Certificates for Properties of Reliability Polynomials of Graphs

by

Rui Chen, BCompSc

Thesis

Submitted by Rui Chen

in partial fulfilment of the Requirements for the Degree of

Bachelor of Computer Science with Honours (1608)

Supervisor: Prof. Graham Farr
Dr. Kerri Morgan

Clayton School of Information Technology
Monash University

November, 2012

Contents

List of Tables

List of Figures

Certificates for Properties of Reliability Polynomials of Graphs

Rui Chen, BCompSc
Monash University, 2012

Supervisor: Prof. Graham Farr
Dr. Kerri Morgan

Abstract

The *reliability polynomial* $\Pi(G, p)$ of a graph $G = (V, E)$ represents the probability that there exists a connected path between any two vertices in G, given a set of independent events that an edge $e \in E$ can randomly fail with probability $1 - p$. A reliability polynomial $\Pi(G, p)$ has a *reliability factorisation* if there exist smaller graphs G_1 and G_2 such that $\Pi(G, p) = \Pi(G_1, p)\Pi(G_2, p)$. Factorisation is a basic property of any polynomial. Reliability factorisation gives a divide-and-conquer approach to compute reliability polynomials that can be reliably factorised.

A *cutvertex* of a graph G is a vertex whose removal increases the number of components of G. It is known that any graph with a cutvertex has a reliability factorisation. This research investigates if there exist reliability factorisations for graphs without cutvertices. We find 581 such reliability factorisations by an exhaustive search over reliability polynomials of all connected graphs with at most 13 edges. We also show that an infinite graph family has a reliability factorisation.

A *certificate* is a sequence of steps based on identities. This research uses certificates to explain reliability factorisations. We give certificates for reliability factorisations of all connected graphs with at most 8 edges. We also show a certificate for a reliability factorisation of an infinite family of graphs. Considering the complexity of computing reliability polynomials, the lengths of these certificates are quite short. We discuss how the upper bound on the lengths of certificates of reliability factorisation is related to the complexity of the decision problem whether a reliability polynomial has a reliability factorisation.

Certificates for Properties of Reliability Polynomials of Graphs

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Rui Chen
November 9, 2012

Acknowledgments

I would like to thank Prof. Graham Farr for his offer and guidance on this research.

I would like to thank Dr. Kerri Morgan for her supervision over the past eight months.

I would like to thank my parents, Xiaobin Chen and Hongling Guo, for their care, support and sponsorship.

Rui Chen

Monash University
November 2012

Chapter 1

Introduction

A graph is a set of vertices and a set of edges that join pairs of vertices. Graph theory is widely used in the study of mass transportation, computational biochemistry, computer networks and social networks as well as other fields of mathematics (Evans et al., 2006; Pirzada and Dharwadker, 2007; Newman et al., 2002; Bertram and Horá, 1999). Graphs are an abstraction of the essential elements in network structures. A network can be represented as a graph by modelling each node such as a terminal, a station or a computer as a vertex and modelling each link between nodes as an edge.

A network is *reliable* if every pair of nodes is connected by a path. The analysis of the reliability of networks such as computer architecture networks and data communication networks has become an increasingly significant field of study (Ball et al., 1995; Chang and Shrock, 2003). Reliability is one of the most important considerations in network design as failures in networks may cause serious damage (Konak et al., 2002).

The reliability of a network can be determined by analysing the reliability of the underlying graph. The *reliability polynomial* (Brown and Colbourn, 1992; Colbourn, 1997; Chang and Shrock, 2003) was introduced to represent algebraically the reliability of graphs. It is the probability of a graph being reliable assuming a set of independent events that each edge can randomly fail with a certain probability. Rather than considering the possibility of failure of vertices, the reliability polynomial depends on the reliability of edges (Chang and Shrock, 2003).

This thesis focuses on an algebraic property of the reliability polynomial, *reliability factorisation*, which refers to the case that a reliability polynomial can be expressed as a product of reliability polynomials of lower degrees. It is well known that any graph with a cutvertex has a reliability factorisation (Wanger, 2000). One objective of this research was to investigate if there exist reliability factorisations for some graphs without cutvertices.

In this research, we compute the reliability polynomials of all connected graphs with at most 13 edges. *Reliability equivalence* refers to the fact that there exist two graphs that have the same reliability polynomial. We identify 581 reliability factorisations of graphs without cutvertices by an exhaustive search over all cases of reliability equivalence. We also give a reliability factorisation for an infinite family of graphs.

The concept of *certificate* (Morgan and Farr, 2009b) is extended by this research to explain cases of reliability factorisation and reliability equivalence. We give certificates for all reliability factorisations of connected graphs with at most 8 edges. We also give a certificate for a reliability factorisation of an infinite family of graphs. Compared with the complexity of computing reliability polynomials, the lengths of these certificates are remarkably short. Motivated by the short lengths of these certificates, we discuss the relationship between the upper bound on the lengths of certificates and the complexity of the decision problem whether a reliability polynomial has a reliability factorisation.

The rest part of this chapter gives definitions related to the reliability polynomial and main contributions of this research. Chapter 2 gives an overview of the literature context of the reliability polynomial. Chapter 3 describes the methods used by this research to generate graphs, compute reliability polynomials and search for reliability factorisations. Chapter 4 lists the results including reliability factorisations of graphs without cutvertices and a reliability factorisation of an infinite family of graphs. Chapter 5 shows certificates for reliability factorisations of graphs without cutvertices and a certificate of reliability factorisation for an infinite graph family. Chapter 6 discusses the relationship between the upper bound on the lengths of certificates and the complexity of the decision problem whether a reliability polynomial has a reliability factorisation. Chapter 7 summarises the main works done in this research and makes some suggestions for further research.

1.1 Definitions

1.1.1 Graph Basics

This section gives some basic knowledge about graphs. It includes the definition of a graph, special elements of graphs such as cutvertices, bridges and multiple edges, different types of graphs such as paths and cycles as well as some operations on graphs such as contraction, deletion and vertex-gluing.

A *graph* G is a pair of sets $G = (V, E)$ where V (or $V(G)$) is the set of vertices and E (or $E(G)$) $\subseteq V^{(2)}$ is the set of edges (Diestel, 2000) where $V^{(2)}$ is the set of unordered pairs of elements of the set V. The *order* of G is $|V|$, denoted by n. The *size* of G is $|E|$, denoted by m. For example, Figure 1.1 displays a graph H on vertex set $V = \{1, 2, 3, 4, 5, 6, 7\}$ with edge set $E = \{\{1, 2\}, \{1, 2\}, \{2, 3\}, \{3, 3\}, \{3, 4\}, \{4, 7\}, \{1, 7\}, \{4, 6\}, \{5, 6\}, \{5, 6\}, \{5, 5\}\}$. The order n of H is 7. The size m of H is 11.

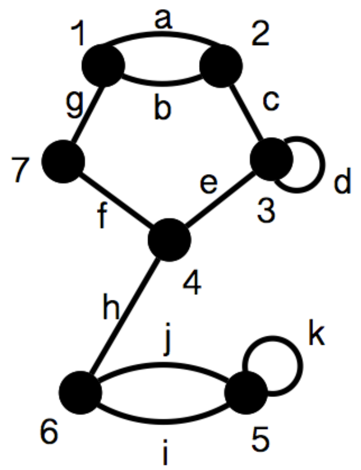
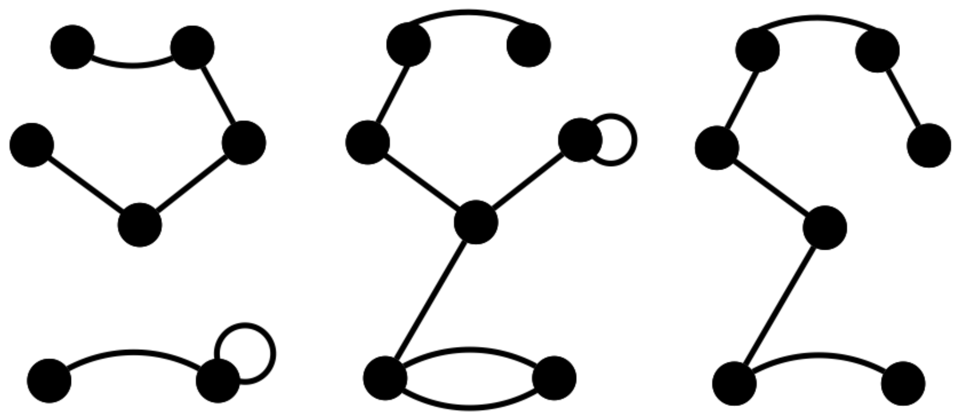


Figure 1.1: A graph H

If both of the vertices incident to an edge $e \in E$ are the same, then e is a *loop*. If both of the vertices incident to an edge e_1 are the same as both of those incident to another edge e_2, then e_1 and e_2 are *multiple* or *parallel* edges. If a graph G contains multiple edges, then G is a *multigraph*. In Figure 1.1, the edges $d = \{3, 3\}$ and $k = \{5, 5\}$ are both loops. The edge pairs $(a, b) = \{1, 2\}$ and $(i, j) = \{5, 6\}$ are multiple edges. The graph H in Figure 1.1 is a multigraph. Multiple edges are allowed on graphs in this research.

A graph $G' = (V', E')$ is a *subgraph* of G, denoted by $G' \leq G$, if $V' \subseteq V$ and

$E' \subseteq E$ (Diestel, 2000). If $V' = V$ then G' is a *spanning subgraph* (Diestel, 2000). If a spanning subgraph G' is a tree, then G' is a *spanning tree* of G. A graph G is *connected* if G contains at least one spanning tree. A graph G is *disconnected* if G has no spanning tree. The scope of this research is limited to connected graphs. Figure 1.2 illustrates three spanning subgraphs of the graph H in Figure 1.1. The graph H_1 is disconnected. The graphs H_2 and H_3 are connected spanning subgraphs of the graph H in Figure 1.1. The graph H_3 is also a spanning tree of H.



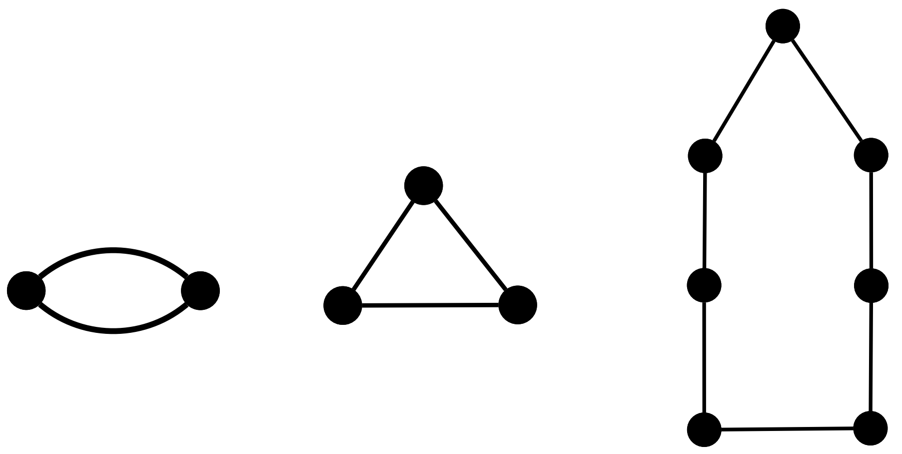
(a) The graph H_1 (b) The graph H_2 (c) The graph H_3

Figure 1.2: Spanning subgraphs of the graph H in Figure 1.1

A *complete graph* K_n is a graph of order n where all pairs of vertices are adjacent (Diestel, 2000). A *null graph* N_n is an edgeless graph of order n. A *path* is a graph $P_n = (V, E)$ of order n where $V = \{v_0, v_1, ..., v_{n-1}\}$ and $E = \{v_0v_1, v_1v_2, ..., v_{n-2}v_{n-1}\}$ (Diestel, 2000). A *θ-graph*, denoted by $\theta_{x,y,z}$, is a graph that can be obtained from three disjoint paths $p_1 = (a_0, a_1, a_2, ..., a_x)$, $p_2 = (b_0, b_1, b_2, ..., b_y)$ and $p_3 = (c_0, c_1, c_2, ..., c_z)$ for $x, y, z \geq 1$ by identifying vertices a_0, b_0 and c_0 and identifying vertices a_x, b_y and c_z (Morgan, 2010). A *cycle* C_n is a graph $C \equiv P_n + v_{n-1}v_0$ of order n where $P = v_0v_1...v_{n-1}$ is a path. Figure 1.3 gives three examples of cycles C_2, C_3 and C_7.

A *component* of a graph G is a maximal connected subgraph of G (Diestel, 2000). By definition, a connected graph has a single component. A vertex $v \in V$ is a *cutvertex* if the removal of v increases the number of components of G. A graph G is *separable* if G has at least one cutvertex. A graph G is a *non-separable* if G has no cutvertex. The graph H in Figure 1.1 is separable with two cutvertices 4 and 6. All cycles in Figure 1.3 are non-separable.

A *block* of a graph G is a maximal connected non-separable subgraph (Diestel, 2000).



(a) The graph C_2 (b) The graph C_3 (c) The graph C_7

Figure 1.3: Cycles

An edge $e \in E$ is a *bridge* if the removal of e increases the number of components of G. A C_2-*bridge* of a graph G is a pair of multiple edges whose removal increases the number of components of G. The graph H in Figure 1.1 has a bridge $h = \{4, 6\}$ and three blocks. The graph J in Figure 1.4 has a C_2-bridge $(e, f) = \{3, 4\}$. We say that J is divided by a C_2-bridge into two graphs each of which is isomorphic to C_3.

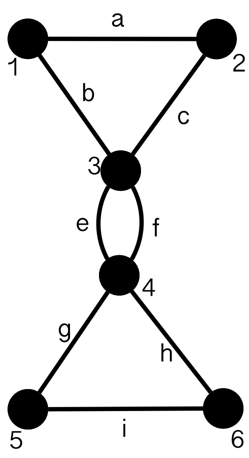
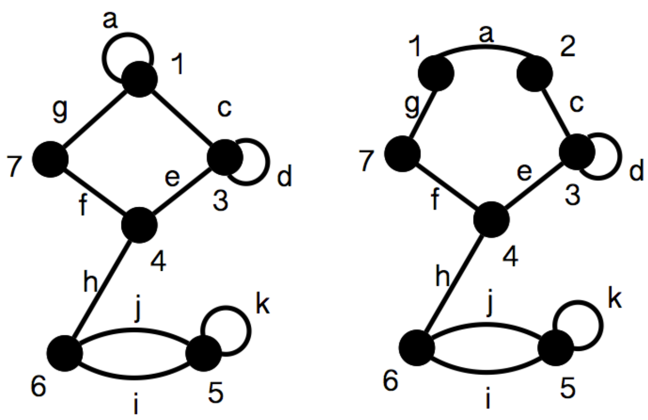


Figure 1.4: A graph J with a C_2-bridge $(e, f) = \{3, 4\}$

Given a set $E^{'} \subseteq E$, the *complement* of $E^{'}$ from E, denoted by $E - E^{'}$, is the set of edges that belong to E and do not belong to $E^{'}$. Given an edge e of a graph

G, the *contraction* of e on G is the graph G/e obtained by identifying the vertices with which e is incident and removing e (Diestel, 2000). The *deletion* of e from G is the graph $G - e$ obtained by removing e. A graph G is a *vertex-gluing* of graphs G_1 and G_2, denoted by $G = G_1 \cdot G_2$, if G can be obtained by identifying a vertex of G_1 with a vertex of G_2. Figure 1.5a shows the contraction of the edge b on the graph H in Figure 1.1. Figure 1.5b shows the deletion of b from H. The graph J in Figure 1.4 is a vertex-gluing of a cycle C_2 and two cycles C_3.



