/4-i$ black and $n/2-j$ white vertices on the right side of the vertical line. Now if we calculate the hcut for each particular shift of the vertical line, the hcut will be:

$$\text{hcut} = (m/4+i)*(n/2-j) + (m/4-i)*(n/2+j)$$

$$\Rightarrow \text{hcut} = 2mn/8 - 2ij$$

$$\Rightarrow \text{hcut} = mn/4 - 2ij$$

As we can see the max hcut of $mn/4$ will occur when $i=0$, $j=0$ or $i,j=0$. Thus the upper bound for the grid cutwidth of this layout must be $mn/4$.

Case 3

For this major case we will prove the rest of the cases where the black vertices are laid out on top of each other. These layouts are different in a few ways, but are

characterized by the fact that vertical line does not partition the black and white vertices equally on both sides. See Figure 22 below for the general descriptions of the three layouts.

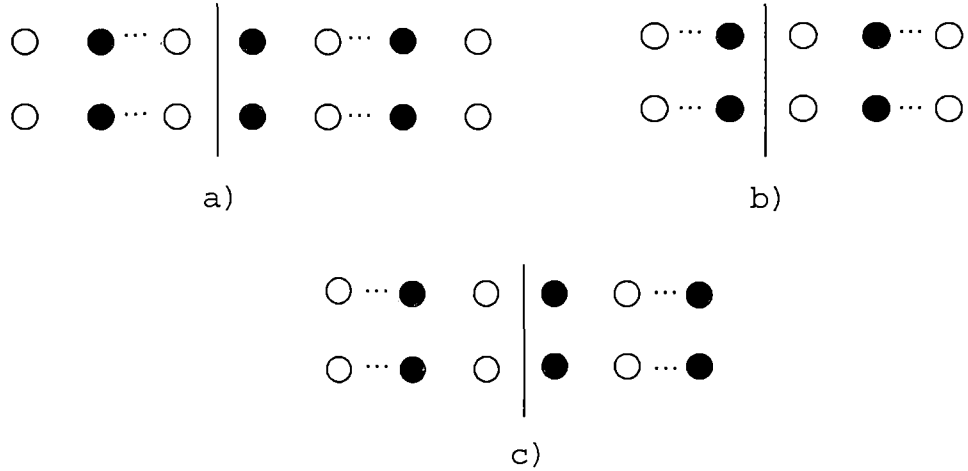


Figure 22. Case 3's Vertex Layout of $K_{m,n}$

Now each particular layout requires its own proof, but we will show the proof of one of the layouts, where the same type of set up could be used to prove the other two. So we will work with the layout on Figure 22a. Notice right away that there are $(m-2)/2$ black and $n/2$ white vertices on the left, and $(m+2)/2$ black and $n/2$ vertices on the right of the vertical line. If we calculate the hcut at the location of the vertical line as we have done before it will be once again $mn/4$. See the calculation below.

$$hcut = [(m-2)/2 * n/2 + (m+2)/2 * n/2] / 2$$

$$\Rightarrow hcut = [(m-2)n + (m+2)n] / 8$$

$$\Rightarrow hcut = [2mn] / 8$$

$$\Rightarrow hcut = mn/4$$

Now let's show that the lower bound for the grid cutwidth cannot be less than $mn/4$. Using the same techniques as in Case 2 we interchange black vertices from the left side of the vertical line with white vertices from the right. Let i represent the number of these particular switches of black and white vertices about the vertical line. So there will be $(m-2)/2-i$ black and $n/2+i$ white vertices on the right, and $(m+2)/2+i$ black and $n/2-i$ white vertices on the left of the vertical line. Now let's determine the $hcut$ at the vertical line for each switch, which is performed below.

$$hcut = [(m-2)/2-i * (n/2+i) + ((m+2)/2+i) * (n/2-i)] / 2$$

$$\Rightarrow hcut = [(m-2)n/4 + (m+2)n/4 - 2i^2] / 2$$

$$\Rightarrow hcut = [(2mn/4 - 2i^2)] / 2$$

$$\Rightarrow hcut = mn/4 + i^2$$

As we can see again from the result of the calculation, when $i=0$ the smallest $hcut$ will be $mn/4$. Thus the lower bound for this particular layout is found to be $mn/4$.

