A greedy algorithm enhancing cage search

by

Douglas Wayne Ray, Jr.

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Mathematics

Program of Study Committee:
Dan Ashlock, Major Professor
David Fernández-Baca
Sung-Yell Song

Iowa State University

Ames, Iowa

2004

Graduate College
Iowa State University

This is to certify that the master's thesis of

Douglas Wayne Ray, Jr.

has met the thesis requirements of Iowa State University

Major Professor

For the Major Program

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

A cubic cage is a regular graph of degree three with given girth and a minimum number of vertices. Current information about cubic cages is surveyed. A greedy algorithm that permits extremely sparse descriptions of known cages is presented. A number of methods for enhancing the algorithm performance is given. Also, a strategy for looking for better bounds on as yet unknown cages is presented.

CHAPTER 1. INTRODUCTION TO GIRTH

We assume the reader is familiar with graph theory. Readers unfamiliar with graph theory will find (8) a good introduction. Girth is defined as the length of the smallest cycle in a graph. The term *graph* will be used to mean a simple, undirected graph (a graph with no loops or multi-edges). A cage is the smallest order regular graph with a specified girth. This paper explores the many aspects of girth. After definitions and relating basic properties of a graph to its girth, we will present current information about cages and several approaches to improve the current best information about certain cages. We will use a greedy algorithm to make conclusions about girth and relate other graphs to the cages. Also, a new technique for expanding graphs into cages will be presented. In particular, $(3, 13)$-graphs, the current best approximation to a $(3, 13)$-cage, will be examined and results on efforts to reduce the number of vertices in it.

1.1 Definitions

A graph $G = G(V, E)$ is an ordered pair, where V denotes a list of distinct vertices, and E denotes the edges between the vertices of V. The order of a graph G is the number of vertices in G; the size of a graph G is the number of edges in G. The degree of a vertex v, denoted $\deg_G(v)$ or just $\deg(v)$ if the graph is unambiguous, is the number of vertices adjacent to it, i.e., the number of vertices which share an edge with v. The minimum degree of all vertices in a graph G is denoted $\delta(G)$ while the maximum degree of all vertices in G is denoted $\Delta(G)$.

A graph is r-regular if every vertex of the graph has degree r. Thus, if G is r-regular, then $r = \delta(G) = \Delta(G)$. Specifically, a 3-regular graph is called cubic or trivalent. The distance from vertices u to v is denoted $d(u, v)$ and is the smallest number of edges in a path from vertex u to v. If there is no path between vertices u and v, the distance is defined to be infinite. The girth of a graph G, $g(G)$, is the smallest cycle length in the graph. An (n, g)-cage is an n-regular graph of minimal order with girth g. An (n, g)-graph is an n-regular graph with girth g, but one that has not been proven to be of minimal order.

The following notation will be used throughout this paper to denote different special graph: E_n denotes the empty graph on n vertices; C_n denotes the cycle with n vertices; K_n denotes the complete graph on n vertices; P_n denotes the path on n vertices.

1.2 Basic Properties of Girth

The girth of a graph is a property that theorists have been able to intertwine with other properties of a graph that would have obvious connections and some properties where the connections with girth are profound.

Since the girth of a graph is the smallest cycle length in the graph and trees do not have cycles, we define the girth of a tree or forest to be infinite (8, p. 185). In 1959, Erdős proved that there exists a graph with arbitrary large girth and large chromatic number (8, p. 408). This result does not appear to be helpful in solving cage problems since such graphs have a large number of vertices.

The diameter of a graph G, denoted $\operatorname{diam}(G)$, is the maximum distance between any two vertices of a graph. For any graph G with cycles, the girth $g(G) \leq 2\operatorname{diam}(G)+1$. (4, p. 8) Thomassen proved that "given an integer k, every graph G with girth $g(G) \geq 4k-3$ and $\delta(G) \geq 3$ has a minor graph H (a graph formed by grouping vertices together to denote one new vertex and then connecting new vertices if and only if there was an edge between two vertices from different groups in the original graph G) with $\delta(H) \geq k$." (4, p. 17, p. 179)

1.3 Girth of Special Graphs

The girth of the n-cycle C_n is n and adding any chord to C_n reduces its girth. The girth of the complete graph on n vertices, K_n, for $n \geq 3$, is 3 since every vertex is adjacent to all other vertices in the graph, meaning any three vertices form a 3-cycle.

Standard constructions of graphs, including the use of Cayley graphs (to be discussed later), appeared at first to be useful in creating cubic graphs with large girth; however, the maximum girth known for these graphs is 14 in the range of orders useful for the $(3, 13)$-graph (3, p. 10).

We now examine some standard constructions adn compute an absolute upper bound for their girths. The generalized Petersen graph and generalized Coxeter graph will be examined.

Definition 1 *The* (n, k)***-generalized Petersen graph*** *is defined as the graph* G *where the vertex set* $V = X \cup Y$, $X = \{x_0, x_1, \ldots, x_{n-1}\}$ *and* $Y = \{y_0,\ y_1,\ \ldots, y_{n-1}\}$. *The edges of* G *must be in one of the following forms:* $\{x_i, x_{i+1}\}$, $\{y_i, y_{i+k}\}$, *or* $\{x_i, y_i\}$, *where all subscripts are taken modulo* n. *The edges of an* (n, k)*-Petersen graph in the form* $\{x_i, y_i\}$ *are called* ***spokes****.*

The generalized Petersen graph has order $2n$ and is 3-regular. The $(5, 2)$-generalized Petersen graph is the well-known Petersen graph and is shown below in Figure 1.1.

Theorem 1 *The girth of the* (n, k)*-generalized Petersen graph is no larger than 8.*

PROOF: To show that the girth of the (n, k)-generalized Petersen graph is no larger than 8, we will find a closed path which must contain a cycle of length no more than 8 for all values of n and k.

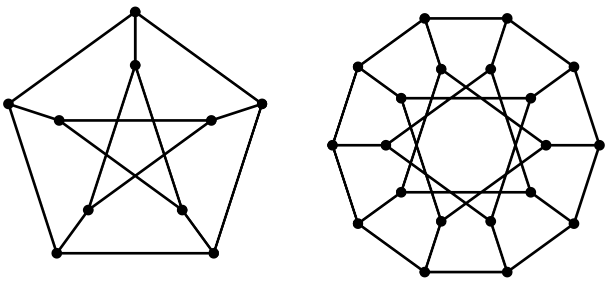

Figure 1.1 $(5, 2)$- and $(10, 3)$-generalized Petersen graphs, also known as the Petersen graph and the Desargues graph.