(a) Graph H/b (b) Graph $H - b$

Figure 1.5: Contraction and deletion of the edge b in the graph H

1.1.2 Concepts of the Reliability Polynomial

For a graph $G = (V, E)$, an edge $e \in E$ *fails* at a particular time t if e is absent at t. An edge $e \in E$ *operates* at a time t if it does not fail at t. This research assumes that vertices never fail and that every edge e can either operate or fail at a certain time. At any time, each edge fails randomly and its failure is independent of the other edges. A *state* $S \subseteq E$ at a time t refers to the set of all operating edges of the graph G at t. In the rest of this document, the time t is omitted.

The *k-terminal reliability* of a graph G (Ball et al., 1995; Colbourn, 1997; Chang and Satyanarayana, 1983; Page and Perry, 1994) is the probability that any two vertices in the set $K \subseteq V$ are connected by a path of edges $e \in S$, that is, between any two vertices in K, there exists at least one path constructed by edges in S. If $K = V$, it is called the *all-terminal reliability* of G (Colbourn, 1997; Chang and Shrock, 2003; Page and Perry, 1994) which is the probability that there exists at least one spanning tree of G constructed by edges $e \in S$. The reliability polynomial in this research is referred as the all-terminal reliability of graphs.

Assuming that every edge in a graph G operates with probability $p \in [0,1]$ and fails with probability $1-p$, the reliability of G can be expressed as a polynomial called the *reliability polynomial* (Brown and Colbourn, 1992; Colbourn, 1997; Chang and Shrock, 2003), denoted by $\Pi(G,p)$. The reliability polynomial represents the probability that there exists a state $S \subseteq E$ such that S contains at least one spanning tree of G. A graph G is *reliably equivalent* to another graph $G^{'}$, denoted by $G \sim G^{'}$, if $\Pi(G,p) = \Pi(G^{'},p)$. A *class of reliably equivalent* graphs is a set of graphs whose reliability polynomials are the same. For example, the graphs G and $G^{'}$ in Figure 1.6 have the same reliability polynomial $p^4(2p-3)^2$. We say that G and $G^{'}$ belong to a class of reliability equivalent graphs.

(a) A graph G (b) A graph $G^{'}$

Figure 1.6: Reliably equivalent graphs

1.1.3 Reliability Factorisation

We say that a graph G or a reliability polynomial $\Pi(G,p)$ has a *reliability factorisation* if there exist graphs G_1 and G_2 such that

$$\Pi(G,p) = \Pi(G_1,p)\Pi(G_2,p).$$

A reliability polynomial that have a reliability factorisation can be computed using a divide-and-conquer approach. Reliability factorisation shows a relationship between reliability polynomials of some graphs. It is known that any separable graph has a reliability factorisation. This research investigates reliability factorisations of non-separable graphs. We find 24,886 reliability factorisations of all connected graphs of size at most 13. In those reliability factorisations, we identify 581 cases of non-separable graphs (see Section 4.1). We show a reliability factorisation of an infinite family of θ-graphs in Section 4.2.2.

1.1.4 Certificate

A *certificate* is a sequence of steps CS_1, CS_2, ..., CS_i, ..., CS_k based on identities (Morgan and Farr, 2009b). Each of these steps is called a *certificate step*. A *certificate of reliability equivalence* is a sequence of steps based on algebraic operations and properties of the reliability polynomial to prove that two reliability polynomials are equivalent. Similarly, a *certificate of reliability factorisation* is a sequence of such steps to explain a reliability factorisation. We find 54,577 classes of reliably equivalent graphs from all connected graphs of size at most 13 (see Section 4.1).

In order to construct certificates, this research generates twelve types of certificate steps (listed in Section 5.1) based on properties of the reliability polynomial. We give certificates for all reliability factorisations of connected graphs of size at most 8 in Section 5.2. We also show a certificate of reliability factorisation for an infinite family of θ-graphs by mathematical induction in Section 5.2.2. A certificate of reliability factorisation shows a sequence of expressions E_0, E_1, ..., E_i, ..., E_k where $E_0 = \Pi(G, p)$, $E_k = \Pi(G_1, p)\Pi(G_2, p)$ and each expression E_i is transformed to the next expression E_{i+1} based on a certificate step CS_{i+1}. The *length* of a certificate of reliability factorisation is the number of steps k.

1.2 Main Contributions

This research mainly involves the following works:

- Initiating the study of reliability factorisation of graphs
- Computing all reliability polynomials of connected graphs of size at most 13
- Finding all reliability factorisations of connected graphs of size at most 13
- Demonstrating the existence of reliability factorisations of non-separable graphs
- Identifying all reliability factorisations for non-separable graphs of size at most 13
- Extending the concept of certificate to explain cases of reliability factorisation and reliability equivalence
- Generating twelve certificate steps used in construction of certificates
- Generating certificates of reliability factorisation for all connected graphs of size at most 8

- Finding a reliability factorisation of an infinite family of θ-graphs
- Generating a certificate of reliability factorisation for an infinite family of θ-graphs.

Chapter 2

Research Context

This chapter shows the literature context of the reliability polynomial. Section 2.1 discusses several combinatorial interpretations of the reliability polynomial. All those interpretations are based on the property that the reliability polynomial can be expressed as a sum over all connected spanning subgraphs. Section 2.2 discusses the roots of the reliability polynomial. The analysis of implications from the real roots was used in the search algorithm for reliability factorisations which will be described in Section 3.3. Section 2.3 states some properties of the reliability polynomial, including the deletion-contraction relation used by this research to compute reliability polynomials. Section 2.4 introduces the Tutte polynomial. The reliability polynomial is a partial evaluation of the Tutte polynomial. Section 2.5 introduces the chromatic polynomial which is another partial evaluation of the Tutte polynomial. The research conducted by Morgan and Farr (2009b) on certificates of chromatic factorisation is analysed in terms of computing methods and certificates of chromatic factorisation, which give some motivation and implication for this research.

2.1 Combinatorial Analysis

This section gives some combinatorial interpretations of the reliability polynomial. These interpretations are based on the fact that the reliability polynomial of a graph G can be expressed as a sum over subsets of edges of G. Every interpretation gives an expression with different coefficients.

2.1.1 Basic Form

The reliability polynomial $\Pi(G,p)$ can be written as a sum over all connected spanning subgraphs (Chang and Shrock, 2003). Given a connected spanning subgraph $G^{'} = (V, E^{'})$, the probability of the existence of such $G^{'}$ is $p^{|E^{'}|}(1-p)^{|E|-|E^{'}|}$. The reliability polynomial $\Pi(G,p)$ is a summation over such probabilities, namely, $\Pi(G,p)$

can be expressed as

$$\Pi(G,p) = \sum_{G'=(V,E')} p^{|E'|}(1-p)^{|E|-|E'|} \tag{2.1}$$

where $E' \subseteq E$ and G' is connected (Colbourn, 1997; Chang and Shrock, 2003; Graver and Sobel, 2005; Page and Perry, 1994; Welsh, 1993). Equation (2.1) shows that the coefficients of the reliability polynomial $\Pi(G,p)$ counts the connected spanning subgraphs of G.

2.1.2 Derived Forms

The reliability polynomial $\Pi(G,p)$ of a graph $G = (V,E)$ can be expressed in the following forms based on Equation (2.1) given that m is the size of G:

(a) *N-form* (Ball et al., 1995; Colbourn, 1997; Moore and Shannon, 1956)

$$\Pi(G,p) = \sum_{i=0}^{m} N_i p^i (1-p)^{m-i} \tag{2.2}$$

where N_i is the number of connected spanning subgraphs of size i. The reliability polynomial $\Pi(G,p)$ in this form sums over all connected spanning subgraphs of size from 0 to m.

(b) *F-form* (Brown and Colbourn, 1992; Ball et al., 1995; Colbourn, 1997; Slyke and Frank, 1971)

$$\Pi(G,p) = \sum_{i=0}^{m} F_i (1-p)^i p^{m-i} \tag{2.3}$$

where F_i is the number of connected spanning subgraphs of size $m-i$. The reliability polynomial $\Pi(G,p)$ in this form sums over all connected spanning subgraphs of size from m to 0. Thus, $F_i = N_{m-i}$.

(c) *M-form* (Colbourn, 1997)

$$\Pi(G,p) = 1 - \sum_{i=0}^{m} M_i p^i (1-p)^{m-i} \tag{2.4}$$

where M_i is the number of disconnected spanning subgraphs of size i. In contrast to the N-from and the F-form, the reliability polynomial $\Pi(G,p)$ in this form gives a summation over all disconnected spanning subgraphs of size from 0 to m. Thus, $N_i + M_i = \binom{m}{i}$.

(d) *C-form* (Ball et al., 1995; Colbourn, 1997; Moore and Shannon, 1956)

$$\Pi(G,p) = 1 - \sum_{i=0}^{m} C_i(1-p)^i p^{m-i} \tag{2.5}$$

where C_i is the number of disconnected spanning subgraphs of size $m-i$. The reliability polynomial $\Pi(G,p)$ in this form gives a summation over all disconnected spanning subgraphs from m to 0. Thus, $F_i + C_i = \binom{m}{i}$.

The above forms are useful for calculation of the reliability polynomial $\Pi(G,p)$ in different ways. Each of these forms gives a different combinatorial interpretation of $\Pi(G,p)$.

2.2 Roots of the Reliability Polynomial

This section describes some aspects of roots of the reliability polynomial. The analysis of the real roots is used by this research to search for reliability factorisations. This reduces the search complexity by omitting impossible cases of reliability factorisation in the search algorithm.

2.2.1 Implications from Real Roots for Finding Reliability Factorisations

Brown and Colbourn (1992) showed that all real roots of the reliability polynomial lie in the unit disc with centre 1, more precisely, in the interval $0 \cup (1,2]$. The reliability polynomial $\Pi(G,p)$ has zero as a root of multiplicity $n-1$ where n is the order of G (Brown and Colbourn, 1992). Thus, $\Pi(G,p)$ can be expressed as

$$\Pi(G,p) = p^{n-1} f(p) \tag{2.6}$$

where $f(p)$ is a polynomial in p. If $\Pi(G,p)$ has a reliability factorisation

$$\Pi(G,p) = \Pi(G_1,p)\Pi(G_2,p),$$

then

$$\begin{aligned} p^{n-1} f(p) &= p^{n_1-1} f_1(p) p^{n_2-1} f_2(p) \\ &= p^{n_1+n_2-2} f_1(p) f_2(p). \end{aligned}$$

The factor p^{n-1} of $\Pi(G,p)$ is unique, thus,

$$n = n_1 + n_2 - 1. \tag{2.7}$$

In Equation (2.7), $n_1 \geq 2$ and $n_2 \geq 2$. Thus, $n \geq 3$. It gives a relation on the possible orders of reliably factorised graphs G_1 and G_2. Both orders of graphs G_1 and G_2 are less than the order of G. This relation is used to reduce the search complexity for reliability factorisations.

2.2.2 Complex Roots

Brown and Colbourn (1992) gave a conjecture that the complex roots of the reliability polynomial of a connected graph lie in $\{z : |z-1| \leq 1\}$. However, Royle and Sokal (2004) divided the Brown-Colbourn conjecture into two parts: a univariate conjecture and a multivariate conjecture and proved that both univariate and multivariate conjectures are false.

A graph is *planar* if it can be drawn on a plane in a way such that no edges intersect (Diestel, 2000). Royle and Sokal (2004) gave a counterexample of the graph K_4 for the multivariate Brown-Colbourn conjecture and a counterexample of a planar graph obtained from K_4 by adding parallel edges for the univariate conjecture (Royle and Sokal, 2004). A loopless graph is *series-parallel* if it can be obtained from a forest by a finite sequence of replacing an edge by two edges in series or two edges in parallel (Royle and Sokal, 2004). Wagner (2000) proved that the scope of the univariate Brown-Colbourn conjecture was limited to all series-parallel graphs. Royle and Sokal (2004) showed that the multivariate Brown-Colbourn conjecture held for all series-parallel graphs as well.

2.3 Properties of Reliablity Polynomial

This section gives some properties of the reliability polynomial $\Pi(G,p)$. These properties are used to compute reliability polynomials and derive basic certificates steps by this research.

If a graph G is disconnected, then G has no spanning tree. Thus,

$$\Pi(G,p) = 0. \tag{2.8}$$

If a graph G has a cutvertex v, then

$$\Pi(G, p) = \Pi(G_1, p)\Pi(G_2, p) \tag{2.9}$$

where $G_1 \cdot G_2 = G$ and $G_1 \cap G_2 = \{v\}$ (Wanger, 2000). It follows from Equation (2.9) that the reliability polynomial $\Pi(G, p)$ of a separable graph G is the product of the reliability polynomials of its blocks (Brown and Colbourn, 1986). Prior to this research, it was only known that separable graphs have reliability factorisations. This research found 24,305 cases of reliability factorisation for separable graphs (see Section 4.1).

For any edge e of a graph G,

(a) if e is a loop then its failure does not affect $\Pi(G, p)$ (Chang and Shrock, 2003), namely

$$\Pi(G, p) = \Pi(G/e, p); \tag{2.10}$$

(b) If e is a bridge then G is disconnected when e fails (Chang and Shrock, 2003), namely

$$\Pi(G, p) = p\Pi(G/e, p); \tag{2.11}$$

(c) If e is neither a loop nor a bridge, $\Pi(G, p)$ comes from the sum of two mutually exclusive possibilities whether e operates or fails (Ball et al., 1995; Colbourn, 1997; Chang and Shrock, 2003; Moore and Shannon, 1956; Welsh, 1993), namely

$$\Pi(G, p) = p\Pi(G/e, p) + (1 - p)\Pi(G - e, p). \tag{2.12}$$

Equation (2.12) is a *deletion-contraction* relation of the reliability polynomial. It is also called the *factoring theorem* (Chang and Satyanarayana, 1983). Equation (2.10) and Equation (2.11) are two special cases of Equation (2.12). One reason to state them separately is providing an inductive explanation for the reliability polynomial which is similar to the inductive definition of the Tutte polynomial (see Equation (2.13)).