Now let's show that the upper bound for the grid cutwidth of our embedding process will be at most $mn/4$, for this layout. Let's start by calculating the hcut on the second row about the vertical line as we have done before with the other cases. So there are $(m-2)/4$ black vertices on the left that will connect to $n/2$ white vertices on the right of the vertical line. Also there are $(m+2)/4$ black vertices on the right that will connect to $n/2$ white vertices on the left of the vertical line. Now let's calculate the hcut about the vertical line.

$$\text{hcut} = (m-2)/4 * n/2 + (m+2)/4 * n/2$$

$$\Rightarrow \text{hcut} = [(m-2)n + (m+2)n]/8$$

$$\Rightarrow \text{hcut} = [2mn]/8$$

$$\Rightarrow \text{hcut} = mn/4$$

So let's show that if we move the vertical line into other locations then the hcut is less than or equal to $mn/4$. In the same way as Case 2, let i represent the number of black and j the number of white vertices that get switched from one side of the vertical to the other side. Again we will shift the vertical towards the right and calculate the result. Shifting the vertical line towards the right implies that there are $(m-2)/4+i$ black and $n/2+j$

white vertices on the left side, and $(m+2)/4-i$ black and $n/2-j$ white vertices on the right side of the vertical line. Now if we calculate the hcut for each particular shift of the vertical line, the hcut will be:

$$\text{hcut} = (m-2)/4+i)*(n/2-j) + (m+1)/4-i)*(n/2+j)$$

$$\Rightarrow \text{hcut} = (m-2)n/8 + (m+2)n/8 - 2ij$$

$$\Rightarrow \text{hcut} = 2mn/8 - 2ij$$

$$\Rightarrow \text{hcut} = mn/4 - 2ij$$

As we can see the max hcut of $mn/4$ will occur when $i=0$, $j=0$, or $i,j=0$. Thus the upper bound for the grid cutwidth of this layout must be $mn/4$.

Finally, we need to check that the max vcut for our embedding algorithm is less than or equal to $mn/4$. Since we are only using hv-edges to connect from one row onto the other row then the max vcut should be found between two adjacent white vertices in a vertical position. The reason is that hv-edges will travel through the adjacent white vertices through the top and the bottom. Since each white vertex has $m/2$ black vertices connecting to it from the opposite row then the max vcut should be m . Let's show that m must be less than or equal to $mn/4$.

$$m \leq mn/4$$

$$4m \leq mn$$

$$4 \leq n$$

As we can see $mn/4$ is greater than or equal to m when $n \geq 4$. Now when $n < 4$ then we need to check the case when $n=2$, since n is even. The $K_{m,2}$ graph classifies the case when $n=2$. Remember from Lemma 2 that the $K_{m,2}$ graph is isomorphic to the $K_{2,m}$ graph, and from Proposition 2 the max hcut is greater than or equal to the max vcut.

In brief, the max hcut is greater than or equal to the max vcut. In addition, with every case we have shown that the upper bound and the lower bound for the grid cutwidth match. Therefore, our embedding algorithm does in fact have the minimum grid cutwidth. \square

Theorem 2

For m even, n odd, and $n \neq 3$ the Complete Bipartite Graph, $K_{m,n}$, can be embedded in a $2 \times \lceil (n+m+1)/2 \rceil$ grid, with the minimum grid cutwidth of: $mn/4$, if m is a multiple of 4, and $(mn+2)/4$ if m is not a multiple of 4.

Proof:

Proving this Theorem will be much easier since the same cases and similar calculations will be performed as in Theorem 1. Much of the setup from Theorem 1 and the same style of proof of matching the lower and upper bound will be utilized in this theorem. We will split the proof of this theorem into two cases. Case 1 will show the cases where m is a multiple of four and the actual cutwidth for these cases is $mn/4$. Case 2 will show the cases where m is not a multiple of four and the actual cutwidth for these cases is $(mn+2)/4$. In addition, by removing the edges of a white vertex on a corner of the layouts of Theorem 1 will yield the layouts for the two cases in this theorem. In general it will not matter which corner this specific vertex is located in.

Case1: When $m = 4x$, for $x = 1, 2, 3, \dots$

For this case there are three layouts of vertices from Theorem 1 that fall under this case. See Figure 23 below for the three different layouts. We will show a proof for one of these layouts, but keep in mind that the same argument can be done for the other two layouts.

