UNIVERSITY OF BAHRAIN

College of Information Technology



Enterprise Network Topology Design with Fuzzy Logic Based Preferences

A Thesis Submitted in Partial Fulfillment of the Requirements for the Master Degree in Information Technology

Submitted by

Ebrahim Ahmed Bamasood

ID number: 20032199

Supervised by

Dr. Salman Khan

(Assistant Professor)

University of Bahrain

Kingdom of Bahrain

December, 2016

DECLARATION

I declare that this work is the result of my own investigation and that it has not already been accepted in substance for any degree, nor is it currently submitted for any degree.

Signed: ---------- Date: 1/12/2016

Ebrahim Ahmed Bamasood (Candidate)

The Thesis defense committee considers the Thesis titled:

Enterprise Network Topology Design with Fuzzy Logic Based Preferences

Submitted by Ebrahim Ahmed Bamasood to the College of Information Technology is satisfactory and acceptable for the Master of Science in Information Technology.

Thesis Defense Committee:

Dr. Salman Khan
(Supervisor)

Department of Computer Science
University of Bahrain

Professor Sadiq M. Sait
(Member)

King Fahd University of Petroleum & Minerals, Saudi Arabia

Dr. Alaudin Al-Omary
(Member)

Department of Computer Engineering
University of Bahrain

Abstract

The design of an enterprise networks consists of a number of network elements which are interconnected in a hierarchal structure in order to design the network. These network elements include local area networks from different geographical locations, routers, and layer 3 switches. Such network topology design is an NP-hard problem. This network topology design consists of simultaneous optimization of several design criteria such as monetary cost, network delay, number of hops between two nodes, and network reliability. Since some of these criteria are in conflict with each other, the problem can be considered a multi-objective optimization problem. In this thesis, fuzzy logic has been employed to address the multi-objective nature of the problem. A specific fuzzy operator, namely, Einstein operator has been utilized to aggregate the aforementioned design criteria into a scalar optimization function. The operator is combined with two optimization algorithms which are Simulated Evolution and Ant Colony Optimization. The study has four major purposes: the first is to propose a multi-objective Simulated Evolution (SE) algorithm to maintain Pareto optimal solutions. The second is to investigate the application and performance evaluation of the Einstein's operator for the enterprise network topology design problem, in the context of SE and Ant Colony Optimization algorithm (ACO). The third is to evaluate the sensitivity of Einstein operator in presence of preferences among criteria. Finally, the fourth purpose is to mutually compare the performance of both algorithms. Results indicated that ACO algorithm was able to produce better results for larger test cases in terms of quality of solution. Furthermore, with respect to analysis of preference, ACO also produced better results than SE algorithm.

Dedication

I would like to dedicate my thesis to my beloved family and to my lovely aunt Mona in United Arab Emirates (UAE).

Acknowledgements

All praise be to Allah Almighty, for his limitless blessing and guidance. It is only because of his will and mercy that this thesis was made possible.

I would like to express my deep gratitude and appreciation to my supervisor Dr. Salman Khan, for his guidance, patience, and sincere advice throughout this thesis. I acknowledge his valuable time and helpful criticism. Also, I would like to appreciate his deep understanding of my health condition that forced me to extend the work for another one year. His support encouraged me to complete the thesis.

All my family members, especially my parents, were a constant source of motivation and support. Their love and care carried me in my illness. Their prayers, guidance and inspiration led to this accomplishment.

I am very thankful to my dear aunt Mona in UAE for her support, love and care during the treatment in Dubai for complete one year. She helped me a lot in my most difficult time in my life.

Table of Contents

Subject	Page number
Abstract	I
Dedication	II
Acknowledgement	III
Table of Contents	IV
List of Tables	VII
List of Figures	VIII
List of Symbols	IX
List of Abbreviations	XI
List of Appendixes	XII
Chapter One Introduction	2
1.1 Introduction	2
1.2 Problem Statement	3
1.3 Motivation	3
1.4 Objectives	7
1.5 Methodology	7
1.6 Contributions	8
1.7 Thesis Structure	9
Chapter Two Background and Literature Review	12
2.1 Multi-objective Optimization	12
2.2 Fuzzy Logic	15
2.2.1 Fuzzy Logic and Multi-objective Optimization	16
2.2.2 Preferences of Objectives in Fuzzy Logic	20
2.3 Metaheuristics	22
2.3.1 Simulated Evolution	25
2.3.2 Ant Colony Optimization Algorithms	30
2.4 Concluding Remarks	34
Chapter Three Topology Design of Enterprise Networks	36
3.1 Background	36
3.2 Literature Review	42

3.3 Problem Definition	45
3.4 Design Objectives	45
3.4.1 Network Reliability	45
3.4.2 Average Network Delay	46
3.4.3 Monetary Cost	46
3.4.4 Number of Hops	47
3.5 Assumptions	47
3.6 Constraints	48
3.7 Einstein Operator based approach for Enterprise Network Topology Design Problem	49
3.7.1 Preference Rules	52
3.7.2 Preference Rules for EN Topology Design	53
3.8 Characteristics of Test Cases	55
3.9 Conclusion	56
Chapter Four **Archived Multi-objective Fuzzy Simulated Evolution Algorithm for EN Topology Design**	58
4.1 Proposed Algorithm	58
4.1.1 Initialization	59
4.1.2 Fuzzy Evaluation	59
4.1.3 Selection	59
4.1.4 Fuzzy Allocation	59
4.2 Experimental Results	60
4.3 Analysis of Preference Rules	67
4.4 Conclusion	70
Chapter Five **Einstein Operator based Fuzzy Ant Colony Optimization Algorithm**	72
5.1 FACO Algorithms and its Design Parameters	72
5.2 Experimental Results	73
5.3 Analysis of Preference Rules	79
5.4 Conclusion	81
Chapter Six **Comparison of Techniques**	84
6.1 Results and Discussion	84
6.1.1 Comparison Based on the Quality of Solutions	85
6.1.2 Comparison Based on Pareto-Optimal Solutions	90
6.2 Conclusion	91

Chapter Seven **Conclusions and Future Work**	93
7.1 Summary and Conclusion	94
7.2 Future Work	95
References	96

List of Tables

No.	Subject	Page
3.1	Network characteristics assumed for experiments	55
3.2	Characteristics of test cases used in experiments. Min Cost is in US$, Min Delay is in milliseconds, and traffic is in Mbps	55
4.1	Average and standard deviation for best solutions for AFSE algorithm. Cost is in US$, Delay is in milliseconds, and Time is in seconds	61
4.2	Regression analysis results for n15 using AFSE	68
4.3	Regression analysis results for n25 using AFSE	68
4.4	Regression analysis results for n33 using AFSE	69
4.5	Regression analysis results for n40 using AFSE	69
4.6	Regression analysis results for n50 using AFSE	69
5.1	Average and standard deviation for best of best solutions for FACO algorithm. Cost is in US$, Delay is in milliseconds, and Time in seconds	73
5.2	Regression analysis results for n15 using FACO	80
5.3	Regression analysis results for n25 using FACO	80
5.4	Regression analysis results for n33 using FACO	80
5.5	Regression analysis results for n40 using FACO	81
5.6	Regression analysis results for n50 using FACO	81
6.1	Comparison of AFSE and FACO	85
6.2	PO solutions for a typical run for AFSE and FACO	90

List of Figures

No.	Subject	Page
2.1	Optimization Methods	23
2.2	Pseudo-code for SE algorithm	26
2.3	Pseudo-code of the ACO meta-heuristic	31
3.1	Typical Enterprise Network	37
3.2	Flat Network Design	38
3.3	Typical Modular Enterprise Network Design	39
3.4	Three-Tier Network Design Model	41
3.5	Two-Tier Network Design Model	41
4.1	AFSE overall goodness for a typical run for n15	62
4.2	AFSE overall goodness for a typical run for n25	62
4.3	AFSE overall goodness for a typical run for n33	63
4.4	AFSE overall goodness for a typical run for n40	63
4.5	AFSE overall goodness for a typical run for n50	64
4.6	AFSE goodness for a typical run for n15	64
4.7	AFSE goodness for a typical run for n25	65
4.8	AFSE goodness for a typical run for n33	65
4.9	AFSE goodness for a typical run for n40	66
4.10	AFSE goodness for a typical run for n50	66
5.1	Overall goodness value of FACO for a typical run for n15	74
5.2	Overall goodness value of FACO for a typical run for n25	75
5.3	Overall goodness value of FACO for a typical run for n33	75
5.4	Overall goodness value of FACO for a typical run for n40	76
5.5	Overall goodness value of FACO for a typical run for n50	76
5.6	Goodness value of FACO for a typical run for n15	77
5.7	Goodness value of FACO for a typical run for n25	77
5.8	Goodness value of FACO for a typical run for n33	78
5.9	Goodness value of FACO for a typical run for n40	78
5.10	Goodness value of FACO for a typical run for n50	79
6.1	Plots of a typical run for AFSE and FACO for (a) n15, (b) n25, (c) n33, (d) n40, and (e) n50, Time in second	89