Equations (2.10), (2.11) and (2.12) describe a recurrence relation for the reliability polynomial. Section 3.2 will describe the algorithms to compute reliability polynomials based on this relation. All the above properties are used to generate certificate steps, which will be given in Section 5.1.

2.4 Relation with the Tutte Polynomial

The study of the Tutte polynomial is an important subject in graph theory. The Tutte polynomial is a generalisation of both the reliability polynomial and the chromatic polynomial (see Section 2.5). This section introduces the inductive definition of the Tutte polynomial and discusses the relation between the Tutte polynomial and the reliability polynomial.

2.4.1 Definition of the Tutte Polynomial

The *Tutte polynomial* $T(G, x, y)$ of a graph $G = (V, E)$ is a two-variable polynomial in x, y that can be inductively defined (Welsh, 1993) as follows:

If G has no edges then $T(G, x, y) = 1$; otherwise for any $e \in E$,

$$T(G,x,y) = \begin{cases} yT(G-e,x,y) & \text{if } e \text{ is a loop,} \\ xT(G/e,x,y) & \text{if } e \text{ is a bridge,} \\ T(G-e,x,y) + T(G/e,x,y) & \text{if } e \text{ is neither a loop nor bridge.} \end{cases} \tag{2.13}$$

Both of the Tutte polynomial and the reliability polynomial have the property called *deletion-contraction* relation.

2.4.2 The Whitney Rank Generating Expression

The Tutte polynomial is closely related to the Whitney rank generating function. If $G^{'} = (V, E^{'})$ is a spanning subgraph of G, then the *rank* of $E^{'}$, denoted by $r(E^{'})$, is expressed (Welsh, 1993) as

$$r(E^{'}) = |V| - k(G^{'}) \tag{2.14}$$

where $k(G^{'})$ is the number of components of $G^{'}$. The Tutte polynomial $T(G, x, y)$ can be expressed (Chang and Shrock, 2003; Welsh, 1993) in the form

$$T(G,x,y) = \sum_{G^{'} \subseteq G} (x-1)^{r(E)-r(E^{'})}(y-1)^{|E^{'}|-r(E^{'})}. \tag{2.15}$$

The *Whitney rank generating function* $R(G, u, v)$ is a polynomial in the variables u, v and is defined (Welsh, 1993) by

$$R(G,u,v) = \sum_{G^{'} \subseteq G} u^{r(E)-r(E^{'})} v^{|E^{'}|-r(E^{'})}. \tag{2.16}$$

By substituting the two variables u and v in Equation (2.16) with $x-1$ and $y-1$ respectively, Equation (2.15) can be derived, that is, $T(G,x,y) = R(G,x-1,y-1)$.

2.4.3 Relation between the Reliability Polynomial and the Tutte Polynomial

The Recipe Theorem defined in (Welsh, 1993, p.48) provides a way to calculate a graph invariant $f(G)$. It states that if $f(G)$ satisfies the following properties:

(1) $f(G) = af(G-e) + bf(G/e)$ for $e \in E$ not a loop or bridge,

(2) $f(G_1 \cdot G_2) = f(G_1)f(G_2)$

then $f(G)$ is given by

$$f(G) = a^{|E|-r(E)} b^{r(E)} T\left(G, \frac{x_0}{b}, \frac{y_0}{a}\right) \tag{2.17}$$

where x_0 and y_0 are the values that $f(G)$ takes when edge e is a bridge and a loop respectively. The invariant $f(G)$ is called *Tutte-Gröthendieck(TG)-invariant.* Based on Equation (2.12) and Equation (2.9) in Section 2.3, the reliability polynomial $\Pi(G,p)$ is a TG-variant with $a = 1-p$ and $b = p$. Thus, $\Pi(G,p)$ can be expressed (Chang and Shrock, 2003; Welsh, 1993) as a partial evaluation of the Tutte polynomial $T(G,x,y)$ namely

$$\Pi(G,p) = p^{|V|-1}(1-p)^{|E|-|V|+1} T\left(G, 1, \frac{1}{1-p}\right). \tag{2.18}$$

The reliability polynomial is a specialisation of the Tutte polynomial. It is given by a partial evaluation of the Tutte polynomial.

2.5 Methodologies in Chromatic Polynomial Research

Both the reliability polynomial and the chromatic polynomial are partial evaluations of the Tutte polynomial (Welsh, 1993). Morgan and Farr (2009b) investigated the factorisation of chromatic polynomials of graphs and introduced certificates to explain chromatic factorisations.

2.5.1 Definitions

For a positive integer λ, a λ-*colouring* of a graph $G = (V, E)$ is a mapping $\phi : V \to \{1, 2, ..., \lambda\}$ such that $\phi(u) \neq \phi(v)$ for all $uv \in E$ (Diestel, 2000; Welsh, 1993). The *chromatic number* of G, denoted by $\chi(G)$ (Diestel, 2000), is the smallest value λ that can be used in a λ-colouring. The *chromatic polynomial* $P(G, \lambda)$ is defined as the number of λ-colourings of G (Welsh, 1993). An r-*clique* is a subgraph $G^{'} = (V^{'}, E^{'})$ such that $|V^{'}| = r$ and each vertex $v^{'} \in V^{'}$ is adjacent to the other vertices in $V^{'}$.

A graph G is a *clique-gluing* of graphs H_1 and H_2 if G can be obtained by identifying an r-clique in H_1 with an r-clique in H_2 (Morgan, 2010). A graph G is *clique-separable* if G is a clique-gluing of two graphs (Morgan, 2010). If G is not clique-separable, then G is a *non-clique-separable* graph (Morgan and Farr, 2009b). Graphs G and $G^{'}$ are *chromatically equivalent* if $P(G, \lambda) = P(G^{'}, \lambda)$. A graph G is *quasi-clique-separable* if $P(G, \lambda) = P(G^{'}, \lambda)$ where $G^{'}$ is clique-separable (Morgan and Farr, 2009b). A graph is *strongly non-clique-separable* if it is not quasi-clique-separable (Morgan and Farr, 2009b).

2.5.2 Computation Method

Morgan and Farr (Morgan and Farr, 2009b) calculated the chromatic polynomials of all non-isomorphic connected graphs of order at most 10. The calculation was based on an algorithm that recursively applied the deletion-contraction relation (Read, 1968, 1987; Read and Tutte, 1988; Tutte, 1972)

$$P(G, \lambda) = P(G - e, \lambda) - P(G/e, \lambda) \tag{2.19}$$

with the base case of computing the chromatic polynomial of null graphs (Morgan and Farr, 2009b). Then PARI, a C library for fast computation (*PARI/GP, version* `2.3.0`, 2006), was used to factorise these chromatic polynomials (Morgan and Farr, 2009b). A search of all the chromatic polynomials of degree at most 10 was conducted to identify which chromatic polynomials have chromatic factorisations (Morgan and Farr, 2009b).

2.5.3 Chromatic Factorisation

A chromatic polynomial $P(G, \lambda)$ of a graph G has a *chromatic factorisation* if there exist graphs H_1 and H_2 such that

$$P(G, \lambda) = \frac{P(H_1, \lambda)P(H_2, \lambda)}{P(K_r, \lambda)} \tag{2.20}$$

where H_1, H_2 are graphs of lower order than G and $r \leq \min\{\chi(H_1), \chi(H_2)\}$ (Morgan and Farr, 2009b). Equation (2.20) shows that a clique-separable graph has a chromatic factorisation. Based on this, the motivation was to find chromatic factorisations of non-clique-separable graphs. In this research, Morgan and Farr (2009b) found 512 chromatic polynomials of strongly non-clique-separable graphs of order at most 10 which have chromatic factorisations.

2.5.4 Certificate of the Chromatic Factorisation

Morgan and Farr (Morgan and Farr, 2009b) introduced the concept of a *certificate of chromatic factorisation*, a series of steps $P_1, P_2, ..., Pi, ..., P_{n-1}, P_n$ using the properties of the chromatic polynomial and some basic algebraic operations in order to explain chromatic factorisations for some graphs. These steps are used to construct certificates of chromatic equivalence when two graphs have the same chromatic polynomial (Morgan and Farr, 2009b).

Certificates, which share some common steps of transformations are grouped into a template, defined as a *schema* (Morgan and Farr, 2009b). Equation (2.20) can be either a single certificate of factorisation for a clique-separable graph or a schema. The first expression E_1 is $P(G, \lambda)$. The second expression E_2 is $P(H_1, \lambda)P(H_2, \lambda)/P(K_r, \lambda)$ obtained from E_1 by applying the step given by Equation (2.20). The graphs that satisfy this certificate have a common structural property called *clique-separability* (Morgan and Farr, 2009b). Non-clique-separable graphs which have chromatic factorisations that satisfy a schema have a common structure as well (Morgan and Farr, 2009b). Here the non-clique-separable graphs are not limited to small graphs. Morgan and Farr (2009a) constructed an infinite family of strongly non-clique-separable graphs whose chromatic polynomials have chromatic factorisations.

A certificate is a sequence of steps that transform a factorisation expression to another equivalent expression. The length of a certificate of chromatic factorisation is related to the complexity of determining a chromatic factorisation. Morgan and Farr (2009b) gave an upper bound $n^2 2^{n^2/2}$ on the lengths of certificates of chromatic factorisation. In comparison to this result, some classes of certificates in their research were much shorter.

2.5.5 Implications for Reliability Polynomial Research

Chapter 3 describes a similar approach applied in this research to investigate the reliability factorisation. Firstly, the reliability polynomials of all connected graphs of size at most 13 were computed. Then an exhaustive search over all reliability polynomials was used to find reliability factorisations (see Section 3.3). Similarly, the motivation of the research on reliability polynomials was to identify cases of reliability factorisation for single non-separable graphs as well as an infinite family of non-separable graphs.

The research on reliability polynomials also generated certificate steps (see Section 5.1) to explain reliability factorisations. Section 5.2.1 gives certificates of reliability factorisations for all non-separable graphs of size at most 8. Section 5.2.2 gives a certificate for a reliability factorisation of an infinite family of graphs. The lengths of certificates of reliability factorisation are analysed in Section 5.2.3.

Chapter 3

Computational Methods

This chapter describes the computational methods used in this research. Section 3.1 introduces the way to generate all connected graphs including both simple graphs and multigraphs of size at most 13. Section 3.2 gives the methods to compute reliability polynomials for these graphs. Section 3.3 describes the search over all reliability polynomials to find reliability factorisations.

3.1 Method to Generate Graphs

This research imports a suite of programs **gtools** included in the **nauty** package (McKay, 2009) to generate all connected simple graphs and multigraphs of size at most 13. There are $1,821,234$ such graphs. Each of these graphs has a canonical label used to uniquely identify non-isomorphic graphs. For example, Figure 3.1 shows the graph with the canonical label 84.

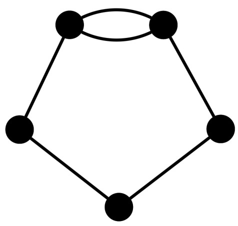


Figure 3.1: Graph 84

All graphs were generated in two main steps: The first step used the program **geng** in **gtools** to generate all connected simple graphs of order at most 14; the second step implemented the program **multig** in **gtools** to generate all connected graphs including both simple graphs and multigraphs of size at most 13 based on the simple graphs generated in the first step. Given a certain order n as input, the program **geng** can generate all connected simple graphs of order at most n (McKay, 2009). Given a certain size m as input, the program **multig** reads a set of simple graphs and replace every edge with multiple edges in all possible ways as long as the size of the generated multigraphs is no greater than m (McKay, 2009). The maximum order of connected graphs of size at most 13 is 14. Thus, the previous two-step approach can generate all connected graphs of size at most 13.

3.2 Methods to Compute Reliability Polynomials

In this research, we computed the reliability polynomials of all connected graphs of size at most 13 using two methods: Recursive and Lookup. Both methods are based on the deletion-contraction relation given in Equation (2.12).

3.2.1 Recursive Method

The Recursive method includes two algorithms *ComputeAllRelPolys1* and *ComputeRelPoly*. The algorithm *ComputeAllRelPolys1* takes as input a list L of all connected graphs of size at most 13 and calls $ComputeRelPoly$ once for each graph G in L. The algorithm $ComputeRelPoly$ takes as input a graph G and recursively computes the reliability polynomial of G with the base case where the input is the graph N_1.

Algorithm 1 *ComputeAllRelPolys1*

Input: Graph list L

foreach G *in* L **do**
  | $ComputeRelPoly(G)$
end

Algorithm 2 *ComputeRelPoly*

```
Input: Graph G
if G has no edges then
    if G has a single vertex then
        return 1
    else
        return 0
    end
else
    e ← an edge of G
    if e is a loop then
        return ComputeRelPoly(G/e)
    else if e is a bridge then
        return p * ComputeRelPoly(G/e)
    else
        return p * ComputeRelPoly(G/e) + (1 − p) * ComputeRelPoly(G − e)
    end
end
```

3.2.2 Lookup Method

The Lookup method includes one function *ComputeAllRelPolys2*. The input list L of *ComputeAllRelPolys2* is required to be sorted in increasing order of m. *ComputeAllRelPolys2* maintains two lists GL and PL which store the processed graphs and their reliability polynomials respectively. The reliability polynomial of the i-th graph in GL is the i-th reliability polynomial in PL. When computing the reliability polynomial of a graph G of size m, the reliability polynomials of both G/e and $G-e$ can be found in PL as GL is a list of all connected graphs of size no greater than m.

Algorithm 3 *ComputeAllRelPolys2*

Input: Graph list L

$GL \leftarrow$ an empty list of graphs

$PL \leftarrow$ an empty list of reliability polynomials

insert N_1 to the first place of GL

insert $\Pi(N_1, p)$ to the first place of PL

foreach G *in* L **do**

  $i_1 \leftarrow$ index of G_1 in GL where G_1 is isomorphic to G/e

  $i_2 \leftarrow$ index of G_2 in GL where G_2 is isomorphic to $G - e$

  $\Pi(G_1, p) \leftarrow PL(i_1)$

  $\Pi(G_2, p) \leftarrow PL(i_2)$

  append G to GL

  append $p * \Pi(G_1, p) + (1 - p) * \Pi(G_2, p)$ to PL

end

This research computed the reliability polynomials of all connected graphs of size at most 13 using the Recursive method. The Lookup method was used to recompute the reliability polynomials of all connected graphs of size at most 12. It was not feasible to use the Lookup method for $m > 12$ due to the large search cost of the program based on the Lookup method. A possible future improvement would be using a hash table to store graphs rather than a list. The reliability polynomials computed by both methods are the same for all connected graphs of size at most 12. The results from the Lookup method are able to confirm the correctness of the Recursive method.