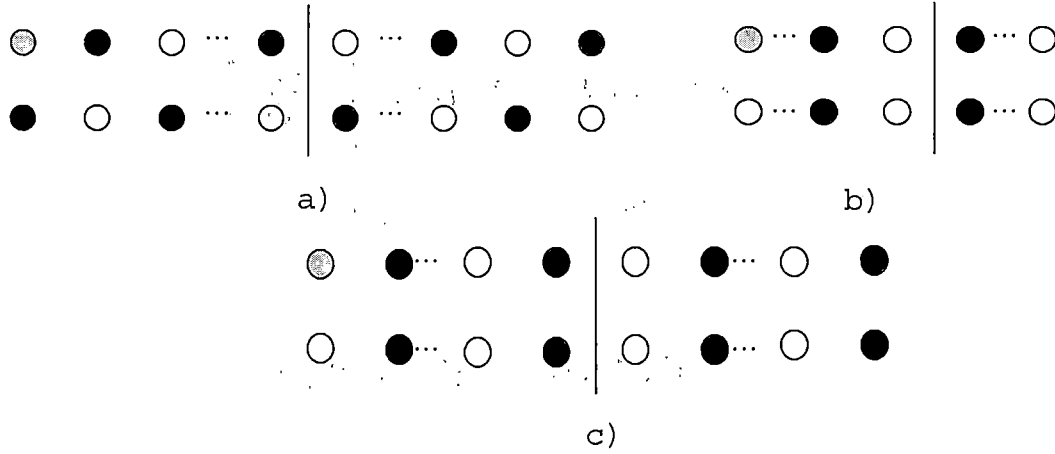


Figure 23. Case 1's Vertex Layouts for n Odd

Let's work with the layout in Figure 23a and determine the lower bound for this layout. Notice that the gray vertex is located in the position (1,1) at a corner. If we calculate the hcut at the location of the vertical line, using the cutwidth counting technique, the result will be $mn/4$. See the calculations below.

$$\text{hcut} = [(m/2)(n+1)/2 + (m/2)(n-1)/2]/2$$

$$\Rightarrow \text{hcut} = [m(n+1)/4 + m(n-1)/4]/2$$

$$\Rightarrow \text{hcut} = [mn + m + mn - m]/8$$

$$\Rightarrow \text{hcut} = [2mn]/8$$

$$\Rightarrow \text{hcut} = mn/4$$

Let's switch black and white vertices about the vertical line as was we did in the previous theorem and

calculate the hcut at every switch. Again let i represent the number of switches. So the hcut is:

$$\text{hcut} = [(m/2+i)(n+1+2i)/2 + (m/2-i)(n-1-2i)/2]/2$$

$$\Rightarrow \text{hcut} = [(m/2+i)(n+1+2i) + (m/2-i)(n-1-2i)]/4$$

$$\Rightarrow \text{hcut} = [mi + 2i + 4i^2]/4$$

$$\Rightarrow \text{hcut} = mn/4 + i/2 + i^2$$

As we can see when $i=0$, the lower bound is $mn/4$.

Now let's calculate the upper bound for this layout. Again using the same ideas from Theorem 1, let's calculate the hcut on the second row on the location of the vertical line. See the calculation below.

$$\text{hcut} = (m/4)(n+1)/2 + (m/4)(n-1)/2$$

$$\Rightarrow \text{hcut} = [m(n+1) + m(n-1)]/8$$

$$\Rightarrow \text{hcut} = [mn + m + mn - m]/8$$

$$\Rightarrow \text{hcut} = [2mn]/8$$

$$\Rightarrow \text{hcut} = mn/4$$

Now let's calculate the hcut, by moving the vertical line towards the right, at every location along the second row as it was done in Theorem 1. Again let i represent the number of black and j the number of white vertices that get switched to the left hand side of the vertical. So the hcut at every location along the second row is:

$$hcut = (m/4+i)(n+1-2j)/2 + (m/4-i)(n-1+2j)/2$$

$$\Rightarrow hcut = [2mn/4 + 2i - 4ij]/2$$

$$\Rightarrow hcut = mn/4 + i - 2ij$$

As we can see when $i=0$ or $i,j=0$ the max $hcut$ is $mn/4$. Thus the upper bound is $mn/4$.

Case 2: When $m \neq 4x$, for $x = 1, 2, 3 \dots$

Very similar to Case 1 let's prove one layout and the same argument can be done for the other two cases. See Figure 24 below for the descriptions of the layouts of this case.