List of Symbols

Symbol	Meaning
s_i	First objective function
s_j	Second objective function
B	Bias Value
E	Complete Solution
e_i	Individual cell in E
C_i	Current Cost of i^{th} cell in E
S	Queue to store the selected cells.
O_i	Lower bound on cost of i^{th} cell in E
g_i	Goodness of i^{th} cell in E
O_i	Optimal cost of the element e_i
C_i	Actual cost of the same element
S	Optimal Solution Set
$P_{ij}^k(t)$	Probability of selecting a link l between node i and j
l	link
N_i	Set of neighbors
i	Node
k	Ant
η_{ij}	Heuristic value
τ_{ij}	Pheromone of the link l
α_{ant}	Coefficient of pheromone content
β_{ant}	Heuristic value
R_i	Reliability of link i
R_s	Reliability of the network
d	Total number of network devices in the network
Υ	Total traffic in the network
L	Number of links in the topology
λ_i	Traffic on link i in the bits per second (bps)
$\lambda_{max,i}$	Capacity of link i in bps
B_{ij}	Delay per bit in a link ij
b_{ij}	Delay per packet
ω	Average packet size in bits

$Length$	Total length of used cable
C_{cable}	Cost per cable unit
μ_c	Membership function of near optimum monetary cost
μ_d	Membership function of near optimum delay
μ_h	Membership function of maximum number of hops
μ_h	Membership function of near optimum reliability

List of Abbreviations

Code	Full Name
ACO	Ant Colony Optimization
AFSE	Archived Fuzzy Simulated Evolution
DM	Decision Maker
EN	Enterprise Network
FACO	Fuzzy Ant Colony Optimization
GAs	Genetic Algorithms
LAN	Local Area Network
MOO	Multi-Objective Optimization
NTD	Network Topology Design
PO	Pareto Optimal
PR	Preference Rule
PSO	Particle Swarm Optimization
RIP	Routing Information Protocol
SA	Simulated Annealing
SE	Simulated Evolution
SOO	Single-Objective Optimization
TS	Tabu Search
UAO	Unified And-Or

List of Appendixes

Number	Appendix
(1)	Linear regression analysis

Chapter One
Introduction

1.1 Introduction	2
1.2 Problem Statement	3
1.3 Motivations	3
1.4 Objectives	7
1.5 Methodology	7
1.6 Contributions	8
1.7 Thesis Structure	9

Chapter One
Introduction

1.1 Introduction

In daily life, decisions is one of the common actions that a person does. For example, what should be eaten for optimal physical growth, which company is the best for car insurance, which smartphone is the best in terms of its features, etc. In order to have the best decision, a number of options are evaluated based on certain criteria or objectives, where the aim is to maximize or minimize those objectives. For example, the picture clarity in a smartphone needs to be maximized; the travel time between two places should be minimized to reach a destination. When a task requires maximization or minimization of the problem objective(s), it is known as an *optimization* problem.

Optimization is an important subject in various areas such as science, engineering, and business. The aim of optimization in these areas is simply to find the best way of doing something. Many problems deal with optimization of single objective. However, there is a huge number of problems where the requirement is to optimize multiple objectives simultaneously. In such situations, the notion of preference may also come into picture whereby some objectives are preferred over the others in the decision process. Using preferences with optimization is evident especially with large number of objectives (Cvetkovic & Parmee, 2002).

This thesis discusses an optimization problem in the domain of computer network design. The specific problem to be dealt with is the enterprise network topology design whereby the best configuration of an enterprise network is sought from a tremendously huge number of

possible solutions. This problem is considered a multi-objective optimization problem requiring simultaneous optimization of various design criteria (a.k.a objectives) such as network monetary cost, network latency, hop count between two nodes, and network reliability. The focus is on applying two optimization techniques, namely, simulated evolution (SE) (R.M. Kling & Banerjee, 1990) and ant colony optimization (ACO) (Colorni, Dorigo, Maniezzo, Varela, & Bourgine, 1992), integrated with the Einstein operator (Li & Yen, 1995). In addition, a seven-level linguistic terms preference approach proposed in previous studies (Salman A. Khan, 2009) will be thoroughly analyzed through its application to the problem.

1.2 Problem Statement

This thesis aims at investigating the application and effectiveness of Einstein's operator in fuzzy logic with a preference method for Multi-Objective Optimization (MOO) problems. The application domain for the work was enterprise networks with focus on design network topology, while considering monetary cost, network reliability, average network delay, and number of hops between source-destination pair as the design objectives. The effectiveness of the proposed approaches were evaluated through empirical analysis and mutual comparisons.

1.3 Motivation

For many years, network topology design (NTD) problems have received a significant attention (da Silva, de Souza Britto Jr., de Oliveira, & Koerich, 2013; Estepa, Estepa, Cupertino, Vozmediano, & Madinabeitia, 2011; Gobjuka & Breitbart, 2010; Sung, Xin, Rao, Xie, &

Maltz, 2011). NTD is a complex problem that traditionally requires optimization of a design objective. A more complex form of NTD is where simultaneous optimization of a number of design objectives is sought (Kumar & Banerjee, 2011) which is classified as a mult-objective optimization problem. Due to the complexities involved in the design process, the NTD problem is categorized as an NP-hard problem (Gobjuka & Breitbart, 2010; Kleeman, Seibert, Lamont, Hopkinson, & Graham, 2012; Kumar & Banerjee, 2011). Enterprise network design, which is a sub-problem of NTD, is equally complex, and therefore has received notable attention from the optimization community and network design researchers. In an enterprise network (EN), several local access networks are connected through routers or layer three switches and may be subdivided into subgroups called network segments.

When it comes to the design of an enterprise network, or any NTD for that sake, optimization of a single design objective cannot produce quality results. It is due to the fact that there are various governing factors in network design, each playing a crucial role in the overall design process. Some important factors include monetary cost, network latency, hop count, and reliability. However, some objectives are conflicting where optimizing one objective leads to a negative impact on other(s). For example, minimizing the monetary cost of a network leads to minimize the reliability. As a result, complexity of the problem increases. Hence, a trade-off between conflicting objectives is required. This aspect has been addressed in many studies (Aparicio-Pardo, Skorin-Kapov, Pavon-Marino, & Garcia-Manrubia, 2012; Chen, Liu, & Chou, 2013; De Leon-Aldaco, Calleja, & Aguayo Alquicira, 2015; Estepa, Estepa, Cupertino, Vozmediano, & Madinabeitia, 2011; Kleeman, Seibert, Lamont, Hopkinson, & Graham, 2012; Yizhong, Jun, Lili, & Qingxin,

2012) where the focus was to come up with efficient multi-objective optimization strategies synergized with optimization algorithms.

An important issue in multi-objective optimization is to deal with "Pareto optimality". Pareto optimality is a state of allocation of resources in which it is impossible to make any one individual better without making at least one individual worse (Salman, 2009). In other words, since there are *contradiction*s of objectives, there does not exist a single solution that would optimize all the objectives simultaneously. Dealing with Pareto optimal solutions is somewhat well-established in the context of population based heuristics such as genetic algorithms (GAs), ant colony optimization (ACO), particle swarm optimization (PSO), and differential evolution (DE), among others. However, when it comes to synergizing multi-objective optimization with single-solution algorithms (also known as gradient based algorithms), dealing with Pareto optimal solution is a challenge. It is due to the fact that these single-solution algorithms, such as simulated evolution (SE), simulated annealing (SA), and tabu search (TS), maintain a single-solution in an iteration and cannot produce Pareto solutions in one iteration. This aspect provides motivation for the work undertaken in this thesis.

Another aspect that deserves attention is handling of preferences among different objectives in multi-objective optimization. In MOO problems, all individual objectives are generally considered as having an equal level of importance and no preferences are considered. In many situations, the desire of the decision maker is to prefer one (or more) design objectives over the others. This puts a burden on the decision maker (Xiaoning, Yu, Qingwei, & Weili, 2010). MOO algorithms ought to reflect the decision maker's preferences by incorporating their feelings about different solutions (J. S. Arora, 2012). However, MOO algorithms themselves are not capable of implementing preferences among

objectives. Therefore, a mechanism is required for implementation of preferences which is used to guide the search direction of the optimization algorithms. This would lead to finding solutions in the desired regions. Many different approaches exist to represent preferences to express information of pair alternatives (Zhou, He, Chen, & Liu, 2014). One of them is the fuzzy preference relation. Part of this work is motivated by the above observations, which inspire the analysis of a seven-level fuzzy preference approach proposed in literature (Salman A. Khan, 2009) in the context of DLAN topology design problem. These preference schemes are studied using two iterative algorithms, namely, SE and ACO. Simulated evolution follows a single-solution based search strategy while ant colony optimization is a population based algorithm.

