Simulation has been increasingly used for the solution of problems in business, engineering, physical sciences, and social sciences. Simulation is valuable both for its relative simplicity and for its potential to study a large number of variants of the original model without incurring the costs associated with physical experimentation or facing the difficulties of re-deriving formulas for output parameters that are inherent in mathematical analysis. For an arbitrary graph G, finding the largest eulerian subgraph of odd order is an NP-complete problem. The asymptotic behavior of the number of eulerian circuits in a complete graph of odd order had been studied, and the probability that all the edges are eventually used in a random maximal circuit has been shown to be asymptotic to $e^{3/4}n^{-1/2}$. However the probabilities for cases in which not all the edges are used remain unknown because of the complexity of the analysis. This thesis is a pilot study of all the cases of random maximal circuits on the complete graphs of odd order by computer simulation. The simulation estimates of the probability that all edges are used conform very closely to the asymptotic value. The probability that three edges are left after the random maximal circuit appears also to be $\Theta(n^{-1/2})$, but this pattern is not observed when 4 or more edges are left. The smaller the number of edges left after the random maximal circuit, the more likely the outcome is to occur. The left over edges tend to form a be connected subgraph.

INDEX WORDS: Euler Graph, Eulerian Circuit, Complete Graph, Random Circuits, Simulation, Probability
SIMULATION OF RANDOM MAXIMAL CIRCUITS IN COMPLETE GRAPHS

by

JUNFENG QU

B.Eng. East China University, P.R. China, 1990
M.Eng. East China University, P.R. China, 1994

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial
Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2000
ACKNOWLEDGEMENTS

In the course of my pursuit of science, I have accumulated a debt of great magnitude to numerous people who have taught, advised, supported and assisted me.

First of all, I would like to thank Dr. Robert W. Robinson, my major professor, for his constant guidance, advice, encouragement and patience. Without his support and help, it would have been impossible for me to complete this project.

I wish to thank my committee members, Dr. Eileen T. Kraemer and Dr. Andrew F. Seila for their participation, help and valuable suggestions.

I have saved the most important acknowledgement for my wife Li Sun, my parents, and my sisters. Their encouragement, willingness to help and love have been my essential support throughout the process.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>ACKNOWLEDGEMENTS</th>
<th>iv</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF TABLES</td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>viii</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Graph, definitions and notations</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Matrices of a graph</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Statement of problems</td>
<td>3</td>
</tr>
<tr>
<td>1.4 Two theorems</td>
<td>7</td>
</tr>
<tr>
<td>2 SIMULATION DESIGN AND TECHNOLOGY</td>
<td>11</td>
</tr>
<tr>
<td>2.1 Random model architecture</td>
<td>12</td>
</tr>
<tr>
<td>2.2 Java technology</td>
<td>12</td>
</tr>
<tr>
<td>2.3 Simulation architecture</td>
<td>13</td>
</tr>
<tr>
<td>3 SIMULATION AND DATA ANALYSIS</td>
<td>17</td>
</tr>
<tr>
<td>3.1 Nature of simulation</td>
<td>17</td>
</tr>
<tr>
<td>3.2 Data analysis</td>
<td>17</td>
</tr>
<tr>
<td>3.3 Validation of simulation</td>
<td>23</td>
</tr>
<tr>
<td>4 CONCLUSIONS AND FUTURE WORK</td>
<td>26</td>
</tr>
<tr>
<td>4.1 Probability analysis</td>
<td>26</td>
</tr>
<tr>
<td>4.2 Degree analysis of remainder graph</td>
<td>27</td>
</tr>
</tbody>
</table>
4.3 Structural analysis of the complementary graph..........................33
4.4 Summary and future work......................................................37
REFERENCES.................................................................................38
## LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 – 1</td>
<td>Exact Results of Random Circuit on $K_5$</td>
</tr>
<tr>
<td>3 – 2</td>
<td>Simulation Results of Random Circuit on $K_5$</td>
</tr>
<tr>
<td>4 – 1</td>
<td>Degree Patterns for Complements of Circuits in $K_n$ of Size $m = 6, 7, 8$</td>
</tr>
<tr>
<td>4 – 2</td>
<td>Degree Patterns for Complements of Circuits in $K_n$ of Size $m = 9, 10$</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 – 1</td>
<td>Complete Graph $K_3$</td>
</tr>
<tr>
<td>1 – 2</td>
<td>Complete Graph $K_7$</td>
</tr>
<tr>
<td>1 – 3</td>
<td>Adjacency Matrix Representation of Complete Graph $K_5$</td>
</tr>
<tr>
<td>1 – 4</td>
<td>Königsberg Bridge Problem and its Graph Representation</td>
</tr>
<tr>
<td>1 – 5</td>
<td>A Maximal Circuit of Minimal Size</td>
</tr>
<tr>
<td>2 – 1</td>
<td>Simulation Architecture Components</td>
</tr>
<tr>
<td>2 – 2</td>
<td>Flowchart of Simulation Architecture</td>
</tr>
<tr>
<td>4 – 1</td>
<td>Probability of Random Circuit on $K_n$</td>
</tr>
<tr>
<td>4 – 2</td>
<td>Converging Process of Random Circuit on $K_5$</td>
</tr>
<tr>
<td>4 – 3</td>
<td>Comparison of Theoretical Analysis with Simulation at Zero Left on $K_n$</td>
</tr>
<tr>
<td>4 – 4</td>
<td>Comparison of Mathematical Solution with Simulation Results</td>
</tr>
<tr>
<td>4 – 5</td>
<td>$e(n, 3)$ and $e(n, 4)$ with Regression Analysis</td>
</tr>
<tr>
<td>4 – 6</td>
<td>Simulation Results of $e(n, k)$, $5 \leq k \leq 12$</td>
</tr>
<tr>
<td>4 – 7</td>
<td>Isomorphic Graphs of Size 4</td>
</tr>
<tr>
<td>4 – 8</td>
<td>Non-isomorphic Graphs of Size 6</td>
</tr>
</tbody>
</table>
1.1 Graphs: definitions and notations

A graph $G$ consists of a finite non-empty set $V(G)$ of elements called vertices and a finite set $E(G)$ of distinct pairs of distinct elements of $V(G)$ called edges (see figure 1-1). The $V(G)$ is called the vertex set of $G$ and $E(G)$ is called the edge set of $G$. The edge \{v, w\} (where v and w are vertices of $G$) is denoted by $vw$ in the thesis, and the graph $G$ is given as $G = (V, E)$, where $V = V(G)$ and $E = E(G)$. Two vertices are adjacent if they are the end vertices of some edges. If the edges $G$ consist of are ordered pairs of vertices, then $G$ is called a directed graph. Otherwise, the edges are unordered and $G$ is called an undirected graph. The definition of graph excludes loops and multiple edges, and coincides with what is sometimes called a simple graph. We define the size of a graph to be the number of edges contained in the graph, and the order of a graph to be the number of vertices in the graph. A graph is called even graph is the degree of all its vertices are of even.

![Complete Graph K₃](image)

Figure 1-1 Complete Graph $K₃$.