3.3 Method to Search for Reliability Factorisations

In order to search for reliability factorisations, Maple (TM) (2011), a computer algebra system, was used to factor reliability polynomials computed by the Recursive program given in Section 3.2.1. There are a large number of cases of reliability equivalence. Table 4.1 shows that the number of connected graphs of size at most 13 is 1,821,234 while the number of reliability polynomials, also known as the number of classes of reliably equivalent graphs, is 54,577. Thus, the search space is reduced significantly by performing the search over reliability polynomials rather than graphs.

As stated in Section 2.2, every reliability polynomial has a factor p^{n-1} where n is the order of G. This research focuses on reliability factorisations of non-separable graphs. If a non-separable graph G has a reliability factorisation $\Pi(G_1, p)\Pi(G_2, p)$, then $\Pi(G, p)$ must have at least three factors including the factor p^{n-1}. Except p^{n-1}, the other factors are not divisible by p. Equation (2.7) shows that any graph that has a reliability factorisation has an order $n \geq 3$. Thus, the search space is further reduced by performing the search over the reliability polynomials that have at least three factors where the related graphs have at least order 3.

Algorithm $FindRelFact$ takes input consisting of a list of factored reliability polynomials. Given that a *multiset* is a generalised set in which elements are allowed to appear more than once, for each reliability polynomial $\Pi(G, p)$, $FindRelFact$ partitions the factors of $\Pi(G, p)$ excluding p^{n-1} into a pair of multisets MS_1 and MS_2 in all possible ways. Each multiset has at least one factor in any partition. Then according to Equation (2.7), for each possible partition, p^i and p^{n-1-i} are distributed to the multisets MS_1 and MS_2 respectively in all possible ways where $i \in [1, n-2]$. Thus, for each reliability polynomial, the multisets MS_1 and MS_2 are listed in all possible distributions of p^i and p^{n-1-i} for all such partitions. The pair of multisets MS_1 and MS_2 in a distribution of a partition is called a *combination*. In a combination, the product of all factors in the multiset MS_i is called a *comb-factor*, denoted by cf_i where i is 1 or 2.

Then $FindRelFact$ searches for a map from each comb-factor to a reliability polynomial in the list of reliability polynomials. If both comb-factors of a combination can be mapped to reliability polynomials, then these comb-factors and the reliability polynomial that generates this combination form a reliability factorisation. Typically, combinations do not include the case in which one com-factor is p^x where $x \in \mathbb{Z}^+$ because in such case, even if the combination formed a reliability factorisation, the reliably factorised graph must be separable. The output from Algorithm $FindRelFact$ possibly includes some duplicate cases of reliability factorisation. A search for duplicate cases of reliability factorisation over the output list of reliability factorisations is then performed to remove those duplicate cases.

Algorithm 4 *FindRelFact*

Input: Reliability polynomial list $PL^{'}$

$FL \leftarrow$ an empty list of reliability factorisations
foreach $\Pi(G, p)$ *in* $PL^{'}$ **do**
    $l \leftarrow$ number of factors of $\Pi(G, p)$
    $n \leftarrow$ order of G
    if $l \geq 3$ & $n \geq 3$ **then**
        $CL \leftarrow$ a list of all combinations for $\Pi(G, p)$
        foreach *combination comb in* CL **do**
            $cf_1 \leftarrow$ one comb-factor in *comb*
            $cf_2 \leftarrow$ the other com-factor in *comb*
            if $find(cf_1, PL^{'})$ **then**
                if $find(cf_2, PL^{'})$ **then**
                    append $\Pi(G, p) = cf_1 cf_2$ to FL
                end
            end
        end
    end
end
return FL

Chapter 4

Research Results

This chapter describes the results of this research. Section 4.1 gives the computational results and some analysis and propositions inspired from these results. Section 4.2 gives some cases of reliability factorisation. These cases include reliability factorisations of all connected graphs of size at most 8 and a reliability factorisation of an infinite family of θ-graphs.

4.1 Computational Results

This section gives the computational results of the number of graphs, the number of reliability polynomials and the number of reliability factorisations with size m of graphs (see Table 4.1). Some graphs have the same reliability polynomial, which is a case of reliability equivalence. By Proposition 1, the reliability equivalence only exists in the case that some graphs have the same order and the same size. A reliability polynomial may or may not have a reliability factorisation. By Proposition 2, the condition that a reliability polynomial has a reliability factorisation depends on the existence of a separable graph belonging to the class of reliably equivalent graphs that have this reliability polynomial.

Proposition 1. *If a graph G of order n and size m is reliably equivalent to another graph $G^{'}$ of order n' and size m', then $n = n'$ and $m = m'$.*

Proof. The reliability polynomial $\Pi(G, p)$ has zero as a root of multiplicity $n - 1$. The reliability polynomial $\Pi(G^{'}, p)$ has zero as a root of multiplicity $n' - 1$. Because

$$\Pi(G, p) = \Pi(G^{'}, p),$$

we have

$$p^{n-1} = p^{n'-1}.$$

Therefore,

$$n = n';$$

According to the deletion-contraction relation, we can say that the degree of a reliability polynomial $\Pi(G,p)$ increases by 1 if $\Pi(G,p)$ applies the deletion-contraction relation once on some edge of G. Thus, the degree of a reliability polynomial $\Pi(G,p)$ equals the size of G. Because

$$\Pi(G,p) = \Pi(G^{'},p),$$

we have

$$\deg(\Pi(G,p)) = \deg(\Pi(G^{'},p)).$$

Therefore,

$$m = m'.$$

□

Proposition 2. *The reliablity polynomial* $\Pi(G,p)$ *of a non-separable graph* G *has a reliability factorisation if and only if* G *is reliably equivalent to a separable graph* $G^{'}$*.*

Proof. If G has a reliability factorisation, then

$$\Pi(G,p) = \Pi(G_1,p)\Pi(G_2,p) \tag{4.1}$$

where G_1 and G_2 are smaller graphs. There exists a graph $G^{'}$ which is isomorphic to $G_1 \cdot G_2$. Therefore,

$$\Pi(G^{'},p) = \Pi(G_1 \cdot G_2,p). \tag{4.2}$$

By Equation (2.9), we have

$$\Pi(G_1 \cdot G_2,p) = \Pi(G_1,p)\Pi(G_2,p). \tag{4.3}$$

By Equation (4.2) and Equation (4.3), we have

$$\Pi(G^{'}, p) = \Pi(G_1, p)\Pi(G_2, p). \tag{4.4}$$

By Equation (4.1) and Equation (4.4), we have

$$\Pi(G^{'}, p) = \Pi(G, p); \tag{4.5}$$

If G is reliability equivalent to $G^{'}$, then

$$\Pi(G, p) = \Pi(G^{'}, p). \tag{4.6}$$

Because $G^{'}$ has a cutvertex, by Equation (2.9) we have

$$\Pi(G^{'}, p) = \Pi(G_1, p)\Pi(G_2, p) \tag{4.7}$$

where G_1 and G_2 are smaller graphs. By Equation (4.6) and Equation (4.7), we have

$$\Pi(G, p) = \Pi(G_1, p), \Pi(G_2, p), \tag{4.8}$$

that is, $\Pi(G, p)$ has a reliability factorisation $\Pi(G_1, p)\Pi(G_2, p)$. □

Table 4.1 lists the main results with size m of graphs. These results include the number of reliability polynomials (# RPs), the number of reliability factorisations (# RFs) as well as the number of reliability factorisations of all separable graphs (# RFs (cutvertex)) and the number of reliability factorisations of all non-separable graphs (# RFs (mixture)). The last two fields count two mutually exclusive results that sum up to the number of reliability factorisations (# RFs).

m	# graphs	# RPs	# RFs	# RFs (cutvertex)	# RFs (mixture)
1	1	1	0	0	0
2	2	2	0	0	0
3	5	4	0	0	0
4	12	8	1	1	0
5	33	16	3	3	0
6	103	35	13	12	1
7	333	76	36	35	1
8	1,183	180	107	106	1
9	4,442	443	285	274	11
10	17,576	1,349	864	841	23
11	72,810	3,314	2,011	1,977	34
12	314,595	10,986	5,690	5,564	126
13	1,410,139	38,163	15,876	15,492	384
total	1,821,234	54,577	24,886	24,305	581

Table 4.1: Experimental results in terms of m

Table 4.2 lists the ratios # RPs/# graphs, # RFs/# graphs and # RFs/# RPs with size m. The proportion # RFs/# RPs increases when m is small, peaks at $m = 9$ and decreases from $m = 10$. Similar tendency happens for # graphs, # RPs and # RFs with order n of graphs in Table 4.3. The details of the results grouped by n are not covered in this research as the reliability polynomial reflects the property of edges of graphs rather than vertices. The reason for the up-and-down tendency of the reliability factorisation may be investigated in further research as it could indicate some relations between the reliability factorisations and the size of graphs.

m	# RPs/# graphs	# RFs/# graphs	# RFs/# RPs
1	1.000	0	0
2	1.000	0	0
3	0.800	0	0
4	0.667	0.083	0.125
5	0.485	0.091	0.188
6	0.340	0.126	0.371
7	0.228	0.108	0.474
8	0.152	0.090	0.594
9	0.100	0.064	0.643
10	0.077	0.049	0.640
11	0.046	0.028	0.607
12	0.035	0.018	0.518
13	0.027	0.011	0.416
total	0.030	0.014	0.456

Table 4.2: Ratios based on Table 4.1

n	# graphs	# RPs	# RFs	# RFs (cutvertex)	# RFs (mixture)
2	13	13	0	0	0
3	109	109	30	30	0
4	1,258	706	251	251	0
5	9,615	3,152	1,202	1,182	20
6	49,232	8,955	3,409	3,354	55
7	158,590	15,046	5,978	5,820	158
8	330,994	14,674	6,521	6,334	187
9	454,635	8,313	4,622	4,498	124
10	419,885	2,864	2,125	2,103	22
11	260,670	630	638	624	14
12	106,619	101	110	109	1
13	26,485	13	0	0	0
14	3,159	1	0	0	0
total	1,821,234	54,577	24,886	24,305	581

Table 4.3: Experimental results in terms of n

4.2 Reliability Equivalence and Reliability Factorisation

According to Proposition 2, a non-separable graph G_1 whose reliability polynomial has a reliability factorisation is reliably equivalent to some separable graph G_2. A certificate of reliability factorisation of G_1 usually includes a certificate of reliability equivalence of G_1 and G_2. This section gives reliability factorisations of all non-separable graphs of size at most 8. It also gives a reliability factorisation of an infinite family of θ-graphs. We use graphs themselves to represent their reliability polynomials in equations.

4.2.1 Cases of Graphs of Small Size

Table 4.1 shows that there are three reliability factorisations of all non-separable graphs of size at most 8. They are corresponding to the cases $m = 6$, $m = 7$ and $m = 8$. Figure 4.1 illustrates the case of reliability factorisation for $m = 6$. Graph 84 is non-separable. It is reliably equivalent to Graph 96 which has a cutvertex. Both Graph 84 and Graph 96 can be reliably factorised into two reliability polynomials of graphs C_3 labelled as Graph 5.

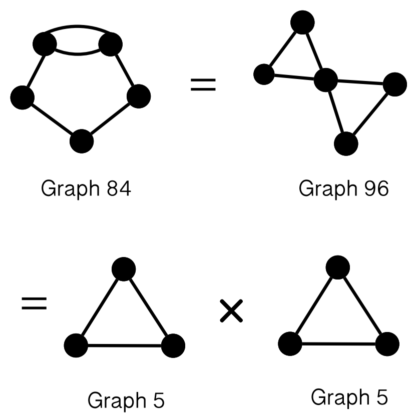
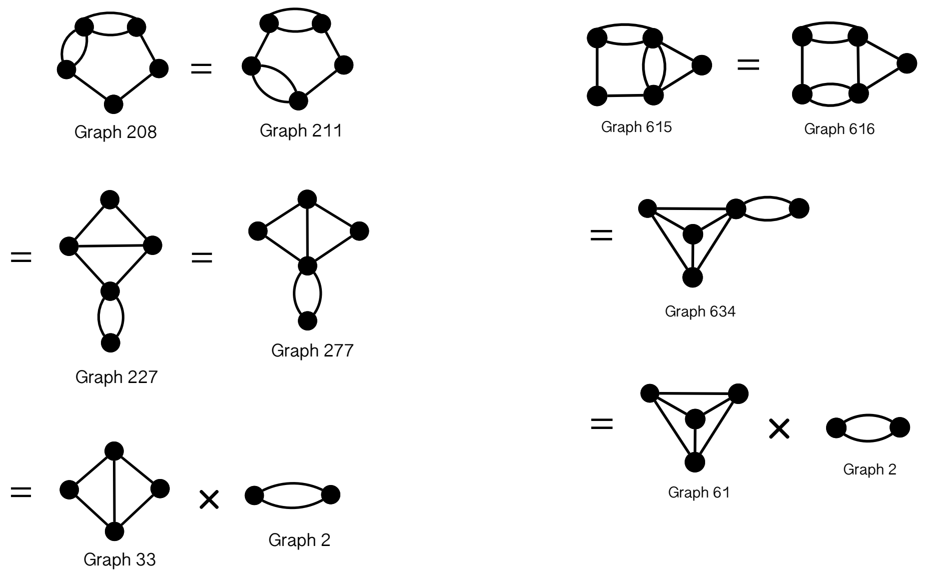


Figure 4.1: Case $m = 6$

Figure 4.2 demonstrates the two cases of reliability factorisation for $m = 7$ and $m = 8$. In Figure 4.2a, Graph 208 and Graph 211 are two reliably equivalent non-separable graphs. They are reliably equivalent to two separable graphs: Graph 227 and Graph 277. They can be reliably factorised into the reliability polynomials of Graph 33 and the reliability polynomial of the graph C_2 labelled as Graph 2. Figure

4.2b gives a reliability factorisation for a class of three reliably equivalent graphs: Graph 615, Graph 616 and Graph 634. Graph 615 and Graph 616 are non-separable and can be reliably factorised into the reliability polynomials of Graph 61 and Graph 2.