The following closed path (see Figure 1.2) contains a cycle of length 8 or less:

$$x_0 \; y_0 \; y_k \; x_k \; x_{k+1} \; y_{k+1} \; y_1 \; x_1 \; x_0$$

□

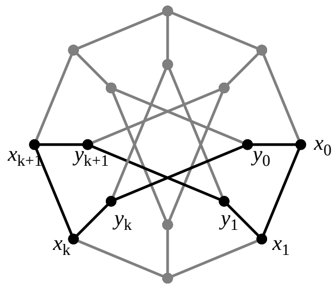

Figure 1.2 $(8, 3)$-generalized Petersen graph showing cycle of length 8

Definition 2 *The (n, k, ℓ)-generalized Coxeter graph is defined as the graph G with vertex set $V = X \cup Y \cup Z \cup C$, with $X = \{x_0, x_1, \ldots, x_{n-1}\}$, $Y = \{y_0, y_1, \ldots, y_{n-1}\}$, $Z = \{z_0, z_1, \ldots, z_{n-1}\}$, and $C = \{c_0, c_1, \ldots, c_{n-1}\}$. The edges of G must be in one of the following forms: $\{x_i, x_{i+1}\}$, $\{y_i, y_{i+k}\}$, $\{z_i, z_{i+\ell}\}$, $\{c_i, x_i\}$, $\{c_i, y_i\}$, or $\{c_i, z_i\}$, where all subscripts are taken modulo n. The edges incident with the vertices of C are called claws.*

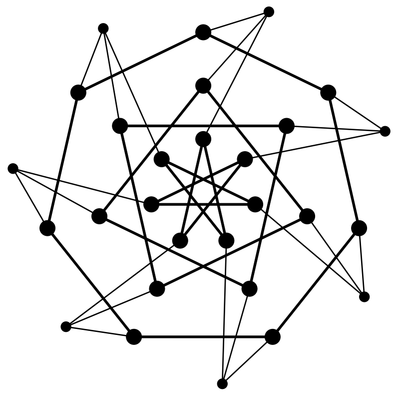

Figure 1.3 The Coxeter graph, a $(7, 2, 3)$-generalized Coxeter graph

Theorem 2 *The girth of the (n, k, ℓ)-generalized Coxeter graph is no larger than 12.*

PROOF: To show that the girth of the (n, k, ℓ)-generalized Coxeter graph is no larger than 12, we will find a closed path which must contain a cycle of length no more than 12.

The following closed path contains a cycle of length 12 or less:

$$x_0\ c_0\ y_0\ y_k\ c_k\ x_k\ x_{k+1}\ c_{k+1}\ y_{k+1}\ y_1\ c_1\ x_1\ x_0$$

□

CHAPTER 2. CUBIC CAGES

2.1 Information

A (k, g)-*cage* is a k-regular graph of girth g with the fewest possible number of vertices.

The quantity $n(k, g)$, sometimes known as the *Moore bound* or the *naive bound* (3), is the lower bound on the number of vertices for which a (k, g)-cage can exist based on the distances between vertices and the regularity of the degree of the graph. For the cubic cages,

$$n(3,g) = \begin{cases} 3 \cdot 2^{(g-1)/2} - 2, & \text{if } g \text{ is odd,} \\ 2^{(g+2)/2} - 2, & \text{if } g \text{ is even.} \end{cases} \tag{2.1}$$

To prove the equation 2.1 for the Moore bound, we will need the following definition.

Definition 3 *A distance tree of G with respect to vertex v is a rooted spanning tree of G with root v. All vertices are listed from left to right in vertical tiers of increasing distance from the root v. If G is not connected, then all vertices outside the component containing v are listed at the far right since the distance from v to such a vertex is infinity. See Figure 2.1 for an example.*

Theorem 3 *The smallest possible number of vertices for which a $(3, g)$-cage with odd girth can exist is* $3 \cdot 2^{(g-1)/2} - 2$.

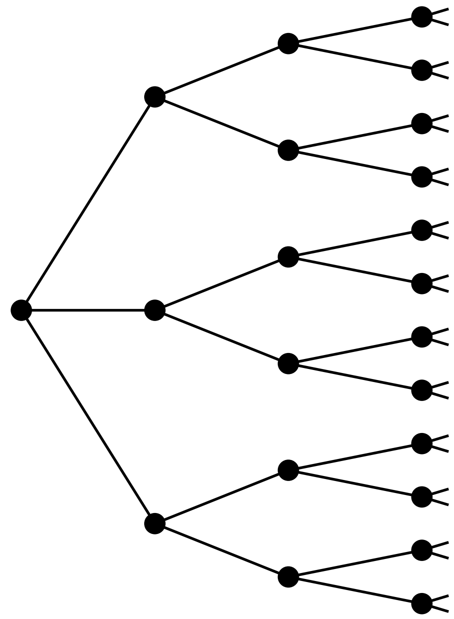

Figure 2.1 Distance Tree of a Cubic Graph with Odd Girth

PROOF: Let T be the distance tree of the graph G with odd girth g, shown in Figure 2.1.

In the first level, there is one vertex, say v_0. Since all the vertices are cubic, there are three vertices adjacent to v_0. Thus, in the distance 1 level of T, there are three vertices. However, in the distance 2 level of T, there are two vertices adjacent to the same vertex in the distance 1 level, since each vertex is cubic. Thus, there are a total of $3 \cdot 2$ vertices in the distance 2 level of T. In general, there are $3 \cdot 2^{i-1}$ vertices in the distance i level of T.

Since the girth g is odd, this means that two vertices in the same distance level from v_0 will be adjacent and thus, in the same distance level but from different forks. This distance is $(g-1)/2$.

Summing up the total number of vertices,

$$\begin{aligned} n(3,g) &= 1+3+3\cdot 2+3\cdot 2^2+\ldots+3\cdot 2^{(g-1)/2-1} \\ &= 1+3+3\cdot 2+3\cdot 2^2+\ldots+3\cdot 2^{(g-3)/2} \\ &= 1+3(1+2+2^2+\ldots+2^{(g-3)/2}) \\ &= 1+3(2\cdot 2^{(g-3)/2}-1) \\ &= 1+3(2^{(g-1)/2}-1) \\ &= 3\cdot 2^{(g-1)/2}-2. \end{aligned}$$

□

Theorem 4 *The smallest possible number of vertices for which a* $(3,g)$*-cage with even girth can exist is* $2^{(g+2)/2}-2$.

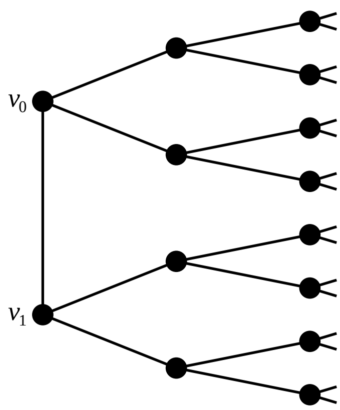

Figure 2.2 T' is a modified distance tree with two adjacent vertices in the first level.