A complete graph is a simple graph in which every pair of vertices is adjacent. We denote the complete graph with $n$ vertices by $K_n$. For example, $K_3$ is shown in figure 1-1 and $K_7$ is shown in figure 1-2. It is clear that $K_n$ has $n(n-1)/2$ edges.
The number of edges incident on a vertex \( v \) is called the *degree of the vertex*, and it is denoted by \( \text{deg}(v) \).

Figure 1-2  Complete Graph \( K_7 \).

A *walk* of length \( m \) on undirected graph \( G = (V, E) \) is a sequence \( v_0, v_1, \ldots, v_m \) of vertices of \( G \) such that \( \{v_i, v\} \in E(G) \) for \( 1 \leq i \leq m \). This walk is a circuit if \( v_m = v_0 \) and \( \{v_i, v\} \neq \{v, v_i\} \) for \( 1 \leq i < j \leq m \), that is, no edge is repeated.

A walk is a *trail* if all its edges are distinct. A trail is *open* if its end vertices are distinct; otherwise, it is *closed*. An open trail is a *path* if all its vertices are distinct. A closed trail is a *circuit* if all its vertices except the end vertices are distinct.

An eulerian circuit in Graph \( G \) is a circuit of length \( |E| \). A graph possessing an Euler circuit is an eulerian graph. Two eulerian circuits are called equivalent if one is a cyclic permutation of the other. It is clear that the size of such an equivalence class equals the common length of the walks in the class. The number of the equivalent classes of eulerian circuits in Graph \( G \) is denoted by \( \text{Eul}(G) \).

A tournament is a digraph in which, for each pair of distinct vertices \( v \) and \( w \), either \( (v, w) \) or \( (w, v) \) is an edge, but not both. A tournament is regular if the in-degree is equal to the out-degree at each vertex. Each eulerian circuit in \( K_n \) induces a regular tournament by orienting the edges according to the directions in which they are traversed.
1.2 Matrices of a graph

Graph can be represented by matrix. The adjacency matrix of a simple graph is a matrix with rows and columns labeled by vertices, with a 1 or 0 in each position according to whether the vertices are adjacent or not. Let \( G \) be a graph with vertex set \( \{ v_1, v_2, \ldots, v_n \} \) and edge set \( \{ e_1, e_2, \ldots, e_m \} \). The adjacency matrix of \( G \) is the \( n \times n \) matrix \( A(G) = (a_{ij}) \), where:

\[
a_{ij} = \begin{cases} 
1, & \text{if } v_i \text{ and } v_j \text{ are adjacent} \\
0, & \text{if not.}
\end{cases}
\]

For a simple graph with no self-loops, the adjacency matrix must have 0s on the diagonal. For an undirected graph, the adjacency matrix is symmetrical. For the complete graph, the adjacent matrix is symmetrical and all values on the diagonal are 0s. Figure 1-3 shows the representation of complete graph \( K_5 \) via adjacent matrix.

\[
A(G) = \begin{bmatrix}
v_1 & v_2 & v_3 & v_4 & v_5 \\
v_1 & 0 & 1 & 1 & 1 \\
v_2 & 1 & 0 & 1 & 1 \\
v_3 & 1 & 1 & 0 & 1 \\
v_4 & 1 & 1 & 1 & 0 \\
v_5 & 1 & 1 & 1 & 0 
\end{bmatrix}
\]

Figure 1-3 Adjacency Matrix Representation of Complete Graph \( K_5 \).

1.3 Statement of problems

As the appearance of Euler’s name in the problem suggests, the problem is not new to graph theory. In 1736, Euler, the great Swiss mathematician, solved the Königsberg bridge problem. The problem was to start at any one of the four land areas (marked as A, B, C, D in figure 4-1), walk across each bridge exactly once, and then return to the starting point, that is, to establish a closed walk over all seven bridges without recrossing any of them. As a generalization of the Königsberg bridge problem, Euler first showed
that the problem was equivalent to establishing a closed trail along the edges of the graph of figure 1-4, where the vertices A, B, C, and D represent the land areas and the edges represent the bridges connecting the land areas [25]. Euler showed in general that the existence of such a closed trail (an eulerian trail) requires that the degree of every vertex be even. Hence there is no solution to the Königsberg bridge problem, as all four vertices of the corresponding graph have odd degree. Euler also asserted (without proof) the converse, that every connected even graph has an eulerian circuit. Hence for complete graphs, \( Eul(K_n) > 0 \) if, and only if \( n \) is odd.

Figure 1-4 Königsberg Bridge Problem and its Graph Representation

In 1873, Hierholzer gave a vigorous proof of Euler’s converse assertion.
Reiss[17] calculated $Eul(K_7)$ as the number of legal circular arrangements of the 21 dominoes (doubles excluded) over the set \{1, 2, ..., 7\}, and the paper was published in 1859. He viewed the problem as that of counting circular sequences from the above set in which each element appears 3 times and each unordered pair appears as a consecutive pair exactly once. Then he broke the enumeration down further by considering all the subsequences that could be formed from \{1, 2, 3\}, then all the ways that subsequences from \{4, 5, 6, 7\} could be interpolated.

Lucas[13] observed that the circular domino arrangements with doubles excluded could be thought of as eulerian circuits in a complete graph. This approach greatly simplified calculation of $Eul(K_n)$. Tarry took this point of view in calculating $Eul(K_n)$. His method was a recurrence based on deleting a vertex and pairing its incident edges in all possible ways. It could be used on the computer to find $Eul(K_n)$ up to at least $n = 15$.

Shishov and Thuan[21] in 1968 showed that $Eul(G)$ could be written as a solution of an upper triangular set of linear equations. Unfortunately, the number of equations for $Eul(K_n)$ grows very quickly as a function of $n$.

Sorokin[24] claimed that he had determined $Eul(K_n)$ exactly in 1969, but he was not correct. First of all, the formula stated for the number of regular tournaments was wrong. This was pointed out by Liskovets[12]. Second, the assumption that different regular tournaments had the same number of eulerian circuits was incorrect.

Wen[27] gave the counts of equivalence class of eulerian circuits up to $n = 7$ in his paper published in 1989. This is an extended notion of equivalence, which takes into account automorphism of $K_n$.

An undirected graph is an euler graph iff every vertex has even degree. The numbers of eulerian graphs with $n = 1, 2, ...$ nodes are $1, 1, 2, 3, 7, 16, 54, 243, ...$ [5, 6, 14, 19, 23]. There is an explicit formula giving these numbers.
As noted earlier, the eulerian graph (without isolated vertices) are precisely the connected even graphs. The numbers of those graphs with \( n = 1, 2, \ldots \) nodes are 1, 0, 1, 1, 4, 8, 37, 184, ... [9, 12, 19, 23].

Finding the largest subgraph of graph having an odd number of vertices which is Eulerian is an NP-complete problem [22].

In 1998, Mckay and Robinson[15] determined the asymptotic behavior of the number of eulerian circuits in a complete graph of odd order. They introduced two models of random circuits on a given even undirected graph \( G \) or balanced bidirected graph. The greedy model stated that we have an eulerian graph or digraph \( G \). Starting at an arbitrary node \( v \), following any incident edge at random, and extend the trail by choosing an unused edge incident to the last vertex visited at random subject to each edge being used at most once. At each step, the choice between the available next edge is made uniformly at random. This random trail will eventually end at the starting node \( v \), forming a circuit that can not further extended. The other model is a random pairing of the edges at each vertex. For an undirected graph, at each vertex divide the incident edges into unordered pairs, with each possible pairing being equally likely and independent at each vertex.