Fuzzy logic is widely applied in many disciplines and real-world applications (Cara, Wagner, Hagras, Pomares, & Rojas, 2013). The logic has also received significant attention by researchers working in the domain of MOO. Since MOO problems require simultaneous optimization of all objectives under consideration, the "AND" operators known as t-norms have been traditionally used since they efficiently incorporate the characteristics of different objectives (through their membership functions) into one fuzzy rule. This aggregation of different objectives into one objective function is commonly known as *scalarization*. A number of t-norm operators have been proposed over the years. However, a majority of these t-norm operators (discussed in Chapter 2) involve a *function variable*, which adds a level of complexity to such operators. There are, however, some t-norm operators that do not involve a function variable and therefore a relatively easier to apply and analyze for scalarization. This motivates the use of one such operator in this work, which is known as the Einstein's operator. It is a simple t-norm

since it does not have any function variable. The performance of this operator is analyzed with application to the SE and ACO algorithms.

1.4 Objectives

The primary objectives of this thesis are summarized as follows:

- To analyze the suitability of the Einstein operator in solving the topology design problem, with the application of the operator to SE and ACO algorithms.
- To propose a multi-objective simulated evolution algorithm that could properly maintain Pareto optimal solutions.
- To analyze the 7-point preference scheme for SE and ACO algorithms.
- To perform mutual comparisons of the two algorithms.

1.5 Methodology

The concepts of multi-objective optimization and the two MOO algorithms analyzed in this thesis are first presented and discussed. It should be noted that the purpose of this thesis is not to formulate the enterprise topology design problem, and neither the thesis opts to suggest approaches as how the SE and ACO algorithms could be used to solve the enterprise network topology design problem; these issues have been studied in detail in previous research works (Premprayoon & Wardkein, 2005; H. Youssef, Sait, & Khan, 2000; Habib Youssef, Sait, & Khan, 2002). The purpose is to analyze the Einstein operator and the 7-point preference scheme, in addition to proposing a multi-objective SE algorithm. Therefore, test cases and algorithm parameter values that produced the best results have been taken from previous studies (Khan &

Engelbrecht, 2009, 2012; Youssef, Sait, & Khan, 2001). In this connection, five test cases were used for experimentation. These test cases included networks of 15, 25, 33, 40, and 50 nodes (local sites).

Due to the stochastic nature of the iterative algorithms considered herein, results are generally reported in terms of averages. However, since the simulations are computationally expensive, averages are calculated for thirty runs, and the average run time for the thirty runs is reported wherever appropriate. The performance is evaluated based on the quality of solutions produced. The results are also statistically validated through linear regression test.

Finally, both algorithms were written in C++ programming language and the simulation was performed based on a Windows 7 personal computer that has i7 processor with speed of 3.00 Ghz and 8GB of RAM.

1.6 Contributions

The following are the key contributions of the proposed work.

1. A multi-objective SE algorithm is developed based on the archiving approach. While archiving has been used in the context of other single-solution algorithms, such as SA, the approach has not been applied to and evaluated for the SE algorithm.
2. Performance of Einstein operator based fuzzy fitness function is studied with regard to its application to the SE and ACO algorithms. While other fuzzy operators, such as Unified And-Or and Werners operators have been studied in previous works, use of Einstein operator for the optimization problem considered herein has not been reported in other studies.

3. Analysis of a 7-point preference scheme proposed in a previous study (Khan & Engelbrecht, 2007) is performed with its application to the SE and ACO algorithms. The analysis is based on 8 preference rules.
4. In order to ascertain the relative performance of the Einstein operator and 7-point preference scheme on the two algorithms, a mutual comparison of the obtained results is carried out.

1.7 Thesis Structure

Chapter 2 provides the background and literature review. An overview of optimization techniques is presented. It starts with short discussion on optimization followed by its classification as single-objective optimization and multi-objective optimization. Then the chapter focuses on of multi-objective optimization. Following this, a primer on fuzzy logic and its use in MOO is discussed, along with some famous operators, including the Einstein operator. Later on, the chapter focuses on preferences and its effectiveness in MOO. This is followed by a description of the eight preference rules used in this thesis in order to evaluate the 7-point preference scheme. The chapter then continues with discussion on iterative algorithms. In this context, comprehensive information is given on simulated evolution algorithm and act colony optimization algorithms which are the focus of this thesis.

Chapter 3 addresses the EN topology design problem in sufficient details. This includes the problem description, assumptions, design objectives and its computation equations. Furthermore, preference rules used in this thesis to evaluate the iterative algorithms are also developed in this chapter.

Chapter 4 provides the details of implementing the archived fuzzy SE algorithm (AFSE) for EN topology design. It starts with describing the fuzzy design parameters of the algorithm. This is followed by a detailed discussion on the empirical results obtained. Later on, the 7-point preference scheme is evaluated using regression analysis for the eight preference rules through the empirical results obtained. Finally, a conclusion is provided based on the above discussion.

Chapter 5 describes the fuzzy ACO (FACO) algorithm for EN topology design. It focuses on the fuzzy design parameters of the algorithm. Then, a detailed discussion on the empirical results is presented. Furthermore, the 7-point preference scheme is evaluated using regression analysis. Finally, a conclusion is provided based on the above discussion.

Chapter 6 compares and summarizes the results obtained in Chapters 5, 6. The focus is to determine in which algorithm the Einstein operator performs better as well as the 7-point preference scheme.

Chapter 7 provides the conclusion and the outcomes of the thesis as well as the future work.

Chapter Two
Background and Literature Review

2.1 Multi-objective Optimization	12
2.2 Fuzzy Logic	15
2.2.1 Fuzzy Logic and Multi-objective Optimization	16
2.2.2 Preferences of Objectives in Fuzzy Logic	20
2.3 Metaheuristics	22
2.3.1 Simulated Evolution	25
2.3.2 Ant Colony Optimization Algorithms	30
2.4 Concluding Remarks	34

Chapter Two

Background and Literature Review

In this chapter, optimization and its various directions are discussed. The chapter starts with a short discussion on optimization, covering the important issues. The chapter then describes the classification of optimization problems as well as the multi-objective optimization. Another focus of this chapter is the background of metaheuristic algorithms. The definition and the derivation of metaheuristics are discussed in details. Later, an overview of simulated evaluation algorithm and ant colony optimization algorithms are given since they will be used in this thesis. The chapter also discusses fuzzy logic in the context of multi-objective optimization, various t-norms operator, and a design preference scheme that will be used to evaluate the performance of the two optimization algorithms.

2.1 Multi-objective Optimization

Optimization is defined as the process of finding the best way to utilize available resources without violating any of the constraints that might exist (De Leon-Aldaco, Calleja, & Aguayo Alquicira, 2015). The main goal of optimization is to achieve certain values such that the objective function is maximized or minimized. A solution that satisfies all problem constraints is called a *feasible solution.* An *optimum solution* is defined as a feasible solution that has optimum value(s) of objective function among other feasible solutions.

An optimization problem generally consists of three main aspects: design variables, objective functions, and constraints (J. Arora, 2004). Design variables are quantities which are chosen by the designer in the

design process. These quantities cannot be chosen arbitrarily since specific functional requirements must be satisfied. The second aspect of the optimization problem is the objective function. It can be defined as the criterion or property that need to be optimized (maximized or minimized) in the design and expressed as a function of the design variables. Constraints are restrictions and conditions that must be satisfied in order to accept the generated solution(s). Mainly, they are related to the performance specifications such as input and output limitations.

For maximization problems, the optimized solution derived from the problem can be classified as global optimum and local optimum. The global maximum is the maximum value of an objective function that can be found in the entire search space. In contrast, the local optimum is the maximum value within the local neighborhood (sub-region) of a solution. Thus, it is possible to have many local optima within the global search space.

Another classification of the optimization problems could be based on the number of objectives to be optimized. Problems which require only one objective to be optimized are called *single-objective optimization (SOO) problems*. In addition, it is possible to have a problem with more than one objective to be optimized, but the problem is still called single optimization problem, if the objectives are non-conflicting. This means if one objective is optimized, all other objectives are optimized implicitly. However, when objectives are *conflicting*, which means that maximizing one objective will negatively affect the others, the problem is referred to as *multi-objective optimization (MOO) problem*. In this case, a trade-off is required between the objectives in order to derive a balanced solution.