(a) Case $m = 7$ (b) Case $m = 8$

Figure 4.2: Cases $m = 7$ and $m = 8$

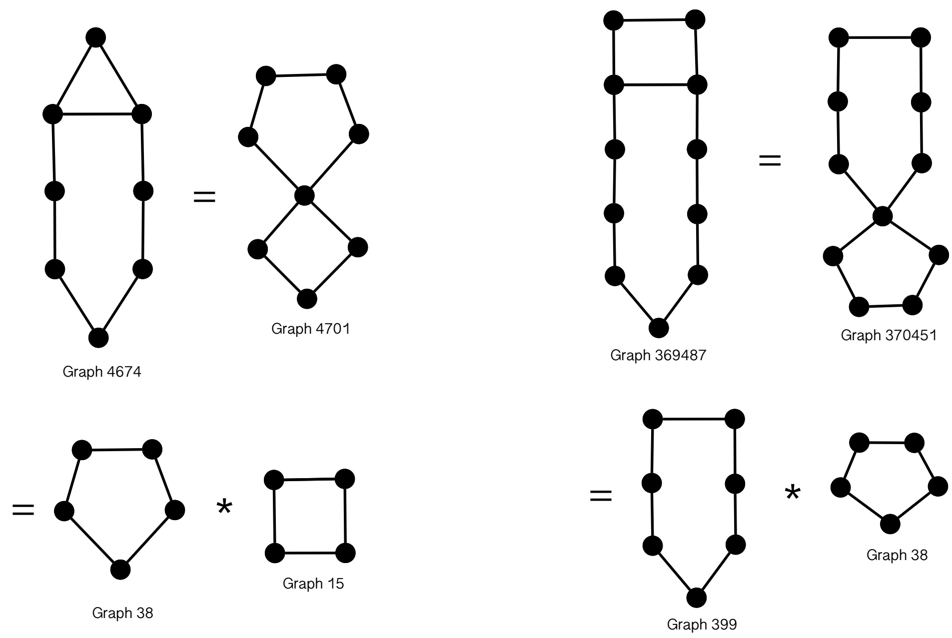
4.2.2 Case of an Infinite Families of θ-graphs

The following theorem gives a reliability factorisation of an infinite graph family $\theta_{1,d,2d+2}$ for $d \in \mathbb{Z}^+$:

Theorem 1. *The reliability polynomial* $\Pi(\theta_{1,d,2d+2}, p)$ *of a graph* $\theta_{1,d,2d+2}$ *has the reliability factorisation* $\Pi(C_{2d+1}, p)\Pi(C_{d+2}, p)$ *for* $d \in \mathbb{Z}^+$*.*

We give the proof of this theorem in Section 5.2.2 by a sequence of certificate steps and mathematical induction. All graphs in the family $\theta_{1,d,2d+2}$ for $d \in \mathbb{Z}^+$ are non-separable. By definition of θ-graphs, a graph $\theta_{1,d,2d+2}$ has $1 + d + 2d + 2 = 3d + 3$ edges. In all graphs of size at most 13, there are three graphs belonging to this family. The simplest graph in this family is $\theta_{1,1,4}$ (labelled as Graph 84 illustrated in Figure 4.1) for $d = 1$. Figure 4.3 gives the reliability factorisations of the other two graphs $\theta_{1,2,6}$ and $\theta_{1,3,8}$ generated in this research.

In Figure 4.3a, the graph $\theta_{1,2,6}$ (labelled as Graph 4674) has a reliability factorisation in terms of the graph C_5 (labelled as Graph 38) and the graph C_4 (labelled as Graph 15). Figure 4.3b shows that the graph $\theta_{1,2,6}$ (labelled as Graph 369487) can be reliably factorised into the graph C_7 (labelled as Graph 339) and the graph C_5 (labelled as 38).



(a) Case of the graph $\theta_{1,2,6}$ (b) Case of the graph $\theta_{1,3,8}$

Figure 4.3: Cases of the graphs $\theta_{1,2,6}$ and $\theta_{1,3,8}$

Chapter 5

Certificates of Reliability Factorisation

This chapter gives certificates for the cases of reliability equivalence and reliability factorisation described in Section 4.2. In order to construct certificates, Section 5.1 gives twelve types of certificate steps. Section 5.2 gives three certificates of reliability factorisation of non-separable graphs for the cases $m = 6$, $m = 7$ and $m = 8$. It also shows a proof of Theorem 1 following an illustration by a certificate of reliability factorisation of an infinie family of graphs $\theta_{1,d,2d+2}$ for $d = 2$.

5.1 Certificate Steps

A *certificate step* is a way to transform an expression E_i to another expression E_{i+1} based on identities in a certificate of reliability factorisation or reliability equivalence. In order to construct certificates, this section defines twelve types of certificate steps. Any certificate step is based on either algebraic operations or a property of the reliability polynomial.

5.1.1 Basic Certificate Steps

The properties of the reliability polynomial stated in Section 2.3 include the cases where an graph has a loop, a bridge or a cutvertex and the general deletion-contraction principle. Based on these properties, we give the following certificate steps:

(CS1) $\Pi(G, p)$ becomes $\Pi(G - e, p)$ for some loop $e \in E(G)$

(CS2) $\Pi(G, p)$ becomes $\Pi(G + vv, p)$ where $v \in V(G)$

(CS3) $\Pi(G,p)$ becomes $p\Pi(G/e,p)$ for some bridge $e \in E(G)$

(CS4) $p\Pi(G,p)$ becomes $\Pi(G^{'},p)$ where $G^{'}$ is isomorphic to $G \cdot K_2$

(CS5) $\Pi(G,p)$ becomes $p\Pi(G/e,p) + (1-p)\Pi(G-e,p)$ for some $e \in E(G)$ where e is neither a loop or a bridge

(CS6) $p\Pi(G_1,p)+(1-p)\Pi(G_2,p)$ becomes $\Pi(G,p)$ where G_1 is isomorphic to G/e, G_2 is isomorphic to $G-e$ and $e \in E(G)$ is neither a loop nor bridge

(CS7) $\Pi(G,p)$ becomes $\Pi(G_1,p)\Pi(G_2,p)$ for some cutvertex $v \in V(G)$ where $G_1 \cdot G_2 = G$ and $G_1 \cap G_2 = \{v\}$

(CS8) $\Pi(G_1,p)\Pi(G_2,p)$ becomes $\Pi(G,p)$ where $G = G_1 \cdot G_2$

(CS9) $\Pi(G,p)$ becomes $\Pi(G^{'},p)$ where $G \sim G^{'}$

(CS10) By applying a sequence of algebraic operations, an expression E becomes another expression $E^{'}$.

5.1.2 Additional Property of the Reliability Polynomial

The properties of the reliability polynomial in the case where an edge is a loop or a bridge can be derived from the deletion-contraction principle. Both cases reduce the number of repeated expressions in certificates. Following this purpose, we give another property of the reliability polynomial in Theorem 2:

Theorem 2. *If a graph G is divided by a C_2-bridge into two subgraphs G_1 and G_2, then*

$$\Pi(G,p) = p(2-p)\Pi(G_1,p))\Pi(G_2,p) \tag{5.1}$$

Proof. Assume the two vertices of C_2-bridge are v_1 and v_2. Thus, v_1 and v_2 are two cutvertices of the graph G. The reliability polynomial $\Pi(G,p)$ can be expressed as

$$\begin{aligned}
\Pi(G,p) &= \Pi(G_1,p)\Pi(C_2 \cdot G_2,p) \qquad \textbf{(CS7)}\\
&= \Pi(G_1,p)\Pi(C_2,p)\Pi(G_2,p) \qquad \textbf{(CS7)}\\
&= \Pi(G_1,p)(p\Pi(C_1,p) + (1-p)\Pi(K_1,p))\Pi(G_2,p) \qquad \textbf{(CS5)}\\
&= \Pi(G_1,p)(p + (1-p)\Pi(K_1,p))\Pi(G_2,p) \qquad \textbf{(CS1)}\\
&= \Pi(G_1,p)(p + (1-p)p)\Pi(G_2,p) \qquad \textbf{(CS3)}\\
&= p(2-p)\Pi(G_1,p))\Pi(G_2,p) \qquad \textbf{(CS10)}
\end{aligned}$$

□

According to Theorem 2, we give the following two certificate steps:

(CS11) $\Pi(G, p)$ becomes $p(2-p)\Pi(G_1, p)\Pi(G_2, p)$ for some C_2-bridge C_2 where G_1 and G_2 are subgraphs of G such that $G_1 \cdot C_2 \cdot G_2 = G$

(CS12) $p(2-p)\Pi(G_1, p)\Pi(G_2, p)$ becomes $\Pi(G, p)$ where $G = G_1 \cdot C_2 \cdot G_2$

5.2 Sample Certificates of Reliability Factorisation

5.2.1 Simple Cases

This section demonstrates three certificates of reliability factorisation in the cases $m = 6$, $m = 7$ and $m = 8$ given in Table 4.1. Figures 5.1 and 5.2 give a certificate of reliability factorisation for the graph $\theta_{1,1,4}$ (labelled as Graph 84 in the case $m = 6$). The length of this certificate is 19. Similarly in the case $m = 7$, Figures 5.3 and 5.4 give a certificate of reliability factorisation of length 16 for Graph 211. The case $m = 8$ demonstrated in Figure 5.5 is a certificate of reliability factorisation with length 18 for Graph 616 using the reliability equivalence of Graph 211 and Graph 227. The certificate of reliability equivalence of these two graphs is included in the first 15 steps of the certificate in Figures 5.3 and 5.4.

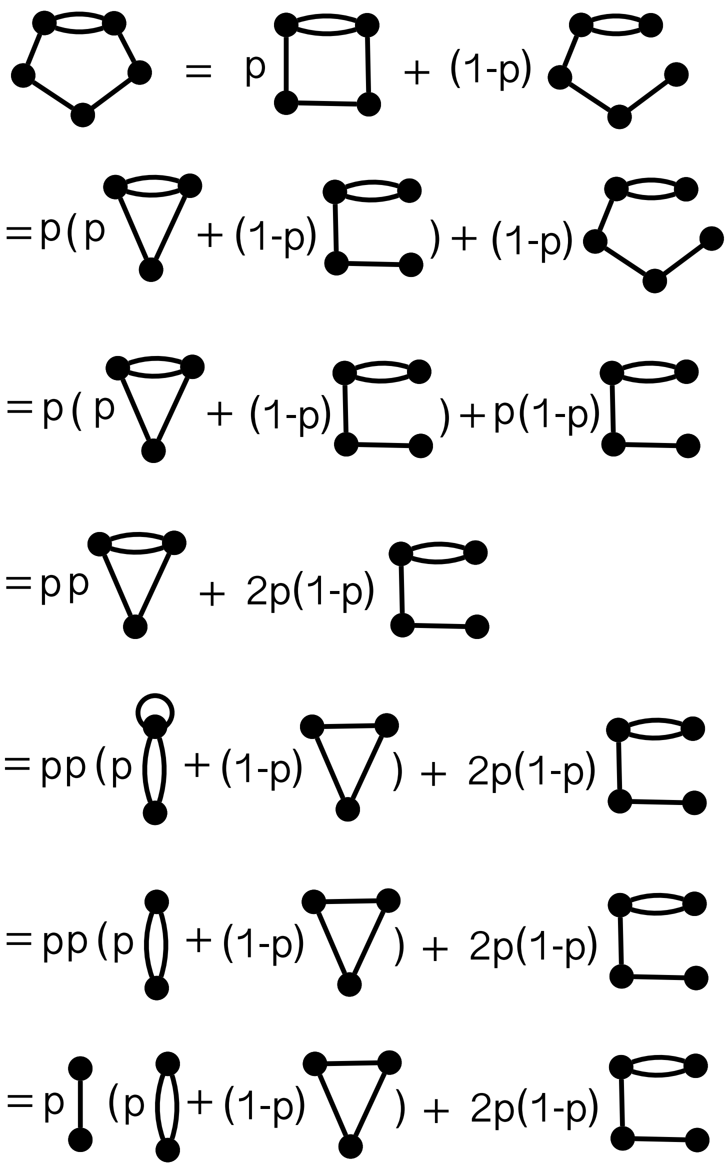


Figure 5.1: Certificate of reliability factorisation for Graph 84 (to be continued)

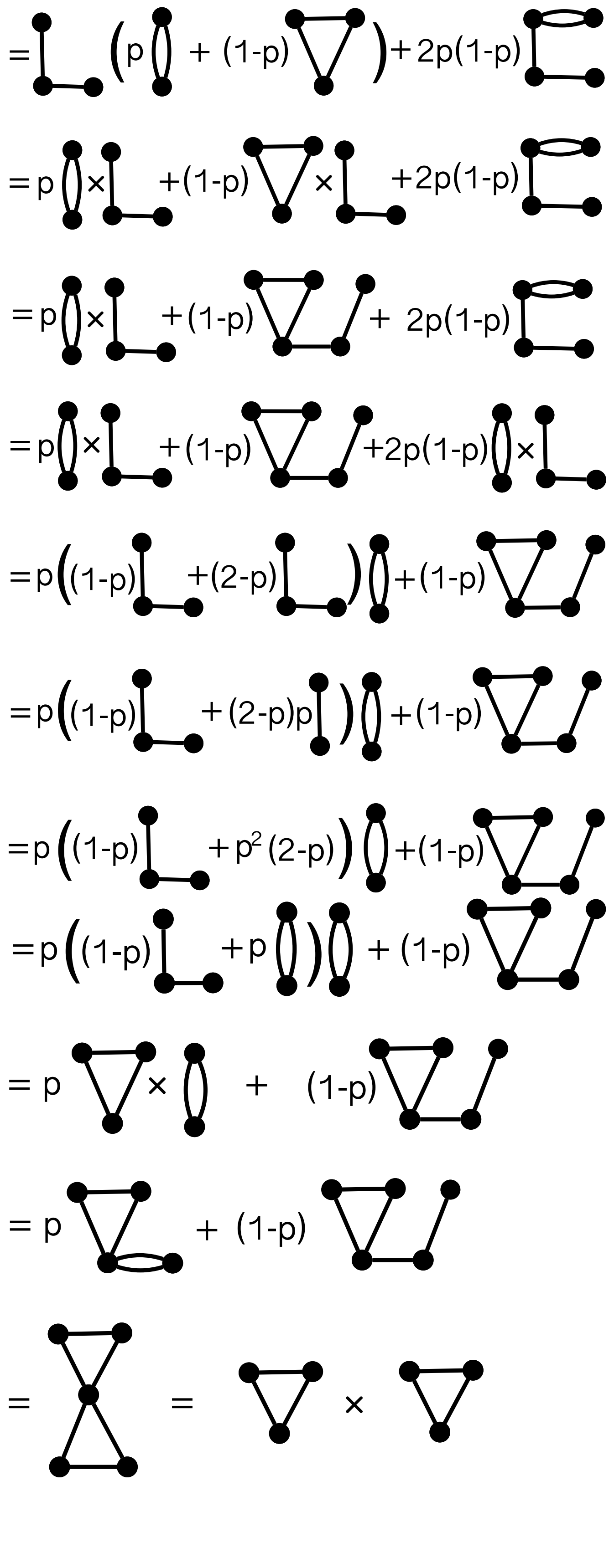


Figure 5.2: Certificate of reliability factorisation for Graph 84 (Continued from Figure 5.1)