McKay and Robinson gave the mathematical analysis in their paper regarding the probability of \( P(K_n) \) that every edge of \( K_n \) was used by this circuit as \( n \to \infty \) where odd \( n \) on both undirected and directed graphs. They concluded that if a random maximal circuit, constrained to use each edge at most once, is taken on \( K_n \), the probability \( P(K_n) \) for undirected \( K_n \) were

\[
P(K_n) \to e^{3/4} n^{-1/2} \quad \text{as } n \to \infty
\]

This thesis uses the greedy model on a complete graph introduced by McKay and Robinson. It is clear that the circuits are formed each time in the greedy model while
traversing the graph. The exact probabilities of 0 edges left on $K_n$ were given up to 21 nodes.

On $K_n$, $Eul(G) = 0$ if $n$ is even. In the thesis, we study random maximal circuits in $K_n$ for odd $n$, where $K_n$ is undirected.

The thesis will study the simulation process of the greedy model stated above. We will let $E(n, m)$ denote the event that $m$ edges are left over after choosing a random maximal circuit in $K_n$ in the greedy model, and $e(n, m)$ be the probability of $E(n, m)$. For example, if $n = 5$, then $e(5, 3)$ means that after the greedy model over $K_5$, the probability of 3 edges left.

Here are some questions which are studied in this thesis:

1. For the greedy model of random maximal circuits, what is $e(n, m)$ for $m = 0, 3, 4, \ldots$?
2. What is expected number of edges left after the random maximal circuits traversing in $K_n$?
3. What are the likely degrees of subgraphs of left-over edges?
4. What is the graph pattern of the leftover subgraph?
5. Does $e(n, m) \to 0$ as $n \to \infty$ for arbitrary fixed $m$?

1.4 Two theorems

**Theorem 1:**

A connected, undirected graph $G$ contains an eulerian circuit if and only if the degrees of the vertices of are all even.

**Proof:**
**Necessity.** Let $T$ be an Euler circuit in graph $G$, where $G$ is connected and undirected. Suppose we traverse $T$ starting from any vertex, say $v_j$, in $G$. Let $T$ be:

$$v_j = x_1, e_1, x_2, e_2, x_3, \ldots, e_{r-1}, x_n, e_n, x_{r+1} = v_j.$$  

All the edges are distinct since $T$ is a circuit. The vertices $x_2, \ldots, x_r$ may not all be distinct and some of these vertices may be $v_j$. Then it is clear that the pair of successive edges $e_i$ and $e_{i+1}$, $1 \leq i \leq r-1$, contributes 2 to the degree of the vertex $x_{i+1}$. In addition, vertex $v_j$ gets a contribution of 2 to its degree from the initial and the final edges $e_1$ and $e_r$. Thus all the vertices are of even degree, since $T$ contains all the edges of $G$.

**Sufficiency.** Let $G$ be a connected graph with all vertices of even degree. Start a path at some arbitrary vertex $v_0$, and proceed along a previously unused edge to the next vertex until the path returns to $v_0$ and a circuit is completed. If all edges have been used then the desired eulerian circuit has been established. If some edges have not been used, then let $\Phi$ be the circuit just completed. Since $G$ is connected, $\Phi$ must have passed through some vertex, say $v_i$, which is the terminal vertex of some hitherto unused edge. If all edges used by $\Phi$ were removed, then the resulting graph would still have all vertex degrees even since $\Phi$ must use an even number of edges incident at every vertex.

Starting from $v_i$, we could traverse a circuit $\Phi'$ starting and finishing at $v_j$. Once more, if all remaining edges are used by $\Phi'$, we are finished. The part of $\Phi$ from $v_0$ to $v_i$ followed by circuit $\Phi'$, followed by the part of $\Phi$ from $v_i$ to $v_0$, would be the required eulerian circuit. If some edges are still unused, let the union of $\Phi$ and $\Phi'$ described above be the new circuit $\Phi$. We could again find a vertex $v_j$ met by $\Phi$ which is the terminal of some unused edge. We could then proceed to form a new circuit $\Phi'$ starting at $v_j$ and so on until finally all edges are used in this way and an eulerian circuit $\Phi$ has been obtained.

**Corollary 1:**

For $n$ odd the undirected complete graph $K_n$ is eulerian.

**Proof:**
Now we observe that for odd \( n \) the complete graph \( K_n \) is undirected, connected, and has even degree \( n - 1 \) at every vertex. Hence, for odd \( n \), the undirected complete graph \( K_n \) is eulerian.

**Corollary 2:**

The undirected complete graph \( K_n \), where \( n \) is odd, has some edge-disjointed circuits

*Proof:*

Since an undirected complete graph \( K_n \) for odd \( n \) is an even graph, from corollary 1 we know that the complement of a random maximal circuit is an even graph. Hence, the complement is eulerian. Even though it may not be connected, each component can be a circuit.

**Theorem 2:**

For the random maximal circuits traversing in undirected \( K_n \), each edge can be only used once and \( n \) is odd. The maximum size of the remainder graph is \((n-1)(n-3)/2\).

*Proof:*

Given an undirected complete graph \( K_n \), where \( n \) is odd, \( K_n \) has \( n(n-1)/2 \) edges. \( K_n \) is also an \((n-1)\)-regular graph, i.e. all the vertices of \( K_n \) are of equal degree of \((n-1)\). For each vertex, there are \((n-1)\) edges incident with the vertex to connect with other \((n-1)\) vertices, so there are \((n-1)*n/2\) total number of edges.

For the random maximal circuit start from an arbitrary vertex \( v_0 \), and proceed along edges connected to \( v_0 \) and uniformly randomly pick one unused edge to the next vertex until a path returns to \( v_0 \) and a circuit is formed. For each circuit formed, it takes one edge to go out of \( v_0 \) and one edge to come back to \( v_0 \). So, the maximum number of circuits that can be formed starting from \( v_0 \) is \((n-1)/2\), i.e. the number of pairs that starting from \( v_0 \). For each circuit starting from \( v_0 \), the shortest length of each circuit consumes one edge for each pair of the edges from \( v_0 \). That is, the number of edges of the
maximum number of circuits starting from \(v_0\) is \(3^{n-1}/2\). Thus, the maximum number of edges left over after the random maximal circuit is

\[
(n-1)n/2 - 3(n-1)/2 = (n-1)(n-3)/2
\]  

(1)

Figure 1-5 shows a maximal circuit of minimum size starting from \(v_0\).

![Figure 1-5 A Maximal Circuit of Minimal Size](image-url)
CHAPTER 2
SIMULATION DESIGN AND TECHNOLOGY

For the past fifty years, simulation has been increasingly used for the solution of problems of human endeavor in business, engineering, science, and social science. Simulation is valuable both for its relative simplicity as compared to analytical (mathematical) techniques and for its potential to study a large number of variants of the original model without incurring the experimental costs associated with experimenting with the real system or difficulties in re-deriving formulas for output parameters that are inherent in mathematical analysis.

