A study of the pleiotropy versus redundancy trade-off, using evolutionary computation

Zhiyang Ong [*]

The University of Adelaide
School of Electrical & Electronic Engineering
HONOURS PROJECT REPORT

Commencement Date: 26 July 2004
Submission Date: 12 May 2005

---

[*] Email: zhiyang@ieee.org

THE UNIVERSITY OF ADELAIDE

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

ADELAIDE, SOUTH AUSTRALIA 5005

A study of the pleiotropy versus redundancy trade-off, using evolutionary computation

B.E. in Electrical and Electronic Engineering

ELEC ENG 4039 A/B HONOURS PROJECT

Each student at Level IV of the course in Electrical and Electronic Engineering, Computer Systems Engineering, and Information Technology and Telecommunications is required to complete this subject. The subject involves approximately 240 hours of project work over the whole academic year. Students are assessed on their performance in the project, a written proposal, this written report, a technical paper, and two seminar presentations.

Supervisor: Associate Professor Derek Abbott (dabbott@eleceng.adelaide.edu.au)

Signature of Supervisor:

Acknowledgments

The author would like to thank his supervisors, Associate Professor Derek Abbott and Mr Matthew Berryman, for their invaluable advice. They encouraged the author to persevere in times of difficulties, whilst noting that a trivial problem is not worth researching about. They also rendered suggestions on improving the *NetSim* software simulator, which the trade-off between between pleiotropy and redundancy in the generic, optimized telecommunications network can be determined from. In addition, they had on countless occasions directed the author to information resources and introduced the author to world renown experts so that the author is able to gather more background information. Consequently, the author was able to seek a better appreciation for this project from a multidisciplinary perception.

He would also like to thank Professor Zbigniew Michalewicz, Dr Matthew Roughan, and Mr Alex Ryan for setting time aside to speak to the author about their research. Professor Michalewicz discussed the state-of-the-art techniques used in evolutionary computation with the author and his supervisors. Dr Roughan provided insights to the author about the idea of telecommunication network design as a optimization problem whilst Mr Ryan gave handy pointers on software simulation in complex systems research and ant colony optimization. The author had also sought the advice of Mr Andrew Allison on the measurement of the resistance of a node in an electrical network using a SPICE program.

The author also wishes to express his gratitude to his project partner Andy Hao-Wei Lo for his valuable collaboration in developing *NetSim* whilst working on this project. He has rendered substantial assistance in the design of *NetSim*'s software architecture; his impromptu tutorial sessions on shortest path graph traversal algorithms, and graphical user interface (GUI) development and multithreading on the *Java* platform has been enormously beneficial to the author's learning experience. He also advised the author on the subtleness of *Java*'s syntax and semantics during code walkthroughs and pair programming sessions. Moreover, he has provided a sounding board for some of suggested software engineering practices, such as automated testing and pair programming. Consequently, the author has been left in awe of the impressive social capital and intellectual virtuosity of Prof. Abbott, Mr Berryman, and Mr Lo. They were able to weave myriad multidisciplinary concepts together, and inspire the author to seek the thrill of exploring the trade-off between pleiotropy and redundancy in telecommunication networks with them.

Next, the author would like to thank the School of Electrical and Electronic Engineering, University of Adelaide, for supporting and funding this work. In particular, he treasures the prompt and effective computer support provided by Mr David Bowler and Mr Greg Pullman. The author also wishes to thank Mr Mohammad Moore, whom is a system administrator with the School of Computer Science, University of Adelaide, for extending his help and providing advice on matters relating to managing your own

server, and using *Subversion* and *Concurrent Versions System* for configuration management on the Computer Science department's servers. The author would also like to thank the South Australian Partnership of Advance Computing (SAPAC), Dr Paul Coddington, and Mr Patrick Fitzhenry for allowing us to run our simulations on one of their Linux clusters, and their assistance in helping us to dispatch jobs for high performance computing.

The author appreciates the encouragement from his friends, whom he has spoken to about this project, and their expression of interest in the relevance of this project to their area of research or expertise that ranges from gene expression analysis and formal methodology to game theory and cultural anthropology. In particular, the author recognizes the contribution of opinions by Mr David Knight, Nikolay Stoimenov, and Joshua Carr. Mr Knight and Nikolay provided their perception on evolutionary computation for network optimization whilst Joshua queried about the relevance of this study of network optimization to *in vivo* networks of nano-scale biocompatible robots that can be used for medical diagnosis and treatment.

Mr Knight, along with Mr Heiko Maurer, Nikolay, and Tishampati Dhar, had also provided insights to the author about academic research, good software development process models, software architecture, and automated testing. The author had also sought the advice of Mr Knight and Mr Matthew Lowry on determining a minimum spanning tree using Dijkstra's algorithm. Mr Lowry mentioned that the shortest path for any node in a graph can be determined from the minimum spanning tree, and the comparison of the initial adjacency matrix with the adjacency matrix obtained from applying Dijkstra's algorithm to every node would yield the minimum spanning tree. However, Mr Knight has advised the author that other algorithms such as Kruskal and Prim-Jarník will yield the same results with much better time order complexity.

The author would like to thank Professor Lynn Margulis from the University of Massachusetts, Amherst, for providing the author a copy of one of her earlier publications on the role of symbiosis in evolution. He would also like to thank Professor Tom Lenaerts from Université Libre de Bruxelles for providing a copy of his Ph.D. dissertation on selection methods and modeling interactions between chromosomes in a genetic algorithm. Along with Professor Michalewicz, Asiri Wijenayaka, Chee-Yong Pek, and Yiu-Ming Khor, they have encouraged and inspired the author to model biological phenomena such as Darwinian evolution and symbiosis in his software simulator.

Craig Eales has provided useful information to the author about GUI development using *Java* Swing. In addition, Luke Nuttall has counseled the author on plotting the results from the computer simulation to compare the performance of the fittest chromosome against the average and/or median performance of the population of chromosomes. The author would also like to thank Jane Bekesi and Sing-Yew Foo for their assistance in defining "aggregated market capitalization." Joseph De Luca and Shyalle Kahawita have also rendered advice about mutualism and how the amount of benefits gained by each entity from the relationship does not matter. The author sought assistance from Alfred Yapp and Anusha Surrender for a word or phrase for describing twins, triplets, quadruplets, and so forth; Alfred came up with the phrases identical siblings, and monozygotic multiple births.

Lastly, the author has also received assistance from Stuart Ekberg, for comments and helping the author to find a paper about social networks from an anthropology journal, and Ms June Field, whom helped the author determine the editor for the 1998 issue of Lincoln College's Stag magazine. The efforts of them helped Andy and the author in their feasibility study about the connectivity of individuals in scientific and engineering collaborations, and in romantic relationships of young adults.

Executive Summary

Since the late 1980s, evolutionary strategies have been used widely in optimization. Not only have evolutionary strategies been employed in the study of sociology, economics, and building construction, they have also been used for communication network optimization. Using recent studies by our predecessors about the trade-off between pleiotropy and redundancy, this report will investigate the trade-off between pleiotropy and redundancy as the telecommunication networks are optimized to improve their connectivity. Several key issues regarding the choice of cost functions and techniques in evolutionary computation will be discussed, and a future research agenda is outlined. Experimental results indicate that pleiotropy and redundancy do not vary significantly, as expected, with evolving networks. This is due to the controlled evolution of networks that is modelled by the steady-state evolutionary strategy; change in telecommunications networks that occur drastically over a very short period of time are rare.

Contents

List of Figures

List of Tables

1 Introduction

An investigation of the trade-off between pleiotropy and redundancy in a generic, optimized telecommunications network is carried out and analyzed with Andy Hao-Wei Lo. This work is a continuation of the research work done by the author's predecessors, and seeks to optimize the telecommunications network using evolutionary computation and lowly correlated cost functions. The motivation and analysis of the problem is provided as follows in this section, followed by a recapitulation of previous work, literature review, and the objectives of this research project.

1.1 Motivation

People desire, need, and want to communicate aurally, visually, and verbally over large distances, and are willing to pay for that. With the advent of interconnection between the different types of communication networks, a seamless transition between different communications services is made available to all telecommunication customers to attempt to satisfy this insatiable need. People would then be able to switch between different telecommunication networks easily and communicate via electronic mail, phone calls, short message service (SMS), multimedia messaging service (MMS), electronic file transfers, and/or video-conferencing [1]. The Internet is an example of a global communication system that has grown exponentially over the last two decades due to the demand for communication over relatively long distances. By 2003, the number of computers attached to the Internet is estimated to be about 160 million [2].

From January 2003 to December 2004, the market value of the worldwide telecommunications industry increased by 28%. On 31 December, 2004, the aggregate market capitalization of these companies is US$2,349 billion [3]. In Australia, Telstra Corporation Limited had a total operating revenue, excluding interest revenue, of AUD$21,280 million for the fiscal year that ended on 30 June, 2004 [4]. Given this large operating cost of Telstra and the market value of the telecommunications industry, there is a worldwide demand for a continual improvement in the topology of telecommunication networks to keep installation, operating, and maintenance costs low.

The pleiotropy of a server in a telecommunications network is defined as the number of clients that it can service whilst the redundancy is described as the number of servers servicing a client [5, 6]. The servers or genetic agents/devices are satellites, routers, switches for telecommunications exchange, antennas, access points for wireless networks, and computers. Complementing the servers are the clients, which are mobile phones, telephones, fax machines, and computers. The words server, clients, agents, and components are used snonymously. Similarly, the service rendered by a server for a client is the function or task of the server component. A client-server network with a mixture of pleiotropy and redundancy can be seen in Figure 1. This indicates good robustness in the system; if server A or B fails, the set of clients {1,2,3} or {2,3,4} can still function with

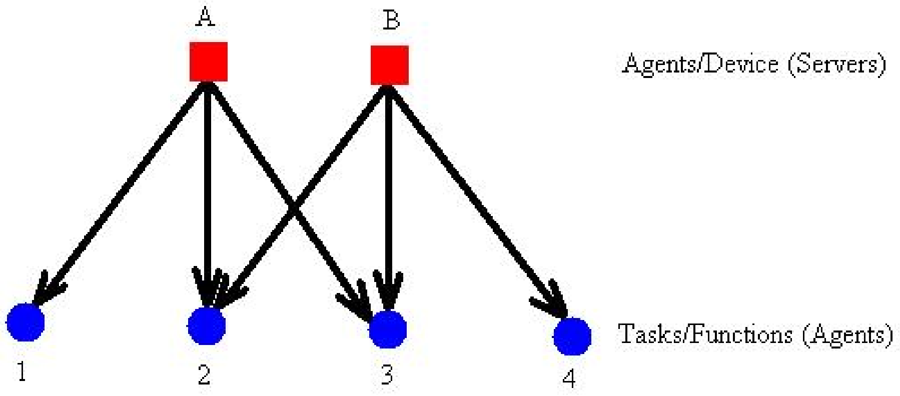


Figure 1: A simple client-server network topology indicating the pleiotropy and redundancy of the servers, which leads to a network topology of good robustness and low costs

the service of the other. On the other hand, allowing the server to serve more than one client reduces the costs of installing expensive servers. Consequently, network topologies can be optimized so that the network can be both robust and have low setup or runtime costs.

An abundance of methods exist for network optimization; hence, a discussion of several candidates for use in our simulation are provided as follows. Monte-Carlo simulations for optimizing telecommunication networks are considered to be inappropriate by the author since there are no uncertain variables that the author seeks to optimize [7]. Each objective function used in the network optimization is deterministic for any given network. The exclusion of random variables in these cost functions renders Monte-Carlo simulations to be unsuitable for the optimization of a telecommunications network. The author rejected the use of dynamic programming tsince the growth of the telecommunications network is not a sequential process [7, 8]. Phone calls are made at random times for various durations [9]. In addition, servers are occasionally replaced, repaired, removed from, or added to the network due to the large investment costs in installing and maintaining these servers and data links. Therefore, it is incorrect to model the transition of a telecommunications network from one stage to the next for network optimization as these networks do not undergo deterministic sequential transformation.

When complex linear programming algorithms, which can be solved with contemporary fast personal computers, are used to optimize multi-objective problems, the solution space may not have been adequately searched to determine the optimum solution [10]. This could be partly due to the conversion of multiple objectives into a weighted sum so that the linear programming algorithms can determine the optimum solution & efficiently [11]. The best possible optimizing solution for the telecommunications network is the global minimum or maximum. However, most nonlinear programming algorithms yield solutions that are local minima or maxima [7, 8, 12]. As a result, alternate local minima or maxima solutions of the telecommunications network must be compared with each other to determine the best possible global solution. In addition, although data can be obtained from telecommunication service providers for specific networks to formulate a nonlinear function representing the multiple objectives of the network optimization process, the author concedes that it is difficult to determine the nonlinear function.

Gradient descent algorithms are used to find the global minimum of a real-valued differentiable function. This requires a formulation of a multi-dimensional function that best describes the interrelation between the various objectives for the optimization of the network [12]. As in the case of nonlinear programming, the determination of the function is challenging. In addition, gradient descent algorithms cannot not capture the growth of the telecommunications network. This is because a changing network topology would require frequent restarting of the gradient descent, and may yield a local minimum instead of a global minimum. Neural networks are also not considered since the desired outputs of each objective or the desired output of a function representing these objectives in the network optimization are not known [12].