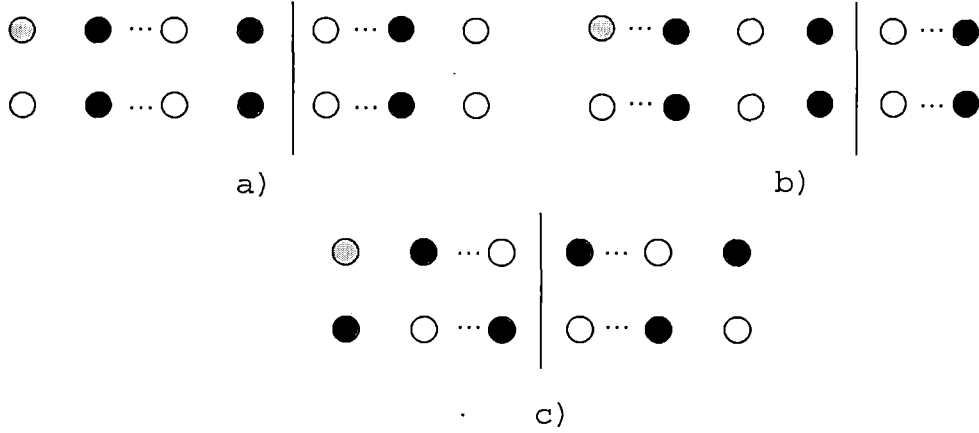


Figure 24. Case 2's Vertex Layouts for n Odd

Let's prove the layout in Figure 24a. Again let's show the lower bound. To save time we will jump to the calculations that help us determine the lower bound. There are $(m+2)/2$ black and $(n-1)/2$ white vertices on the left hand side of

the vertical line. In addition, there are $(m-2)/2$ black and $(n+1)/2$ white vertices on the right hand side of the vertical line. If we switch black and white vertices about the vertical line as we did in the previous case, then the calculation below, using i to represent every switch, should yield the hcut at every switch.

$$\text{hcut} = [(m+2+2i)/2 * (n+1+2i)/2 + (m-2-2i)/2 * (n-1-2i)/2]/2$$

$$\Rightarrow \text{hcut} = [(m+2+2i)(n+1+2i) + (m-2-2i)(n-1-2i)]/8$$

$$\Rightarrow \text{hcut} = [2mn + 4 + 8i^2]/8$$

$$\Rightarrow \text{hcut} = (mn+2)/4 + i^2$$

As we can see when $i=0$ the minimum hcut is $(mn+2)/2$. Thus the lower bound is $(mn+2)/2$.

In the same way as Case 1, let's determine the upper bound for the second row. Again i and j represent the number of black and white vertices that get switched from the right hand side of the vertical line to the left side as a consequence of moving the vertical line to the right. The calculations below will determine the max hcut.

$$\text{hcut} = (m+2+2i)/4 * (n+1-2j)/2 + (m-2-2i)/4 * (n-1+2j)/2$$

$$\Rightarrow \text{hcut} = [(m+2+2i)(n+1-2j) + (m-2-2i)(n-1+2j)]/8$$

$$\Rightarrow \text{hcut} = [2mn + 4 - 8j + 4i - 8ij]/8$$

$$\Rightarrow \text{hcut} = (mn+2)/4 + (-2j+i-2ij)/2$$

As we can see when $i, j > 0$ then $(-j+i-ij)/2 < 0$, thus for $i=0$ the max hcut will be $(mn+2)/4$. Therefore, the upper bound for this layout is $mn+2)/4$.

Finally, we need to check that the max vcut is less than or equal to $mn/4$ and $(mn+2)/4$. Using the same reasoning as Theorem 1, the max vcut should be m between two vertically aligned white vertices. Doing the same checks we get:

$$m \leq (mn+2)/4$$

$$4m \leq mn+2$$

$$4m-2 < mn$$

$$4-2/m < n$$

and

$$m \leq mn/4$$

$$4m \leq mn$$

$$4 \leq n$$

As we can see $(mn+2)/4$ and $mn/4$ are greater than or equal to m when $n \geq 4$. Since n is odd and $n \neq 3$, we only need to check the case when $n=1$. The $K_{m,1}$ graph classifies the case when $n=1$. Remember from Lemma 2 that the $K_{m,1}$ and $K_{1,m}$ graphs are isomorphic, and in Proposition 1 we proved that the max hcut is greater than or equal to the max vcut.

So we have proven that the minimum grid cutwidths can be achieved in Case 1 and Case 2, and that the max vcut is less than or equal to the max hcut. Thus this concludes the proof for this theorem. \square

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

This project began with the goal to break the ground on a conjecture: Through a given algorithm any complete bipartite graph, could be embedded in such a way that the minimum grid cutwidth could be achieved. Therefore, we decided to look into an idea in Alvin Sacdalan's paper [9], where he proved a general problem that a complete graph could be embedded into grid in such a way that the minimum grid cutwidth could be achieved. Sacdalan's first approach was to prove that a complete graph can be embedded into a $2 \times n$ grid in such a way that minimum grid cutwidth could be achieved. Once a proof was generated he then used the techniques of the proof to prove the general problem. As a result, we decided to take the same approach and first look into proving that a complete bipartite graph could be imbedded into a $2 \times n$ grid in such a way that the minimum grid cutwidth could be achieved. In this paper we broke the problem into two cases, where the m vertices of set A is even and where the m vertices of set A is odd, for the $K_{m,n}$ graph. In this paper we managed to prove the even

case, but only managed to prove a few cases for the odd n case.