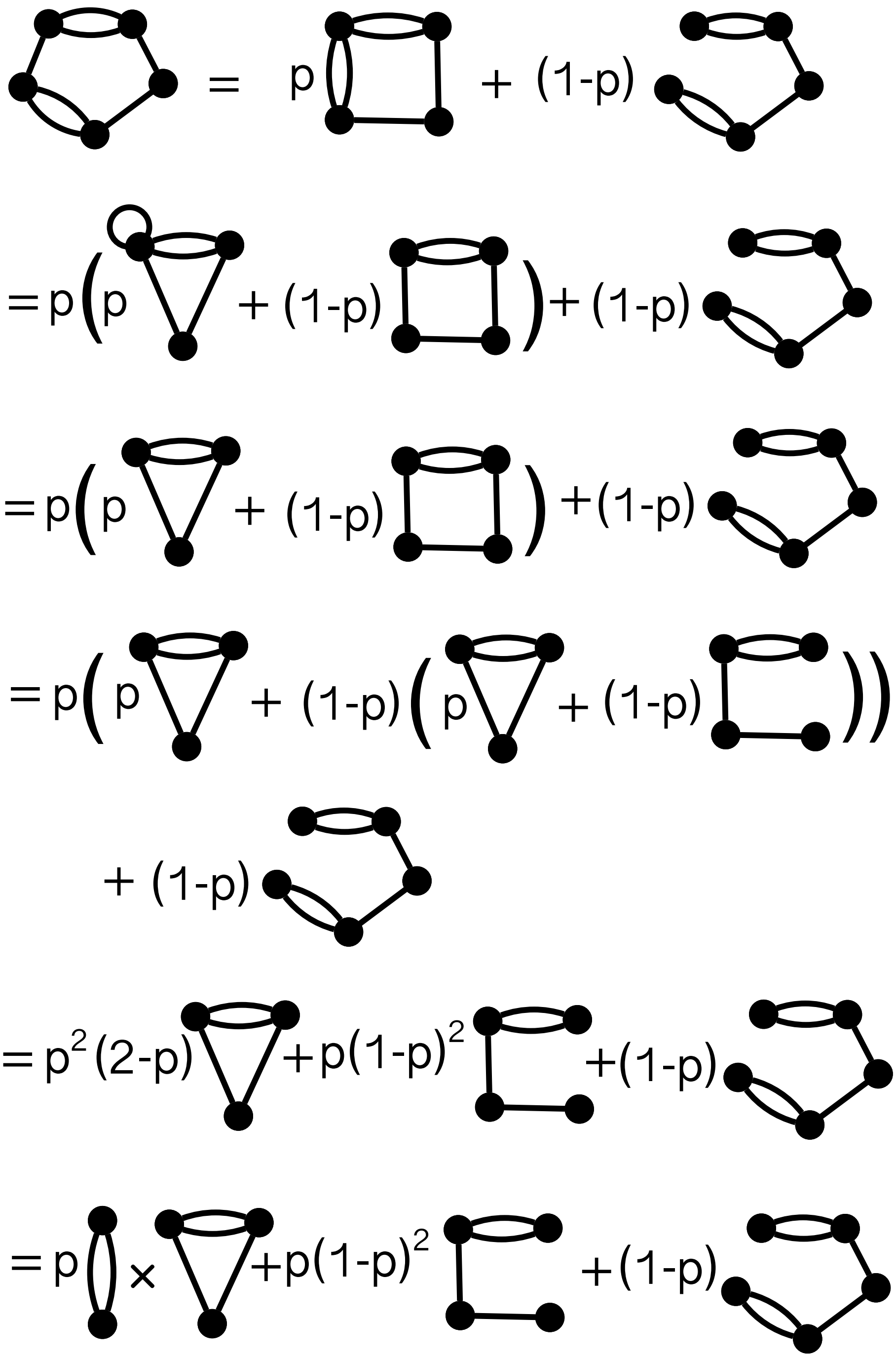


Figure 5.3: Certificate of reliability factorisation for Graph 211 (to be continued)

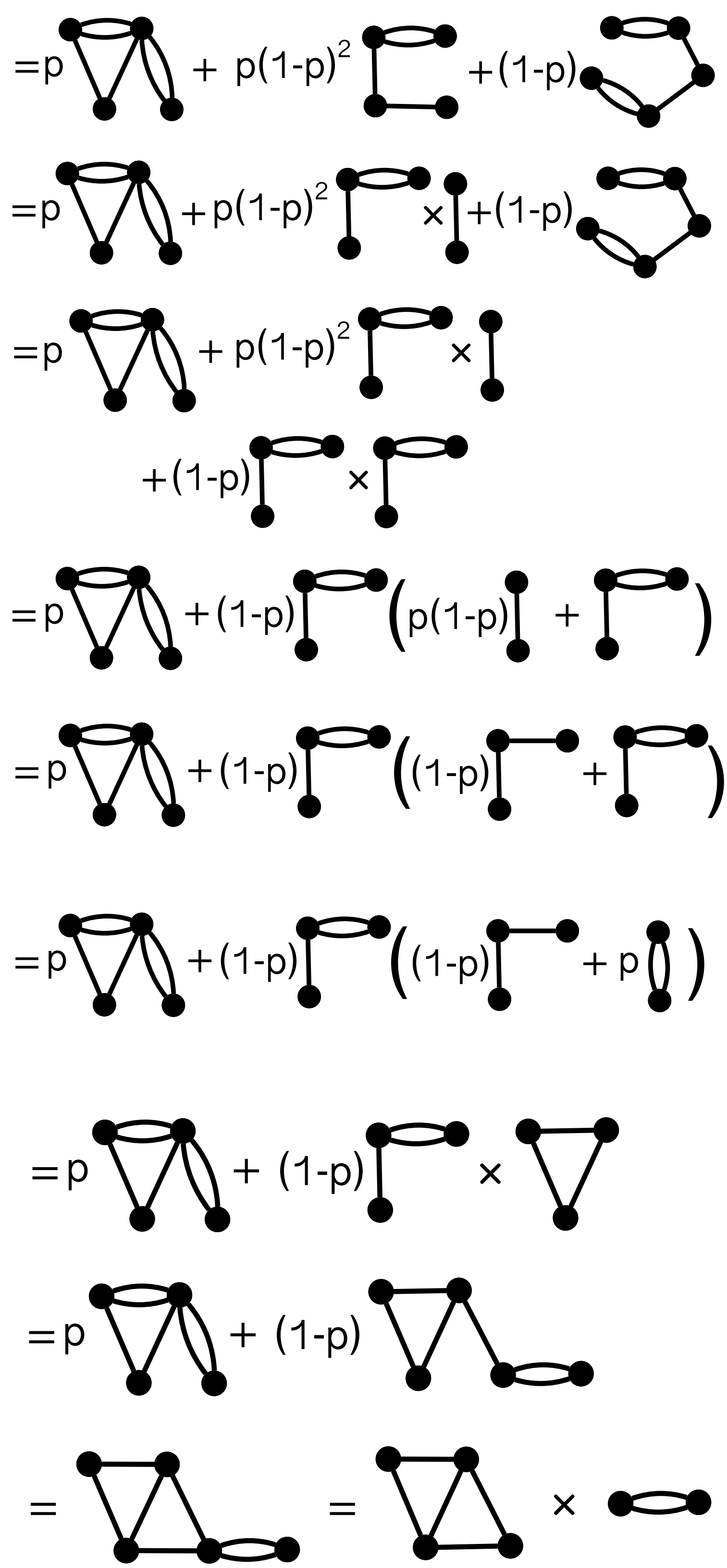


Figure 5.4: Certificate of reliability factorisation for Graph 211 (Continued from Figure 5.3)

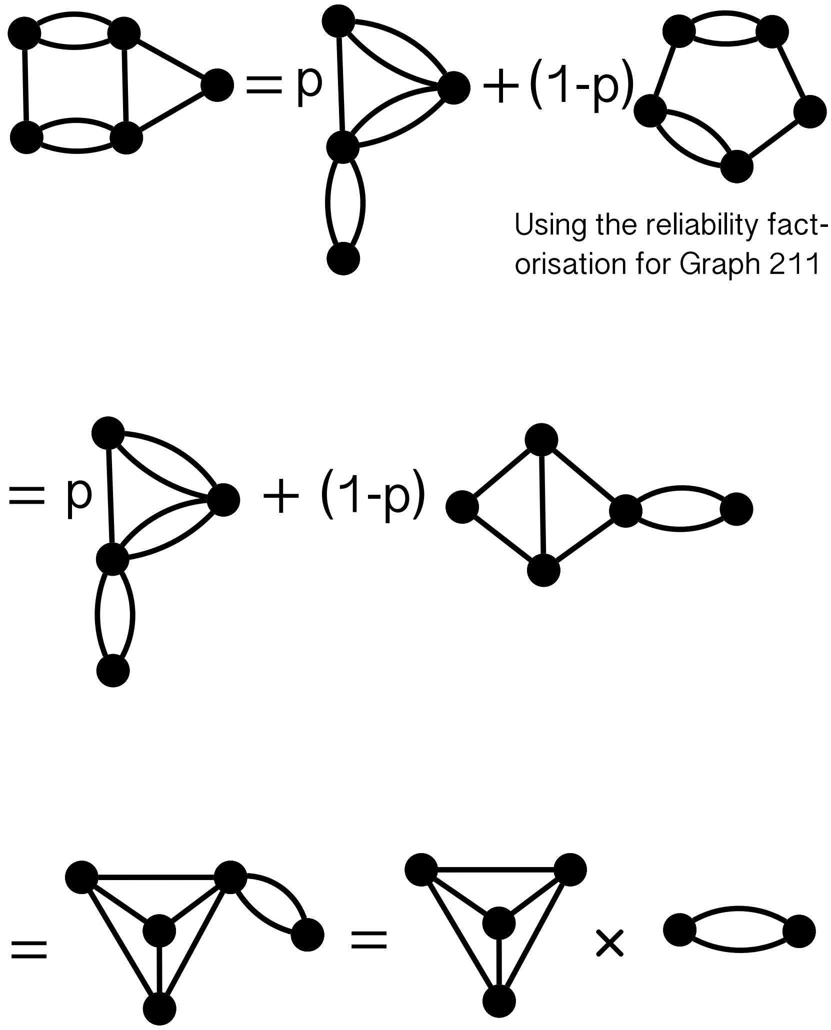


Figure 5.5: Certificate of reliability factorisation for Graph 616

5.2.2 Case of the Infinite Graph Family $\theta_{1,d,2d+2}$

This section proves Theorem 1 by a sequence of certificate steps and mathematical induction. Before showing the proof, we introduce the following fact:

Fact 1. *The reliability polynomial* $\Pi(C_n, p)$ *of the cycle* C_n *for* $n \geq 3$ *can be expressed as*

$$\Pi(C_n, p) = p\Pi(C_{n-1}, p) + (1-p)\Pi(P_{n-1}, p) \qquad \textbf{\textit{(CS6)}}$$

where P_{n-1} *is a path of order* $n-1$.

By providing Fact 1, we give the following proof for Theorem 1

Proof. We use a proof by induction on d. When $d = 1$, the reliability polynomial $\Pi(\theta_{1,1,4}, p) = \Pi(C_3, p)\Pi(C_3, p)$. This is the only case of reliability factorisation for $m = 6$ as shown in Figures 5.1 and 5.2; Then given the hypothesis that $\Pi(\theta_{1,d-1,2d}, p) = \Pi(C_{2d-1}, p)\Pi(C_{d+1}, p)$ is true for $d \geq 2, d \in \mathbb{Z}^+$, the reliability polynomial

$$
\begin{aligned}
\Pi(\theta_{1,d,2d+2}, p) &= p\Pi(\theta_{1,d,2d+1}, p) + (1-p)\Pi(C_{d+1} \cdot P_{2d+1}, p) \qquad \textbf{(CS5)} \\
&= p\left[p\Pi(\theta_{1,d,2d}, p) + (1-p)\Pi(C_{d+1} \cdot P_{2d}, p)\right] \\
&\qquad + (1-p)\Pi(C_{d+1} \cdot P_{2d+1}, p) \qquad \textbf{(CS5)} \\
&= p\left[p\Pi(\theta_{1,d,2d}, p) + (1-p)\Pi(C_{d+1} \cdot P_{2d}, p)\right] \\
&\qquad + (1-p)p\Pi(C_{d+1} \cdot P_{2d}, p) \qquad \textbf{(CS3)} \\
&= p^2\Pi(\theta_{1,d,2d}, p) + 2p(1-p)\Pi(C_{d+1} \cdot P_{2d}, p) \qquad \textbf{(CS10)} \\
&= p^2\left[p\Pi(\theta_{1,d-1,2d}, p) + (1-p)\Pi(C_{2d+1} \cdot P_{d-1}, p)\right] \\
&\qquad + 2p(1-p)\Pi(C_{d+1} \cdot P_{2d}, p) \qquad \textbf{(CS5)} \\
&= p^3\Pi(\theta_{1,d-1,2d}, p) + (1-p)p^2\Pi(C_{2d+1} \cdot P_{d-1}, p) \\
&\qquad + 2p(1-p)\Pi(C_{d+1} \cdot P_{2d}, p) \qquad \textbf{(CS10)} \\
&= p^3\Pi(C_{2d-1}, p)\Pi(C_{d+1}, p) + (1-p)p^2\Pi(C_{2d+1} \cdot P_{d-1}, p) \\
&\qquad + 2p(1-p)\Pi(C_{d+1} \cdot P_{2d}, p) \qquad \textbf{(Inductive Hypothesis)} \\
&= p^3\Pi(C_{2d-1}, p)\Pi(C_{d+1}, p) + (1-p)\Pi(C_{2d+1} \cdot P_{d+1}, p) \\
&\qquad + 2p(1-p)\Pi(C_{d+1} \cdot P_{2d}, p) \qquad \textbf{(CS4 Twice)} \\
&= p^3\Pi(C_{2d-1}, p)\Pi(C_{d+1}, p) + (1-p)\Pi(C_{2d+1} \cdot P_{d+1}, p) \\
&\qquad + 2p(1-p)\Pi(C_{d+1}, p)\Pi(P_{2d}, p) \qquad \textbf{(CS7)} \\
&= p\Pi(C_{d+1}, p)\left[p^2\Pi(C_{2d-1}, p) + (1-p)\Pi(P_{2d}, p) + (1-p)\Pi(P_{2d}, p)\right] \\
&\qquad + (1-p)\Pi(C_{2d+1} \cdot P_{d+1}, p) \qquad \textbf{(CS10)} \\
&= p\Pi(C_{d+1}, p)\left[p^2\Pi(C_{2d-1}, p) + (1-p)p\Pi(P_{2d-1}, p) + (1-p)\Pi(P_{2d}, p)\right] \\
&\qquad + (1-p)\Pi(C_{2d+1} \cdot P_{d+1}, p) \qquad \textbf{(CS3)} \\
&= p\Pi(C_{d+1}, p)\{p\left[p\Pi(C_{2d-1}, p) + (1-p)\Pi(P_{2d-1}, p)\right] + (1-p)\Pi(P_{2d}, p)\} \\
&\qquad + (1-p)\Pi(C_{2d+1} \cdot P_{d+1}, p) \qquad \textbf{(CS10)} \\
&= p\Pi(C_{d+1}, p)\left[p\Pi(C_{2d}, p) + (1-p)\Pi(P_{2d}, p)\right] \\
&\qquad + (1-p)\Pi(C_{2d+1} \cdot P_{d+1}, p) \qquad \textbf{(CS6)} \\
&= p\Pi(C_{d+1}, p)\Pi(C_{2d+1}, p) + (1-p)\Pi(C_{2d+1} \cdot P_{d+1}, p) \qquad \textbf{(CS6)} \\
&= p\Pi(C_{d+1}, p)\Pi(C_{2d+1}, p) + (1-p)\Pi(C_{2d+1}, p)\Pi(P_{d+1}, p) \qquad \textbf{(CS7)} \\
&= \Pi(C_{2d+1}, p)\left[p\Pi(C_{d+1}, p) + (1-p)\Pi(P_{d+1}, p)\right] \qquad \textbf{(CS10)} \\
&= \Pi(C_{2d+1}, p)\Pi(C_{d+2}, p). \qquad \textbf{(CS6)}
\end{aligned}
$$

Therefore, $\Pi(\theta_{1,d,2d+2}, p)$ has a reliability factorisation $\Pi(C_{2d+1}, p)\Pi(C_{d+2}, p)$ for all $d \in \mathbb{Z}^+$. □

The above proof is also a certificate of reliability factorisation for the infinite graph family $\theta_{1,d,2d+2}$. To illustrate this proof, Figures 5.6-5.8 give a certificate of reliability factorisation for the graph $\theta_{1,2,6}$.