Evolutionary computation, in particular evolutionary strategies, are used since their cost functions can be easily modified to model the incremental growth of the telecommunication networks as an organic process [5, 6, 13, 14]. It allows us to capture the dynamic of the telecommunications network as data links, servers, and clients are added to the network, removed from it, fail, or get repaired. It also allows us to model the routing of bursts in data packet traffic, which are asynchronous and apparently random. Thus, the modification of the problem specifications, constraints, and/or objective functions do not require the optimization process to be restarted as evolutionary strategies can adapt to the changes. Moreover, evolutionary algorithms search for a solution from a diverse set of solution space, as opposed to searching from a starting point [15, 16].

Since a guaranteed optimal solution is not provided by evolutionary strategies, evolutionary strategies are used as a tool in the design process as opposed to a tool for determining the optimum topology of a telecommunications network [17]. This is because evolutionary computation is heuristic, and any set of selected cost functions is only an approximation to the telecommunication network provider's preferences. It is difficult to accurately model all objectives in the evolutionary algorithm since subtle attractive qualities of the network are difficult to formulate. Consequently, the solution obtained from optimizing a population of telecommunications using evolutionary computation is not a unique solution, and there is as much justification for the use of heuristic algorithms as exact optimization methods [11].

In addition, there does not exist any general purpose optimization technique that is better than all others, and the selection of appropriate optimization techniques and objective functions are problem specific [18]. This is attributed to the No Free Lunch (NFL) theorem, which states that all possible optimization techniques have the same performance on average upon evaluation of all possible objective functions.

The author notes the existence of other methods to optimize generic or telecommunications networks. However, a detailed analysis of these methodologies is beyond the scope of this research project.

Hence, the author intends to use evolutionary computation to optimize a generic telecommunications network using multi-objectives that a telecommunication service provider may consider employing. This enables the author to model the network design process more accurately, where several conflicting goals of network optimization are not rare. For example, the telecommunication service provider may desire the network to be very reliable, provide multimedia content at high data rates, cover a large geographical area, and provide telecommunication services at affordable prices [19]. It is desired that the outcomes of this ongoing research can facilitate the reduction of the installation, operating, and maintenance costs of the telecommunication service provider. Generic telecommunication networks within a layer in a network reference model is considered for optimization since it is difficult to obtain data for any specific network, as such information are considered proprietary to telecommunication service providers. Examples of network reference models include the International Organization for Standardization's (ISO) open systems interconnection (OSI) seven-layer communication architecture reference model and Transmission Control Protocol/Internet Protocol (TCP/IP) network architecture [2,9,20,21].

Finally, the connections and disconnections between servers and clients are part of a distributed, asynchronous, complex, and uninterrupted process that appears to be random to external observers [22]. These connections and disconnections will influence the pleiotropy and redundancy of the telecommunications network and hence their trade-off. By appropriately modeling the dynamics of links addition and removal for different types of telecommunication networks, substantial advancements in scientific network research can be made. Subsequently, commercial firms in the telecommunications industry may profit from these by reducing their operating costs. Moreover, the abstraction of common features between different types of networks will also enhance research about biological systems, social-economic networks, and integrated circuit design.

1.2 Previous Studies and Literature Survey

To the author's knowledge, there has hardly been any work done on studying the trade-off between pleiotropy and redundancy in telecommunication networks. The work of our predecessors included the use of the following fitness function F [5,6]:

$$F = \frac{R_e}{C_o} \tag{1}$$

where F is the fitness function of the telecommunications network, R_e is its reliability function, and C_o is its cost function. The flaw with this fitness function is that when costs are minimized and the reliability of the network is maximized, the fitness of the telecommunications network will be infinite. Hence, a new objective function for the telecommunications network needs to be obtained.

The reliability of a telecommunications network cannot be determined by additively adding up the reliability of each client and server in the network. This is due to their complex interactions, which may cause problems in terms of the network's susceptibility to overloads [21, 23]. For example, bursts of data packet traffic during peak hours may lead to short-term resource overloads on the network if a poorly implemented routing protocol keeps transmitting error or incorrectly directed messages between a subset of nodes in the network. Conversely, people have the ability to predict and determine the reliability of the telecommunications network without much knowledge of the reliability of each data link, client, or server. Consequently, contemporary complex systems research is important to this study; it can be used to optimize the telecommunications network in terms of cost and reliability, and study its trade-off between pleiotropy and redundancy.

1.3 Objectives

The aims of the project is to investigate the trade-off between pleiotropy and redundancy in a telecommunications network as it evolves under the optimization process, using steady-state evolutionary strategy [24]. Being a continuation of the work done by our predecessors, this study will focus on developing multi-objective optimization techniques to improve the model of the telecommunications network, and enhance the genetic operators involved. The evolution of telecommunication networks will be subjected to a variety of constraints and conflicting objectives.

2 Theory and Proposed Approach

In this section, a formulation for optimizing a generic telecommunications network in generalized goal programming terms and an evolutionary strategy are provided. These will be used to determine the best possible solution based on the selected cost functions for this multiobjective optimization problem.

2.1 Steady-state Genetic Algorithm

The steady-state genetic algorithm is selected for the optimization process since it does not cause rapid changes in population diversity or average fitness value of the population. This is because it generates a fixed number of offsprings each generation, and only replaces a constant small number of chromosomes each generation. Consequently, this requires a large population to provide a diverse search space. A population of chromosomes is initialized at the start of the evolutionary strategy, and a random number of edges is added at random to each chromosome. As the population evolves, edges are added to or removed from each selected chromosome after the mating process and during mutation. At each generation, tournament selection is carried out only once to speed up the evolutionary process. A pseudo-random generated number will be compared with the probability of crossover to determine if the selected chromosomes should mate. If the former is smaller, the pair shall mate and produce one or two offsprings [17].

Subsequently, another pseudo-random generated number will be compared with the probability of symbiosis. If the former is smaller, these fitter chromosomes, or their offsprings if any were born, will interact and cooperate with each other. Finally, a pseudo-random generated number will be compared with the probability of mutation to determine if these chromosomes should mutate. Once again, they will mutate if the third pseudo-random generated number is smaller. The resultant mutants are then inserted back into the population via replacement of the chromosomes that have fitness values closest to theirs [17, 25].

This ensures that unconnected nodes during the initial generations of the evolution will not remain disconnected from the network permanently [22]. Using cellular telephone systems as an example, the connection of individuals in remote areas to base stations in regional macrocells can be modeled as linking isolated individuals to the communications network [9, 20, 21, 26, 27]. As people need to obtain medical aid, and food and fuel supplies, or to trade, they need to communicate from remote areas. Hence, the existence of solitary individuals/group of individuals that may be represented as isolated client nodes in a telecommunication network is negligible.

The terms individuals, chromosomes, graphs, and telecommunication networks are used interchangeably. Similarly, the terms set and population of chromosomes are used synonymously. This is a consequence of modeling a telecommunications network as a directed graph G=(N_{set}, E_{set}), where N_{set} is the set of nodes and E_{set} is the set of directed edges. with its [28]. The nodes denote the geographical locations of the clients and servers in the network whilst the edges represent transmission, or data, links between clients and servers/other clients.

As aforementioned, pseudo random generated numbers are used to create random edges in each network, where the number of edges for each network is randomly generated. Since these chromosomes are created and modified in a pseudo-random manner, the author expects that the number of existing monozygotic multiple births and identical twins during the evolutionary process to be negligible. That is, the probability of creating two networks of the same topology during the population initialization process is considered to be negligible by the author.

The selection and reproduction processes, and variation of networks to improve population diversity models the three fundamental principles of Darwinian evolution in this stochastic optimization approach [15, 16, 29]. Symbiosis allows the modeling of interaction between .... Other evolutionary algorithms were not considered for the following reasons: Evolutionary programming does not involve recombination since different species are evolved under organic evolution, and different species do not naturally mate with each other [30].

Evolution strategies (ESs) use real valued object or decision variables to model individuals and phenotypes, and have mutation rates that self-adapt to speed up the search for optimal solutions [30]. The variants of ES include the following: (1+1)-ES that evolves a basis of one parent and one descendent per generation, $(\mu + 1)$-ES that replaces the the worst parent in the population of μ chromosomes with an offspring, if the offspring is fitter, $(\mu + \lambda)$-ES that form the parents of the next generation by selecting the best μ individuals from μ parents and λ descendants, and (μ, λ)-ES that forms the new parent set with the μ best offsprings [31, 32]. Here, μ refers to the number of parents in a generation and λ refers to the number of descendants resulting from applying the genetic operators on the parents. (1+1)-ES does not optimize the telecommunication network with a large search space; similarly, $(\mu + 1)$-ES favors elitism and does not promote population diversity. On the other hand, $(\mu+\lambda)$-ES searches through a large and diverse population but often tends to focus on local search properties. It also allows misadapted ES parameters to last for a significant number of generations. (μ, λ)-ES does not allow fit parents to be retained in the population.

2.2 Implementation of Genetic Operators

A description for the basic models of biological phenomena employed in the optimization of telecommunication networks within a network layer is provided [30].

2.2.1 Methods of selection

Rank method of selection was first considered for the selection of chromosomes from the population for mating and mutation [33]. A suggested means of determining the probability of selecting the chromosome, $P(i)$, ranked i^{th} in the sorted population is:

$$P(i) = \frac{P(1) \cdot (1 - P(1))^{i-1}}{1 - (1 - P(1))^{Pop}}, i \neq 1 \tag{2}$$

where $P(i)$ is the probability of selecting the chromosome that is ranked i^{th} in the sorted population, and Pop is the size of the population. The value of $P(1)$ has to be determined before $P(i)$ can be determined; $P(1)$ can be obtained as follows:

$$P(1) = 1 - P(Pop)^{\frac{1}{Pop}} \tag{3}$$

where $P(Pop)$ is the probability of selecting the chromosome with the worst fitness. This values is determined by the user. A drawback for this method is that if the population size is reasonably large, for example, $Pop = 100$, $P(1)$ would be very close to one and $P(Pop)$ is negligible. If a pseudo-random number generated from a uniform or normal distribution, the chances of selecting the fittest chromosome would be very low. This prevents the fittest group of chromosome from being selected for mating and mutation, which leads to a poor model of Darwinian evolution [34].

Roulette Wheel selection assigns the probability of each individual chromosome being selected based on its fitness [16]. This leads to a robust but computationally expensive operation to sort the population, and maintain its sorted order. Normalizing the fitness values of each cost function can prevent premature convergence at an optimal solution. Elitist strategies were avoided to prevent the population from converging too quickly at a solution, and allow the search space to be more diverse [17].

Tournament selection was used to select a pair of chromosomes from the population for the possibility of mating, symbiosis interaction, and mutation. A pair of chromosomes are picked at random using *Java*'s pseudo random number generator that follows a uniform distribution. The fitter chromosome of the pair is kept for genetic modification via mating and mutation. Another pair of chromosomes is selected at random, and the fitter chromosome is picked again to join the other fitter chromosome to form the mating pool. The selective pressure, which is measured by the difference between the average fitness of the mating pool and the average population fitness, can be adjusted with the number of chromosomes selected from the population (tournament size). This allows the selection operator to be more efficient and robust whilst overcoming the problems associated with scaling the fitness of each chromosome for every cost function [16, 17, 29]. Tournament selection is also an efficient procedure for implementation in both non-parallel and parallel architectures.

2.2.2 Mating (Crossover)

To get two chromosomes to mate, two crossover points are selected within the width of the shorter chromosome, if they differ in length. The edges of the nodes lying in between those two crossover points are swapped between the two chromosomes. If the chromosomes have the same outgoing edge at a node between the two crossover points, the edge is not copied over [25].

2.2.3 Mutation

Chromosome selected for mutation undergo the following process: A pseudo-random number is generated to determine the number of edges in a network that will be affected. For each affected edge, another pseudo random number will be generated to determine if an edge will be removed from or added to the network [25]. If the second pseudo random number is greater than a threshold, an edge will be added. Else, an edge will be removed.

2.3 Metrics for measuring Pleiotropy and Redundancy

Metrics for measuring pleiotropy and redundancy are provided as follows:

The metric used for measuring the pleiotropy of a server is:

$$P_{leio} = \frac{\sum O_{s,c}}{|S|} \tag{4}$$

where $\sum O_{s,c}$ is the outgoing edge of a server with a client node as its destination node and $|C|$ is the number of clients in the telecommunications network.

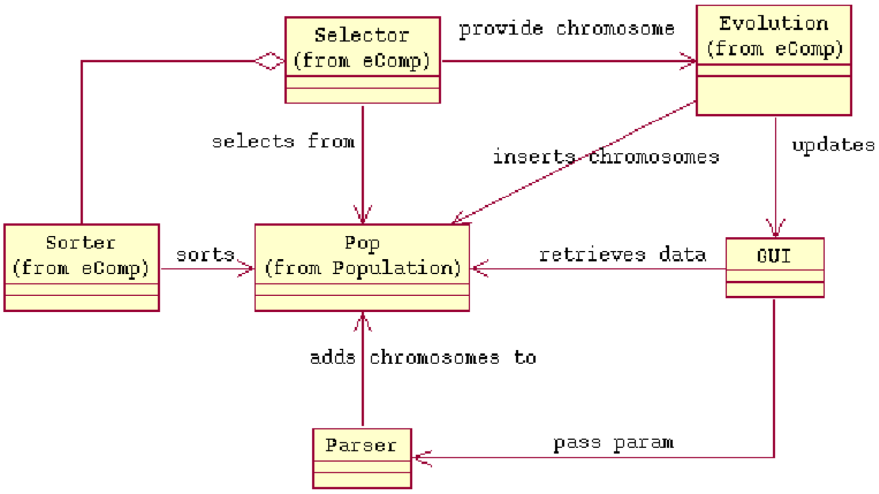


Figure 2: Proposed software architectural design; it is represented using the Unified Modeling Language (UML) notation

The redundancy of a server can be measured as:

$$R_{edun} = \frac{\sum I_{c,s}}{|C|} \tag{5}$$

where $\sum I_{c,s}$ is the incoming edge of a server with a client node as its source node and $|S|$ number of servers in the telecommunications network

2.4 Proposed Software Architecture

Figure 2 indicates the software architecture of the *NetSim* software that is used to optimize a generic telecommunications network using evolutionary computation [24]. It consist of three main packages: graphical user interface (*GUI*), *eComp*, and *Population*. The *GUI*, which the user operates the program with, includes a *Parser* that reads input from the user via the *GUI* and processes the input to create a set, or population, of generic telecommunication networks. This population is indicated as *Pop* in the figure. Next, the package *eComp* is responsible for performing the bulk of the computation, which includes simulating evolution. Its *Java* Classes include *Selector*, which simulates the natural selection of individuals from the sorted population, *Sorter* that sorts the population and maintains the resultant sorted order, and *Evolution* that models the mating and mutation of individuals and offsprings, and inserts them back into the population.