Proof: Let T' be a modified distance tree so that two adjacent vertices, v_0 and v_1, sit in the root level of the tree. Since g is even, there exists two adjacent vertices u_0 and u_1 in the same distance level from the pair v_0 and v_1 but they necessarily come from different forks and $d(v_0,u_0)=d(v_1,u_1)=(g-2)/2$.

Summing up the number of vertices from the distance levels, we get

$$\begin{aligned} n(3,g) &= 2 + 2(2) + 2(2^2) + \cdots + 2(2^{(g-2)/2}) \\ &= 2\left(1 + 2 + 2^2 + \cdots + 2^{(g-2)/2}\right) \\ &= 2\left(\frac{1 - 2^{(g-2)/2+1}}{1-2}\right) \\ &= 2\left(2^{(g-2)/2+1} - 1\right) \\ &= 2^{(g+2)/2} - 2. \end{aligned}$$

The following chart shows the most current information available on $(3,g)$-cages (Royle). The $(3,3)$-, $(3,4)$-, $(3,5)$-, $(3,6)$-, $(3,8)$-, and $(3,12)$-cages all meet the Moore bound. In the chart, the numbers in square brackets represent the de facto lower bound on the number of vertices needed to produce the given cage, as checked by extensive computer search.

Table 2.1 Current cubic cage information

Cage	Smallest Known	$n(3,g)$	Number	Special Name
$(3,3)$-cage	4	4	1	K_4
$(3,4)$-cage	6	6	1	$K_{3,3}$
$(3,5)$-cage	10	10	1	Petersen
$(3,6)$-cage	14	14	1	Heawood
$(3,7)$-cage	24	22	1	McGee
$(3,8)$-cage	30	30	1	Tutte 8-cage
$(3,9)$-cage	58	46	18	
$(3,10)$-cage	70	62	3	
$(3,11)$-cage	112	94 [112]	1	Balaban's graph
$(3,12)$-cage	126	126	1	Generalized Hexagon
$(3,13)$-cage	272	190 [202]	1+	
$(3,14)$-cage	384	254 [258]	1+	
$(3,15)$-cage	620	382	1+	
$(3,16)$-cage	960	510	1+	
$(3,17)$-cage	2176	766	1+	
$(3,18)$-cage	2640	1022	1+	
$(3,19)$-cage	4324	1534	1+	
$(3,20)$-cage	6072	2046	1+	

2.2 Survey of Cubic Cages

All cubic cages of girth up to 8 are well-established as most of them meet the Moore Bound and are vertex transitive. The one exception is the $(3,7)$-cage, which is not vertex transitive and exceeds the Moore bound by 2 vertices. The automorphism group has orbits of size 8 and 16 (Royle).

The most interesting contemporary research comes with the discovery of the cages with girth larger than 8. Biggs and Hoare discovered the first graph that was subsequently proven to be a $(3,9)$-cage by McKay. An exhaustive search by Brinkmann and Saager found all 18 $(3,9)$-cages (Royle).

A $(3,10)$-cage was described by Wong and O'Keefe in 1980. There are three nonisomorphic $(3,10)$-cages with order 70. (Royle).

For the $(3,11)$-cage, McKay and Myrvold showed it must contain 112 vertices. They then proved that the graph provided by Balaban was the unique $(3,11)$-cage. (Royle)

To understand the information provided about the $(3,13)$- and $(3,14)$-cages, we first look at Cayley graphs. Let H be a group with identity 1 and let Ω be a set of generators of H such that $1 \notin \Omega$ and if $x \in \Omega$, then $x^{-1} \in \Omega$. A Cayley graph is a graph G whose vertex set is set of the elements of a finite group H with a set of generators Ω, described above, and whose edge set is

$$E(G) = \big\{\{g,h\} \mid g^{-1}h \in \Omega\big\}.$$

(2, 123)

To determine the edge set, we must satisfy the condition that $g^{-1}h$ is an element of Ω. Notice that if Ω is written as the set $\big\{\lambda_1, \lambda_2, \ldots, \lambda_k\big\}$, then for each g, there is a set $\big\{h_1, h_2, \ldots h_k\big\}$ such that $g^{-1}h_1 = \lambda_1$, $g^{-1}h_2 = \lambda_2$, $\ldots$, $g^{-1}h_k = \lambda_k$. This is equivalent to $h_1 = g\lambda_1$, $h_2 = g\lambda_2$, $\ldots$, $h_k = g\lambda_k$. Therefore, to find all neighbors of vertex g, right multiply g by each element of the generating set Ω.

Table 2.2 Cayley graph adjacency computation

Vertex v of D_8	Ω	Product	Neighbors of v
1	$\{r, s, r^3\}$	$\{r, s, r^3\}$	$\{r, s, r^3\}$
r	$\{r, s, r^3\}$	$\{r^2, rs, r^4\}$	$\{r^2, sr^3, 1\}$
r^2	$\{r, s, r^3\}$	$\{r^3, r^2s, r^5\}$	$\{r^3, sr^2, r\}$
r^3	$\{r, s, r^3\}$	$\{r^4, r^3s, r^6\}$	$\{1, sr, r^2\}$
s	$\{r, s, r^3\}$	$\{sr, s^2, sr^3\}$	$\{sr, 1, sr^3\}$
sr	$\{r, s, r^3\}$	$\{sr^2, srs, sr^4\}$	$\{sr^2, r^3, s\}$
sr^2	$\{r, s, r^3\}$	$\{sr^3, sr^2s, sr^5\}$	$\{sr^3, r^2, sr\}$
sr^3	$\{r, s, r^3\}$	$\{sr^4, sr^3s, sr^6\}$	$\{s, r, sr^2\}$

For example, let $H = D_8 = \langle r, s \mid r^4 = s^2 = 1 \text{ and } sr = r^{-1}s\rangle$, the dihedral group with eight elements (5) with the generating set $\Omega = \{r, s, r^3\}$. To find the edge set of this Cayley graph, we will multiply each element of D_8 by the set Ω.

This gives the following familiar graph in Figure 2.3.

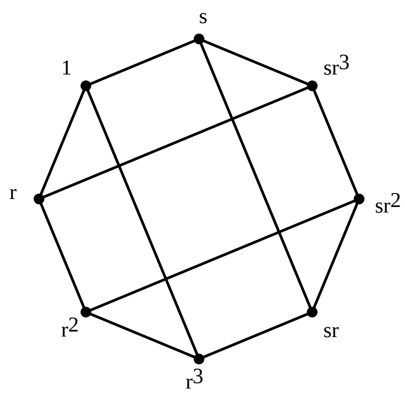

Figure 2.3 Cayley graph of D_8 with $\Omega = \{r, s, r^3\}$.