There are two ways to study problems of this sort [20]:

a. Use mathematical analysis, and

b. Use Simulation

There is an important difference between these two approaches. A mathematical analysis will yield formulas or a computational procedure to produce an exact value of the model’s performance measure, whereas a simulation will yield a sample of observations that can be used to compute confidence intervals for the performance measure, and therefore to estimate the value of the performance measure.

Static simulations have been used frequently by mathematicians, physicists, engineers, and mathematical statisticians to solve mathematical problems that could not be solved using mathematical analysis. Static simulations operate by generating random variants and transforming them according to formulas or rules that compose the model. This process is repeated independently many times to produce a set of independent, identically distributed observations that are then used to study the characteristics of the
transformed random variable. Thus, time does not play an essential role in static simulation, although time may be involved in the transformation that is used to produce each observation.

2.1 Random model architecture

The random maximal circuit model for the complete graph $K_n$ was introduced in chapter one, and questions to be addressed were also introduced. We designed the random maximal circuit pseudocode as follows:

1. Initialize all $edges[n]$ to be unmarked, $current\_vertex \leftarrow v_0$, $next\_vertex \leftarrow v_0$
2. **While** ($edges[current\_vertex]$ has incident unused edges) {
   
   $next\_vertex \leftarrow$ randomly choose one unused edge incident to $current\_vertex$
   
   Mark the edge $\{current\_vertex, next\_vertex\}$
   
   $current\_vertex \leftarrow next\_vertex$
   
   }

2.2 Java technology

By the end of the 1980s and in the early 1990s, object-oriented programming using C++ took hold. Indeed, for a brief moment it seemed as if programmers had finally found the perfect language. Because C++ blended the high efficiency and stylistic elements of C with the object-oriented paradigm, it was a language that could be used to create a wide range of programs. In 1991, Sun Microsystems funded an internal corporate research project code-named Green. The language was initially called “Oak” but was renamed “Java” in 1995. The Internet helped catapult Java to the forefront of programming, and Java in turn has had a profound effect on the Internet, because Java expands the universe of objects that can move about freely in cyberspace.
In the software development world, reliable reusable classes become building blocks. The building blocks are software components. Component-based software development systems may support some or all the following capabilities[16].

**Object-Oriented Program:** Under OOP, application software is developed as objects consisting of attributes (data members) and methods (member functions). The advantages of encapsulation, inheritance and polymorphism have been well documented.

**Persistence:** Mechanisms are provided to save and restore the state of executing objects with little or no programming.

**Platform Independence:** While “Write-Once, Run Anywhere” is not a requirement, it certainly simplifies the developer’s job. If this is fully supported, any object can be downloaded to any machine and executed without recoding or even recompiling.

**Distribution:** Distribution is not a requirement, but it is useful to be able to have components work together even if they are not executing on the same machine. This is facilitated by providing high-level mechanisms for distributed objects to communicate, typically through remote method calls or remote handling of events.

At present the two widely used platforms for building software from components are ActiveX and Java. Java has the advantages of platform independence, a simple programming environment and ease of GUI development. For these reasons, Java has been used to develop the simulation and implemented in this thesis.

2.3 Simulation architecture

The simulation application is composed of five components. These are the control module, the virtual circuit module, the random circuit module, the data collection and analysis module and the documentation module. The Architecture construction is shown in figure 2-1
The control module exercises overall supervision of the execution of the simulation. After the user control parameters are obtained from the GUI, it starts the virtual circuit module, the random circuit module and the data collection module. After the random circuit finishes the simulation, the data collection module collects the required data, and analyzes the results. If the specified requirements are not met, it notifies the control module to continue the random circuit. Once the requirements are met, it notifies the control module to stop all the simulations and start the documentation module to display the results.

The virtual circuit module visually demonstrates the progress of random circuit on the complete graph $K_n$. When the order $n$ is input by the user, the virtual module sets up a display canvas on which is drawn $K_n$. When the start control button is pressed, it displays the progress of random circuit process on the graph by changing the color of the vertex visited and drawing a line to show the edge just traversed.

The Random circuit module does the real simulation work at run time. The random module workflow was introduced in section 2.1. This pseudocode was implemented in
Java. To represent the complete graph, an adjacency matrix was used. For the random circuit process, all previous transition history should be remembered in order to allow the next transition to be calculated. A boolean set was used to remember all the visited edges of the graph $K_n$.

The data analysis and collection module’s function was to collect results of each random walk. The information includes the number of edges left after each random walk, the accumulated number of left edges appearing under certain constraints input by the user, and the degree pattern of the left-over subgraph. After the simulation was done, the module analyzes the data and makes the decision about whether the simulation should continue with additional replications or stop. If the decision to stop is made, it fires an event to the control module and transfers the results analyzed to the documentation module for display.

The main role of the documentation module is to display the simulation results to the end user on the GUI after the completion of simulation.

The flowchart of the algorithm used by the simulation is shown in figure 2-2.
Figure 2-2 Flowchart of Simulation Architecture
3.1 Nature of simulation

Even though simulation is a powerful tool for solving mathematical problems, it has inherent limitations.

First, simulation can not compute an exact value of parameters that are to be studied. Instead, simulation will produce a sample of data that can be used to estimate those parameters.

Second, as the number of replications is increased, the point estimates for the parameters approach the true values. If we could perform an indefinitely large number of replications, the estimates would converge arbitrarily closely to the true values.

The simulation approach will give a valid, correct result if the model is correct and the number of replications is sufficiently large.

3.2 Data analysis

The primary purpose of most simulations is to approximate certain parameters in order to study them. Simulation data analysis involves computing the estimates of parameters of modeling data generated by the simulation model. Suppose that $X_1, X_2, \ldots, X_n$ represent a sequence of output data. The following three forms of convergence exit for such a sequence of random variables on the same probability space[1]:

1. Almost sure convergence. The sequence $X_1, X_2, \ldots$ converges to the random variable $X$ almost surely (or with probability one) (denoted as $X_n \rightarrow X \,(a.s.)$) if for every $\varepsilon > 0$,

   $$P(\left| X_k - X_n \right| \leq \varepsilon \text{ for all } k \geq n ) \rightarrow 1 \text{ as } n \rightarrow \infty$$
(2) Convergence in probability. The sequence $X_1, X_2, \ldots$ converges to $X$ in probability (denoted as $X_n \xrightarrow{(p)} X$) if for every $\epsilon > 0$,

$$P( |X_k - X_n| \leq \epsilon ) \to 1 \quad \text{as } n \to \infty$$

(3) Convergence in distribution. The sequence $X_1, X_2, \ldots$ converges to the random variable $X$ in probability (denoted as $X_n \xrightarrow{(d)} X$) if

$$P( X_n \leq x ) \to P( X \leq x) \quad \text{as } n \to \infty$$

at all points $x$ where the cumulative distribution function $P( X \leq x)$ is continuous.

Among the above three forms, the first form is the strongest while the last is the weakest and easiest to establish.