Finally, the *Population* package includes the class *Pop* and another class that represents the chromosomes, which are potential solutions to this optimization process. In each chromosome, the topology of the network can be represented as an adjacency matrix if only one or none directed edge is allowed to be connected from node A to node B. That is, the number of directed edges going from A to B is either one or zero, and the number of directed edges going from B to A can be one or zero. This is because parallel edges cannot be represented in adjacency matrices [35]. By using adjacency matrices, storage space in memory can be saved by storing numerical values that represent the edge cost connecting any two nodes instead of Node objects. The Node objects belong to a package that models the network topology as a graph; this package is not depicted in Figure 2. However, savings in memory storage comes at the expense of increased complexity and computation time. This bodes well since the focus is on meeting the constraints in the memory capacities of the computers that the simulations are run on, as opposed to the duration of simulations.

2.5 Proposed Fitness Function

Discuss our original proposed approach, and then discuss their flaws...

The author proposed two initial dual-objective fitness functions [24]; the first fitness function is:

$$F = \mu \cdot R_e - \nu \cdot C_o \tag{6}$$

where R_e and C_o are the reliability and cost functions of the telecommunications networks. The constant coefficients μ and ν are weights to indicate the contribution of the reliability and cost of the network to the fitness function.

The second proposed fitness function is:

$$F = \eta \cdot (\phi \cdot R_e + \psi \cdot C_o - \chi \cdot R_e \cdot C_o) \tag{7}$$

where R_e and C_o are the reliability and cost functions of the telecommunications networks as before. The $R_e \cdot C_o$ term is used to measure the interaction, or correlation, between the reliability and cost factors. The constant coefficients ϕ, ψ, and χ are weights to indicate the contribution of the of each factor to the fitness function whilst the coefficient η is used to indicate the type of the network.

A proposed measure for the reliability R_e of the telecommunications network is:

$$R_e = E_{working}/m - fail_{avg} \tag{8}$$

where $E_{working}$ and $fail_{avg}$ represent the total number of working data links and average percentage of link failures per generation respectively. Dividing the number of link failures during a generation over the total number of data links will yield the percentage of link failures for that generation. The symbols m and n, not referred to above in Equation (8), refer to the number of edges E and number of nodes N in the graph, see Equations (9) and (10).

$$m = |E| \tag{9}$$

$$n = |N| \tag{10}$$

The reliability of the network can also be determined, using the concept of system reliability, from the reliabilities of servers and clients, their interrelations, and topology of the network [36,37]. This models the probability of network functioning above a specified level of performance based on stochastically failing servers and clients. Since there are no start and end points in the telecommunications network, the network's reliability can be determined by the average reliability for all possible paths in the minimum spanning tree representation of the telecommunications network. Unfortunately, the time order complexity for such a simulation will be very high. For example, if the breath-first search was used [38], the time order complexity will be of order $O(n \cdot m^2)$.

The proposed measure for the cost C_o of the telecommunications network is:

$$C_o = \alpha \cdot fail_{avg} + \varepsilon \cdot (\beta \cdot dist + \gamma \cdot amp + \delta \cdot sat) \tag{11}$$

where *dist*, *amp*, and *sat* represent the total cost of transmitting signals over all links, amount of amplification required in repeaters and/or satellites, and cost of orbiting a satellite around the earth. Once again, the coefficients α, ε, β, γ, and δ are used to weight the contribution of these factors.

Classical design and analysis of experiments can be used to determine the coefficients of these cost functions. The response variables would be the terms on the left-hand side of Equations (6) and (7), whilst the controllable variables will be the factors on the right-hand side of those equations [37,39]. By determining appropriate numbers of levels that these factors can have, a fraction factorial experiment can be used to determine the coefficients of main effects, which are the factors, and their interactions in these additive (Equations (6) and (11)) and nonadditive linear models (Equations (7)).

An alternative means to determine the coefficients is to collect data about a specific type of network. For example, data from wireless local area networks (WLANs) or Global System for Mobile communications (GSM) mobile telephone systems can be collected and averaged over a few similar networks [21]. Subsequently, values for estimating *dist*, *amp*, and *sat* and their coefficients can be obtained.

2.6 Improved Objective Functions

Upon further investigation, Andy Lo and the author realized that a more useful means to approach this problem is to optimize a telecommunications network based on common performance metrics [25]. Consequently, they have proposed the following objective functions, more specifically cost functions, that can be used to optimize the telecommunications network.

The first cost function measures the total cost of all edges in the minimum spanning tree [40]. Dijkstra's algorithm was used to determine the set of shortest paths for each node, where the set of shortest paths for each node was used to construct the minimum spanning tree.

$$C_o = \sum_{i \in mst} E_{mst_i} \tag{12}$$

where mst and E_{mst_i} refer to the set of edges belonging to the minimum spanning tree and the i^{th} edge in this set of edges for the minimum spanning tree.

A telecommunications network installed using a topology that is a minimum spanning tree enables all the client and servers to be connected using the least total cost of data links, which is determined by adding up the cost of each data link in that telecommunications network. Note that costs here refer to the costs of purchasing and installing the data links, and the delay of the data links. The delay of a data link is the duration taken for a packet of data to travel along that link [2]. Since data links are installed in telecommunications network as a wireless communications channel of a specified bandwidth, transmission line, or optical cable, this results in minimizing the installation costs of the network whilst allowing users of the telecommunications network to enjoy a good performance of data transmission [20].

A spanning tree, represented mathematically in the graph G, is an acyclic connected spanning subgraph of G, given the existence of this subgraph. A subgraph of G refers to a graph $G_{sub} = (N_{sub}, E_{sub})$, where N_{sub} and E_{sub} are subsets of N and E respectively. The spanning subgraph $G_{span} = (N, E_{sub})$ is a subgraph with the same set of nodes as G. Hence, the acyclic connected spanning subgraph refers to the spanning subgraph G_{span} that has no cycles, or closed directed paths with repeated nodes; this is also called the spanning tree. Here, a closed path refers to the alternating sequence of nodes and edges from a node to itself. Therefore, the minimum spanning tree is the spanning tree in G, which is weighted and directed, if $\sum E_{mst_i} < \sum E_{sp_j}, \forall\ i \in mst$ and $\forall\ j \in edge_{st}$, for every spanning tree st in G. $\sum E_{sp_j}$ refers to the sum of all edge costs in a spanning tree st, and $edge_{st}$ is the set of edges in st [41].

Given that the shortest path for any node in a graph can be determined from the minimum spanning tree, the minimum spanning tree can be determined as follows: Dijkstra's algorithm is applied to each node in the graph. An adjacency matrix A_{adj} is used to keep the shortest path for each node. The initial adjacency matrix $A_{adj_{initial}}$ of G prior to applying Dijkstra's algorithm is compared with the adjacency matrix $A_{adj_{dijkstra}}$ obtained from applying Dijkstra's algorithm to every node. If a_{ij} in $A_{adj_{initial}}$ is same as that in $A_{adj_{dijkstra}}$, a_{ij} is entered into the adjacency matrix $A_{adj_{mst}}$ of the minimum spanning tree. This is because the minimum spanning tree gives the shortest path from node i to node j. See sub-Section **??** for more details on the implementation of Dijkstra's algorithm.

The second cost function measures the average cost of all shortest paths from node i to node j in the minimum spanning tree, $\forall i$ and $\forall j\ i \neq j$ [40].

$$C_o = \frac{1}{|mst|} \cdot \sum_{k \in mst} E_{mst_k} \tag{13}$$

where $|mst|$ is the number of edges in the minimum spanning tree. Here, the cost of the transmission line represent the propagation delay of the data packets traveling along the data line [2]. Therefore, minimizing the average cost of all shortest paths in the minimal spanning tree allows the destination nodes to receive information faster.

The next cost function measures the total cost of all edges in the graph G.

$$C_o = \sum_{i \in E_{set}} E_i \tag{14}$$

where E_i is the i^{th} edge in the graph, belonging to E_{set}. By reducing the data links in the telecommunications network, the cost of installing and maintaining the transmission links will be reduced. However, this has the trivial solution of evolving a network that has less edges added to it than removed since the lesser the edges, the fitter the network will be.

The fourth cost function measures the average degree of separation between [23, 42].

$$C_o = \frac{1}{|N_{set}|} \cdot \sum_{i \in |N_{set}|} D_{oS} \tag{15}$$

where D_{oS} is the degree of separation between any two nodes; this is the number of links connecting them. By minimizing the average degree of separation, the delay in transmitting packets along paths and the delay of processing data packets at each node will be reduced.

The fifth cost function measures the maximum degree of separation between [23, 42].

$$C_o = max(D_{oS_i}), \ \forall \ i \in E_{set} \tag{16}$$

As before, minimizing the maximum degree of separation shall reduce delay of transmitting data packets.

The sixth cost function measures the load factor between [25].

$$C_o = \sum_{i \in E_{set}} E_i \tag{17}$$

The seventh cost function measures the average clustering coefficient of servers in the telecommunications network [23].

The seventh cost function measures the average clustering coefficient of servers in the telecommunications network [23, 43].

$$C_o = \frac{1}{N_2} \cdot \sum_{i} \frac{2 \cdot E_i}{k_i \cdot (k_i - 1)} \tag{18}$$

where N_2 refers to the number of network nodes with more than or equal to two neighbours, E_i is the number of connections, and k_i is the number of neighbors of the node. The greater the average clustering coefficient of servers, the greater will be their ability to connect to more clients in the network.

The eight cost function measures the resistance of the network [25].

$$C_o = (\sum \frac{1}{path_{ij_k}})^{-1} \tag{19}$$

where $path_{ij_k}$ refers to the k^{th} path between nodes i and j. The lower the resistance, the easier it is for the data packets to be transmitted between nodes in the network.

The adjacency matrix, which need not be symmetric, of a directed graph G is the $n \cdot n$ matrix $A_{adj} = [a_{ij}]$, where a_{ij} represents the cost of the data link connecting node i to node j [35,41]. If there are no data links connecting node i to node j, a_{ij} is infinite. This implies that node i cannot connect to node j. Similarly, a_{ii}, which is any diagonal entries of A_{adj}, is zero. This is because it is not economically viable for the telecommunication service providers to install, operate, and maintain a data link that connects a node to itself since there is no information that can be gained by the node from this data link.

The increased complexity and computation time can be reduced from $\mathrm{O}(n^2 \cdot \log n)$ to $\mathrm{O}(n^2)$ by ignoring calculations involving infinity. Note that we still have to reference that we have implemented an approximation of D's algorithm.

The overall objective fitness function is used to determine how good the solution is for this problem described by the models employed [29]. A scalar overall objective function obtained with a linear combination of weighted objectives requires knowledge about each of the aforementioned cost functions to determine its weight. This also results in a loss of information due to the absence of knowledge about their correlation and interaction.

Hence, pythagorean sum is used. The multi-cost function model is:

$$C_overall = \sqrt{\sum C_{o_i}} \tag{20}$$

where $C_overall$ is the overall fitness of the chromosome, and C_{o_i} is the i^{th} cost function.

2.7 Implementation Details

The lowly coupled and well-defined interfaces of the software modules of *NetSim* facilitates the addition of alternative specifications for adjacent channel interference, load balancing, load shedding, multipath fading, network routing, signal attenuation, , and [2, 9, 20, 21, 27, 44]. Similarly, the removal of cost functions is trivial.

3 Experiments and Results

The simulations of the *Java*-based *NetSim* software were carried on the following machines: an Apple PowerBook G4 notebook computer running Mac OS X 10.3.8 with DDR RAM of 1 GB capacity and a PowerPC G4 processor with a speed of 1.50 GHz, a desktop computer running Gentoo Linux 1.4 with a AMD Sempron 2600 processor and 2.0 GB DDR RAM, and the Hydra Linux cluster. Hydra is an IBM eServer 1350 Linux cluster with 128 nodes and a header node for cluster management, with each of the 128 nodes having dual 2.4 GHz Intel Xeon processors of 512 KB of L2 cache memory, and 2.0 GB DDR RAM [45]. Hydra was measured with a Linpack benchmark result of 682 GFlops [46].