The current candidate for the $(3, 13)$-cage using Cayley graphs was found by Hoare and published by Biggs. In his paper (3), Biggs describes two types of Cayley graph constructions based on the generating set Ω. These types are

$$\Omega = \{\alpha, \beta, \gamma \mid \alpha^2 = \beta^2 = \gamma^2 = 1\}$$

and

$$\Omega = \{\alpha, \delta, \delta^{-1} \mid \alpha^2 = 1 \text{ and } \delta^2 \neq 1\},$$

where each of α, β, γ, and δ is a permutation. The permutations α, β, and γ are known as involutions since they are of order 2.

For the $(3, 13)$-cage, Biggs uses $\alpha(x) = b - x$ and $\delta(x) = cx$, where $b, c \in \mathbb{Z}_p$ for prime p with $c \neq 0$, generates the affine group of transformations of $\mathbb{Z}_p$. Specifically, Biggs chose $p = 17$, $b = 1$, and $c = 3$. These numbers produce a group of order $17 \times 16 = 272$. The shortest word, or string of permutations, representing a path in the Cayley graph that gives the identity word has length 13. This implies that the girth is at most 13, and in fact, the girth is 13.(3)

For the $(3, 14)$-cage, this approach was also applied with $p = 29$, $b = -1$ and $c = 4$. This yielded a graph with 406 vertices of girth 14.(3) However, this value of the order of the $(3, 14)$-cage has been reduced by a result from Exoo, lowering the number to 384.(Royle) Even worse, any construction of this second type always has an identity word of length 14, thus rendering this method useless in the search for cages of larger girth. (3)

The latest update in the knowledge of cages at the time of this writing is found in the work by Exoo on the $(3, 18)$-cage. In his paper (6), Exoo modifies the first type of Cayley graphs methods presented above by adding in two more relations, namely $bcabababcbcbcabc = 1$ and $cacbcababcbcbabacb = 1$, in addition to $\alpha^2 = \beta^2 = \gamma^2 = 1$. The construction yields a Cayley graph of order 2640 and the best-known upper bound on the order of the $(3, 18)$-cage.

CHAPTER 3. THE ALGORITHM

Having surveyed the current state of knowledge, we now focus on new information. In this chapter, we will examine a greedy algorithm that can be used to create cubic cages. Following this, we will looks at modifications of the algorithm and their connection to a particular class of graphs. We will also look at when the algorithm fails and ways of improving the performance of the algorithm.

Definition 4 *A greedy algorithm is an algorithm that iteratively selects objects based on certain local optimality criteria.*

Given an initial graph G_0, called the "hint" graph, of even order greater than or equal to 4 whose vertices all have degree 3 or less, the algorithm given below adds edges to G_0 with the goal of creating a cubic graph. Under less than ideal circumstances, the resulting graph is not cubic. For the purposes of this paper, we will place restrictions on the hint graph. This chapter explores the algorithm and its behavior on classes of graphs. In particular, we will see how the algorithm is linked to the cages.

3.1 Implementing the Algorithm

Let $G_0 = (V_0, E_0)$ be the hint graph with vertex set V_0 of order n, n even, with vertices labelled $\{0, 1, 2, \ldots, n-1\}$ and edge set E_0.

Algorithm 1

- *Begin with* $G = G_0$*, a hint graph of even order.*

- *Repeat the following steps until G is cubic or the graph is one such that the algorithm can no longer add edges to it.*
  - *Let v be the smallest numbered vertex of G with degree less than 3.*
  - *Find the smallest numbered vertex u ($\neq v$) of degree less than 3 such that $d(u, v)$ is maximal. If no such vertex exists, stop.*
  - *Add edge $\{u, v\}$ to the edge set.*
- *Loop.*

Breaking down the algorithm, we notice that if the hint graph G_0 is already cubic, algorithm 1 returns the graph G_0. Also, the ordering of the vertices is important. The algorithm can return different graphs when the vertices are in a different order.

In the algorithm, the distance $d(u, v)$ could be infinite if there is no path connecting vertices u and v, i.e., if G_0 is disconnected. The algorithm completes in a finite number of steps since at most the algorithm will add $3n/2$ edges to any graph and fail if it cannot add an edge in a given iteration.

Definition 5 *A cubic component is a component of a graph with all degree 3 vertices.*

If G_0 is disconnected with cubic components, then we will take the hint graph to be G_0 without these cubic components as the algorithm cannot alter these components in any way. Furthermore, the ultimate purpose for working with the algorithm is to produce cages, and thus we will eventually need connected graphs.

The algorithm has several possible non-cubic terminal states. They are (1) having all cubic vertices except one degree 1 vertex or (2) the more common two adjacent degree 2 vertices (see figure 3.1).

To show that the algorithm has these non-cubic terminal states, we first begin by showing that the algorithm can fail to produce a cubic graph, and then show that these are the only failure states.

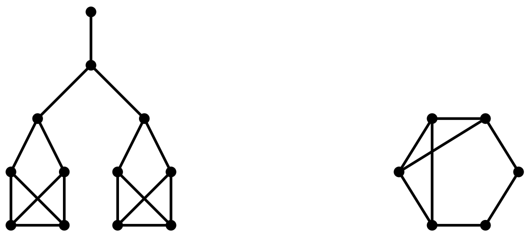

Figure 3.1 Algorithm fails when G has one degree 1 vertex (left) or G has two adjacent degree 2 vertices (right).

Observation 1 *Algorithm 1 does not always produce a cubic graph.*

Proof: The reader is invited to check that running algorithm 1 on hint graphs isomorphic to cycles of orders from 4 to 1000 provides ample witnesses to the incompleteness of the algorithm. In particular, the following orders of cycles do not produce a cubic graph: 16, 46, 60, 66, 70, 94, 108, 114, 180, 198, 252, 272, 292, 334, 410, 600, 666, 672, and 870. □

Theorem 5 *If Algorithm 1 fails to produce a cubic graph, then it must either produce a graph with one degree 1 vertex or two adjacent degree 2 vertices.*

Proof: We will prove that these are the only failure modes for the algorithm by examining possible situations for a graph near the end of the execution of the algorithm. First, we know that we cannot have four noncubic vertices, as the algorithm would connect any nonadjacent pair. If all four vertices were mutually adjacent, then each vertex would have degree 3 and that completes the algorithm and this is not a failure mode for the algorithm. Notice that the number of noncubic vertices can always be reduced by the algorithm to 4 vertices or less and then the considerations in this proof apply.