Now, let’s suppose that the random variables $X_1, X_2, \ldots$ are from some distribution with an unknown parameter $\theta$ and the objective is to estimate a function $f(\theta)$. For fixed $n$, let $g_n = g_n(X_1, X_2, \ldots X_n)$ be an estimator of $f(\theta)$. If $E(g_n) = f(\theta)$, then $g_n$ is an unbiased estimator, and if $g_n \xrightarrow{(p)} f(\theta)$ then $g_n$ is a consistent estimator.

Suppose that $X_1, X_2, \ldots X_n$ are independent, identically distributed (i.i.d.) observations produced by the random model simulation. Let

$$\overline{X}_n = \frac{1}{n} \sum_{i=1}^{n} X_i$$

be the sample mean of the $X_i$’s. Since $E( \overline{X}_n ) = \mu$, the mean of the distribution, $\overline{X}_n$ is an unbiased estimator of $\mu$. At same time, by the strong law of large numbers, we have

$$\overline{X}_n \xrightarrow{(a.s.)} \mu \quad \text{as } n \to \infty$$

so $\overline{X}_n$ is also a consistent estimator of $\mu$.

From the central limit theorem, we know that

$$\frac{\overline{X}_n - \mu}{s/\sqrt{n}} \xrightarrow{(d)} N(0,1) \quad \text{as } n \to \infty$$

where $N(0, 1)$ denotes a normal random variable with mean 0 and variance 1. That is,
\[ P \left( \frac{X_n - \mu}{\sigma / \sqrt{n}} \leq z \right) \to \Phi(z) \quad \text{as } n \to \infty \]

where \( \Phi \) is the distribution function of the standard normal random variable.

The central limit theorem remains valid when the potentially unknown parameter \( \sigma^2 \) is replace by its unbiased and consistent estimator

\[ S_n^2 = \frac{1}{n-1} \sum_{i=1}^{n} \left( X_i - \bar{X}_n \right)^2 \quad (2) \]

therefore, for sufficiently large \( n \),

\[ P \left( \left| \frac{X_n - \mu}{S_n(X)/\sqrt{n}} \right| \leq z_{1-\alpha/2} \right) \approx 1 - \alpha \]

where \( z_{1-\alpha/2} \) denotes the 1- \( \alpha/2 \) quantile of \( N(0,1) \).

Now, solving the above equation treating \( \mu \) as unknown, we can obtain the 1- \( \alpha \) confidence interval:

\[ \bar{X}_n \pm z_{1-\alpha/2} \frac{S_n(X)}{\sqrt{n}} \quad (3) \]

By replacing the normal quantile \( z_{1-\alpha/2} \) by the larger quantile \( t_{n-1, 1-\alpha/2} \) of the t-distribution with \( n-1 \) degrees of freedom, the estimate will be approximately valid for small \( n \). Thus, \( \bar{X}_n \) is an unbiased estimator for \( \mu \), and an approximate 100(1- \( \alpha \)) percent \(( 0 < \alpha < 1 \) ) confidence interval for \( \mu \) can be expressed as:

\[ \bar{X}_n \pm t_{n-1, 1-\alpha/2} \frac{S_n(X)}{\sqrt{n}} \quad (4) \]

where \( S_n(X) \) can be calculated from equation (2). This procedure is sometimes called fixed-sample-size procedure. [10]

The options for analyzing simulation experiments depend on the type of simulation at hand. There are two principal types of simulations[1, 10], terminating or nonterminating depending on whether or not there is an obvious way for determining the run length. A terminating simulation is one for which there is a “natural” event E that
specifies the length of each run (replication). Since different runs use independent random numbers and the same initialization rule, this implies that comparable random variables from the different runs are i.i.d.. The event \( E \) often occurs at a time point when the system is “cleaned out” or at a time point beyond which no useful information is obtained. The event should be specified before any runs are made. A nonterminating simulation is one for which there is no natural event \( E \) to specify the length of a run.

Our random maximal circuit model is a terminating simulation since it is halted when all edges incident to the current vertex are used.

Let’s suppose that \( n \) independent replications of a random maximal circuits had been done and each replication is terminated by the event \( E \) and is begun with the “same” initial conditions, Let \( X_1, X_2, \ldots, X_n \) be the resulting i.i.d. random variables, the point estimate and confidence interval for the mean \( \mu = E(X) \) can be easily computed from equations (2) and (4) for the fixed-sample-size procedure.

One disadvantage of the fixed-sample-size procedure based on \( n \) replications is that the analyst has no control over the confidence-interval half-length (or the precision of \( \bar{X}_n \)). For fixed \( n \), the half-length will depend on \( Var(X) \), the population variance of the \( X_i \)’s.

In fact, the number of replications may be a random variable. If the estimate \( \bar{X} \) is such that \(| \bar{X} - \mu | = \varepsilon \), then we say that \( \bar{X} \) has an absolute error of \( \varepsilon \). If we continue replications of a simulation until the half-length of the 100(1 - \( \alpha \)) percent confidence interval stated in equation (4) is less than or equal to \( \varepsilon \) (where \( \varepsilon > 0 \)), then

\[
1 - \alpha = P(\bar{X} - \text{half-length} \leq \mu \leq \bar{X} + \text{half-length}) \\
= P(| \bar{X} - \mu | \leq \text{half-length}) \\
\leq P(| \bar{X} - \mu | \leq \varepsilon )
\]

Then \( \bar{X} \) has an absolute error of at most \( \varepsilon \) with a probability of approximately \( 1 - \alpha \). That is, if we construct 100 independent 90 percent confidence intervals using the above
stopping rule, we would expect $X$ to have an absolute error of at most $\varepsilon$ in about 90 out of the 100 cases; in about 10 cases the absolute error would be greater than $\varepsilon$.

Suppose that based on the fixed number $n$ of replications, we had already constructed a confidence interval for $\mu$. An approximate expression for the total number of replications $N_a(\varepsilon)$ required to obtain an absolute error of $\varepsilon$ is given by equation (5) under the assumption that the estimate $S_n$ of the population variance will not change as the number of the replications increases.

$$N_a(\varepsilon) = \min \left\{ i \geq n : t_{i-1,1-\alpha/2} S_n / \sqrt{i} \leq \varepsilon \right\}$$  \hspace{1cm} (5)

The value of $N_a(\varepsilon)$ can be determined iteratively by increasing $i$ by 1 until a value of $i$ is obtained for which $t_{i-1,1-\alpha/2} S_n / \sqrt{n} \leq \varepsilon$. If $N_a(\varepsilon) > n$, and if we make $N_a(\varepsilon) - n$ additional replications of the simulation, then the estimate $\bar{X}$ based on all $N_a(\varepsilon)$ replications should have an absolute error of approximately $\varepsilon$.