Preliminary simulation runs indicate that cost function 8 takes up too much memory space and cannot be simulated on a personal computer. In addition, cost functions 3 and

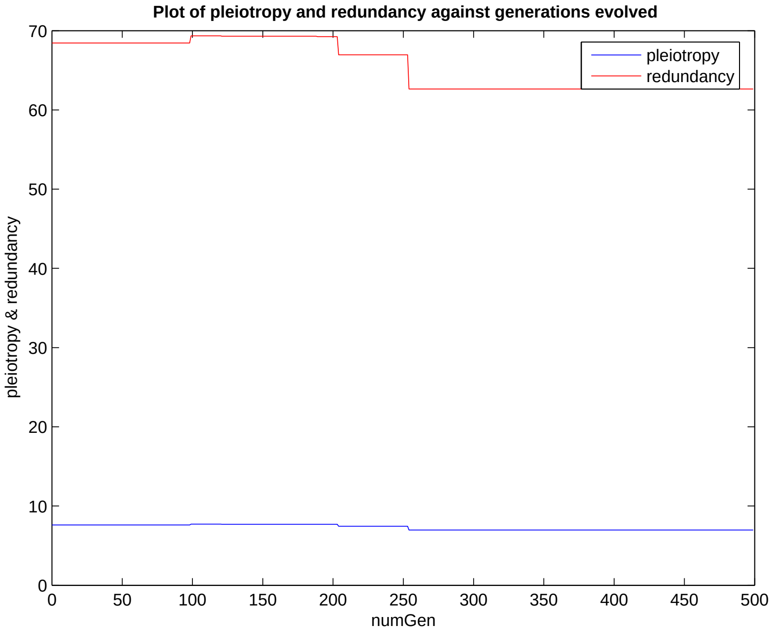


Figure 3: Plot of pleiotropy and redundancy against generations evolved

5 were ignored since cost function 3 was badly designed and cost function 5 yielded a constant value of fitness. Subsequently, only cost functions 1,2,4,6, and 7 were used.

Simulation runs were carried out for each cost function, and their cross-correlation was determined using a MATLAB script. The values for the correlation were: For cost functions 1 and 2, the correlation is: 0.3330 For cost functions 1 and 4, the correlation is: -0.1260 For cost functions 1 and 6, the correlation is: -0.2696 For cost functions 1 and 7, the correlation is: 0.2882 For cost functions 2 and 4, the correlation is: 0.1932 For cost functions 2 and 6, the correlation is: 0.0990 For cost functions 2 and 7, the correlation is: 0.1465 For cost functions 4 and 6, the correlation is: 0.2292 For cost functions 5 and 7, the correlation is: 0.1772 For cost functions 6 and 7, the correlation is: -0.1432

The P-values for these correlations are zero; hence, the author concludes that the correlations are statistically significant and the cost functions are independent of each other.

The simulation results are indicated as shown:

The constant average value, or a gently decreasing average value of fitness was due to the implementation of the steady-state evolutionary strategy. Since only two chromosomes are selected at each generation, hardly any significant changes were made to the population of chromosomes. Cost functions 1, 2, and 6 are correlated, and cost functions 4 and 7 are correlated. The pleiotropy values remain fairly constant whilst the redundancy values decrease gently.

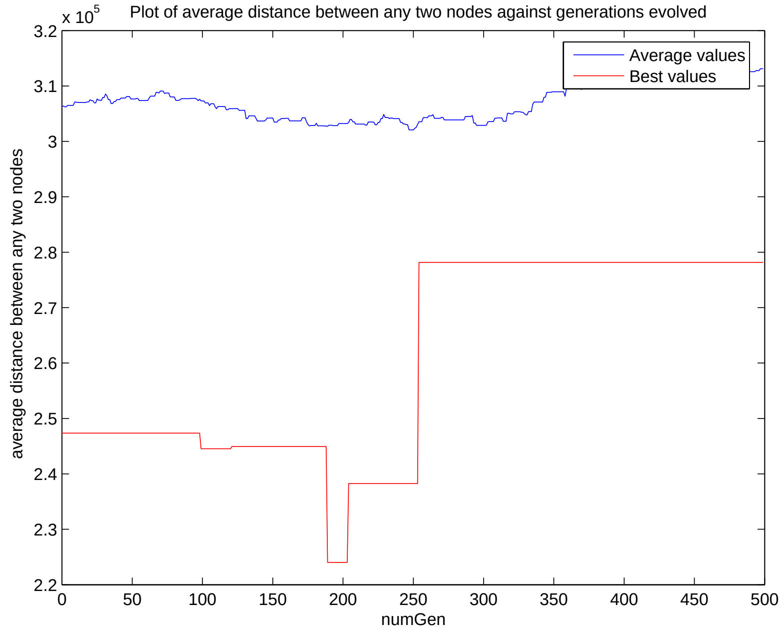


Figure 4: Plot of average distance between any two nodes against generations evolved

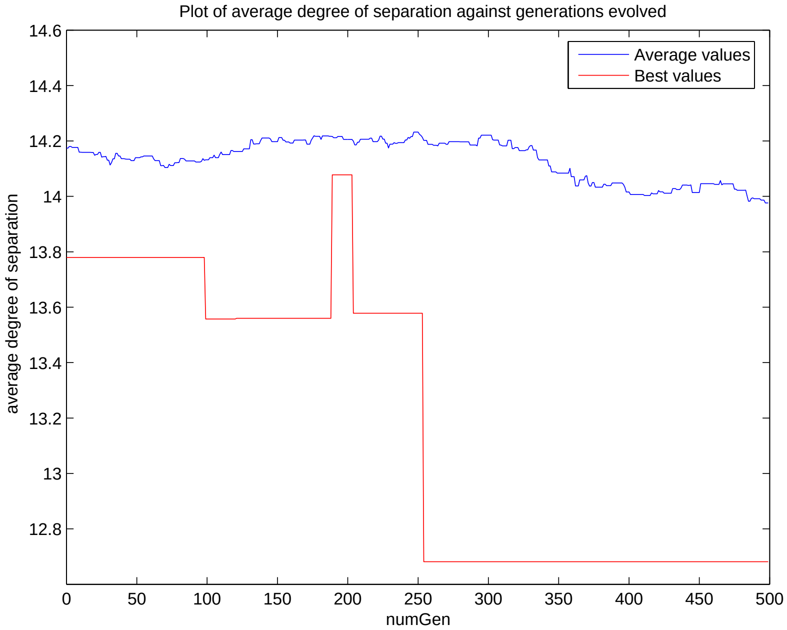


Figure 5: Plot of average degree of separation against generations evolved

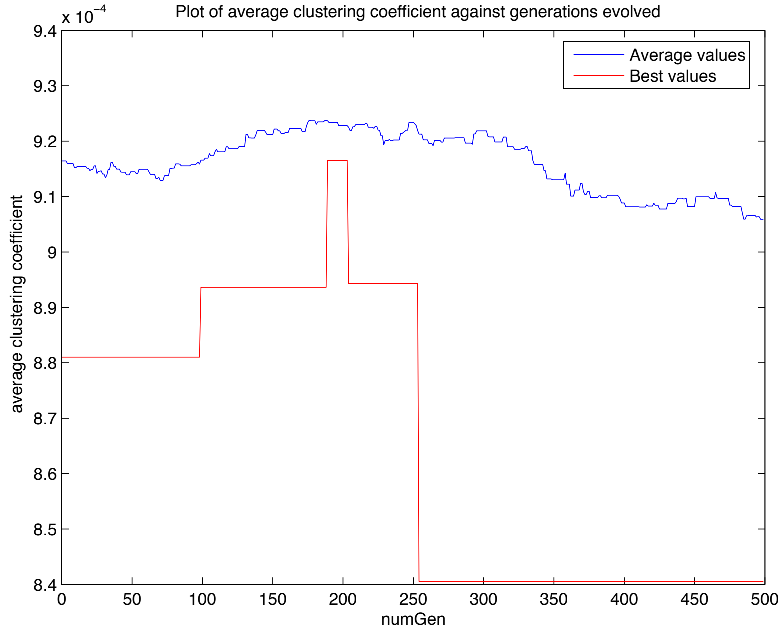


Figure 6: Plot of average clustering coefficient against generations evolved

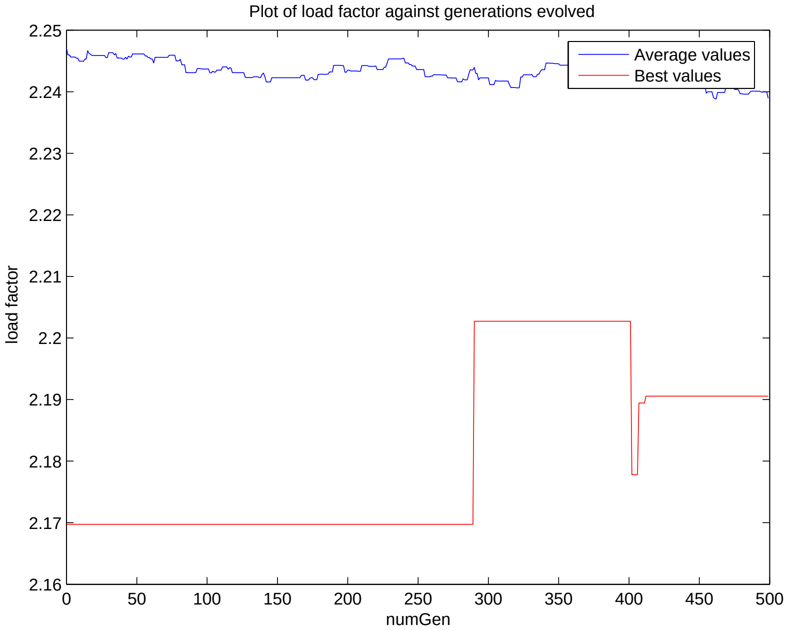


Figure 7: Plot of load fcator against generations evolved

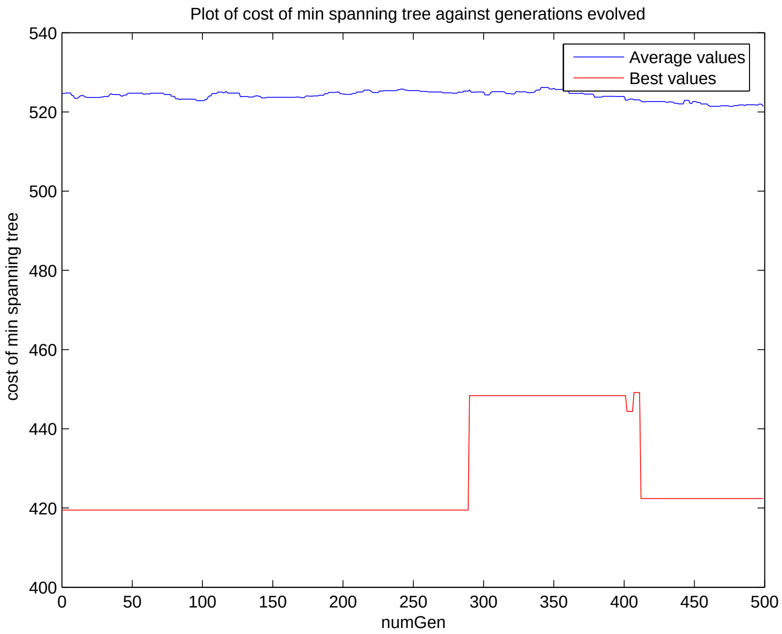


Figure 8: Plot of cost of minimum spanning tree against generations evolved

4 Project Management

A discussion and evaluation of the management for this research project is given. It briefly describes the problems encountered during the management of the project, and how the project plans were consequently modified. Subsequently, an enumeration of the completed milestones and tasks, the timeline for the achievement of goals, and division of work, is given. This is followed by a discussion of the software development process model, the budget for this project's expenses, and the risk management involved in this project.

4.1 Difficulties Encountered and Deviation from Project Plan

4.1.1 Configuration Management

Andy Lo and the author faced a problem with configuration management of the software simulator that they were developing. The development of a software requires many builds, and a management plan is needed to efficiently and effectively manage changes made to the software. Various builds need to be categorized appropriately, and have succinct descriptions provided for to facilitate the identification of changes made from the previous builds. This will facilitate code reuse, and software debugging and refactoring.

Early in the first semester of the project's duration, Andy and the author submitted a request to the computer support staff of the School of Electrical and Electronic Engineering, Adelaide University, pertaining to the provision of dual student accounts with its servers and a configuration management tool throughout the entire duration of the project to them. This request was granted, but the provision of the student accounts and access to a configuration management tool was disrupted in the summer break that occurred in the middle of the project's duration. Owing to the geographical separation of the authors, it was difficult to continue the development of the software over the summer break without a configuration management tool that is accessible by them.

After several requests to have the services restored, they were able to gain access to their accounts and the tool a month into the second semester of this project. By then, they had written a fairly extensive amount of code and the software had become fairly complex. Hence, it was difficult for them to update their repository with the builds accumulated during the service downtime. Consequently, they decided to use their electronic mail accounts to transfer the source code and software documentation to each other. To facilitate the management of the software builds, and software debugging and refactoring, they spend time developing the software together in fall break. This also facilitate the communication of ideas, and assistance to each other.

4.1.2 High memory requirements of *NetSim* simulations

The author concedes that the performance requirements of the software simulations were grossly overestimated. He did not expect the memory requirements for the software simulations to be very significant. This is attributed to the author's inexperience with project planning in software development. With an Apple PowerBook G4 laptop operating at a speed of 1.5 GHz with an PowerPC G4 processor and 1.0 GB of memory, he was able to simulate a population of 80 chromosomes with 90 nodes each for up to 3000 generations in about an hour. Hence, this performance measures fall significantly short of the estimated performance [24]. The duration of the simulation runs were not as important as the memory requirements since the authors can afford to let the computer perform the simulations whilst working on other tasks.