To solve a MOO problem, several methods exist in literature such as weighted sum, goal programming, lexicographic ordering, and ϵ-

constraint, among others (J. Arora, 2004; El-Ghazali, 2009; Miettinen et al., 2001). All techniques have their pros and cons. For example, weighted sum method is simple in terms of implementation but suffers from sensitivity of the weights assigned to each objective; a small change in weights may result in big changes in the objective vectors. In goal programming, targets, or goals that need to be achieved, are assigned by the decision-maker The main advantage of goal programming is its computational efficiency, provided that the target values are known, and if the goals are in a feasible region. However, if the targets are wrong, then a feasible region is difficult to approach, in which case goal programming could be very inefficient. Lexicographic ordering ranks the objectives in order of importance and optimizes the objectives with respect to their importance, starting with the optimization of the most important objective first. The major advantage of the technique is its simplicity. However, one major disadvantage is that the technique tends to favor certain objectives when many are present, since randomness is involved in the process. This may cause the solutions to be concentrated in a particular part of the Pareto front rather than the complete front. The ϵ-constraint method is based on the minimization of one objective function (the most preferred or primary, as chosen by the decision-maker), and considering the other objectives as constraints bound by some allowable levels. The main strength of the technique is its simplicity but the most prominent disadvantage of the above approach is that it is time-consuming.

Fuzzy logic has also been used extensively for MOO. Since the focus of this thesis is on fuzzy logic based MOO, a detailed discussion on fuzzy logic is provided in the following section.

2.2 Fuzzy Logic

Fuzzy logic was first introduced by Zadeh in 1965 (L.A. Zadeh, 1965). It is a mathematical discipline that expresses human reasoning in precise mathematical notation. Also, the logic is known for providing a convenient tool for interfacing linguistic categories with numerical data and for expressing user's preference in a gradual and qualitative way (Dubois, Prade, & Sedes, 2001). Fuzzy logic statements can be partially true and partially false at the same time. The level of acceptance of a statement is measured by a membership function, μ, in the range [0, 1], where $\mu = 0$ indicates that the statement is pure false and $\mu = 1$ shows that the statement is pure true. A membership function provides a measure of the degree of membership (membership value) for every element in the fuzzy set.

As mentioned earlier, fuzzy logic expresses human reasoning. This is done through a *linguistic variable*. A linguistic variable, as defined by Zadeh (L. A. Zadeh, 1975), is a variable whose values are words or sentences in a natural or artificial language. Such variable could be like, "monetary cost", "network average delay", "maximum number of hops", and "network reliability". Each linguistic variable has a set of values. Each value has a meaning associated by a membership function. For example, the 7-point preference scheme (discussed later) consists of seven linguistic values such as "strongly preferred" which has a range of values from 0.916 to 0.999 that is identified through a membership function.

Fuzzy rules are one of the most important components in a fuzzy system. They are of type "IF- THEN" rules. It can be identified through a relation between linguistic values of condition (the antecedent) and linguistic values of outcome (the consequent). For example,

IF *monetary cost is low* AND *network average delay is low* AND *maximum number of hops are low* AND *network reliability is high* THEN *the solution is good.*

Where *monetary cost, network average delay, maximum number of hops, network reliability,* and *solution* are linguistic variables and *low, high*, and *good* are linguistic values.

In order to combine one or more linguistic variables in a fuzzy rule, fuzzy operators are employed. Many operators for fuzzy union and fuzzy intersection exist. Fuzzy union operators are known as s-norm operators which depend basically on "ORing" functions since they implement the OR operation between the membership functions. Some examples of s-norm operators are maximum operator, algebraic sum operator, and bounded sum operator. On the other hand, fuzzy intersection operators are known as t-norm operators. They work on "ANDing" properties since they apply the AND operations between the membership functions. Some examples of t-norm operators are: minimum operator, algebraic product operator, and bounded product operator.

2.2.1 Fuzzy Logic and Multi-objective Optimization

In recent years, fuzzy logic has received a notable attention by researchers in various areas such as energy management strategies (Ahmadi & Bathaee, 2015), stock market forecasting (Majhi & Anish, 2015), intelligent transportation systems (Onieva, Hernandez-Jayo, Osaba, Perallos, & Zhang, 2015), decision policies implementation (de Salles, Gonçalves Neto, & Marujo, 2016), and aggregated production planning (Gholamian, Mahdavi, Tavakkoli-Moghaddam, & Mahdavi-Amiri, 2015). Fuzzy logic framework implements a scalar function that allows to describe uncertainties in terms of conditional probabilities for

the majority of practical cases. Therefore, fuzzy logic approach is strongly considered for solving MOO problems.

Another reason to emphasize the use of fuzzy logic in MOO is the nature of algorithms used to solve MOO problems. MOO problems are NP-hard in nature. In addition, the complexity of the problem is further increased by incorporating design constraints. To solve these NP-hard problems, heuristics are used which are based on human knowledge gained by experience and understanding of problems. Fuzzy logic has a suitable approach for expressing such knowledge.

The third reason behind utilizing fuzzy logic in MOO problems is the objectives of the problem themselves. MOO problems need to optimize a number of objectives for a particular problem. Those objectives are conflicting and require simultaneous optimization. Fuzzy logic provides a framework to handle the problem by incorporating the characteristics of each objective into one fuzzy membership function and transforms the problem from MOO into a single-objective optimization problem.

The original t-norm operator, the "min" function, was a pure ANDing operator that reflects the effect of the objective with the lowest membership only, completely ignoring the effect of all membership values associated with other objectives. In order to reflect the effect of all objectives simultaneously, special t-norm operators play an important role in defining a fuzzy rule. These operators are commonly referred to as "soft AND" operators. A number of soft AND operators have been proposed in literature, such as Hamacher's operator (Hamacher, 1978), Frank's operator (Frank, 1979), Schweizer's operators (Schweizer & Sklar, 1963), Dombi's operator (Dombi, 1982), Weber's operator (Weber, 1983), Dubois and Prade's operator (Dubois & Prade, 1979, 1982), Werners operator (Weber, 1983), Unified And-Or (UAO) operator (Khan

and Engelbrecht, 2007), and Einstein's operator (Li & Yen, 1995). Mathematically, these operators are expressed as follows:

Hamacher:
$$H(\mu_A,\mu_B)=\frac{\mu_A\mu_B}{\beta+(1-\beta)(\mu_A+\mu_B-\mu_A\mu_B)},\beta\geq 0 \quad (2.1)$$

Frank:
$$F(\mu_A,\mu_B)=\log_\beta\left(1+\frac{(\beta^{\mu_A}-1)(\beta^{\mu_B}-1)}{\beta-1}\right),\ \beta>0,\beta\neq 1 \quad (2.2)$$

Schweizer1:
$$S_1(\mu_A,\mu_B)=\sqrt[\beta]{max\left\{0,\left(\mu_A^\beta+\mu_B^\beta-1\right)\right\}},\beta>0 \quad (2.3)$$

Schweizer2:
$$S_2(\mu_A,\mu_B)=1-\sqrt[\beta]{(1-\mu_A)^\beta+(1-\mu_B)^\beta-(1-\mu_A)^\beta(1-\mu_B)^\beta},\beta>0 \quad (2.4)$$

Dombi:
$$D(\mu_A,\mu_B)=\frac{1}{1+\left(\left(\frac{1}{\mu_A}-1\right)^\beta+\left(\frac{1}{\mu_B}-1\right)^\beta\right)^{\frac{1}{\beta}}},\beta>0 \quad (2.5)$$

Weber1:
$$W_1(\mu_A,\mu_B)=max\left\{0,\left(\frac{\mu_A+\mu_B-1+\beta\mu_A\mu_B}{1+\beta}\right)\right\},\ \beta>-1 \quad (2.6)$$

Weber2:
$$W_2(\mu_A,\mu_B)=max\{0,(1+\beta)\mu_A+(1+\beta)\mu_B-\beta\mu_A\mu_B-(1+\beta)\},\beta>-1 \quad (2.7)$$

Dubois and Prade:
$$DP(\mu_A,\mu_B)=\frac{\mu_A\mu_B}{max\{\mu_A,\mu_B,\beta\}},0\leq\beta\leq 1 \quad (2.8)$$

Werners:
$$Y(\mu_A,\mu_B)=\beta\times min\{\mu_A,\mu_B\}+(1-\beta)\times\frac{1}{2}(\mu_A+\mu_B),0\leq\beta\leq 1 \quad (2.9)$$

UAO:
$$U(\mu_A,\mu_B)=\frac{\mu_A\mu_B+\nu\max(\mu_A,\mu_B)}{\nu+\max(\mu_A,\mu_B)},0\leq\nu\leq 1 \quad (2.10)$$

Einstein: $$E(\mu_A, \mu_B) = \frac{\mu_A \mu_B}{2 - (\mu_A + \mu_B - \mu_A \mu_B)} \quad (2.11)$$

From the above operators, Einstein's operator given in Equation (2.11) is the only operator that does not depend on a function variable. In general, function variable adds a level of complexity to the operator since it is a cumbersome process to tune the function variable itself in order to get the best results.