Another way of measuring the error in $\bar{X}$ is to define the relative error $\beta$, given by $\beta = \frac{|X - \mu|}{|\mu|}$, so that the percentage error in $\bar{X}$ is $100\beta$ percent. Suppose that the model continues replication of the simulation until the half-length of the confidence interval given in equation (4), divided by $|X|$, is less than or equal to $\beta (0 < \beta < 1)$. The ratio is an estimate of the actual relative error. Then we have

$$1 - \alpha = P(|X - \mu| / |X| \leq \text{half-length} / |X|)$$

$$\leq P(|X - \mu| \leq \beta |X|)$$

$$= P(|X - \mu| \leq \beta |X - \mu + \mu|)$$

$$\leq P(|X - \mu| \leq \beta |X - \mu + \mu|)$$

$$\leq P(|X - \mu| \leq \beta (|X - \mu| + |\mu|))$$

$$= P((1 - \beta) |X - \mu| \leq \beta |\mu|)$$

$$= P(|X - \mu| / |\mu| \leq \beta / (1 - \beta))$$

So, $X$ will have a relative error of at most $\beta / (1 - \beta)$ with a probability of approximately $1 - \alpha$. For example, if we construct 100 independent 90 percent confidence intervals using the above stopping rule, we will expect $\bar{X}$ to have a relative error of at
most $\beta / (1 - \beta)$ in about 90 of 100 cases; in about 10 cases the relative error would be greater than $\beta / (1 - \beta)$.

If we have constructed a confidence interval for $\mu$ based on a fixed number $n$ replications, and if we assume that our estimates of both the population mean and population variance will not change as the number of replications increases, an approximate equation for the number of replications required to obtain a relative error of $\beta$ is given by equation (5).

$$N_r(\beta) = \min \left\{ i \geq n : \frac{t_{1-\alpha/2}S_n}{\sqrt{\bar{x}_r}} \leq \beta \right\}$$

(5)

where $\beta' = \beta/(1-\beta)$ is the “adjusted” relative error needed to get an actual relative error of $\beta$. If $N_r(\beta) > n$, and if we make $N_r(\beta) - n$ additional replications of the simulation, then the estimate $\bar{X}$ should have a relative error of approximately $\beta$ based on all of the $N_r(\beta)$ replications.

In order to directly obtain an estimate $\bar{X}$ with a relative error of $\beta$, a sequential procedure, in which the new replications are added one at a time, was employed so that an estimate with a specified relative error takes only as many as replications as are actually needed.

The specific objective of the procedure is to obtain an estimate of $\mu$ with a relative error of $\beta$ ($0 < \beta < 1$) and a confidence level of 100(1-$\alpha$) percent. Let’s choose an initial number of replications $n_0 > 2$ and let

$$\delta(n, \alpha) = t_{n-1,1-\alpha/2} \frac{S_n}{\sqrt{n}}$$

be the usual confidence-interval half-length. Then the sequential procedure is as follows:

1. Make $n_0$ replications of the simulation and set $n = n_0$.
2. Compute $\bar{X}_n$ and $\delta(n, \alpha)$ from $X_1, X_2, \ldots, X_n$. 

3. If \( \delta(n, \alpha) \mid |\bar{X}_n| \leq \beta' \), where \( \beta' = \beta/(1-\beta) \), use \( \bar{X}_n \) as the point estimate for \( \mu \) and stop. Otherwise, replace \( n \) by \( n+1 \), make an additional replication of the simulation, and go to step 2.

The confidence interval \( \text{CI}(\alpha, \beta) = [\bar{X}_n - \delta(n, \alpha), \bar{X}_n + \delta(n, \alpha)] \) is an approximate 100(1-\( \alpha \)) percent confidence interval for \( \mu \) with the desired precision.

In the thesis’ data collection module, this sequential procedure is used to halt the simulation when the precision or relative precision specified by the user is reached.

3.3 Validation of simulation

Validation is the process of bringing to an acceptable level the user’s confidence that any inference about the results derived from the simulation is correct. It is impossible to prove that any simulator is a correct or “true” model of the real system. Fortunately we are seldom concerned with proving the “truth” of a model. Instead, we are mostly concerned with validating the insights we have gained or will gain from the simulation. Thus, it is the operational utility of the model and not the truth of its structure that usually concerns us.

Validation of a simulation is extremely important, because simulators look real and both modelers and users find them easy to believe. Unfortunately, simulators often conceal their assumptions from the casual observer and sometimes even form the modeler. Therefore, if validation and evaluation are not carried out carefully and thoroughly, erroneous results may be accepted with disastrous consequences.

There is not such thing as a “test” for validity. The experimenter must conduct a series of tests throughout the process of developing the model in order to build up this confidence. There are some tests that may be used to validate the model[18]. First, we must ascertain that the model has \textit{prima facie} validity. We must make sure that the results of the model appear to be reasonable by comparing the results of the simulation with actual outputs from the real system.
The second method of validation is to test the assumptions and the third is the testing of input-output transformations. These latter two may entail the use of statistical tests of means and variances, analysis of variance, regression, factor analysis, spectral analysis, auto-correlation, chi-square, and nonparametric tests. Since each of these statistical tests make assumptions about the underlying process, the use of each raises questions of validity. Some statistical tests require fewer assumptions that others, but in general, the power of the test decreases as assumptions are relaxed.

Fisherman and Kiviat [7] divide the evaluation of simulation into three categories:

1. verification – insuring that the model behaves the way an experimenter intends;
2. validation – testing the agreement between the behavior of the model and that of the real system; and
3. problem analysis – the drawing of statistically significant inferences from the data generated by the computer simulation.

In the thesis, the complete graph $K_5$ was used to validate the computer simulated random maximal circuit model discussed in chapter one. The tree structure of the random maximal circuit model on the $K_5$ starting from vertex zero was studied. The number of leftover edges could be 0, 3, or 4, which we denoted at $E(5, 0)$, $E(5, 3)$, $E(5, 4)$ respectively. The exact results are shown in table 3.1.

<table>
<thead>
<tr>
<th>Subgraph Pattern</th>
<th>Num. Of Appearance</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e(5, 0)$</td>
<td>512</td>
<td>82.1%</td>
</tr>
<tr>
<td>$e(5, 3)$</td>
<td>90</td>
<td>14.4%</td>
</tr>
<tr>
<td>$e(5, 4)$</td>
<td>22</td>
<td>3.5%</td>
</tr>
<tr>
<td>Sum</td>
<td>624</td>
<td>100%</td>
</tr>
</tbody>
</table>
These exact probability are compared to the results obtained by computer simulation of 1,000,000 random circuits in $K_5$, shown in table 3.2.

<table>
<thead>
<tr>
<th>Subgraph Pattern</th>
<th>Num. Of Appearance</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e(5, 0)$</td>
<td>814,786</td>
<td>81.5%</td>
</tr>
<tr>
<td>$e(5, 3)$</td>
<td>148,479</td>
<td>14.8%</td>
</tr>
<tr>
<td>$e(5, 4)$</td>
<td>36,735</td>
<td>3.7%</td>
</tr>
<tr>
<td>Sum</td>
<td>1000,000</td>
<td>100%</td>
</tr>
</tbody>
</table>

The relative errors of the simulation results for $e(5, 0)$, $e(5, 3)$ and $e(5, 4)$ are 0.73%, 0.28% and 0.57% respectively, so the simulation results are quite accurate.
CHAPTER 4
CONCLUSIONS AND FUTURE WORK

4.1 Probability analysis

The probability of \( e(n, m) \), for \( m = 0, 3, 4, \ldots \) can be estimated by the simulations run on the undirected complete graph \( K_n \), where \( n \) is odd. The simulation results for \( n \) ranging from \( 5 \leq n \leq 21 \) are displayed in figures 4-1 to 4-6.