Recommendations

If one wants to pursue the general problem this paper broke ground into; one needs to come up with a proof for the odd case. We recommend looking into the specific cases we have already investigated for the odd case. Nevertheless, one should pay close attention to this paper's propositions and theorems, where we used Matt Johnson's vertex formula for linear grids to successfully prove some results for the $2 \times n$ grid. In actuality, we used Johnson's vertex formula to place vertices of complete bipartite graphs into appropriate positions of the $2 \times n$ grid, which allowed us to embed complete bipartite graphs with more facility.

Once a proof has been achieved for the odd case, one should investigate if the proof for the odd and even cases could be extended to other grid sizes.

APPENDIX
TECHNICAL TERMINOLOGY

APPENDIX

TECHNICAL TERMINOLOGY

Bipartite Graph - A graph that consists of two disjoint sets of vertices, A and B, where $|A| = m$ and $|B| = n$, such vertices in A are joined by edges from vertices in B.

Cut - The number of edges running between adjacent vertices.

Cutwidth - The maximum of all the cuts in a given graph.

Complete Graph - A graph where each vertex is joined by an edge from all the other vertices.

Complete Bipartite Graph - A bipartite graph that is complete. Denoted as a $K_{m,n}$ Graph.

Embedding - The process or rearranging a graph's known form onto a host graph.

Graph - A graph, $G = (V, E)$, consists of a finite set of vertices, V, and a finite set of edges, E, joining pairs of distinct vertices.

Grid - a set of vertices in a rectangular form.

Grid Cutwidth - Same as cutwidth, but particularly reserved for a grid.

Horizontal Cut - The number of edges running between adjacent vertices, which lay in a horizontal layout. Denoted as the $vcut$.

Horizontal Vertical Edge - a curved edge that runs horizontally first and vertically second. Denoted as a hv-edge.

Linear Graph - A graph where the vertices are arranged in a linear fashion.

$m \times n$ Grid - A grid with m rows and n columns.

Tree - A graph that consist of branches and paths with no complete cycles.

Vertical Cut - The number of edges running between adjacent vertices, which lay in a vertical layout. Denoted as the $vcut$.

Vertical Horizontal Edge - A curved edge that runs vertically first and horizontally second. Denoted as a vh-edge.

REFERENCES

- 1) Chavez, Joseph D., Trapp, Rolland, The Cyclic Cutwidth of Trees. Discrete Applied Mathematics, 87, Elsevier Science, 1998, 25-32.
- 2) Chung, F.R.K., Labeling of Graphs, Graph Theory. 3, Academic Press. 1988.
- 3) Clarke, Dwayne W., The Cyclic Cutwidth of Mesh Cubes. Preprint, 2002.
- 4) James, Beatrice, The Cyclic Cutwidth of the Three-Dimensional and Four-Dimensional Cubes. California State University, San Bernardino McNair Scholar's Program, Summer Research Journal, 1996.
- 5) Hernandez, Sara, Lower Bounds for the Bandwidth of Binary Trees Embedded into Grids. California State University, San Bernardino McNair Scholar's Program, Summer Research Journal, 1998.
- 6) Johnson, Matt, The Linear and Cyclic Cutwidth of the Complete Bipartite Graph. Preprint, 2002.
- 7) Rios, Francisco R., Complete Graphs as a First Step toward Finding the Cyclic Cutwidth of the n-Cube. California State University, San Bernardino McNair Scholar's Program, Summer Research Journal, 1996.

8) Rocha, Mario, Graph Theory: The Embedding of Complete m-ary Trees into Grids as Means of Finding the Minimum Cutwidth. California State University, San Bernardino McNair Scholar's Program, Summer Research Journal, 1998.

9) Sacdalan, Alvin, Proof of the Grid Cutwidth for a Complete Graph. Preprint, 2002.

10) Tucker, Alan, Applied Combinatorics. 3rd edition. John Wiley & Sons, Inc, 1995.

11) Wang, Annie, The Cutwidth of the Complete Graph on $2n$ Vertices when Embedded into a $2 \times n$ Grid. Preprint, 2001.