Next, Suppose we have three noncubic vertices. We can assume the algorithm has produced a triangle between these three vertices. However, since we require the order of the graph to be at least four, there must be some other vertex adjacent to one vertex of the triangle. Thus, this vertex is cubic and this is not a failure mode.

Suppose there are two nonadjacent degree 2 vertices. The algorithm will connect these vertices and complete the graph. If there are two nonadjacent degree 1 vertices, the algorithm will connect these vertices and complete the graph.

Therefore, the only failure modes are two adjacent degree 2 vertices or one degree 1 vertex. □

Theorem 6 *Algorithm 1 yields the vertex transitive cubic cages that meet the Moore bound when the hint graph* $G_0 = E_n$*, where* E_n *is the empty graph on* n *vertices and* $n = n(3, g)$*.*

Proof: Run algorithm 1 on the following: E_4 produces K_4, E_6 produces $K_{3,3}$, E_{10} produces the Petersen graph, E_{14} produces the Heawood graph, and E_{30} produces Tutte's 8-cage. These are all the vertex transitive cubic cages that meet the Moore bound. (Royle) □

3.2 Minimum Hint Graphs

The purpose of the algorithm is to produce the cubic cages from the least amount of information about the cages as possible. As Theorem 6 reveals, the only information needed about the vertex transitive cubic cages that are Moore graphs is their order. Then, when the algorithm is applied to the empty graph with that order, the cubic cage emerges. This does not apply to the $(3, 12)$-cage, which is not vertex transitive.

Observation 2 *All cages have a minimum hint graph not equal or isomorphic to the cage itself.*

Proof: To realize this, take one edge away from the cage. Run the algorithm on this hint graph. Since all vertices of the hint graph are cubic except for the vertices incident with the removed edge, the algorithm is forced to reconnect these vertices, thus re-establishing the cage. □

The question then becomes how much information is needed to produce the cages that are not witnessed by executing the algorithm on the empty graph or correct order? One amazing result is that the $(3, 7)$-cage needs only a graph on 24 vertices with two edges as the hint graph.

Theorem 7 *Given the empty graph on 24 vertices in the order* $\{0, 1, 2, \ldots, 23\}$ *with edges* $\{2, 8\}$ *and* $\{11, 16\}$ *as the minimum hint, algorithm 1 returns the* $(3, 7)$*-cage.*

PROOF: Let G_0 be the graph with $V(G_0) = \{0, 1, \ldots, 23\}$ and $E(G_0) = \big\{\{2, 8\}, \{11, 16\}\big\}$. Running the algorithm on G_0 produces the $(3, 7)$-cage. □

3.3 Actions of the Algorithm on Cycles

The hint graphs of particular interest to use with Algorithm 1 are the empty graphs. By modifying Algorithm 1, we obtain a new algorithm. When given the empty graph of order n, this new algorithm (described below) contains the cyclic graph of order n as one of its intermediate graphs.

Algorithm 2

- *Begin with a hint graph of even order. For our purposes, use* E_n*, the empty graph on* n *vertices.*
- *Let* v *be the initial vertex 0.*
- *Repeat the following steps until* G *is cubic or the graph is one such that the algorithm can no longer add edges to it.*
  - *If* v *has degree less than 3, do the following.*

  * *Find the smallest numbered vertex u ($\neq v$) of degree less than 3 such that $d(u, v)$ is maximal.*
  * *Add edge $\{u, v\}$ to the edge set.*

- $v := v + 1 \pmod{n}$.

- *Loop.*

Theorem 8 *Algorithm 2 produces the cyclic graph of order n as an intermediate graph when given the hint empty of order n.*

PROOF: Let $G_0 = E_n$ where $V(E_n) = \{0, 1, \ldots, n-1\}$. Running Algorithm 2 on G_0, we start with vertex 0. The algorithm finds the smallest numbered vertex at maximal distance from vertex 0. At this stage, all vertices have infinite distance from 0, since the graph is the empty graph. Thus, the smallest numbered vertex is vertex 1. The algorithm connects vertices 0 and 1 with an edge and then moves on to vertex 1. All vertices except vertices 0 and 1 have infinite distance from 1, so the smallest numbered vertex with maximal distance is 2. The algorithm connected vertices 1 and 2 with an edge and then moves on to 2. The algorithm repeats this process, connecting vertices i to $i + 1$ for $i = 0, 1, \ldots, n-2$.

When we reach vertex $n-1$, the intermediate graph G is a path of length $n-1$. Thus G is connected and the maximal distance from vertex $n-1$ is finite. Since G is just a path and one end of this path is vertex $n-1$, the vertex of maximal distance away from vertex $n-1$ must be the other end of the path, vertex 0. Thus, the algorithm connects vertices 0 and $n-1$ by edge. The resulting graph is the cycle on n vertices. □

The cycle is an important graph to study with respect to the algorithm for several reasons. For small values of girth g, running Algorithm 1 with the hint graph isomorphic to E_n, where $n = n(3, g)$, returns the $(3, g)$-cage. Also, the cycle gives prime examples of graphs where the algorithm fails to produce a cubic graph. Finally, as proved below, the first actions of the algorithm when applied to the cyclic graph can be easily characterized.

Theorem 9 *When Algorithm 1 is applied to the cyclic graph of order n, the following edges are the first produced: (1) Vertex 0 is connected to vertex $n/2$ by an edge, and (2) vertex 1 is connected to vertex $\lfloor \frac{3n}{4} - \frac{1}{2} \rfloor$ by an edge.*

PROOF: For (1), in the beginning, the hint graph is C_n, the cycle on n vertices. From any vertex in C_n, the maximal distance is half-way around the cycle, $n/2$. Specifically, the vertex furthest away from 0 is $n/2$. Therefore, the algorithm adds the edge $\{0, n/2\}$.

For (2), the graph is two cycles, C and C', sharing one edge, namely $\{0, n/2\}$ (see figure 3.2). The vertex at maximal distance from vertex 1 cannot be between 2 and $n/2$ since vertex 1 is closer to all vertices in C than some of the vertices in C'. Looking at cycle C', we see that its length is $n/2+1$. Thus, the maximum distance between any two vertices on this cycle is less than or equal to $\frac{1}{2}\left(\frac{n}{2}+1\right)$. To find the maximal distance from vertex 1, which is not on cycle C', we will find the vertex with maximal distance from 0 and then adjust.

Going around the cycle from $0, n-1, n-2, \ldots$, we can compute which vertex will be at distance $\frac{1}{2}\left(\frac{n}{2}+1\right)$ from 0. Recognizing that $0 \equiv n \pmod{n}$, we will use n is place of 0 to simplify calculations. Then, $n - \frac{1}{2}\left(\frac{n}{2}+1\right) = \frac{3n}{4} - \frac{1}{2}$.