Figure 4-1 (a), (b) and (c) list the simulation results of the undirected complete graph \( K_n \), where \( n \) is odd and \( 5 \leq n \leq 21 \). From the figures we can conclude that the random maximal circuits on the undirected complete graph with zero edges left has the highest probability. As the number of vertices in the complete graph increases, the probability decreases; as the number of the leftover edges increases, the probability decreases dramatically.

The probability of zero edges left random maximal circuit has been determined mathematically by McKay and Robinson to the asymptotic to \( e^{3/4} n^{-1/2} \). The comparison of the results of simulation with the mathematical solution was shown in figure 4-4. The regression function was obtained from the random maximal circuit simulation on the complete graph of \( 5 \leq n \leq 21 \). Regression analysis of the simulation results gives:

\[ e(n, 0) = 1.65n^{-0.4304} \] (1)

A comparison of the asymptotic value with the simulation function (1) for higher values of number \( n \) of vertices is shown in figure 4-4. This shows that as \( n \) increases, the probability that a random maximal circuit has zero edges left decreases exponentially, and that the simulation results fit with the mathematical solution very well.
The random maximal circuit results of three edges left and four edges left and are shown in figure 4-5. We can conclude from the simulation results that for the three edges left, the probability decreases exponentially as the number of vertices $n$ increases in the undirected complete graph $K_n$. The regression function for the three edges left is given below:

$$e(n, 3) = 0.313n^{-0.4743}$$ \quad (2)

The r-square is 0.9983. Four edges left can’t be described via exponential function very well. We know from the graph that the probability increases at the beginning when $n$ is small and decreases as $n$ increases.

For the edges left over more than 4, the simulation showed the similar results of four edges left shown in figure 4-6 with $k$ edges left of $5 \leq k \leq 12$, which indicated that the probability of $k$ edges left increases at smaller $n$ and then decreases at bigger $n$.

The converging process as the number of replication increases of $K_5$ was shown in figure 4-2. The converging process was compared with the exact probability of the corresponding zero edges left, three edges left and four edges left. We can conclude that as the number of replications increases, the simulation results converges to the true value. Simulation can’t determine the exact true value for each $e(n, k)$. However, as the number of replications increases, the simulation comes arbitrarily close to the true values, so that accurate estimates can be based on the simulation results.

4.2 Degree analysis of the remainder graph

The complement of a circuit in $K_n$ for odd $n$ is an even graph.

*Proof:*

Start a path at some arbitrary vertex $v_0$, and proceed along a previously unused edge to the next vertex uniformly at random until the random maximal circuits eventually returns
(a) $K_5 - K_9$

(b) $K_{11} - K_{15}$
Figure 4-1  Probability of Random Circuit on $K_n$

(a) $e(5, 0)$

(c) $K_{17} - K_{21}$
Figure 4-2 Converging Process of Random Circuit on $K_5$

(b) $e(5, 3)$

(c) $e(5, 4)$
Figure 4-3 Comparison of Theoretical Analysis with Simulation at Zero Left on $K_n$.

Figure 4-4 Comparison of Mathematical Solution with Simulation Results.
Figure 4-5 $e(n, 3)$ and $e(n, 4)$ with Regression Analysis

Figure 4-6 Simulation Results of $e(n, k)$, $5 \leq k \leq 12$
to \( v_0 \) and a circuit cannot be further extended. Each edge can be used only once. For each vertex, the walk will use one edge to arrive and a different edge to leave. Thus, every vertex in the circuit will have even degree. Because the undirected complete graph is of odd order, each of its vertices has even degree. Thus at every vertex the number of incident edges remaining after the completion of the greedy circuit is the difference of two even numbers, and hence is even.

4.3 Structural analysis of the complementary graph

The structure of the reminder remainder graph is not determined solely by its size. Two graphs may be seemingly different, even though one can be redrawn to look exactly like the other. Thus, these two graphs would actually be structurally “equivalent”. Two graphs \( G_1 \) and \( G_2 \) are said to be isomorphic if there exists a one-to-one correspondence between their vertex sets and a one-to-one correspondence between their edge sets so that the corresponding edges of \( G_1 \) and \( G_2 \) are incident on the corresponding vertices of \( G_1 \) and \( G_2 \).[3, 4, 27] For example, the subgraph of \( K_5 \) complementary to a random maximal circuit could contain four edges. The subgraph pattern for two different circuits could be isomorphic as shown in figure 4-7.

![Isomorphic Graphs of Size 4](image)

Figure 4-7: Isomorphic Graphs of Size 4

The correspondences between the vertex sets and edges are as follows:

**Vertex correspondence:**

\[
\begin{align*}
v_1 & \leftrightarrow v_1' \quad v_2 & \leftrightarrow v_3' \\
v_3 & \leftrightarrow v_2' \quad v_4 & \leftrightarrow v_4'
\end{align*}
\]
Edge correspondence:

\[ e_1 \leftrightarrow e_1' \quad e_2 \leftrightarrow e_2' \]
\[ e_3 \leftrightarrow e_3' \quad e_4 \leftrightarrow e_4' \]

In order to extract structural information about the complement of a random maximal circuit, the degree of each vertex of the remainder graph is recorded and these degrees are grouped into connected components. For example, the graphs in figure 4-7 can be recorded as \([2, 2, 2, 2]\) and \([2, 2, 2, 2]\). Of course isomorphic graphs must exhibit the same degree pattern. Especially on the subgraph of size \(\geq 6\), our expectation is that there will be more possible non-isomorphic graphs in the subgraph after a random maximal circuit. Let’s take size 6 as an example. The isomorphism classes of even graphs of size six excluding isolated vertices are shown in figure 4-8.

Figure 4-8 Non-isomorphic Graphs of Size 6

The degree notation for (a), (b) and (c) respectively are \([2, 2, 2, 2, 2, 2]\), \([4, 2, 2, 2, 2]\) and \([2, 2, 2, 2, 2, 2, 0, 0]\).

In general the degree pattern does not determine the isomorphism class of the graphs. For instance, a graph with pattern \([4, 2, 2, 2, 2, 2]\) might consist of two 4-cycles joined at the vertex of degree 4, or else a 3-cycle with a 5-cycle.

The simulation records the degree pattern of the complementary subgraph when it contains \(\geq 6\) edges. The results were listed in table 4-1 and 4-2 for random maximal circuit on \(K_n\), where \(n = 7, 9, 11,\) or 13. From the tables, we make the following observations:
1. The random maximal circuit traversing tends to leave a connected complementary subgraph.

2. The complements of a random maximal circuit include all of the non-isomorphic even graphs of each possible size.

3. For a fixed number m of edges in the complement of a random circuit, pattern having fewer vertices incident to leftover edges are the more likely to occur.