4.1.3 Development of *NetSim*'s GUI simulations

The author was unable to develop a graphical user interface that the user would be able to use easily. Whilst users are able to determine the set of objective functions to optimize the population of telecommunication networks with and enter key input parameters, they were unable to view the evolution of the network's topology. The only network topology was that when the simulation ended. They were also not able to pause or stop the simulation. However, this is not regarded as important as getting the software to emulate biological evolution and optimize a telecommunication networks. A Java Class file has been written by the author to simulate the evolution without the graphical user interface.

4.2 Milestones, Time-line, and Division of Work

The milestones and division of work are indicated as follows:

- The author wrote the project proposal with the help of Andy Lo, whom carried out the formal technical review for the proposal.
- Andy Lo and the author presented the project proposal together to their supervisors Professor Derek Abbott and Mr Matthew Berryman. Andy Lo saw oversaw the presentation of the presentation slides and the rehearsal for the seminar.
- Andy Lo and the author delivered the first version of *NetSim*, which is the software simulator for optimizing the telecommunications network, 9 academic weeks late on 18 April 2005. This is approximately half a year later than the expected date of completion for the first version of our software [24]. However, the first version was completed with four more objective functions than targeted (see Equations (12) - (17)); they had expected to come up with an objective function for each of the first two versions. This version also allowed the repair and failure rates of the clients and servers to vary based on a uniform distribution. Hence, the first version of *NetSim* was completed with the milestones for the first two versions of the software accomplished.

  Andy Lo was assigned to develop the *ecomp* and *graph* packages whilst the author was assigned to work on the *gui* and *population* packages, and the scripts that are used to build the software and process the simulation results. Both of them had also made contributions to the development of the *utility* package. Each of them wrote test suites for the classes they developed, with Zhiyang assisting Andy in testing two of his classes when Andy was ill.
- Andy Lo and the author completed the second version of the software with two additional cost functions, see Equations (18) and (19). This was delivered on 9 May 2005, which was delivered 6 weeks late. However, the turnover for this version was completed in the same time duration was expected [24].

  Due to the late delivery of the software versions, the author was unable to deliver the third version of *NetSim*. Andy and him were unable to model the limitation placed on the maximum amount of servers that can be running at any given time, nor were they able to model different types of servers. They were also unable to improve the software to cope with simulations of 5000 generations, and 80 chromosomes of 200 servers and 20,000 clients [24]. This task was also left out because of its time and memory order complexity challenge.
- Andy Lo and the author will present their research work about this study in a seminar that is expected to be held on 27 May, 2005.
- Andy Lo and the author have completed their project reports separately, and collaborated on a joint technical paper together.

A Gantt indicating the timeline of the author's progress in completing the milestones were indicated in Figure 9. This also indicated how far the authors were behind schedule at various phases of the development of *NetSim*, and the duration of each completed task. The tasks of literature review, maintenance of the logbook, and updating the final report were carried out concurrently with the software development to cover more milestones at each phase of the research project.

Please refer to Appendix B for additional tasks that were completed.

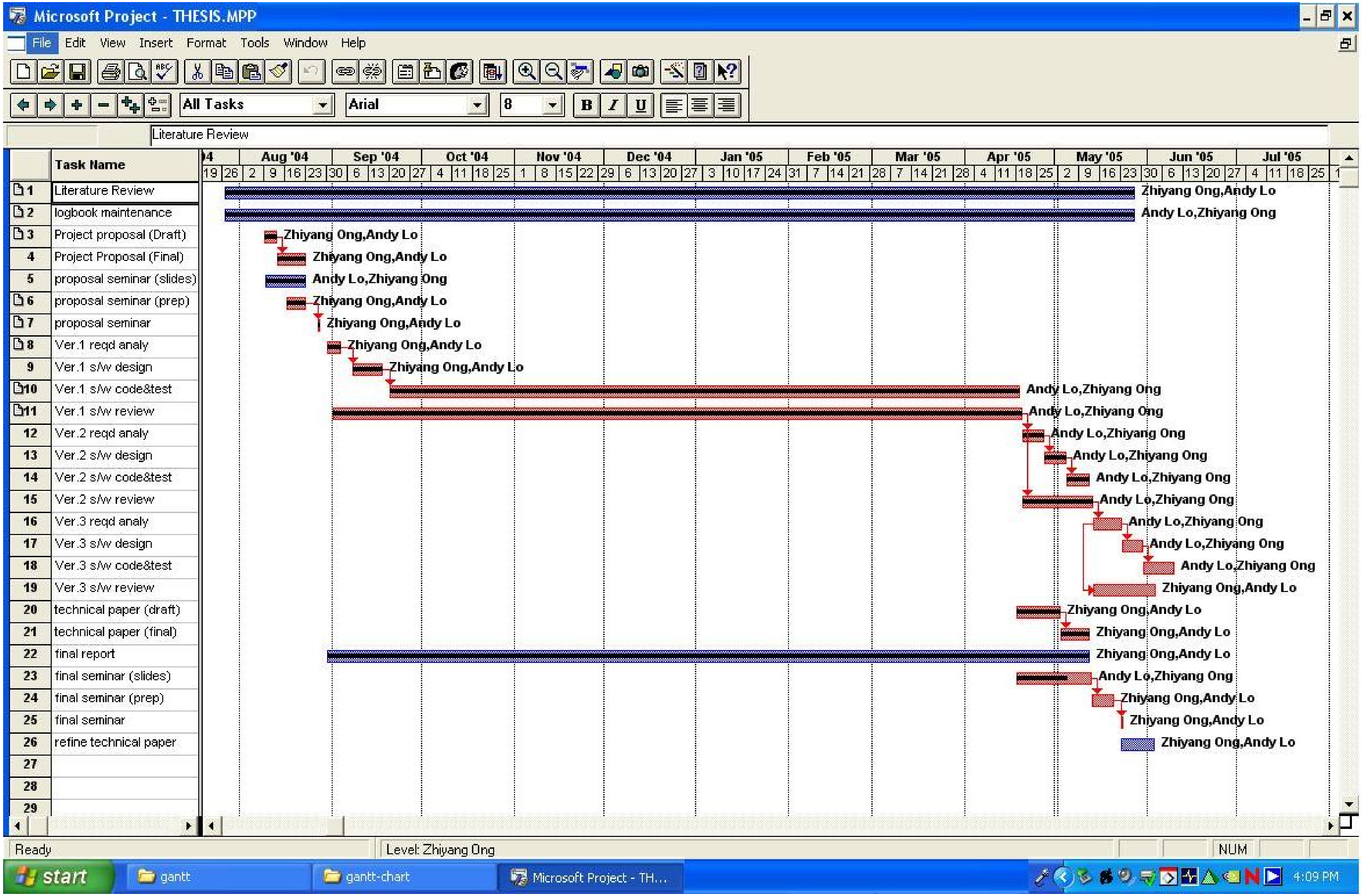
Minutes to meetings with Professor Abbott and Andy Lo were not kept after the first three weeks, as the author felt that it was difficult and ineffective to maintain a separate electronic copy of the minutes for all meetings to keep track of the software development. In lieu of that, the author had kept handwritten copies of minutes to meetings in the log books. Results obtained from running the test suites had not been kept either, since they could always be regenerated from running the test suites of each build.

4.3 Software Development Process Model

Andy and the author had developed the *NetSim* software using the extreme programming software development process. They had regular communication with each other and their supervisors, and often sough feedback to find guage their progress. They have also endeavoured to keep the software as simple as possible to allow modifications be made to the software architecture easily and quickly [47–49]. They had also made use of scripts and automated testing to help them build the system rapidly and frequently [50].

However, the lack of software development experience resulted in the development of the first version of *NetSim* with a fairly complex software architecture. This caused the development of the first version to resemble a single-pass linear project development process. Andy and the author realized that they were attempting to do too much in the first version and scaled down the development of the second version. This resulted in a smaller incremental version update; thus, the author did not develop the software incrementally and iteratively as desired [51–54]. Software documentation was kept to a minimum to reduce man-hours spent on tasks that are not required for the assessment of this subject [55, 56]. Coding standards agreed upon by Andy Lo and the author were adhered to so that assumed information were documented, and they can read and comprehend the source code written by each other [24].

They had used monitoring and controlling mechanisms such as regular discussion about problems encountered, and document and code reviews throughout the duration of the project to validate and verify the *NetSim* software and documents produced together [24]. They had also provided test suites for most classes to test the software. These tests were used to test the key methods and constructors of the Java Classes developed; the test suites for a package are kept in its subdirectory with the directory name "*tests*". Besides checking to see if the software has fulfilled its requirements, results from the test suites were handy in determining if modifications to the software were made correctly. This



	Task Name
1	Literature Review
2	logbook maintenance
3	Project proposal (Draft)
4	Project Proposal (Final)
5	proposal seminar (slides)
6	proposal seminar (prep)
7	proposal seminar
8	Ver.1 reqd analy
9	Ver.1 s/w design
10	Ver.1 s/w code&test
11	Ver.1 s/w review
12	Ver.2 reqd analy
13	Ver.2 s/w design
14	Ver.2 s/w code&test
15	Ver.2 s/w review
16	Ver.3 reqd analy
17	Ver.3 s/w design
18	Ver.3 s/w code&test
19	Ver.3 s/w review
20	technical paper (draft)
21	technical paper (final)
22	final report
23	final seminar (slides)
24	final seminar (prep)
25	final seminar
26	refine technical paper
27	
28	
29	

Figure 9: Gantt chart indicating the timeline for the completion of the milestones.

enabled the author to be less adverse to making changes to the code, and allowed him to incrementally and iteratively make progress on the software modules he was working on. In addition, the test suites provided a basic means of automated testing, where the author can run the test suite and work on something else till testing has been completed. This removes the need for manual testing, where the author would have to manually enter different input to test the software, and record the test inputs and outputs.

Informal code reviews were carried out when Andy or the author had problems verifying the software. This allowed the person in concern to inspect the code with a different perspective and determine the source of the problems. Document review were used to find errors, omissions, and inconsistencies in the project proposal and technical paper. The author decided not to keep monthly records of software metrics to indicate the progress he is making in developing the software with Andy Lo. This is because most of the errors discovered by the author whilst running the simulations with the graphical user interface and the test suites were corrected by the author prior to depositing the developed or modified source code into the repository or emailing it to Andy and Mr Berryman. Hence, the low number of errors that have not been properly tested for were not a good indication that the development of the software has been relatively free of defects. Similarly, the metric for measuring the efficiency of defect removal has hardly any significance. Since the author was learning about *Java* Swing whilst developing the software's graphical user interface, he did not produce much code in the first three months. Hence, the metric measuring the number of lines of code was not used as it would not have measured the amount of effort that the author has put into learning new programming techniques.

4.4 Budget

There were no project expenses claimed from the budget of AUD$320 that is given by the School of Electrical and Electronic Engineering, Adelaide University. The author had purchased a book written by Professor Albert-László Barabási to comtemplate his work on "the science of networks" [42]. Personal expenses incurred were not claimed from the budget since they were difficult to justify. Such expenses included printing and binding documents, and downloading of journal articles and conference papers, and software from the world-wide web. The software were available for free downloads under the GNU's Not Unix (GNU) General Public License.

4.5 Risk Management

The practice of extreme programming allows the author to minimize risk of not being able to complete the software development on schedule. A fairly incremental and iterative development allowed the author to complete two software versions, and avoid seeking extensions to complete the software development.

Risks were associated with the learning of new knowledge and skills by the author include taking a long time to comprehend the material or grasp the skill. The author endeavored to reduce this risk by seeking prompt assistance or a new perspective from his supervisors, Andy Lo, and other acquaintances after a significant period of time spent at learning the material. This reduced the occasions or prolonged unproductiveness where time was spent inefficiently and ineffectively at working out solutions to problems or comprehending new concepts. Finally, risks associated with the loss of data was prevented with regular backups of the software builds and versions, project documentation, and electronic copies of journal articles and conference papers.

5 Future Work

Avenues that are worth exploring include the following:

The method to obtain the minimum spanning tree of the telecommunications network, see Equations (12) and (13), can be improved by using a greedy method such as the Kruskal's and Prim-Jarník algorithms [40]. The method for determining the minimum shortest tree by using Dijkstra's algorithm has a time order complexity of $\mathrm{O}(n^3 \cdot \log n)$, whilst the Kruskal's and Prim-Jarník algorithms would significantly reduce the time order complexity to $\mathrm{O}(m \cdot \log n)$.

Queueing theory and Markov processes can be used to model the unscheduled demand for variable transmission bandwidth and processing capacity, which may lead to delay or loss in service when these resources are unavailable [7–9,57,58]. Therefore, delay and loss performance can be quantified and used as a cost function. From these models, a grade of service (GoS) measure can also be specified to indicate the probability of a phone call being blocked due to unavailable network resources; hence, GoS can be specified as a fitness function to be maximized [26].

The reliability or "resistance" of the telecommunications network can be modeled more accurately for use as cost functions with better algorithms to determine the set of all paths in the network with less time and memory complexity. The equations of the electrical network model for the telecommunication networks, describing its state, meshes and nodal connections, can be represented in matrix form for easier manipulation; graph reduction techniques can also be applied to simplify the networks [59]. Similarly, the reliability of the telecommunication network can also be determined using system reliability theory and state-space models. By modeling communication between two nodes in a telecommunication network as a pair of incoming and outgoing electromagnetic waves, the scattering matrix can be used to model the telecommunication network as a large connected web of N-Port waveguide networks [60–62].