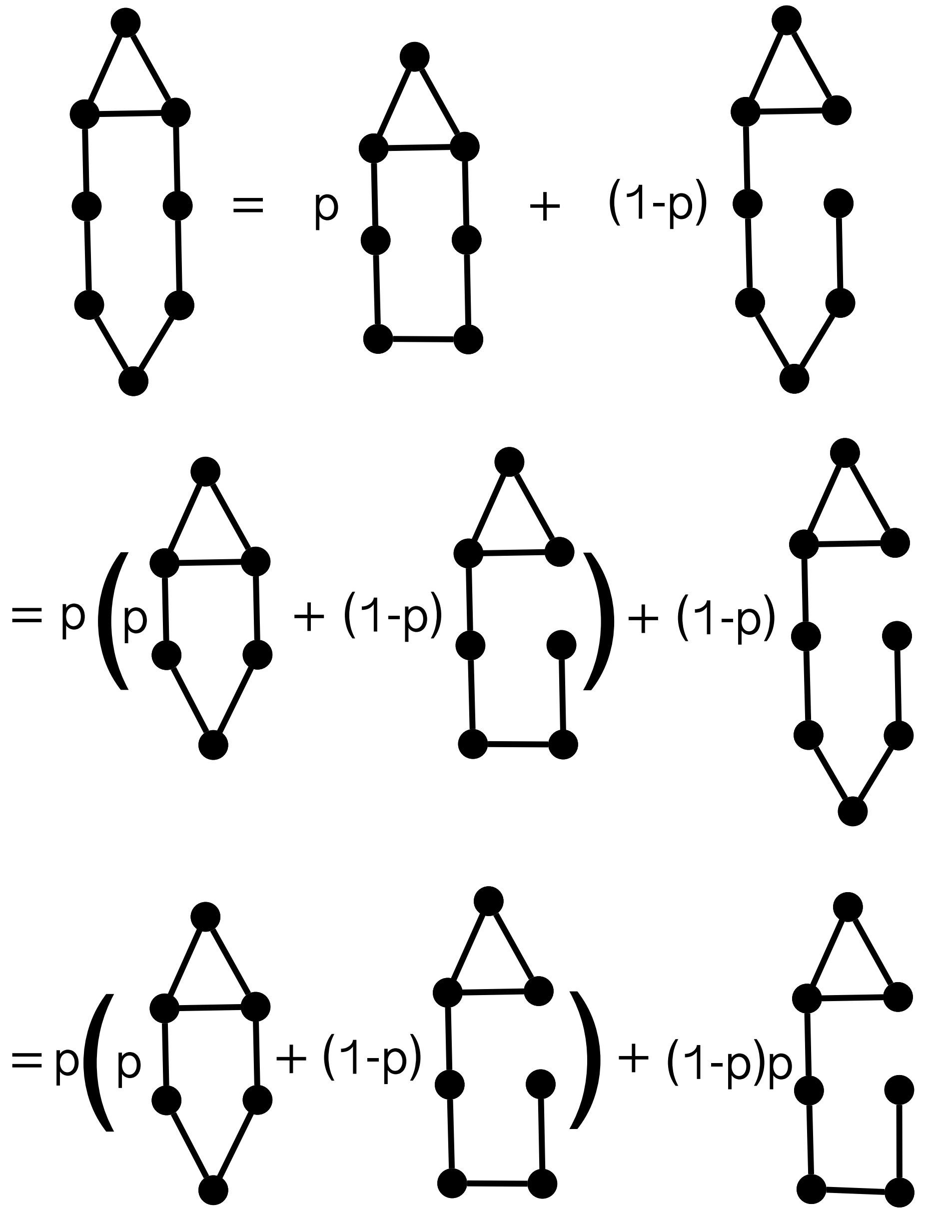


Figure 5.6: Certificate of reliability factorisation for Graph 4674 (to be continued)

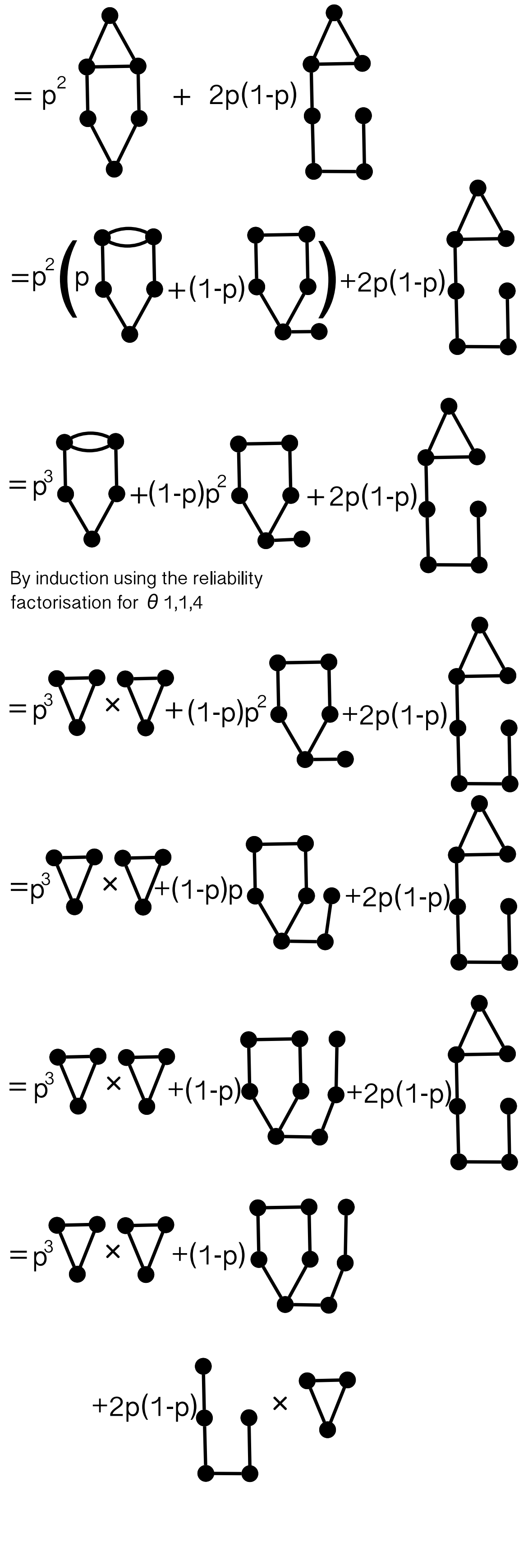


Figure 5.7: Certificate of reliability factorisation for Graph 4674 (Continued from Figure 5.6)

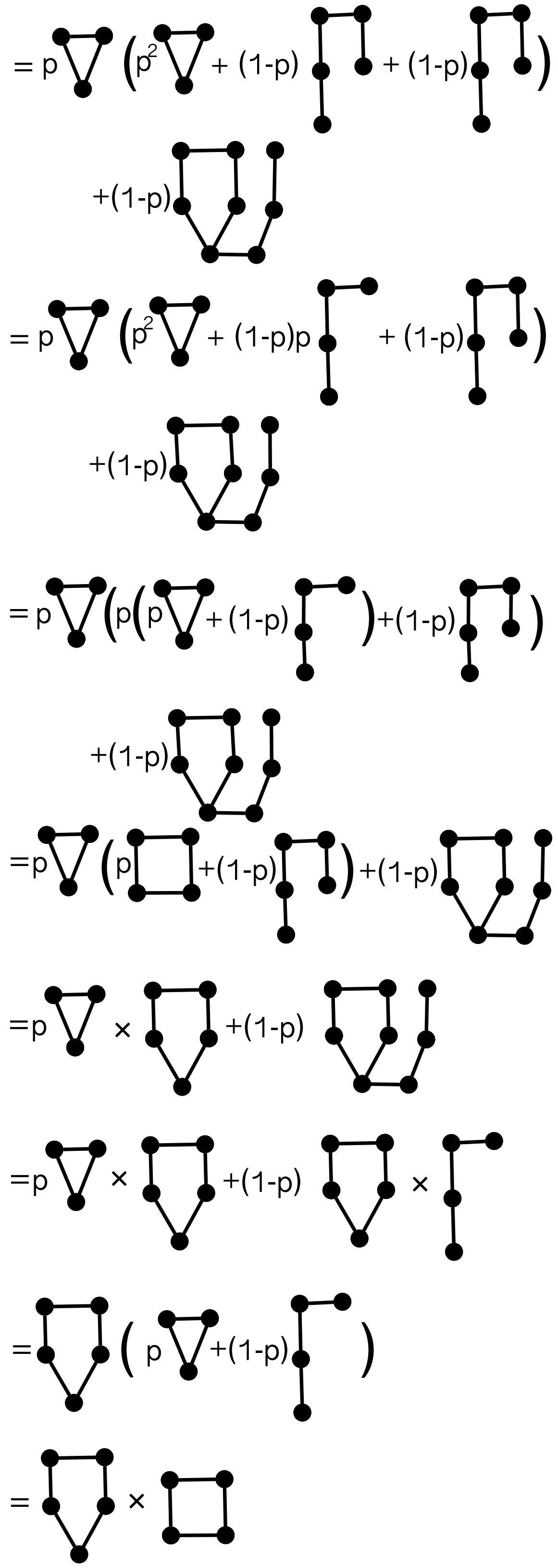


Figure 5.8: Certificate of reliability factorisation for Graph 4674 (Continued from Figure 5.7)

5.2.3 Lengths of Certificates of Reliability Factorisation

This section discusses lengths of certificates of reliability factorisation. The upper bound on the lengths of certificates of reliability factorisation could be exponential. However, the lengths for the cases of certificates of reliability factorisation in Sections 5.2.1 and 5.2.2 are remarkably short.

Table 5.1 gives the lengths of certificates of reliability factorisation for the cases $m = 6$, $m = 7$ and $m = 8$. Table 5.2 describes the lengths of certificates of reliability factorsation for the infinite family of graphs $\theta_{1,d,2d+2}$. In terms of the size m of graphs, the length appears to be linear, which is similar to the results shown by Morgan and Farr (2009b) that the lengths of certificates of chromatic factorisation are much shorter in practice. Considering that computing reliability polynomials are known as hard problems, a certificate could be a approach with smaller complexity to verify a reliability factorisation for a reliability polynomial.

m	Length of certificate
6	19
7	16
8	18

Table 5.1: Lengths of certificates for the cases $m \leq 8$

d	m	Length of certificate
1	6	19
2	9	36
.	.	.
.	.	.
.	.	.
i	$3i + 3$	$17(i - 1) + 19$

Table 5.2: Lengths of certificates for the inifinite graph family $\theta_{1,d,2d+2}$

Chapter 6

Complexity Analysis

As motivated by the short lengths of certificates of reliability factorisation found in this research, this chapter discusses the relationship between the complexity of the problem **Reliability Factorisation** and the upper bound on the lengths of certificates of reliability factorisation. The lengths of certificates are related to the complexity of decision problems. Section 6.1 defines the problem **Reliability Factorisation** and some complexity classes. Section 6.2 analyses the complexity of the problem **Reliability Factorisation** using both an oracle of the problem **Compute Reliability Polynomial** and the upper bound on the lengths of certificates of reliability factorisations.

6.1 Preliminaries

A *decision problem* T is defined as a problem consisting of a set D_T of instances and a subset $Y_T \subseteq D_T$ of yes-instances (Garey and Johnson, 1979). It can be specified into two parts: The first part is a generic instance of the problem; the second part is a yes-no question in terms of this generic instance (Garey and Johnson, 1979). The decision problem **Reliability Factorisation** is defined as follows:

Reliability Factorisation

Input: Connected graph G

Question: Does $\Pi(G, p)$ have a reliability factorisation?

Three popular complexity classes of decision problems are P, NP and NP-complete. *Polynomial time* refers to time complexity functions which are $O(p(n))$ for some polynomial function p of input length n (Garey and Johnson, 1979). A function $f(n)$ is $O(g(n))$ whenever there exists a constant c such that $f(n) \leq c \cdot g(n)$ for all sufficiently large n (Garey and Johnson, 1979).

A decision problem T is in P if there exists a polynomial time DTM (Deterministic Turing Machine) program M such that for every $I \in D_T$, $I \in Y_T$ if and only if M accepts I (Garey and Johnson, 1979), i.e. T can be solved by M.

A decision problem T is in *NP* if there exists a polynomial time DTM program $M(-,-)$ such that for every $I \in D_T$, $I \in Y_T$ if and only if there exists a C such that $M(I,C)$ accepts (Crama and Hammer, 2011; Garey and Johnson, 1979). Such a C is called a *certificate* for I. In other words, T is in *NP* if it can be verified by $M(-,-)$.

A *polynomial transformation* from a decision problem T to another decision problem $T^{'}$ is a function $f : D_T \rightarrow D_{T'}$ such that f is computable in polynomial time and for all $I \in D_T$, $I \in Y_T$ if and only if $f(I) \in Y_{T'}$ (Garey and Johnson, 1979). A decision problem T is *NP-complete* if T satisfies two conditions (Garey and Johnson, 1979):

(1) $T \in NP$;

(2) All other decision problems $T^{'} \in NP$ can be polynomially transformed to T.