Going around the cycle from $0, n/2, n/2+1, \ldots$, we can compute which vertex will be at distance $\frac{1}{2}\left(\frac{n}{2}+1\right)$ from 0. In this case, 0 is represented by $\frac{n}{2}-1$. Thus, we have $\left(\frac{n}{2}-1\right) + \frac{1}{2}\left(\frac{n}{2}+1\right) = \frac{3n}{4} - \frac{1}{2}$.

Therefore, the vertex closest to vertex 1 should be the vertex numbered $\frac{3n}{4} - \frac{1}{2}$. The obvious problem here is that $\frac{3n}{4} - \frac{1}{2}$ may not be an integer. When this number is not an integer, this indicates that there are two adjacent vertices in cycle C' that are the same distance from vertex 1. We then round this number down to provide the smaller of these two numbers, as called for by the algorithm. □

Algorithm 2 finds both the $(3,3)$- and $(3,4)$-cages.

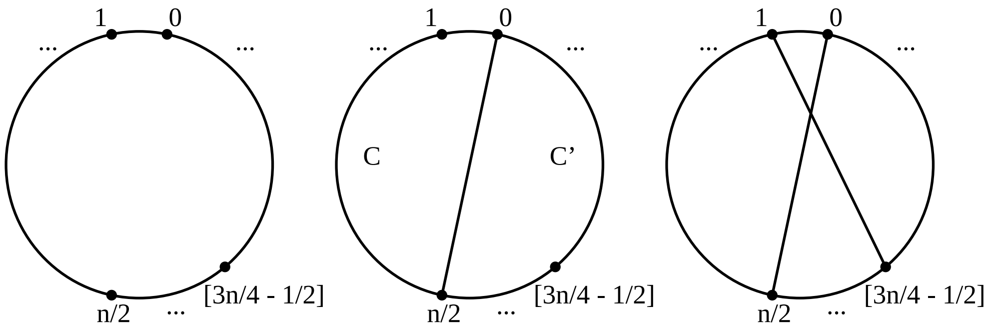

Figure 3.2 First stages of Algorithm 1 acting on a cycle of order n

Finally, we note that algorithm 1 also produces the $(5, 6)$-cage given E_{42} as the hint graph.

CHAPTER 4. TECHNIQUES FOR SEARCHING FOR $(3,13)$-CAGE

The smallest girth for which it is undecided whether or not the cubic cage has been found is 13. The eventual goal of my research is to provide a smaller number on the order of the $(3,13)$-cage or to show that the current $(3,13)$-cage is actually the cage. However, this problem is very challenging. The current value of 272 vertices was found by Hoare in 1981 (3). Biggs even expressed surprise that this value has not been improved at the time of his writing (3) and some seven years have elapsed since then. It has been shown by brute computer force that although the Moore bound for the $(3,13)$-cage is 190, the minimum number of vertices needed to produce the cage is at least 202. This result is due to McKay, Myrvold, and Nadon (3).

So far, my attempts to improve the cage has not succeeded. However, some of the techniques developed are ripe for improvements and may still be fruitful.

First of all, running Algorithm 1 on E_n, for $202 \leq n \leq 272$, does not produce the cage. It should also be noted that running algorithm 1 on E_{272} does not even produce a cubic graph. It is still an open question as to which minimum hint graph is needed to produce the current $(3,13)$-cage through the algorithm.

Some other methods developed by Ashlock may produce more results, such as those developed in (1) along with Schweizer. One other technique is the melter (Ashlock), also known as *edge reduction* (7, p. 17). To melt an edge off a graph is ideal for improving current results on cages since the process lowers the number of vertices in the graph, as shown in Theorem 10.

Algorithm 3 *The Melter*

- *Given a cubic graph, pick an edge with vertices u and v and denote the vertices adjacent to u and v as u_1, u_2, v_1, and v_2, appropriately as shown, where u_1 is not adjacent to u_2 and v_1 is not adjacent to v_2.*
- *Remove vertices u and v and all edges incident to u and v.*
- *Connect vertices u_1 and u_2 by edge. Connect vertices v_1 and v_2 by edge.*

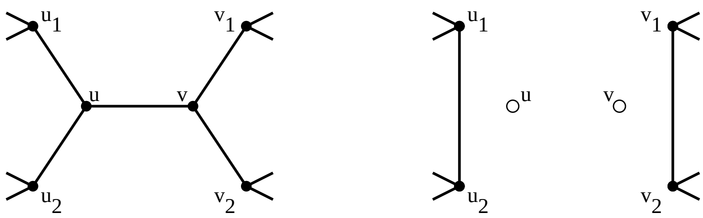

Figure 4.1 The Melter

Theorem 10 *Given a cubic graph G of order n, the Melter returns a cubic graph G' of order $n-2$.*

PROOF: Let G be a cubic graph of order n. Then, let $\{u, v\}$ be an edge of G where u is adjacent to u_1 and u_2 and v is adjacent to v_1 and v_2. This lowers the degree of vertices u_1, u_2, v_1, and v_2 to 2, since these vertices were cubic before the removal and only one incident edge is removed from each of them by the removal. We require u_1 and u_2 to be non-adjacent from the beginning, and thus they are not adjacent immediately after the edge removal. Similarly, v_1 and v_2 are not adjacent. Finally, add the edges $\{u_1, u_2\}$ and $\{v_1, v_2\}$. This raises the degree of all four vertices back to 3. Since no other vertices are perturbed by this method, all vertices of the graph are cubic. Since vertices u and v were completely removed from the graph, the order is reduced by 2. □

Ashlock and Schweizer used melting in (1) to produce $(3, 9)$-cages from the three non-isomorphic $(3, 10)$-cages.

4.1 Icos-explosion

One method that I am developing to resolve the $(3, 13)$-cage issue is the icos-explosion. The method follows.

Algorithm 4 *Icos-explosion*

- *Let I be the dual of the graph of the icosahedron.*
- *"Explode" each vertex of I into copies of C_{13}, cycles of order 13.*
- *Connect the cycles to the other cycles, replicating the adjacency of the original vertices that the cycles replaced.*
- *Run Algorithm 1 on this graph with additional restrictions that prevent the girth from dropping below 13.*

This method has lots of promise as it can be made more flexible than some of the other algorithms. For example, the restrictions mentioned in the last stage can require Algorithm 1 to keep track of which edges it adds to the graph G. If the algorithm runs into a situation where there is no edge that can be added without lowering the girth below 13, then it can backtrack and replace a previous edge with another choice and then continue.