Rerouting of data packets when data communication links fail or are overloaded can also be modeled. This will require the appropriate abstractions of routing protocols, such as Border Gateway Protocol (BGP), and routers. In addition, repeaters, hubs, bridges, switches, and gateways at different layers of the communications network may also be modeled. However, the author notes that it is a challenging task to model different routing protocols and various reference models for network architectures (such as Transmission Control Protocol/Internet Protocol (TCP/IP) and International Standards Organization Open Systems Interconnection (ISO OSI)) [21].

Hence, the development of the software from simulating simple network models to complex systems is a non-trivial task. Modeling concrete parameters of a particular telecommunication network with its specific constraints on client-server, server-server, and client-client interactions may yield more results that are pertinent to this study of the pleiotropy versus redundancy trade-offs. For example, an event-driven model of how cellular telephone calls are made can be incorporated into utilizing the communications link between the mobile stations (MS), base station subsystem (BSS), and network and switching subsystem (NSS) [26, 27]. Furthermore, improvement can be made to simultaneously satisfy competing design objectives by considering multi-objective optimization with evolutionary strategies [13]. For example, Pareto-based multiobjective genetic algorithms can be considered since it does not combine the objectives to derive the fitness values of the solutions; it obtains the fitness values of solutions by comparing the respective objective vectors projected into their objective space [29]

In modeling beneficial symbiotic relationships, mutualism and commensalism, links can be added to nodes with greater connectivity. This preferential treatment leads to the rich-get-richer phenomenon where nodes with greater connectivity have more chances of having edges added to them [22, 42]. As a result, the telecommunications network should follow the power-law relationship. Amensal and parasitic relationships can also be modeled in the evolution of the telecommunications networks. The author believes this would allow the population to be more diverse and take a reasonable time for the fittest chromosome in the population to converge at an optimal solution. However, the author notes that it is natural for entities (individuals, firms, non-profit organizations, and governmental organizations) in social-economic and organisms in biological networks to avoid other organisms, such as parasites, that will cause them to be disadvantaged. Hence, a chromosome should keep a list of previous intolerable partners that it would prefer not enter a relationship with again. This allows former parasitic chromosomes to renew relationships with previous partners as current commensal or mutualistic partners. Consequently, the objective of parasitic chromosomes is to convince its hosts that they are not currently parasitic or amensal [63].

Gene expression can be modeled in *NetSim* to capture patterns of similarity for desired regions within the search space [64]. This will help facilitate the partitioning of the search space by exploiting its nearly-invariant symmetric properties using a divide-and-conquer approach. Other genetic operators such as multi directional crossover and uniform crossover can be considered as well [16, 17] .

It will also be interesting to determine the pleiotropy and redundancy of entities in social-economic networks and, further the research of this study for organisms in biological networks. For example, a very large-scale integration (VLSI) circuit design company may purchase their electronic design automation (EDA) tools from various vendors such as Cadence Design Systems, Synopsys, and Mentor Graphics. It may also choose to purchase their hardware, such as computer servers, workstations, and personal computers, from different firms like Sun Microsystems, the Hewlett-Packard Company, and International Business Machines Corporation (IBM). This increases the redundancy of the design company's suppliers. Should the relationship between a supplier and the VLSI design company sour, the design company is still able to obtain resources from other suppliers and develop integrated circuits designs for its clients. Similarly, the design company should target different market segments such as radio frequency integrated circuits for the telecommunications industry, and System-on-Chips (SoCs) for the automotive industry to its pleiotropy. If and when the automotive or telecommunications industry goes into a slump, the design company would still be able to sell its designs to automotive companies.

An economic network model of individuals, firms, and governments can be developed for the comparison of alternative instruments of monetary policy that governments in developed and developing countries use. An investigation of how the interaction between individuals, firms, and governments in various developed and developing countries affect the effectiveness of those policies can be carried out using evolutionary computation to evolve the network. These entities of the economic network are, in turn, affected by those policies. This research can be coupled with game theory to study phenomena such as the "tragedy of the commons" and the "iterated prisoner's dilemma" [57, 65].

The network topology of the fittest chromosome that is displayed on the graphical user interface of *NetSim* can be saved as a picture file every 10 generations for comparisons of their evolving topology at the end of simulations. In addition, the *NetSim* software can be made to run independently if it can plot graphs on its own without having to access another software, such as MATLAB. Alternatively, the *NetSim* software can also be run from MATLAB, or have scripts written to run the *NetSim* program and plot the graphs on *Gnuplot, Open Office, or Microsoft Office.*

6 Conclusions

Experimental results indicate that pleiotropy and redundancy do not vary significantly, as expected, with evolving networks. This is due to the controlled evolution of networks that is modelled by the steady-state evolutionary strategy; change in telecommunications networks that occur drastically over a very short period of time are rare.

An investigation of the pleiotropy versus redundancy trade-off in networks could be applied to a multitude of networks. Networks can model air routes and airports in aviation [57], reference some other network!, or activities and events in project management [7, 8, 66, 67]. Whilst extending this research for social-economic and transportation networks may bring improve the profit margins of various companies, the author believes that delving into the exploration of pleiotropy and redundancy in telecommunication networks and VLSI circuit topologies would turn out to be more economically viable [68–70].

A Appendix: Investigation of Social Networks

A feasibility study was carried out for the determination of mathematical properties of social networks. Two types of social networks wFere considered: a network of romantic relationships and scientific collaboration networks. Data for the first social network was obtained from the annual magazines of the Lincoln College Club in Adelaide, Australia [71–75]. The years covered from 1997 till 1999, and from 2001 till 2002; data was not available from the 2000 annual magazine, Stag 2000.

The term "*score*" was used to denote a romantic relationship between two individuals for this network; a score is defined as an exchange of bodily fluids, which could be saliva or semen. Since there are at least about 100 people on each score tree, the graph representing the topology of the social network, the author estimates that there will be at least 400 people in this network should the data on each graph be merged together. This is because about 100 of the 240 students in Lincoln College will be replaced each year, and exchange students will arrive from different countries around the world each semester. Hence, there will be occasions where the same person will appear in different graphs, spread over the years.

When data of these graphs is entered into computer network models, the mathematical properties of these graphs can be determined. For example, the clustering coefficient of hubs and average degree of separation of individuals can be determined [23, 42]. Hubs refer to people represented in these graphs as being highly connected to other people, and they can be considered to be popular people in the social network. Sociology theory can be used to further analyze the data.

The author had also considered studying the network properties of scientific collaborations, starting by looking at the publications of academics and students in the School of Electrical and Electronic Engineering, Adelaide University. Some lecturers in this school have collaborated with other academics and research engineers from other departments in Adelaide University, and other universities and research organizations. These research organizations and universities include those from outside of Australia. The author has published papers with Dr Said F. Al-Sarawi from this school, and Professor Wieslaw Nowinski from the Bioinformatics Institute, Singapore. Professor Nowinski has collaborated with many other research scientists and engineers from China, Europe, and India. In turn, the collaborators of Dr Said Al-Sarawi and Professor Nowinski would have collaborated with many more people.

Consequently, it is easy to see how research engineers in a reasonably small research department can be part of a global scientific research collaboration network. An explosion in the number of people in this social network makes it difficult to obtain all the collaborators of each person in the network. Therefore, the author deemed that the further venture into social networks would not yield new scientific discoveries, and considers the enormous task of doing data entry for scientific collaboration networks to be an inefficient and ineffective pursuit of research interest.

B Appendix: Additional tasks

The following tasks were also performed throughout the duration of this project:

- Setting up a software development environment that allowed the author to work efficiently and effectively. This included the writing of scripts to efficiently develop and build the software.
- Maintenance of a logbook to indicate the thought processes of the author. The author has accumulated five logbooks that are submitted together with this thesis and a technical paper.
- Submitting a project proposal and presenting the project seminar with Andy Lo.
- Carrying out a literature review, and regularly seeking and processing information pertaining to this project.
- Carried out a feasibility study of determining mathematical properties of social networks, see Appendix A.
- Read books written by Albert-László Barabási and Duncan J. Watts on "the science of networks" [23, 42].
- Attended a workshop titled "Getting to Present at a Conference: Part 1 - Picking a Topic and Writing an Abstract," held at the office of Engineers Australia South Australian Division, on 4 August 2004. It was organized by the College of Biomedical Engineering (CBmE) South Australian Branch. The titles of the relevant sessions are "Choosing an appropriate topic" and "How to write an engaging abstract."
- Attended a research seminar on "Complex Adaptive Systems: Self-organization to the Edge of Chaos" that was presented by Mr Alex Ryan on 17 August 2004.
- Attended a talk on "Biological information - the essence of life" by Professor Richard Ivell. Prof. Ivell mentioned the use of networks to help him and his peers analyze biological systems; this talk was given on 16 September 2004.
- Attended a talk by Professor Nigel Bean on Mathematics and the telecommunications industry what do ants have to do with it? He talked about how the ant colony algorithm is applicable to telecommunications modelling; this was presented on 22 September 2004.
- Searched for a number of conferences that the author can submit coauthored abstracts, with his supervisors and Andy Lo, to.

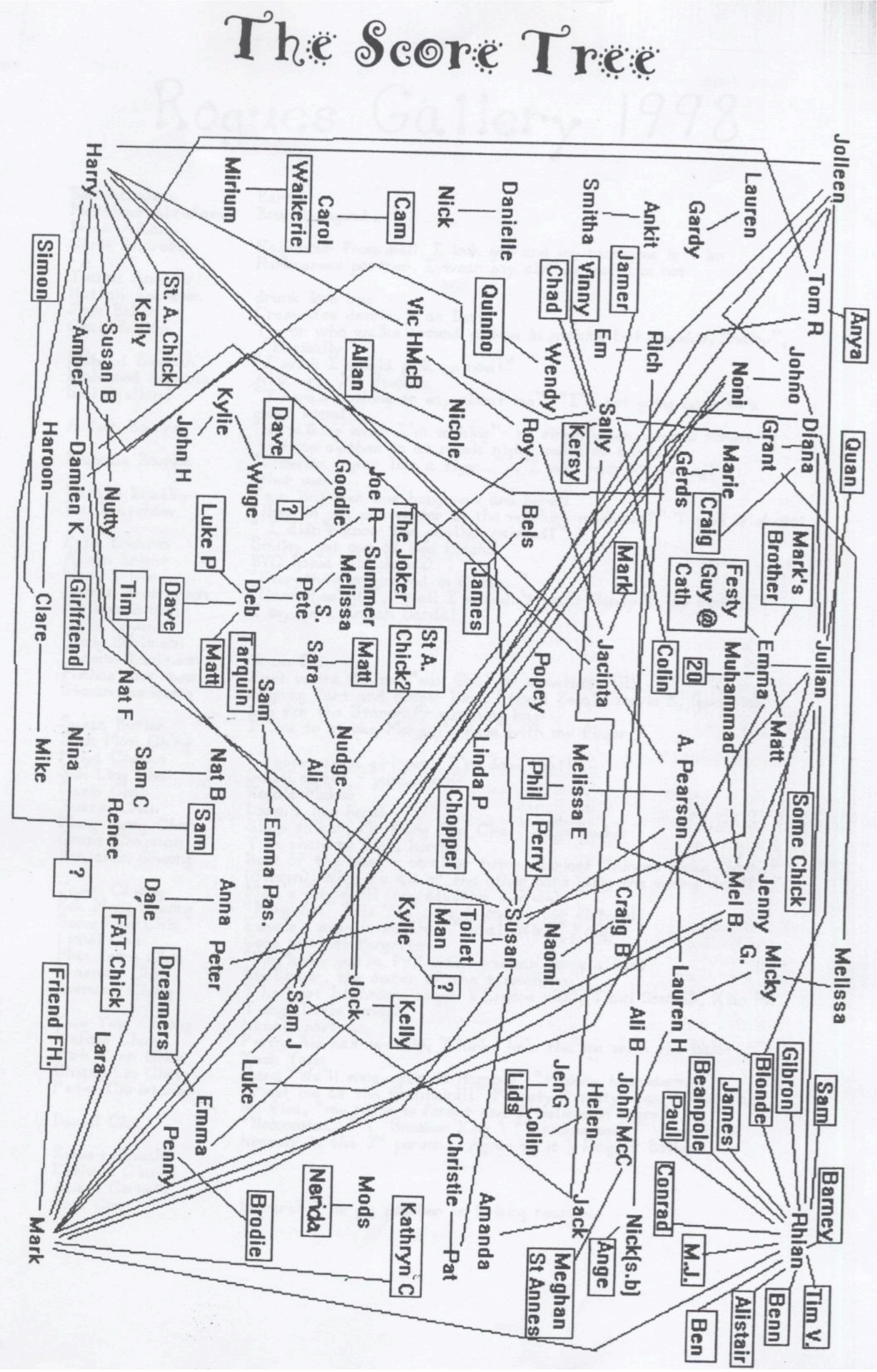


Figure 10: The topology of a social network of romantic relationships at Lincoln College, Adelaide, Australia in 1998