In 1997, Cordon and others (Cordón, Herrera, & Peregrín, 1997) have applied the Einstein's operator to two fuzzy control applications: the Inverted Pendulum problem and the fuzzy modeling of the real curve $Y = X$. In 2009, the Einstein's operator has been applied to a study for Arabic web pages classification based on a fuzzy similarity approach on 50 different Arabic web pages (Al-Taani & Al-Awad, 2009). In this study, five other operators were analyzed such as MinMax, Hamacher, Algebraic, Bounded Difference, and Special case fuzzy. The Einstein's operator recorded best performance among the others. Furthermore, the Einstein's operator is utilized in the intuitionistic fuzzy set, which is an extension to the Zadeh's fuzzy set theory, where the non-membership function is considered besides the membership function for each element (Xu, 2007). Based on the operator, three intuitionistic fuzzy aggregation operators are developed. They are the interval-valued intuitionistic fuzzy Einstein weighted geometric operator, interval-valued intuitionistic fuzzy Einstein ordered weighted geometric operator and interval-valued intuitionistic fuzzy Einstein hybrid weighted geometric operator (Wang & Liu, 2013; Weize & Xinwang, 2013). The operators were successfully applied to a multi-criteria decision making problem of a multinational company (Garg, 2016).

2.2.2 Preferences of Objectives in Fuzzy Logic

In MOO problems, the objective is to find Pareto optimal set that has best possible values of all objectives among a set of feasible solutions. In these MOO problems, all individual objectives are considered as having an equal level of importance and no preferences are considered. As a result, many Pareto optimal solutions exist at the end of execution of the algorithm. In such situations, the decision maker (DM) often decides which solution is preferred. This exerts a burden on the DM (Xiaoning et al., 2010). Although the DM's preferences are given, they are quite often not well stated and the function governing the preferences is imprecise or even arbitrary (Cvetkovic & Parmee, 2002). Sometimes, DMs are only willing or able to provide uncertain information because of time pressure, insufficient data, lack of knowledge, or their limited experience related to the problem domain (Zhou et al., 2014). MOO algorithms should reflect the DM's preferences by incorporating their feelings about different solutions (J. Arora, 2004). Information of preference about objectives is used to guide the searching direction of the optimization algorithms which leads to find solutions in the desired regions and reduce the burden of the DM. Many different representation formats of preferences exist to express the information of pair alternatives such as the multiplicative preference relation, the multiplicative linguistic preference relation, the additive linguistic preference relation, the intuitionistic preference relation, and the fuzzy preference relation (Zhou et al., 2014).

A number of proposals to handle preferences have been reported in the literature. For example, Fonseca and Fleming (Fonseca & Fleming, 1993) utilized the goal attainment technique to accommodate preferences in the search process. Cvetkovic and Parmee (Cvetkovic & Parmee, 2002) used binary preference relations which are expressed as words and then

translated into weights to narrow the search. Greenwood *et al.* (Greenwood, Hu, & D'Ambrosio, 1997) used the idea of imprecisely specified multi-attribute value theory (ISMAVT) from imprecise ranking of objectives. Shen *et al.* (Xiaoning et al., 2010) proposed is a nine grade linguistic method. Each grade is assigned an appropriate numerical value according to its importance. For example, if f_i is more important than f_j, then f_j is minor than f_i. Terms used in this method (like a little more important, between a little more important and more important) are not properly chosen. Clear understanding and meaning between terms is not well expressed. In addition, numerical values require fuzzifier to transform them into the range of 0 and 1, thus introducing more computational complexity and additional time. Wang (Y.J. Wang, 2014) proposed a relative preference relation based on fuzzy preference relation in order to solve fuzzy multi-criteria decision making (FMCDM) problem. Generally, operation complexity of fuzzy preference relation is high due to fuzzy comparison on pair-wise. To overcome this problem, technique for order preference by similarity to ideal solution (TOPSIS) is aggregated with fuzzy preference relation. Zhou *et al.* (Zhou et al., 2014) developed a compatibility for the uncertain multiplicative linguistic preference relations based on linguistic continuous ordered weighted geometric averaging (LCOWGA) operator, and utilized it to determine the optimal weights of experts in the group decision making. The advantage of the compatibility index based on LCOWGA operator is that the decision maker has more flexibility to choose different values. Khan and Engelbrecht (Khan and Engelbrecht, 2007) proposed a new preference handling approach with seven linguistic terms for topology design of distributed local area network.

One of the objectives of this thesis is to analyze the 7-point preference scheme proposed by Khan and Engelbrecht (Khan and

Engelbrecht, 2007) through application to the problem of topology design of EN, as stated earlier. Therefore, a detailed discussion of the scheme is given below.

$$\mu P_m(s_i, s_j) = \begin{cases} 1 & \text{if } s_i \text{ is definitely preferred to } s_j \\ c \in (0.916, 0.999) & \text{if } s_i \text{ is strongly preferred to } s_j \\ d \in (0.832, 0.915) & \text{if } s_i \text{ highly preferred to } s_j \\ e \in (0.748, 0.831) & \text{if } s_i \text{ is considerably preferred to } s_j \\ f \in (0664, 0.747) & \text{if } s_i \text{ is moderately preferred to } s_j \\ g \in (058, 0.663) & \text{if } s_i \text{ is slightly preferred to } s_j \\ h \in (501, 0.579) & \text{if } s_i \text{ is mildly preferred to } s_j \\ 0.5 & \text{if there is no preference (i. e., indifference)} \\ i \in (0.421, 0.499) & \textit{if } s_j \textit{ is mildly preferred to } s_i \\ j \in (0.337, 0.42) & \text{if } s_j \text{ is slightly preferred to } s_i \\ k \in (0.253, 0.336) & \text{if } s_j \text{ is moderately preferred to } s_i \\ l \in (0.169, 0.252) & \text{if } s_j \text{ is considerably preferred to } s_i \\ m \in (0.085, 0.168) & \text{if } s_j \text{ is highly preferred to } s_i \\ n \in (0.001, 0.083) & \text{if } s_j \text{ is strongly preferred to } s_i \\ 0 & \text{if } s_j \text{ is definitely preferred to } s_i \end{cases}$$

The evaluation values can be seen as consisting of two groups: 1) s_i preferred to s_j and 2) s_j preferred to s_i. Here, s_i and s_j represent two optimization objectives. It is realized that the two groups are similar and equivalent since if the first objective is preferred to the second, then the second objective is not preferred to the first. This approach uses values between 0 and 1, which make it simple and easy to use with fuzzy logic, and consists of seven terms.

2.3 Metaheuristics

The word *heuristic* comes from an old Greek word *heuriskein*, which means the art of discovering new strategies (rules) to solve problems. The prefix *meta* is also a Greek word means upper level methodology (El-

Ghazali, 2009). The term *metaheuristic* can be defined as a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems. In other words, metaheuristic is a general purpose algorithmic framework that can be applied to different optimization problems with relatively few modification (Dorigo, Birattari, & Stutzle, 2006).

Optimization problems can be classified based on their complexity to *P* and *NP*. The class *P* consists of all problems that can be solved by a deterministic algorithm in a polynomial time. In other words, a known polynomial time algorithm exists to solve the problem. Hence, problems of class P are relatively easy to solve. On the other hand, NP class represents all problems that can be solved by using a non-deterministic algorithm in non-polynomial time. Many of the real world optimization problems are of type NP-hard where efficient algorithms do not exist. Generally, metaheuristic algorithms are used to solve optimization problems of the complexity class NP.

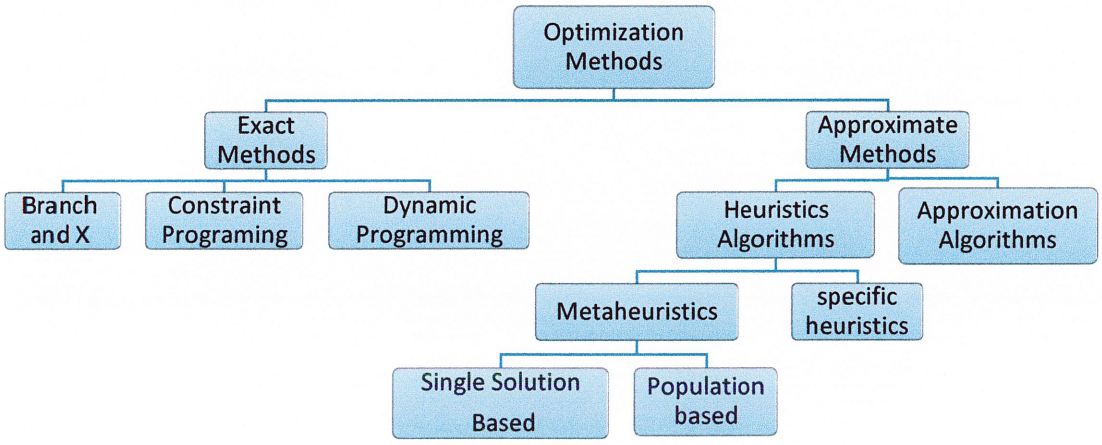


Figure (2.1)
Optimization Methods

Based on the complexity, optimization problems can be solved by exact methods and approximate methods as shown in Figure (2.1) . Exact methods obtain optimal solutions and guarantee the results, normally used for P problems. In contrast, approximate (or heuristic) methods produce high quality solutions in acceptable time, but there is no guarantee of finding a global optimal solution. In addition, heuristics methods are used to solve NP-complete problems. Furthermore, approximate methods can be classified to two sub classes: heuristic algorithms and approximate algorithms. Unlike the heuristics, the approximate algorithms provide provable solutions in provable time.