An *Oracle Turing Machine* (OTM) is a DTM that is allowed to use an oracle (Garey and Johnson, 1979). A *polynomial time Turing reduction* from a problem T to a problem $T^{'}$ is a polynomial time algorithm A that solves T by using an oracle O for solving $T^{'}$ such that each oracle call is counted as one time-step. This means that A solves T in polynomial time by using O for solving $T^{'}$. A problem T is *NP-hard* if there exists a polynomial time Turing reduction from some NP-complete problem $T^{'}$ to T (Garey and Johnson, 1979).

The class *#P* is the set of functions $f : \Sigma^* \rightarrow \mathbb{N} \cup \{0\}$ such that there exists a polynomial time algorithm $M(-,-)$ such that for all input $I \in \Sigma^*$, $f(I)$ is the number of certificates C such that $M(I,C)$ accepts.

A decision problem T is in $P^{\#P}$ if there exists a polynomial time OTM M which uses an oracle for a function in #P such that for every $I \in D_T$, $I \in Y_T$ if and only if M accepts I (Welsh, 1993). The class *FP* is the set of problems T that can be solved by a polynomial time DTM (Welsh, 1993). Here T can be a decision problem or a problem of other types. A problem T is in $FP^{\#P}$ if there exists a polynomial time OTM M which uses an oracle in #P such that T can be solved by M. A decision problem $T^{'}$ is in $NP^{\#P}$ if there exists a polynomial time OTM $M(-,-)$ which uses an oracle in *#P* or $FP^{\#P}$ such that for every $I \in D_{T'}$, $I \in Y_{T'}$ if and only if there exists a certificate C such that $M(I,C)$ accepts.

6.2 Complexity Analysis of Reliability Factorisation

6.2.1 Oracle for Computing Reliability Polynomial

The problem **Reliability Factorisation** can be verified in polynomial time by Algorithm M_1 which uses an oracle for solving the following problem:

Compute Reliability Polynomial
Input: Connected graph G
Output: The reliability polynomial $\Pi(G, p)$

Algorithm 5 M_1

Input: Connected graph G
Certificate: Connected graphs G_1, G_2

```
1: compute Π(G, p), Π(G1, p) and Π(G2, p) using oracle
2: Π(G', p) ← Π(G1, p)Π(G2, p)
3: simplify Π(G', p) → Π(G'', p)
4: if coefficients between Π(G, p) and Π(G'', p) are the same then
5:     return accept
6: else
7:     return reject
8: end if
```

In Algorithm M_1, Step **(1)** includes three oracle calls to the problem **Compute Reliability Polynomial** for $\Pi(G, p)$, $\Pi(G_1, p)$ and $\Pi(G_2, p)$. Each oracle call takes time 1. Given that m is the size of graph G, the degree of $\Pi(G, p)$ is m. The degree of both $\Pi(G_1, p)$ and $\Pi(G_2, p)$ is less than m. In Steps **(2)** and **(3)**, it takes time at most $2 \cdot m^2$ to multiply $\Pi(G_1, p)$ by $\Pi(G_2, p)$ and simplify the expression. In Step **(4)**, it takes time at most m to compare the coefficients between $\Pi(G, p)$ and $\Pi(G'', p)$. Thus, Algorithm M_1 takes time at most $3 \cdot 1 + 2 \cdot m^2 + m = 2m^2 + m + 3$. Thus, M_1 takes polynomial time.

The coefficients of the reliability polynomial $\Pi(G, p)$ count the number of connected subgraphs of G (Beichl et al., 2011). The problem **Compute Reliability Polynomial** can be solved in polynomial time by using an oracle in $FP^{\#P}$ to compute the coefficients of the reliability polynomial. The problem **Reliability Factorisation** can be verified by the polynomial time OTM M_1 which uses an oracle in $FP^{\#P}$ for solving **Compute Reliability Polynomial**. Thus, **Reliability Factorisation** belongs to the complexity class $NP^{\#P}$.

6.2.2 Lengths of Certificates of Reliability Factorisation

Algorithms M_2 and M_3 verify the problem **Reliability Factorisation** using the concept of a certificate of reliability factorisation. Each step of a certificate of reliability factorisation is based on some properties of the reliability polynomial or some algebraic operations. There is no need to compute the reliability polynomial in this case. We will see that processing a reliability polynomial takes at most a linear time in the size of the input graph.

Algorithm 6 M_2

Input: Connected Graph G

Certificate: Connected graphs G_1, G_2 and a certificate C of reliability factorisation for G

1: $k \leftarrow$ length of C
2: $E_0 \leftarrow \Pi(G, p)$
3: **for** $i \leftarrow 1$ to k **do**
4:    **apply** C_i to E_{i-1}
5:    **get** E_i
6:    $i \leftarrow i + 1$
7: **end for**
8: **if** $E_k = \Pi(G_1, p)\Pi(G_2, p)$ **then**
9:    **return** accept
10: **else**
11:    **return** reject
12: **end if**

In Algorithm M_2, Steps **(1)** and **(2)** include two assignments which take time 2. Given that m is the size of graph G, it take time at most $cm \cdot k$ to perform the sequence of certificate steps in Steps **(3)-(7)** where cn is the upper bound on the time taken to get the next expression by applying a certificate step given a constant c. Step **(8)** takes time 1 to confirm that the final expression is $\Pi(G_1, p)\Pi(G_2, p)$. Thus, Algorithm M_2 takes time at most $cm \cdot k + 2$.

Algorithm 7 M_3

Input: Connected Graph G

Certificate: Connected graphs G_1, G_2

1: $k \leftarrow$ upper bound on the certificate lengths
2: $E_0 \leftarrow \Pi(G, p)$
3: $t \leftarrow 0$
4: **while** $t \leq k - 1$ **do**
5:     **for all** E_t^i in E_t **do**
6:         **apply** each possible operation on E_t^i
7:     **end for**
8:     **get** an expression list E_{t+1}
9:     **for all** $E_t^j \in E_t$ **do**
10:         **if** $E_t^j = \Pi(G_1, p)\Pi(G_2, p)$ **then**
11:             **return** accept
12:         **end if**
13:     **end for**
14:     $t \leftarrow t + 1$
15: **end while**
16: **return** reject

In Algorithm M_3, Steps **(1)-(3)** take time 3 for three assignments. If m is the size of graph G and i is the upper bound on the number of operations that can be applied to an expression, it takes time at most $i \cdot cm + i^2 \cdot cm + ... + i^k \cdot cm = cm \cdot \frac{i(i^k-1)}{(i-1)}$ to check if the expression is $\Pi(G_1, p)\Pi(G_2, p)$ by going through all possible operations in Steps **(4)-(15)** where cm is the upper bound on the time taken to get the next expression by applying a certificate step given c is a constant. Thus, Algorithm M_3 takes time at most $cm \cdot \frac{i(i^k-1)}{(i-1)} + 3$.

If the upper bound on the lengths of certificates of reliability factorisations were a constant, then Algorithm M_3 would run in polynomial time. The total number of all certificates would be a polynomial in this case. Thus, the complexity class of **Reliability Factorisation** would be P because this problem were polynomial time solvable.

If the upper bound on the lengths of certificates of reliability factorisations were a polynomial, then Algorithm M_2 would run in polynomial time. Thus, the complexity class of **Reliability Factorisation** would be NP because this problem were polynomial time verifiable.

If the upper bound on the lengths of certificates of reliability factorisations were exponential, **Reliability Factorisation** would neither be verified in polynomial time by Algorithm M_2 or solved in polynomial time by Algorithm M_3. The complexity class of **Reliability Factorisation** would be $NP^{\#P}$ as the result from Algorithm M_1.

The current known upper bound on the lengths of certificates of reliability factorisations is exponential. This research find some short certificates illustrated in Section 5.2.3. Further research may investigate if there exists a better upper bound on the lengths of certificates of reliability factorisations.

Chapter 7

Conclusion

7.1 Results

This research investigates reliability factorisations for all non-separable graphs of size at most 13. Prior to this, the only known graphs that have reliability factorisations were separable graphs. We compute the reliability polynomials of all connected graphs of size at most 13. Then we identify 581 reliability factorisations of non-separable graphs by an exhaustive search over all reliability polynomials of connected graphs of size at most 13. We also find a reliability factorisation of an infinite family of graphs $\theta_{1,d,2d+2}$ for $d \in \mathbb{Z}^+$.

We extend the concept of certificates introduced by Morgan and Farr (2009b) to explain reliability equivalence and reliability factorisation. We define twelve certificate steps to construct certificates. We give certificates for all reliability factorisations of non-separable graphs of size at most 8. We also give a certificate of reliability factorisation for an infinite family of graphs $\theta_{1,d,2d+2}$ for $d \in \mathbb{Z}^+$ by mathematical induction.

The upper bound of certificates could be exponential while the lengths of certificates given in this research are remarkably short. As computing reliability polynomials is hard, we discuss the relationship between the complexity of the problem **Reliability Factorisation** and the upper bound on the lengths of certificates of reliability factorisation.

7.2 Further Work

Further research may investigate how the number of reliability factorisations varies with size of graphs. This may real a relationship between the size of graphs and the reliability factorisation of graphs. Some more infinite families may be investigated such as other patterns of θ-graphs. It is of interest if there exists a better upper bound on the lengths of certificates of reliability factorisations. Compared with the complexity of computing reliability polynomials, certificates of reliability factorisation tend to be a better approach to decide whether a reliability polynomial has a reliability factorisation if the length is short.

The Lookup method used to compute reliability polynomials in this research could be improved by implementing a hash table to store the processed graphs. We could extend the maximum size of input graphs by using the improved Lookup method. The search algorithm for reliability factorisations could be improved by looking for reliability polynomials of separable graphs rather than giving all possible combinations of factors of reliability polynomials.

Bibliography

Ball, M., Colbourn, C. and Provan, J. (1995). *Network Reliability*, Handbook of Operations Research, Elsevier North-Holland, Cambridge, chapter 11, pp. 673–762.

Beichl, I., Harris, D. and Sullivan, F. (2011). Linear algebra and sequential importance sampling for network reliability, *in* S. Jain, R. Creasey, J. Himmelspach, K. White and M. Fu (eds), *Proc. 2011 Winter Simulation Conf.*, pp. 3339–3347.

Bertram, E. and Horá, P. (1999). Some applications of graph theory to other parts of mathematics, *Math. Intelligencer* **21**(3): 6–11.

Brown, J. I. and Colbourn, C. J. (1986). A combinatorial study of the reliability polynomial, *Proc. Sixteenth Manitoba Conf. Num. Math. and Computing*, Vol. 56 of *Congressus Numeratium*, pp. 71–89.

Brown, J. I. and Colbourn, C. J. (1992). Roots of the reliability polynomial, *SIAM J. Disc. Math.* **5**(4): 571–585.

Chang, M. K. and Satyanarayana, A. (1983). Network reliability and the factoring theorem, *Networks* **13**(1): 107–120.

Chang, S. C. and Shrock, R. (2003). Reliability polynomials and their asymptotic limits for families of graphs, *J. Stat. Phys.* **112**: 1019–1077.

Colbourn, C. J. (1997). Reliability polynomials: A survey, *J. Comb. Inf. Syst. Sci.* **22**: 177–193.

Crama, Y. and Hammer, P. L. (2011). *Boolean Functions: Theory, Algorithms and Applications*, Cambridge University Press, New York.

Diestel, R. (2000). *Graph Theory*, Graduate Texts in Mathematics, second edn, Springer, New York.

Evans, M. R., Majumdar, S. N. and Zia, R. K. P. (2006). Factorized steady states in mass transport models on an arbitrary graph, *J. Phys. A: Math. Gen.* **39**: 4859–4873.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability : a Guide to the Theory of NP-completeness*, W. H. Freeman, New York.

Graver, J. and Sobel, M. (2005). Graph theory and reliability: You may rely on the reliability polynomial for much more than you might think, *Communications in Statistics–Theory and Methods* **34**: 1411–1422.

Konak, A., Srivaree-Ratana, C. and Smith, A. E. (2002). Estimation of all-terminal network reliability using an artificial neural network, *Computers and Operations Research* **29**: 849–868.

Maple, version `15` (2011). Copyright (c) Maplesoft, a division of Waterloo Maple Inc.

McKay, B. D. (2009). Nauty, version `2.4(r2)`. Available from `http://cs.anu.edu.au/~bdm/nauty/` [accessed July, 2012].

Moore, E. F. and Shannon, C. E. (1956). Reliable circuits using reliable relays, *J. Franklin Inst.* **262**: 191–208.

Morgan, K. J. (2010). *Algebraic Aspects of the Chromatic Polynomial*, Monash University, Melbourne.

Morgan, K. J. and Farr, G. E. (2009a). Certificates of factorisation for a class of triangle-free graphs, *Electron. J. Comb.* **16**: Research Paper R75.

Morgan, K. J. and Farr, G. E. (2009b). Certificates of factorisation for chromatic polynomials, *Electron. J. Comb.* **16**: Research Paper R74.

Newman, M., Watts, D. and Strogatz, S. (2002). Random graph models of social networks, *Proc. Nat. Acad. Sci. USA of America* **99**(Supplementary 1): 2566–2572.

Page, L. B. and Perry, J. E. (1994). Reliability polynomials and link importance in networks, *IEEE T. Reliab.* **43**(1): 51–58.

PARI/GP, version `2.3.0` (2006). Available from `http://pari.math.u-bordeaux.fr/` [accessed April, 2012].

Pirzada, S. and Dharwadker, A. (2007). Applications of graph theory, *J. Korea SIAM.* **11**(4): 19–38.

Read, R. (1987). Connectivity and chromatic uniqueness, *Ars Combinatoria* **23**: 209–218.

Read, R. C. (1968). An introduction to chromatic polynomials, *J. Combin. Theory* **4**: 52–73.

Read, R. C. and Tutte, W. T. (1988). Chromatic polynomials, *in* L. W. Beineke and R. J. Wilson (eds), *Selected Topics in Graph Theory*, Vol. 3, Academic Press, London, pp. 15–42.

Royle, G. and Sokal, A. D. (2004). The Brown-Colbourn conjecture on zeros of reliability polynomials is false, *J. Comb. Theory B* **91**: 345–360.

Slyke, R. V. and Frank, H. (1971). Network reliability analysis: Part I, *Networks* **1**: 279–290.

Tutte, W. (1972). Chromials, *in* C. Berge and D. Ray-Chaudhuri (eds), *Hypergraph Seminar*, Vol. 411 of *Lecture Notes in Mathematics*, Springer-Verlag, Berlin, pp. 243–266.

Wanger, D. G. (2000). Zeros of reliability polynomials and f-vectors of matroids, *Combin. Probab. Comput.* **9**: 167–190.

Welsh, D. J. A. (1993). Complexity: Knots, colourings, and countings, number 186 in *London Mathematical Society Lecture Note Series*, Cambridge University Press, Cambridge.