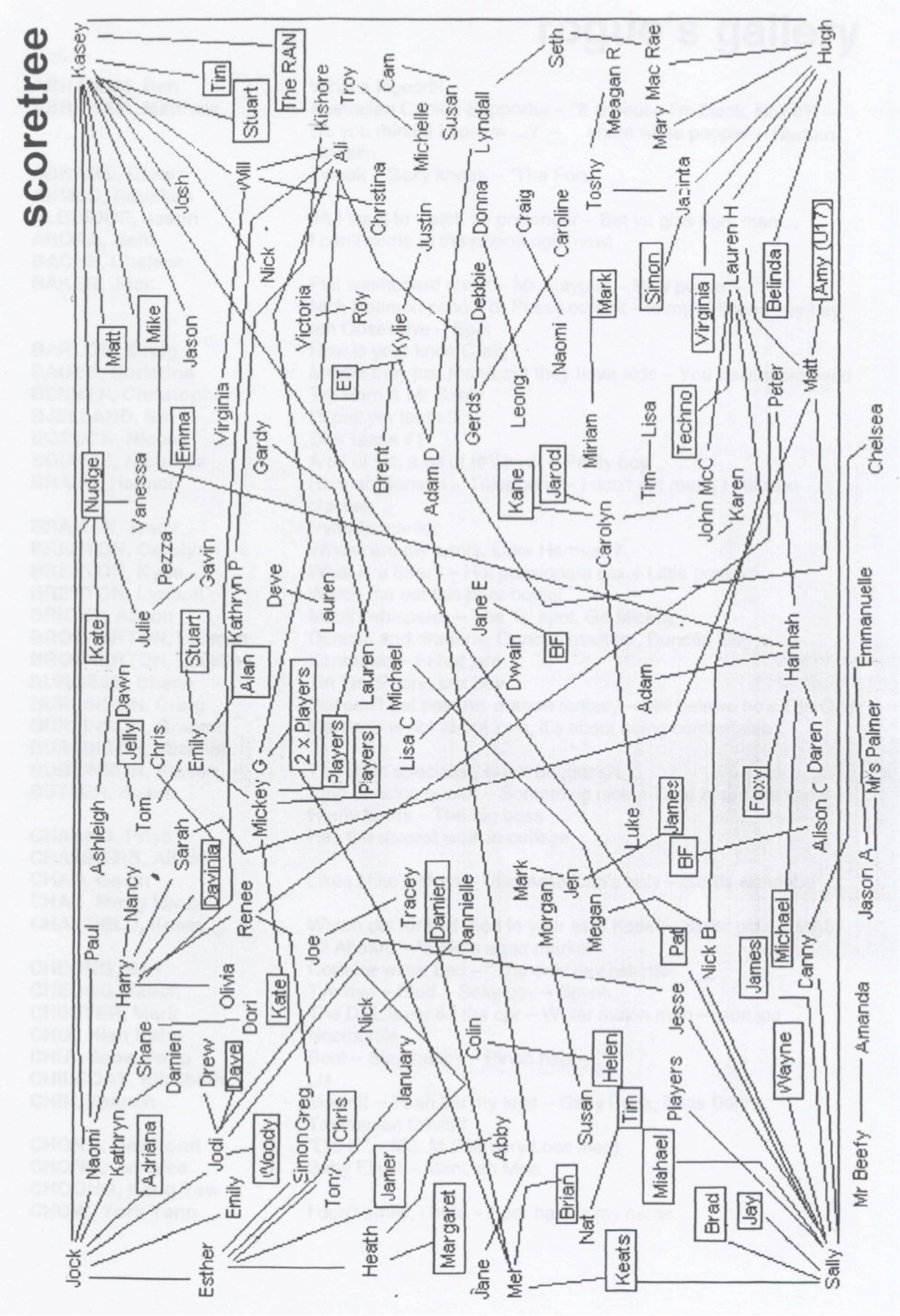


Figure 11: The topology of a social network of romantic relationships at Lincoln College, Adelaide, Australia in 1999

- Learned to create simple *Java* applications with Threads [76–80].
- Andy Lo and the author designed an algorithm to draw an arrow in it Java Swing applications that could correctly indicate the direction and magnitude of the vector arrow. This algorithm can be found in the *drawArrowLine* method of the *Java* Class *NetworkTopology*.
- Read about sorting algorithms to determine a quick and efficient algorithm for sorting data [79, 81]. Since sorting is commonly used with genetic algorithms and evolutionary strategies, the author was wanted to obtain a fast sorting algorithm that will help speed up simulation times [82].

  The Bucket-Sort algorithm can be used to sort a sequence of n items whose keys are in the range $\{0, N-1\}$, where N $\geq$ 2. Whilst it has a fast time order complexity of $\mathrm{O}(n+N)$, it uses up $\mathrm{O}(n+N)$ memory space [40]. Apart from the large memory usage, this was not selected since it required the keys to be kept within a range. Since chromosomes are sorted based on their varying fitness, the fitness values should take up the entire range of numbers of the data type used so that the fitness of chromosomes can vary by a lot for multiobjective optimization. The odd-even transposition sort and shear sort are parallel algorithms that run on n processors. Their worst case time order complexity is $\mathrm{O}(n)$ and $\mathrm{O}(\sqrt{n} \cdot \log n)$ [83]. They cannot be implemented in our software since a large number of processors, determined by the size of the population of chromosomes, will be required. Consequently, the merge sort algorithm from the *Java* Application Package Interface was used instead.
- Perform a feasibility examination, with Andy Lo, of the psossibility of extending and modifying the software developed by the author's predecessors Wei-Li Khoo and Hiep Nguyen.

  Andy Lo and the author split the important classes of the predecessors' software between themselves and reviewed their source code. They also wrote comments into the source code to describe what they believe the predecessors were doing or had intended to do. In addition, they noted that Mr Berryman had added functionalities to the software to include new fitness functions.

  They did not consider the software to be worth improving for the following reasons:

  - The classes in the software were large and contained several thousands of lines of uncommented code. These highly coupled and massive classes were placed in a single package, which included a sub-package for representing the telecommunications network as a graph. This makes the software hard to refactor for the improvement of its functionality, and the addition of genetic modifiers and cost functions.
  - Andy Lo and the author had to guess the intentions of the predecessors, since they had no means of communicating with them and comments in the source

code were absent. There were also no documentation provided to describe the assumptions and decisions made during the development of the software.

  - Test suites and documentation were absent. Andy Lo and the author had difficulty determining which part of the software works and which part needs modification. Some errors in the software were obvious, but the sources of these errors were not.

C Appendix: Attachments

The source code and the *Javadoc* documentation for the *NetSim* software is attached to the end of this thesis.

D Appendix: CD Index

The attached CD ROM includes:

1. a copy of this report
2. a folder containing all experimental data
3. a folder with all the weekly reports
4. a folder containing the project proposal
5. a folder containing the slides for proposal seminar
6. a folder containing the slides for the final seminar
7. a folder with all of the original graphics files
8. any other documents

References

[1] D. Walker and J. Solomon, "The interconnection imperative: E pluribus unum," *Telecommunications Policy*, vol. 17, no. 4, pp. 257–280, May/June 1993.

[2] D. E. Comer, *Computer Networks and Internets with Internet Applications*, 4th ed. New Jersey: Pearson Prentice Hall, 2004.

[3] Deloitte. (2005) Global telecommunications industry index 2005: Reconnected to growth. Global Telco Index 2005(2).pdf. Deloitte Touche Tohmatsu. Zurich. Viewed 30 April, 2005. [Online]. Available: http://www.deloitte.com/dtt/article/0,1002,sid%253D5638%2526cid%253D77173,00.html

[4] Telstra. (2004) Annual report 2004. operating.pdf. Telstra Corporation Limited. Melbourne. Viewed 30 April, 2005. [Online]. Available: http://www.telstra.com.au/communications/shareholder/financial{_}reports.cfm?ReportDate=2004{\&}ReportType=1

[5] M. J. Berryman, W.-L. Khoo, H. Nguyen, E. O'Neill, A. Allison, and D. Abbott, "Exploring tradeoffs in pleiotropy and redundancy using evolutionary computating," in *Proceedings of SPIE: BioMEMS and Nanotechnology*, D. V. Nicolau, Ed., vol. 5275, Perth, Australia, 10–12 December 2003.

[6] M. J. Berryman, A. Allison, and D. Abbott, "Optimizing genetic algorithm strategies for evolving networks," in *Proceedings of SPIE: Noise in Communication*, L. B. White, Ed., vol. 5473, Maspalomas, Gran Canaria, Spain, 26–27 May 2004.

[7] H. G. Daellenbach and J. A. George, *Introduction to Operations Research Techniques.* Boston, MA: Allyn and Bacon, Inc., 1978.

[8] D. T. Phillips, A. Ravindran, and J. Solberg, *Operations Research: Principles and Practice.* New York, NY: John Wiley & Sons, 1976.

[9] A. Leon-Garcia and I. Widjaja, *Communication Networks: Fundamental Concepts and Key Architectures*, 2nd ed. New York, NY: McGraw-Hill, 2004.

[10] T. J. Stewart, "Use of piecewise linear value functions in interactive multicriteria decision support: A Monte Carlo study," *Management Science*, vol. 39, no. 11, pp. 1369–1381, November 1993.

[11] T. J. Stewart, R. Janssen, and M. van Herwijnen, "A genetic algorithm approach to multiobjective land use planning," *Computers & Operations Research*, vol. 31, no. 14, pp. 2293–2313, December 2004.

[12] E. K. P. Chong and S. H. Żak, *An Introduction to Optimization*, ser. Discrete Mathematics and Optimization. New York, NY: John Wiley & Sons, 1996.

[13] A. Chipperfield and P. Fleming. An overview of evolutionary algorithms for control systems engineering. An Overview of Evolutionary Algorithms for Control Systems Engineering.pdf. The University of Sheffield. Sheffield, UK. Viewed 8 May, 2005. [Online]. Available: www.shef.ac.uk/content/1/c6/03/34/64/AnOverviewofEvolutionaryAlgorithmsforControlSystemsEngineering.pdf

[14] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, 2nd ed. Piscataway, NJ: IEEE Press, 2000.

[15] S. Obayashi, "Pareto genetic algorithm for aerodynamic design using the navier-stokes equations," in *Genetic Algorithms and Evolution Strategy in Engineering and Computer Science: Recent Advances and Industrial Applications*, D. Quagliarella,

J. Périaux, C. Poloni, and G. Winter, Eds. Chichester, West Sussex, England: John Wiley & Sons, 1998, ch. 12, pp. 245–266.

[16] C. Poloni and V. Pediroda, "Ga coupled with computationally expensive simulations: tools to improver efficiency," in *Genetic Algorithms and Evolution Strategy in Engineering and Computer Science: Recent Advances and Industrial Applications*, D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, Eds. Chichester, West Sussex, England: John Wiley & Sons, 1998, ch. 13, pp. 267–288.

[17] G. Winter, M. Galán, P. Cuesta, and D. Greiner, "Genetic algorithms: A stochastic improvement technique. tools, skills, pitfalls, and examples," in *Genetic Algorithms in Engineering and Computer Science*, G. Winter, J. Périaux, M. Galán, and P. Cuesta, Eds. Chichester, West Sussex, England: John Wiley & Sons, 1995, ch. 12, pp. 217–249.

[18] D. Whitley, S. Rana, and R. Heckendorn, "Representation issues in neighborhood search and evolutionary algorithms," in *Genetic Algorithms and Evolution Strategy in Engineering and Computer Science: Recent Advances and Industrial Applications*, D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, Eds. Chichester, West Sussex, England: John Wiley & Sons, 1998, ch. 3, pp. 39–57.

[19] M. Roughan. (2005, March) Network optimization: Goals and constraints. lecture05.pdf. School of Mathematical Sciences, University of Adelaide. Adelaide, Australia. Viewed 29 April, 2005. [Online]. Available: http://www.maths.adelaide.edu.au/people/mroughan/Lecture_notes/Comm_Net_Design/files/

[20] W. Tomasi, *Electronic Communications Systems: Fundamentals through Advanced*, 4th ed. New Jersey: Prentice-Hall, 2001.

[21] A. S. Tanenbaum, *Computer Networks*, 4th ed. New Jersey: Pearson Education, Inc., 2003.

[22] D. M. Pennock, G. W. Flake, S. Lawrence, E. J. Glover, and C. L. Giles, "Winners don't take all: Characterizing the competition for links on the web," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 8, pp. 5207–5211, April 2002.

[23] D. J. Watts, *Six Degrees: The Science of a Connected Age.* New York, NY: W. W. Norton & Company, 2003.

[24] Z. Ong and A. H.-W. Lo, "Honours project proposal: A study of the pleiotropy versus redundancy trade-off, using evolutionary computation," 2004, School of Electrical and Electronic Engineering, The University of Adelaide. Adelaide, SA.

[25] A. H.-W. Lo, “A study of the pleiotropy versus redundancy trade-off, using evolutionary computation,” School of Electrical and Electronic Engineering, The University of Adelaide, Adelaide, SA, 2004.

[26] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. New Jersey: Prentice-Hall, 2002.

[27] J. Schiller, *Mobile Communications*, 2nd ed. Essex, England: Pearson Education, 2003.

[28] P. Mirchandani and D. Simchi-Levi, *Networks and Flows: Communication Network Design Models*, ser. Discrete Mathematics and its Applications: Handbook of Graph Theory, J. L. Gross and J. Yellen, Eds. Boca Raton, FL: CRC Press, 2004.

[29] J. Périaux, M. Sefrioui, and B. Mantel, “Ga multiple objective optimization strategies for electromagnetic backscattering,” in *Genetic Algorithms and Evolution Strategy in Engineering and Computer Science: Recent Advances and Industrial Applications*, D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, Eds. Chichester, West Sussex, England: John Wiley & Sons, 1998, ch. 11, pp. 225–243.