The other major benefit to this technique is a target number of vertices. There are 20 vertices in the dual of the icosahedron and replacing them with cycles of length 13 results in 260 vertices for the whole graph. This is 12 vertices shy of the current best estimate on the $(3, 13)$-cage. Thus, if the method works, we will have a better bound on the cage. If the algorithm gets bogged down and runs into the situation where it cannot add an edge without lowering the girth, we can instruct the algorithm to find the edge $\{u, v\}$ that lowers the girth by the least amount, and then insert vertices into this edge (in pairs) to raise the girth back to 13 or better. Since there are 12 vertices left to play with it is conceivable that inserted extra vertices could led to better results.

Finally, the method can be generalized to produce results for other cages.

Algorithm 5 *Generalized Icos-explosion for obtaining $(3, g)$-cage*

- *Let G be a cubic graph of order n.*
- *Let Γ_v, $v \in V(G)$, be any sequence of graphs, where the graph Γ_v corresponds to vertex v in graph G, with the following properties:*
  - *The girth of each Γ_v is greater than or equal to g, the girth.*
  - *The maximum degree of any vertex in Γ_v is 2.*
- *"Explode" G by replacing each vertex v in G with its corresponding graph Γ_v.*
- *Connect the Γ_v graphs to each other appropriately, preserving the overall original structure of G.*
- *Run Algorithm 1 to change this new graph into a cubic graph, applying restrictions to the algorithm as mentioned above to preserve the girth g.*

The improvements on the original Icos-explosion can be applied to other cages; it is just a matter of finding the appropriate underlying structure G and the sequence of graphs Γ_v to explode with. By allowing the explosion graphs Γ_v to take on different structures depending on the vertex i in the original graph G, we provide more freedom to the Icos-explosion, which may produce better results.

4.2 Examples of Icos-explosions Resulting in Cages

We will use the generalized Icos-explosion method to generate the Heawood graph and McGee graph, the $(3,6)$- and $(3,7)$-cages, both from K_4, the $(3,3)$-cage.

4.2.1 Generating the Heawood Graph by Icos-explosion

Let $G = K_4$ with vertices a, b, c, and d. The Heawood graph has 14 vertices. Therefore, we will partition the vertices $\{0, 1, \ldots, 13\}$ and give them the following structures to act as the Γ_v graphs:

Vertex of G	Γ_v
a	$\Gamma_a = P_3$ with path 0-1-2
b	$\Gamma_b = P_4$ with path 3-4-5-6
c	$\Gamma_c = P_3$ with path 7-8-9
d	$\Gamma_d = P_4$ with path 10-11-12-13

Next, we need to re-establish the original overall structure of K_4. To do so, we create the following edges: $\{0, 9\}$, $\{1, 6\}$, $\{2, 11\}$, $\{3, 8\}$, $\{5, 10\}$, and $\{7, 12\}$. At this stage, 16 of the 21 edges of the Heawood graph are in place. Finally, we run Algorithm 1 on this graph and it completes to the Heawood graph. See Figure 4.2.

4.2.2 Generating the McGee Graph by Icos-explosion

Let $G = K_4$ with vertices a, b, c, and d. The McGee graph has 24 vertices. Therefore, we will partition the vertices $\{0, 1, \ldots, 23\}$ and give them the following structures to act as the Γ_v graphs.

Vertex of G	Γ_v
a	$\Gamma_a = P_6$ with path 0-1-2-3-4-5
b	$\Gamma_b = P_6$ with path 6-7-8-9-10-11
c	$\Gamma_c = P_6$ with path 12-13-14-15-16-17
d	$\Gamma_d = P_6$ with path 18-19-20-21-22-23

Next, we need to re-establish the original overall structure of K_4. To do so, we create the following edges: $\{0, 23\}$, $\{1, 8\}$, $\{3, 15\}$, $\{6, 18\}$, $\{10, 17\}$, and $\{13, 20\}$. Finally, running Algorithm 1 on this graph creates the McGee Graph. See Figure 4.3

4.3 Conclusions

The search for cubic cages continues and as this paper has shown, it is a lively research area with many different aspects from which to approach it. The main algorithm presented gives researchers an easy way to take a given graph, complete it (if possible) into a cubic graph and then finally check its girth to determine if it is a cage. My research from here will focus on identifying more minimum hint graphs for the cages we already know, determining which properties these graphs share and if these properties help researchers classify all minimum hint graphs for cages. This research could then lead us to find more cages. Finally, I will be working on relating cages to other cages through icos-explosions.