Heuristic algorithms are capable of obtaining acceptable solutions at acceptable costs in a wide range of problems. They can be classified into two families: specific heuristic and metaheuristic. Specific heuristics are specially designed to solve a specific problem, while metaheuristics are general purpose algorithms that can be applied to solve almost any optimization problem.

Metaheuristics are capable of handling large problem instances by providing satisfactory solutions in a reasonable time. Finding a global optimal solution or even bounded solutions are not guaranteed. Metaheuristics have received significant attention by researchers in different areas such as system modeling, engineering design, topology optimization, machine learning, data mining, scheduling (Al-Betar, Khader, & Zaman, 2012; Chun-Wei & Rodrigues, 2014; De Leon-Aldaco et al., 2015; El-Ghazali, 2009; Hsu-Chih, 2015; Lam & Li, 2010). Metaheuristic may be classified in different ways. One of the popular way is to classify into two families: single-solution (gradient) based and population-based. Single solution based algorithms like simulated annealing and simulated evolution maintain a single solution during the search. They have the power to concentrate the search in local regions.

On the other hand, population-based metaheuristics allow a better variation in the whole search space. For the purpose of this thesis, one single-solution algorithm which is 'simulated evolution' and one population-based algorithm which is 'ant colony optimization' will be used in the study. Therefore, a discussion is provided about these two algorithms.

2.3.1 Simulated Evolution

Simulated evolution (SE) is a general stochastic iterative meta-heuristic algorithm (R. M. Kling & Banerjee, 1989). It belongs to the category of algorithms which focus on the behavioral link between parents and offspring rather than the genetic link. SE operates on a single solution where the solution is seen as a population of movable elements. In addition, SE combines iterative improvement and constructive perturbation. By following a stochastic perturbation approach, it saves itself from getting trapped in local minima (Sait & Khan, 2003). This meta-heuristic has a lower execution time than genetic algorithm (Youssef, Sait, & Ali, 2003).

The pseudo-code for SE (Youssef, Sait, & Ali, 2003) is given in Figure (2.2). For one solution, it operates a sequence of three steps: evaluation, selection, and allocation in each iteration until some stopping condition is satisfied. These steps are discussed below.

Initialization

In this initial step, a feasible solution that satisfied all conditions and constraints of the specified problem is generated. This initial solution may be generated randomly or using a constructive algorithm.

Algorithm *Simulated_Evolution*$(B, E_{initial}, StoppingCondition)$

NOTATION

B = Bias Value.

e_i= individual cell in E.

C_i= Current Cost of i^{th} cell in E.

S = Queue to store the selected cells.

E = Complete Solution.

O_i= Lower bound on cost of i^{th} cell in E.

g_i= Goodness of i^{th} cell in E.

ALLOCATE (e_i, E_i) = Function to allocate e_i in partial solution E_i

Repeat

EVALUATION: **ForEach** $e_i \in E$ **DO**
begin
$g_i = \frac{O_i}{C_i}$
end

SELECTION: **ForEach** $e_i \in E$ **DO**
begin
IF $Random > Min(g_i + B, 1)$ **THEN**
begin
$S = S \cup e_i$;
Remove e_i from E.
end
end

Sort the elements of S

ALLOCATION: **ForEach** $e_i \in S$ **DO**
begin
ALLOCATE (e_i, E_i)
end

Until *Stopping Condition is satisfied*

Return Best solution.

End $(Simulated\_Evolution)$

Figure (2.2)
Pseudo-code for SE algorithm (Youssef, Sait, & Ali, 2003)

Evaluation

A solution generated by SE is seen as a population of movable elements, where an element is considered as an individual component of the solution. Different combinations of these elements lead to have different unique solution. Each element has a goodness (fitness) value in its current location. The goodness value g_i is a ratio between its optimum cost and its actual cost and therefore, the goodness value belongs to the interval of [0, 1]. It is a scale of how near each element is to its optimum position. The higher the goodness of an element, the closer is that element to its optimum location with respect to the current configuration (Youssef et al., 2003). The goodness value is defined as below.

$$g_i = \frac{O_i}{C_i} \tag{2.12}$$

where O_i is the optimal cost of the element e_i and C_i is the actual cost of the same element in its current location. Note that O_i remains constant through all iterations. Generally, the value of O_i is obtained in the initialization phase by mathematical approximation. However, C_i is calculated by a user-defined evaluation function and changes in each iteration.

Selection

Selection is the process of selecting those individual elements which are unfit (in bad position) in the current solution. For this purpose, the input for this phase is the existing solution and the goodness value for each element e_i. Elements with low goodness values have higher probabilities of getting selected for removal from the solution. Another important parameter is involved in the selection process, which is the *bias (B)*. It is used to correct errors during the estimation of goodness measure

by inflating or deflating the goodness of elements. A high positive value of B decreases the probability of selecting large number of elements for removal. As a result, the selection set is small which speeds-up the algorithm, but reduces the quality of solution and increase the risk of being trapped in local minima. Similarly, a low or negative value of B leads to have a large set of selected elements for removal, which causes to have longer runtimes, but may produces better solutions (Youssef et al., 2003). It is important to mention herein that the bias value is problem specific and user defined one in the interval of [-1, 1] and the best suitable value is found based on trial and error. Furthermore, in this phase, a random number is generated in the range of [0, 1]. Finally, the values of the goodness of element, the bias, and the random value is used to perform a comparison as shown in Figure (2.2). The outcome of this operation decides whether a particular element is to be removed or not.

Allocation

The allocation phase is the most important step in the algorithm. It has a huge impact on the quality of the solution. The objective of this phase is to mutate the population by altering the positions of selected elements in the selection phase. For each selected element, several trial moves are performed in order to reduce the cost of a new solution. Several allocation schemes to perform the moves are proposed in (R. M. Kling & Banerjee, 1989). From these schemes, *sorted individual best fit* is the best in terms of complexity, run time requirements, and quality of solution. The sorted individual best fit is simpler, have an accepted run-time, and with marginal improvement in the quality of solution (Youssef et al., 2003). The scheme works as follows: at first, the selected elements are sorted in descending order and placed in a queue, with respect to their connectivity with the existing solution. The sorted elements are moved

one by one and trial moves are made for all available empty locations. Finally, the element is placed in a position where low cost is achieved. The process is repeated until the queue is empty.

In SOO, there is only one global optimum solution. However in MOO, there is a set of solutions known as *Pareto Optimal* (PO) or *Pareto Front set* which are trade off non-dominated solutions and all of them are the global optimum solutions (Bandyopadhyay, Saha, Maulik, & Deb, 2008). SE is an iterative single solution based algorithm that generates a single solution in each iteration and cannot produce PO solutions by its nature. However, the generated solutions in all iterations are known as *Optimal Solution Set* (S). To overcome this problem, a number of metrics have been proposed in MOO for defining PO set and for the assessment of MOO algorithms (Jiang, Ong, Zhang, & Feng, 2014). The most simple and easiest one is the *Capacity* metric. According to this method, the set S must be archived and then the number of non-dominated solutions (having highest goodness value with unique objectives) in S are counted that conforms to the predefined requirements. This metric is also known as Overall Non-dominated Vector Generation (ONVG) metric as shown in Equation (2.13).

$$ONVG(S) = |S| \tag{2.13}$$

SE was originally proposed to solve the design automation of very large scale integration (R. M. Kling & Banerjee, 1989; Sait, El-Maleh, & Al-Abaji, 2006; Sait & Khan, 2003; Youssef et al., 2003). In 2009, (Salman A. Khan, 2009) has adopted the algorithm to design the network topology and produced solution with good quality. In 2013, Mutingi *et al.* (Mutingi & Mbohwa, 2013) presented a fuzzy SE algorithm in the context of staff planning and scheduling in a home care environment in

order to provide medical services to patients at their own homes. The objectives were to decide which patients to assign to each staff and to generate the best route trip for each worker to satisfy the time window preference of the patients. The results showed that the approach was promising. In 2014, Thabti *et al.* (Thabti, Youssef, Mahjoub, & Meddeb, 2014) used SE in Virtual Private Network Design Problem in order to generate a better network topology with the Brach and Cut algorithm and produced optimal solutions.