Table 4-1 Degree Patterns for Complements of Circuits in $K_n$ of Size $m = 6, 7$

<table>
<thead>
<tr>
<th>Degree Pattern</th>
<th>$n = 7$</th>
<th>$n = 9$</th>
<th>$n = 11$</th>
<th>$n = 13$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m = 6$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[4, 2, 2, 2, 2]</td>
<td>2.494%</td>
<td>2.118%</td>
<td>1.727%</td>
<td>1.409%</td>
</tr>
<tr>
<td>[2, 2, 2, 2, 2]</td>
<td>0.340%</td>
<td>1.005%</td>
<td>1.504%</td>
<td>1.811%</td>
</tr>
<tr>
<td>[2, 2, 2, 2, 2, 0, 0]</td>
<td>0.056%</td>
<td>0.164%</td>
<td>0.249%</td>
<td>0.300%</td>
</tr>
<tr>
<td>$m = 7$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[4, 4, 2, 2, 2]</td>
<td>1.678%</td>
<td>0.860%</td>
<td>0.498%</td>
<td>0.316%</td>
</tr>
<tr>
<td>[4, 2, 2, 2, 2]</td>
<td>1.021%</td>
<td>1.864%</td>
<td>1.928%</td>
<td>1.791%</td>
</tr>
<tr>
<td>[2, 2, 2, 2, 2, 2]</td>
<td>0.253%</td>
<td>0.575%</td>
<td>0.843%</td>
<td></td>
</tr>
<tr>
<td>[2, 2, 2, 2, 2, 2, 0, 0]</td>
<td>0.077%</td>
<td>0.167%</td>
<td>0.243%</td>
<td></td>
</tr>
<tr>
<td>$m = 8$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[4, 4, 2, 2, 2]</td>
<td>1.125%</td>
<td>1.210%</td>
<td>0.920%</td>
<td>0.665%</td>
</tr>
<tr>
<td>[4, 2, 2, 2, 2]</td>
<td>0.811%</td>
<td>1.298%</td>
<td>1.493%</td>
<td></td>
</tr>
<tr>
<td>[2, 2, 2, 2, 2, 2]</td>
<td>0.031%</td>
<td>0.165%</td>
<td>0.338%</td>
<td></td>
</tr>
<tr>
<td>[2, 2, 2, 2, 2, 2, 2, 0, 0]</td>
<td>0.013%</td>
<td>0.066%</td>
<td>0.132%</td>
<td></td>
</tr>
</tbody>
</table>
Table 4-2  Degree Patterns for Complements of Circuits in \( K_n \) of Size \( m = 9,10 \)

<table>
<thead>
<tr>
<th>Degree Pattern</th>
<th>( n = 7 )</th>
<th>( n = 9 )</th>
<th>( n = 11 )</th>
<th>( n = 13 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m = 9 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[6, 2, 2, 2, 2, 2, 2]</td>
<td>0.076%</td>
<td>0.071%</td>
<td>0.059%</td>
<td></td>
</tr>
<tr>
<td>[4, 4, 4, 2, 2]</td>
<td>0.743%</td>
<td>0.446%</td>
<td>0.241%</td>
<td>0.132%</td>
</tr>
<tr>
<td>[4, 4, 2, 2, 2, 2]</td>
<td>0.927%</td>
<td>1.054%</td>
<td>0.952%</td>
<td></td>
</tr>
<tr>
<td>[4, 2, 2, 2, 2, 2, 2]</td>
<td>0.162%</td>
<td>0.580%</td>
<td>0.905%</td>
<td></td>
</tr>
<tr>
<td>[2, 2, 2, 2, 2, 2, 2, 2, 2]</td>
<td>0.038%</td>
<td>0.114%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[2, 2, 2, 2, 2, 2, 2, 2, 0, 0]</td>
<td>0.007%</td>
<td>0.026%</td>
<td>0.035%</td>
<td></td>
</tr>
<tr>
<td>[2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0]</td>
<td></td>
<td></td>
<td>0.002%</td>
<td></td>
</tr>
<tr>
<td>[2, 2, 2, 2, 2, 2, 2, 2, 0, 0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( m = 10 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[6, 4, 2, 2, 2, 2, 2]</td>
<td>0.185%</td>
<td>0.126%</td>
<td>0.078%</td>
<td></td>
</tr>
<tr>
<td>[6, 2, 2, 2, 2, 2, 2, 2]</td>
<td>0.031%</td>
<td>0.077%</td>
<td>0.084%</td>
<td></td>
</tr>
<tr>
<td>[4, 4, 4, 4, 4]</td>
<td>0.190%</td>
<td>0.020%</td>
<td>0.004%</td>
<td>0.001%</td>
</tr>
<tr>
<td>[4, 4, 4, 4, 2, 2]</td>
<td>0.567%</td>
<td>0.216%</td>
<td>0.070%</td>
<td>0.034%</td>
</tr>
<tr>
<td>[4, 4, 4, 2, 2, 2]</td>
<td>0.700%</td>
<td>0.568%</td>
<td>0.389%</td>
<td></td>
</tr>
<tr>
<td>[4, 4, 2, 2, 2, 2, 2]</td>
<td>0.288%</td>
<td>0.738%</td>
<td>0.885%</td>
<td></td>
</tr>
<tr>
<td>[4, 2, 2, 2, 2, 2, 2, 2, 2]</td>
<td></td>
<td>0.158%</td>
<td>0.415%</td>
<td></td>
</tr>
<tr>
<td>[2, 2, 2, 2, 2, 2, 2, 2, 2]</td>
<td></td>
<td>0.005%</td>
<td>0.030%</td>
<td></td>
</tr>
<tr>
<td>[4, 4, 2, 2, 2, 2, 2, 2, 0, 0]</td>
<td></td>
<td>0.003%</td>
<td>0.009%</td>
<td>0.009%</td>
</tr>
<tr>
<td>[4, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0]</td>
<td></td>
<td></td>
<td>0.016%</td>
<td>0.038%</td>
</tr>
<tr>
<td>[2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0]</td>
<td></td>
<td></td>
<td>0.000%</td>
<td>0.001%</td>
</tr>
<tr>
<td>[2, 2, 2, 2, 2, 2, 2, 2, 0, 0]</td>
<td></td>
<td></td>
<td>0.002%</td>
<td>0.013%</td>
</tr>
</tbody>
</table>
4.4 Summary and future work

Simulation is a very convenient for the study of complex problems. The simulation framework developed in this thesis could easily be adapted for application to other problems concerning walks and connectivity in graphs or networks. The logic of the simulation is simple compared to the complexities of a mathematical analysis, so the simulation results can be developed rapidly in order to guide or confirm a quantitative analysis.

We conclude that simulation can shed light on the random maximal circuit by estimating parameters for which there is no known mathematical analysis. However, there are several ways in which future work could improve on the current implementation.

1. A way to record the degree pattern might be found so that all isomorphic graphs can be classified, displayed and counted as they occur as complements of random maximal circuits.

2. Distributed computing or parallel computing may ease the problem posed by the huge amount of time and space needed for computing the simulations of larger graphs and/or higher accuracies. Enterprise Java Beans (EJB) [2] will allow interacting modules to be run on multiple machines just as easily as they are now run on a single machine. Furthermore, the simulation can be run as distributed systems on heterogeneous platforms. In addition to EJB, Remote Method Invocation (RMI), and Java IDL (CORBA) technology should be explored.

3. Database support may also be considered for storing all the simulation results for later statistical analysis.
REFERENCES


http://www.javasoft.com/product/ejb/