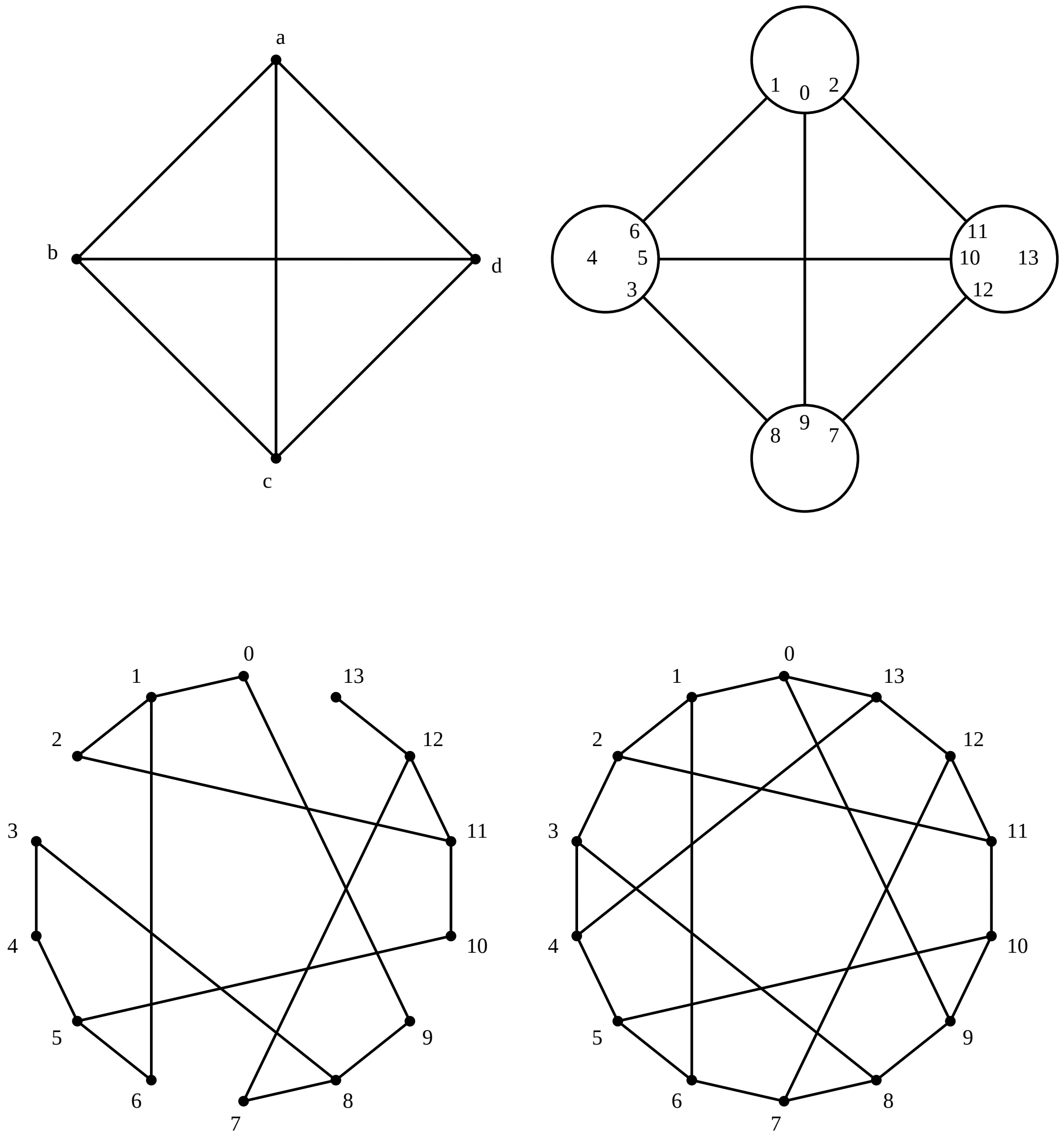

Figure 4.2 K_4; K_4 icos-exploded into 14 vertices; Regular drawing of K_4 icos-exploded before algorithm and after algorithm

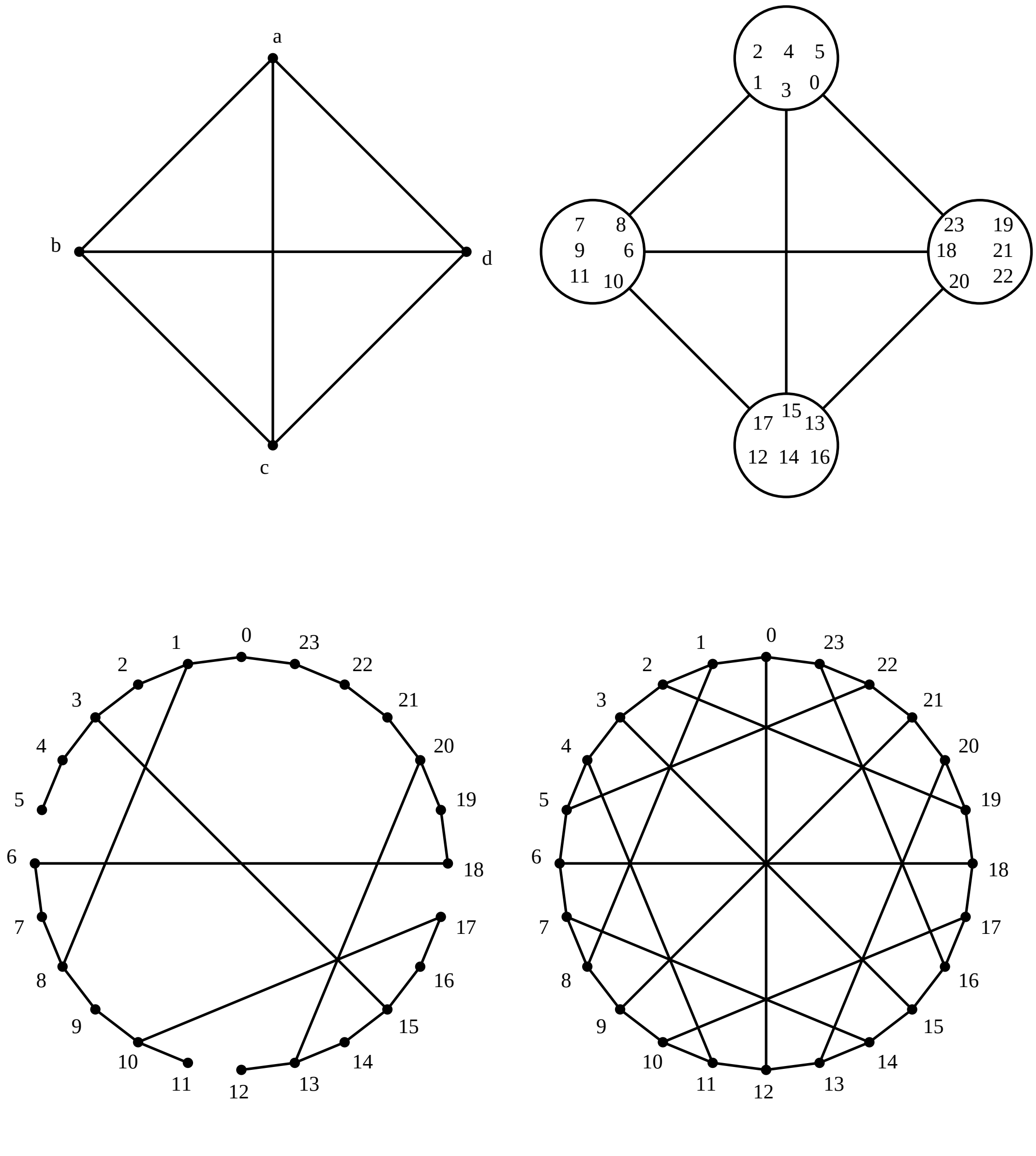

Figure 4.3 K_4; K_4 icos-exploded into 24 vertices; Regular drawing of K_4 icos-exploded before algorithm and after algorithm

Bibliography

[Ashlock] Ashlock, D. Personal Communication.

[1] Ashlock, D. and Schweizer, D. (1995). Graphical construction of cubic cages.

[2] Biggs, N. (1993). *Algebraic Graph Theory.* Cambridge University Press.

[3] Biggs, N. (1998). Constructions for cubic graphs with large girth. *Electronic Journal of Combinatorics*, page #A1.

[4] Diestel, R. (2000). *Graph Theory.* Springer.

[5] Dummit, D. S. and Foote, R. M. (1999). *Abstract Algebra.* John Wiley and Sons, Inc., 2nd edition.

[6] Exoo, G. (2004). Voltage graphs, group presentations and cages. *Electronic Journal of Combinatorics*, 11:#N2.

[7] Holton, D. A. and Sheehan, J. (1993). *The Petersen Graph.* Cambridge University Press.

[Royle] Royle, G. Cubic cages. http://www.cs.uwa.edu.au/~gordon/cages/index.html.

[8] West, D. B. (1996). *Introduction to Graph Theory.* Prentice Hall.