2.3.2 Ant Colony Optimization Algorithms

The idea behind the ant colony optimization algorithm (ACO) was inspired from the behavior of the real ants in order to solve optimization problems. The algorithm was proposed by (Dorigo & Stützle, 2004). It is a population based meta-heuristic that can be seen as multi-agent system in which each agent is represented by a real ant. ACO tries to mimic the behavior of real ants to find the shortest path between two points.

In reality, ants try to find the shortest path from the food to a nest. To do so, ant colonies use a positive feedback mechanism called *stigmergy*. During this process, a chemical volatile substance trail called *pheromone* is left on the ground. This trail is used to guide the coming ants to choose the appropriate path. The probability of selecting a particular path increases by increasing the amount of pheromone on the path. Therefore, a path is chosen based on the quantity of pheromone. The following ants will release new pheromone on the selected path, which allow the path to be more attractive for future ants. At last, all ants will use the same path which is the shortest path (Birattari, Pellegrini, & Dorigo, 2007; El-Ghazali, 2009).

Set parameters, Initialize phenome trails

While *termination condition not met* ***do***

Solution construction using the pheromone trail

Daemon actions (optional)

Update the pheromone trails

End while

Output*: Best solution found or a set of solutions*

Figure (2.3)
Pseudo-code of the ACO meta-heuristic

To simulate the behavior of real ants, a number of computational agents called artificial ants are considered and the optimization problem is transformed to a graph, where each ant generate a solution by walking though the vertices. The accepted solution is stated in terms of feasible paths on the graph. In 2006, ACO has been generalized into meta-heuristic for combinatorial optimization problems (Dorigo et al., 2006). In addition, it is the most successful example of swarm intelligence and it has been applied successfully to many combinatorial optimization problems such as designing, routing, scheduling, and assignment (El-Ghazali, 2009). ACO is an iterative algorithm where the framework consists of three phases excluding the initialization. They are: solution construction, daemon actions (optional), and pheromone update as shown in Figure (2.3). Briefly, after initialization, in each iteration, the artificial ants construct a number of solutions. Then these solution are improved by local search (optional). Finally, the pheromone is updated. The following is the detailed information about the three phases:

Construction of solutions

In this phase, each ant in the set constructs a solution iteratively by adding elements which are the feasible solution components (paths) to the current partial ones until a complete solution (graph) is derived without violating any of the constraints of the problem. The construction process is a stochastic mechanism and is biased by a local decision policy. According to this policy, the selection of a particular path on the graph depends on pheromone trail and heuristic value (Birattari et al., 2007; El-Ghazali, 2009). The pheromone trail is used for memorizing the past experience by storing the characteristics of good solutions. This will help and guide the ants to construct new solutions.

The other part of information required in constructing the solutions is the problem-dependent heuristic information. This information gives more clues and hints to ants in their decisions. In each step, the ants make a probabilistic decision on their movement on the graph. This information is based on pheromone and heuristic information, where the higher the pheromone and heuristic information for a given edge, the higher the probability of that particular edge to be selected. In addition, this information is changed dynamically by ants during the run time. Various pheromone adaption processes exist across different ACO systems (Dorigo et al., 2006; El-Ghazali, 2009). The best rule among them is the one of ant system (AS) as in Equation (2.14):

$$P_{ij}^{k}(t) = \frac{\left[\tau_{ij}(t)\right]^{\alpha_{ant}}\left[\eta_{ij}\right]^{\beta_{ant}}}{\sum_{l\in N_i}\left[\tau_{ij}(t)\right]^{\alpha_{ant}}\left[\eta_{ij}\right]^{\beta_{ant}}} \tag{2.14}$$

where $P_{ij}^{k}(t)$ is the probability of selecting a link l between node i and j in the set of neighbors N_i of the node i for the ant k, η_{ij} is the heuristic

value and τ_{ij} is pheromone of the link l. α_{ant} and β_{ant} are the coefficient of pheromone content and heuristic value respectively.

Daemon Actions

This phase is an optional process and highly problem specific. Daemon actions can be applied after solutions construction and before pheromone update. The aim is to improve the solutions by performing local search (Dorigo et al., 2006).

Pheromone Update

The goal of this phase is to increase the pheromone values associated with good solutions and to drop down those are related to bad ones. Usually, this can be achieved by applying *pheromone evaporation* to decrease all unwanted pheromone values and by increasing the pheromone levels associated with selected set of good solutions. In evaporation process, the pheromone trail drops automatically, where each pheromone value is declined by a fixed proportion.

In literature, several pheromone updating schemes have been proposed (El-Ghazali, 2009). One of those schemes is the elitist ant, which has been applied in this thesis. The elitist ant strategy was initially proposed by (Dorigo, Maniezzo, & Colorni, 1996). The pheromone updating idea is that the trail of the best tour, so reinforced, will direct the search of all the other ants in probability toward a solution composed by some edges of the best tour itself (Dorigo et al., 1996). In another way, the best solution found so far will update the pheromone and direct the search, where a maximum pheromone value has a maximum probability to be selected.

2.4 Concluding Remarks

This chapter reviewed the literature relevant to multi-objective optimization, fuzzy logic and metaheuristics algorithms. First, the chapter discussed the literature related to optimization and its important issues such as the classification of optimization problems and the multi-objective optimization. Later, the chapter provided an overview of fuzzy logic in the context of multi-objective optimization. In addition, preferences of objectives along with the preference scheme that used in this work were discussed. Later on, the chapter provided the literature related to the definition and the derivation of metaheuristics. Finally, the chapter provided an overview of the literature related to Simulated Evaluation algorithm and Ant Colony Optimization algorithms.

Chapter Three
Topology Design of Enterprise Networks

3.1 Background	36
3.2 Literature Review	42
3.3 Problem Definition	45
3.4 Design Objectives	45
3.4.1 Network Reliability	45
3.4.2 Average Network Delay	46
3.4.3 Monetary Cost	46
3.4.4 Number of Hops	47
3.5 Assumptions	47
3.6 Constraints	48
3.7 Einstein Operator based approach for Enterprise Network Topology Design Problem	49
3.7.1 Preference Rules	52
3.7.2 Preference Rules for EN Topology Design	53
3.8 Characteristics of Test Cases	55
3.9 Conclusion	56

Chapter Three

Topology Design of Enterprise Networks

In this chapter, the optimization problem considered in this thesis is discussed. It is important to mention that the problem model considered herein has already been proposed in previous studies. The model here is presented in detail for the sake of completeness and better understanding of the problem domain. Initially, an overview of the problem with concepts and terminology is given. This is followed by the literature review of the problem. Then, a formal description of the problem is provided. Later on, a detailed discussion on the four design objectives is provided. Finally, assumptions and constraints related to the problem are discussed.

3.1 Background

In EN, a large number of computing devices are interconnected to provide services such as resource sharing and information exchange. There are two types of these devices. Hosts or end-systems and network active elements. Hosts may consist of personal computers, workstations, mobile phones, faxes, printers, etc. Network elements such as routers, switches, hubs, and gateways are used to interconnect hosts. As the network grows, it may be necessary to connect two or more sub-networks. These sub-networks are often called network segments. The *root* network is the one that interconnects network segments and is referred to as "internetwork". In general, internetwork can be divided into two levels. The first level is called local area network (LAN), which allows users to access hosts or local servers. The second level is the

backbone, which is responsible for packet delivery from source to destination.

In modern enterprise companies, a hierarchal structure is applied in designing the root network. In such a network, sub-networks or segments in the same physical location are connected together into a LAN. Then, several LANs in different geographical locations are connected through routers and layer 3 switches in order to establish the backbone of the network. The resulting network is known as and Enterprise Networks (EN). Network components (LANs and routers or switches) must be arranged in a specific way in order to work in best performance. Basically, the organization depends on hardware specifications and other characteristics and requirements of the network. Such organizing design is known as network topology design (NTD) problem. The typical layout of EN is shown as in Figure (3.1).