[30] H.-P. Schwefel and T. Bắck, “Artificial evolution: How and why?” in *Genetic Algorithms and Evolution Strategy in Engineering and Computer Science: Recent Advances and Industrial Applications*, D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, Eds. Chichester, West Sussex, England: John Wiley & Sons, 1998, ch. 1, pp. 1–19.

[31] T. Bäck and H.-P. Schwefel, “Evolution strategies i: Variants and their computational implementation,” in *Genetic Algorithms in Engineering and Computer Science*, G. Winter, J. Périaux, M. Galán, and P. Cuesta, Eds. Chichester, West Sussex, England: John Wiley & Sons, 1995, ch. 6, pp. 111–126.

[32] H.-P. Schwefel and T. Bäck, “Evolution strategies ii: Theoretical aspects,” in *Genetic Algorithms in Engineering and Computer Science*, G. Winter, J. Périaux, M. Galán, and P. Cuesta, Eds. Chichester, West Sussex, England: John Wiley & Sons, 1995, ch. 7, pp. 127–140.

[33] Z. Michalewicz. (1995) Evolution systems. 14-17GeneticAlgorithms.pdf. School of Computer Science, Adelaide University. Viewed 28 March 2005. [Online]. Available: http://www.cs.adelaide.edu.au/users/third/ai/

[34] P. H. Raven and G. B. Johnson, *Biology*, 6th ed. New York, NY: McGraw-Hill, 2002.

[35] N. Deo, *Graph Theory with Applications to Engineering and Computer Science.* Englewood Cliffs, NJ: Prentice-Hall, Inc., 1974.

[36] E. M. Scheuer, *Reliability*, ser. Handbook of Operations Research: Models and Applications, J. J. Moder and S. E. Elmaghraby, Eds. New York, NY: Van Nostrand Reinhold Company, 1978, vol. 2.

[37] R. S. Kenett and S. Zacks, *Modern Industrial Statistics: Design and Control of Quality and Reliability.* Pacific Groove, CA: Duxbury Press, 1998.

[38] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows*, ser. Handbooks in Operations Research and Management Science: Optimization, G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, Eds. Amsterdam, The Netherlands: Elsevier Science B. V., 1989, vol. 1.

[39] G. G. Vining, *Statistical Methods for Engineers.* Pacific Groove, CA: Brooks/Cole Publishing Company, 1998.

[40] M. T. Goodrich and R. Tamassia, *Data Structures and Algorithms in Java*, 2nd ed. New York, NY: John Wiley & Sons, Inc., 2001.

[41] V. K. Balakrishnan, *Graph Theory*, ser. Schaum's Outline. New York, NY: McGraw-Hill, 1997.

[42] A.-L. Barabási, *Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life.* New York, NY: Plume, 2003.

[43] P. O. Boykin and V. P. Roychowdhury, "Leveraging social networks to fight spam," *IEEE Computer*, vol. 38, no. 4, pp. 61–68, April 2005.

[44] J. G. Proakis and M. Salehi, *Communication Systems Engineering*, 2nd ed. New Jersey, NJ: Prentice-Hall, 2002.

[45] SAPAC. (2005) Hydra user's guide - hardware. South Australian Partnership for Advanced Computing. Adelaide, Australia. Viewed 2 May, 2005. [Online]. Available: http://pollux.sapac.edu.au/hydra/Hardware.html

[46] SAPAC. (2005) Hydra user's guide - overview. South Australian Partnership for Advanced Computing. Adelaide, Australia. Viewed 2 May, 2005. [Online]. Available: http://pollux.sapac.edu.au/hydra/Overview.html

[47] B. Pyritz, "Extreme programming in the telecommunications domain," *Bell Labs Technical Journal: Software Technologies for Telecommunications Competitiveness*, vol. 8, no. 3, pp. 97–100, December 2003.

[48] R. Jeffries. (2000) Are we doing xp? XProgramming.com. Viewed 11 August, 2004. [Online]. Available: http://www.xprogramming.com/xpmag/are_we_doing_xp.htm

[49] J. Rasmusson, "Introducing xp into greenfield projects: Lessons learned," *IEEE Software*, vol. 20, no. 3, pp. 21–28, May/June 2003.

[50] J. K. Ousterhout, “Scripting: Higher-level programming for the 21st century,” *IEEE Computer*, vol. 31, no. 1, pp. 23–30, March 1998.

[51] B. Boehm and R. Turner, “Using risk to balance agile and plan-driven methods,” *IEEE Computer*, vol. 36, no. 6, pp. 57–66, June 2003.

[52] M. Cohn and D. Ford, “Introducing an agile process to an organization,” *IEEE Computer*, vol. 36, no. 6, pp. 74–78, June 2003.

[53] C. Larman and V. R. Basili, “Iterative and incremental development: A brief history,” *IEEE Computer*, vol. 36, no. 6, pp. 47–56, June 2003.

[54] M. Lycett, R. D. Macredie, C. Patel, and R. J. Paul, “Migrating agile methods to standardized development practice,” *IEEE Computer*, vol. 36, no. 6, pp. 79–85, June 2003.

[55] L. Williams, “Guest editor’s introduction: The xp programmer–the few-minutes programmer,” *IEEE Software*, vol. 20, no. 3, pp. 16–20, May/June 2003.

[56] O. Murru, R. Deias, and G. Mugheddu, “Assessing xp at a european internet company,” *IEEE Software*, vol. 20, no. 3, pp. 37–43, May/June 2003.

[57] J. S. Croucher, *Operations Research: A First Course.* Rushcutters Bay, N.S.W., Australia: Pergamon Press, 1980.

[58] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, 4th ed. New York, NY: McGraw-Hill, 2002.

[59] S.-P. Chan, *Introductory Topological Analysis of Electrical Networks*, ser. HRW Series in Electrical Engineering, Electronics, and Systems. New York: Holt, Rinehart, and Winston, Inc., 1969.

[60] S. Ramo, J. R. Whinnery, and T. Van Duzer, *Field and Waves in Communication Electronics*, 3rd ed. Hoboken, NJ: John Wiley & Sons, 1994.

[61] D. Pozar, *Microwave and RF Design of Wireless Systems.* New York, NY: John Wiley & Sons, 2001.

[62] R. E. Collin, *Foundations for Microwave Engineering*, 2nd ed. Singapore: McGraw-Hill, Inc., 1992.

[63] R. A. Watson and J. B. Pollack, “Symbiotic combination as an alternative to sexual recombination in genetic algorithms,” in *Proceedings of Parallel Problem Solving from Nature*, ser. Lecture Notes in Computer Science, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, Eds., vol. 1917. Paris, France: Springer-Verlag, September 2000, pp. 425–434.

[64] H. Kargupta, "Gene expression: The missing link in evolutionary computation," in *Genetic Algorithms and Evolution Strategy in Engineering and Computer Science: Recent Advances and Industrial Applications*, D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, Eds. Chichester, West Sussex, England: John Wiley & Sons, 1998, ch. 4, pp. 59–83.

[65] L. Tesfatsion, "An evolutionary trade network game with preferential partner selection," in *Evolutionary Programming V: Proceedings of the Fifth Annual Conference on Evolutionary Programming*, L. J. Fogel, P. J. Angeline, and T. Bäck, Eds. Cambridge, MA: MIT Press, February – March 1996, pp. 45–54.

[66] K. G. Lockyer, *An Introduction to Critical Path Analysis*, 3rd ed. London: Sir Isaac Pitman and Sons Ltd, 1969.

[67] W. E. Souder, *Project Selection, Planning and Control*, ser. Handbook of Operations Research: Models and Applications, J. J. Moder and S. E. Elmaghraby, Eds. New York, NY: Van Nostrand Reinhold Company, 1978, vol. 2.

[68] J. F. Miller, P. Thomson, and T. Fogarty, "Designing electronic circuits using evolutionary algorithms. arithmetic circuits: A case study," in *Genetic Algorithms and Evolution Strategy in Engineering and Computer Science: Recent Advances and Industrial Applications*, D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, Eds. Chichester, West Sussex, England: John Wiley & Sons, 1998, ch. 6, pp. 105–131.

[69] D. Floreano, "Evolutionary mobile robotics," in *Genetic Algorithms and Evolution Strategy in Engineering and Computer Science: Recent Advances and Industrial Applications*, D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, Eds. Chichester, West Sussex, England: John Wiley & Sons, 1998, ch. 17, pp. 341–365.

[70] R. H. Möhring, D. Wagner, and F. Wagner, *VLSI Network Design*, ser. Handbooks in Operations Research and Management Science: Network Routing. Amsterdam, The Netherlands: Elsevier Science B. V., 1995, vol. 8.

[71] J. Gray and V. Huntington-McBurney, "Stag 1997," November 1997, Lincoln College Club. Adelaide, Australia.

[72] V. Huntington-McBurney, "Stag 1998," November 1998, Lincoln College Club. Adelaide, Australia.

[73] V. Huntington-McBurney and A. Diggens, "Stag 1999," November 1999, Lincoln College Club. Adelaide, Australia.

[74] J. Bergamin, "Stag 2001," November 2001, Lincoln College Club. Adelaide, Australia.

[75] J. Bergamin, "Stag 2002," November 2002, Lincoln College Club. Adelaide, Australia.

[76] D. Flanagan, *Java Examples in a Nutshell: A Tutorial Companion to Java in a Nutshell.* Sebastopol, CA: O'Reilly, 1997.

[77] C. Xavier, *Programming with Java 2.* Chennai, India: Scitech Publications (India) Pvt Ltd, 2000.

[78] R. Cadenhead, *SAMS Teach Yourself Java 2 in 24 Hours*, 2nd ed. New Delhi, India: Techmedia, 2001.

[79] J. Lewis and W. Loftus, *Java Software Solutions: Foundations of Program Design*, 2nd ed. Reading, MA: Addison-Wesley, 1997.

[80] H. Deitel and P. Deitel, *Java How to Program*, 3rd ed. New Jersey, NJ: Prentice-Hall, 1999.

[81] R. L. Kruse, B. P. Leung, and C. L. Tondo, *Data Structures and Program Design In C.* New Jersey, NJ: Prentice-Hall, 1991.

[82] W. M. Spears, K. A. De Jong, T. Bäck, D. B. Fogel, and H. de Garis, "An overview of evolutionary computation," in *Proceedings of the European Conference on Machine Learning*, ser. Lecture Notes In Computer Science, P. Brazdil, Ed., vol. 667. London, UK: Springer-Verlag, April 1993, pp. 442–459.

[83] A. Kitchen. (1995) Sorting algorithms. Rochester Institute of Technology. Viewed 26 March 2005. [Online]. Available: http://www.cs.rit.edu/~atk/Java/Sorting/sorting.html

Glossary and Symbols

SMS: Short message service

MMS: Multimedia message service

G: Directed graph that is used to model the topology of the telecommunications network

N_{set}**:** Set of nodes in the graph G

E_{set}**:** Set of directed edges in the graph G

$P(i)$**:** Probability of selecting a chromosome that is ranked i^{th} in the sorted population

Pop**:** Size of the population

P_{leio}**:** The pleiotropy of a server

$\sum O_{s,c}$**:** Outgoing edge of a server with a client node as its destination node

$|C|$**:** Number of clients in the telecommunications network

R_{edun}**:** The redundancy of a server

$\sum I_{c,s}$**:** Incoming edge of a server with a client node as its source node

$|S|$**:** Number of servers in the telecommunications network

F: Fitness function

$\mu, \nu, \eta, \phi, \psi, \chi, \alpha, \varepsilon, \beta, \gamma, and\ \delta$**:** Weights (constant coefficients) for the functions

R_e**:** Reliability function of the telecommunications network

C_o**:** Cost function of the telecommunications network

$R_e \cdot C_o$**:** Interaction factor between the reliability and cost functions of the telecommunications network

$E_{working}$**:** Total number of working data links

$fail_{avg}$**:** Average percentage of link failures per generation

E: Edge E

N: Node N

m: Number of edges in the graph

n: Number of nodes in the graph

dist: Total cost of transmitting signals over all links

amp: Amount of amplification required in repeaters and/or satellites

sat: Cost of orbiting a satellite around the earth

mst: Set of edges in the minimum spanning tree

E_{mst_i}**:** i^{th} edge in the set of edges belonging to the minimum spanning tree

$\sum E_{sp_j}$**:** Sum of all edge costs in the spanning tree *st*

E_{sp_j}**:** j^{th} edge in the set of edges belonging to the spanning tree *st*

st**:** Spanning tree in G

$edge_{st}$**:** Set of edges in *st*

$|mst|$**:** Number of edges in the minimum spanning tree

E_i**:** i^{th} edge in the graph, belonging to E_{set}

A_{adj}**:** Adjacency matrix of a directed graph

a_{ij}**:** Cost of the data link connecting node i to node j

$path_{ij_k}$ k^{th} path between nodes i and j

AUD$: Australian dollars

GNU: GNU's Not Unix

WLAN: Wireless local area network

GSM: Global System for Mobile communications

GHz: Giga-Hertz

DDR: Double Data Rate

RAM: Random Access Memory

GB: Giga-Bytes

KB: Kilo-Bytes

GFlops: Giga-floating point operations per second

GoS: Grade of service

BGP: Border Gateway Protocol

TCP/IP: Transmission Control Protocol/Internet Protocol

ISO OSI: International Standards Organization Open Systems Interconnection

MS: Mobile station

BSS: Base station subsystem

NSS: Network and switching subsystem

VLSI: Very large-scale integration

EDA: Electronic Design Automation

IBM: International Business Machines Corporation

SoCs: System-on-Chips