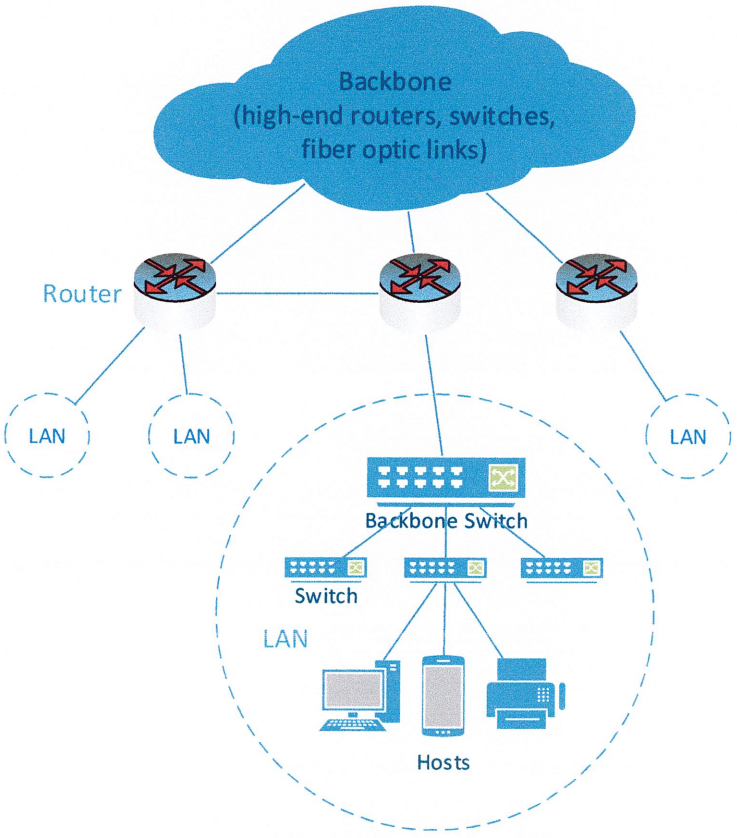


Figure (3.1)
Typical Enterprise Network

In order to design a topology of a network, different network design architecture may be applied (McMillan, 2011). An overview of each type is given as follows:

Flat Network Design

It is an early network design, where layer 2 devices such as hubs and switches in a flat network provide little opportunity to control broadcasts or to filter undesirable traffic. As more devices and applications are added to a flat network, response times degrade until the network becomes unusable as shown in Figure (3.2).

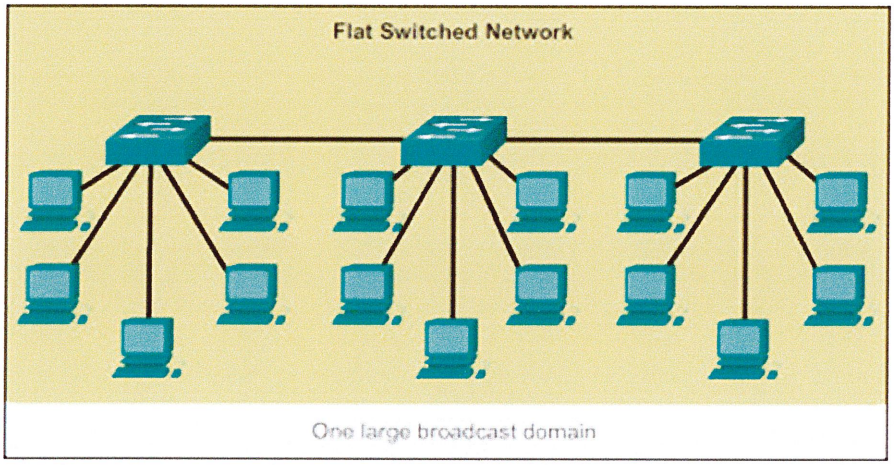


Figure (3.2)
Flat Network Design

Modular Network Design

A module is a component of a composite structure. Modular network design involves creating modules that can then be put together to meet the requirements of the entire network. A modular design for a network has many benefits, including the following:

- It is easier to understand and design smaller, simpler modules rather than an entire network.
- It is easier to troubleshoot smaller elements compared to the entire network.

- The reuse of blocks saves design time and effort, as well as implementation time and effort.
- The reuse of blocks allows the network to grow more easily, providing network scalability.
- It is easier to change modules rather than the entire network, providing flexibility of design.

Modular network design consist of three layers:

1. Core layer: Provides optimal transport between sites and high-performance routing. Due the criticality of the core layer, the design principles of the core should provide an appropriate level of resilience that offers the ability to recover quickly and smoothly after any network failure event with the core block.
2. Distribution layer: Provides policy-based connectivity and boundary control between the access and core layers.
3. Access layer: Provides workgroup/user access to the network.

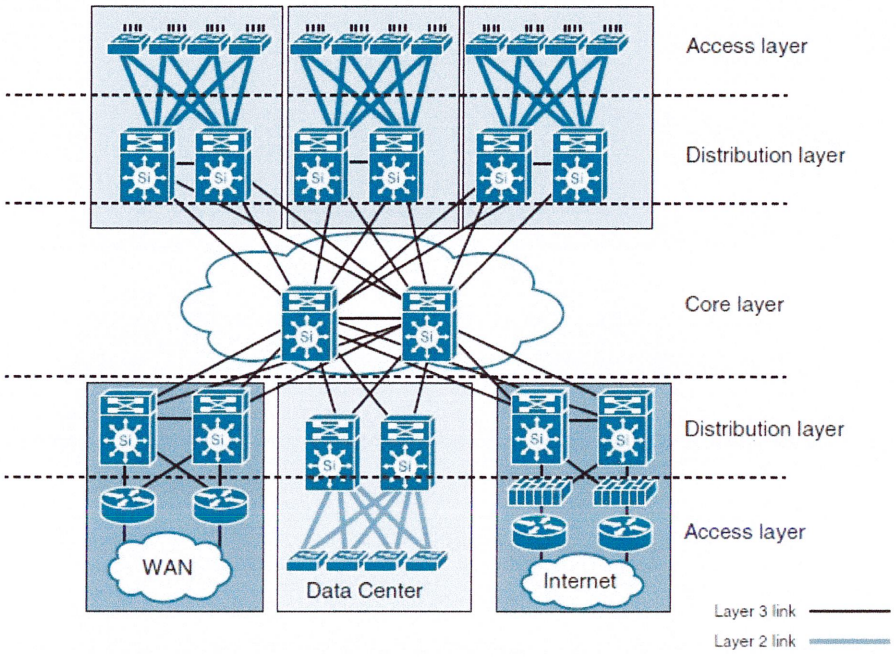


Figure (3.3)
Typical Modular Enterprise Network Design

Hierarchical Network Design

The hierarchical network design model is a subset of the modular network design. It breaks the complex flat network into multiple smaller and more manageable networks. Each level or tier in the hierarchy is focused on a specific set of roles. This design approach offers network designers a high degree of flexibility to optimize and select the right network hardware, software, and features to perform specific roles for the different network layers. It consists of three layers as in modular design. There are several benefits of the hierarchical design model:

- Networks based on the hierarchical model are easy to manage, troubleshoot, and support.
- Networks based on a hierarchical model are efficient in performance because of the use of advanced routing features such as route summarization.
- Because of its simple design, the hierarchical model offers benefits such as cost efficiency and ease of manageability.
- The hierarchical model is easy to scale as compared to other network architecture models because it is divided into modules.
- The hierarchical model makes it possible to provide high-speed switching of data between workstations and internetwork services.

The two primary and common hierarchical design architectures of enterprise networks are the three-tier and two-tier layers models. Three-tier model is typically used in large enterprise networks, which are constructed of multiple functional distribution layer blocks as shown in Figure (3.4). On the other hand, the two-tier model is more suitable for small to medium-size networks (ideally not more than three functional disruption blocks to be interconnected), where the core and distribution

functions can be combined into one layer, also known as collapsed core-distribution architecture as shown in Figure (3.5).

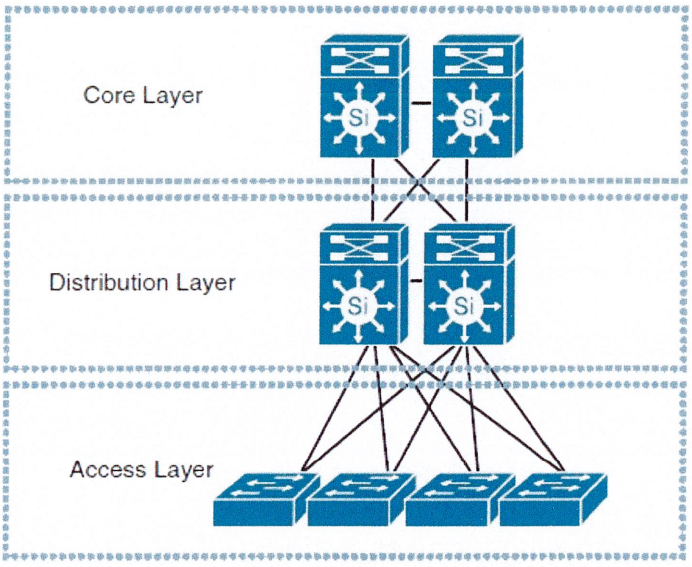


Figure (3.4)
Three-Tier Network Design Model

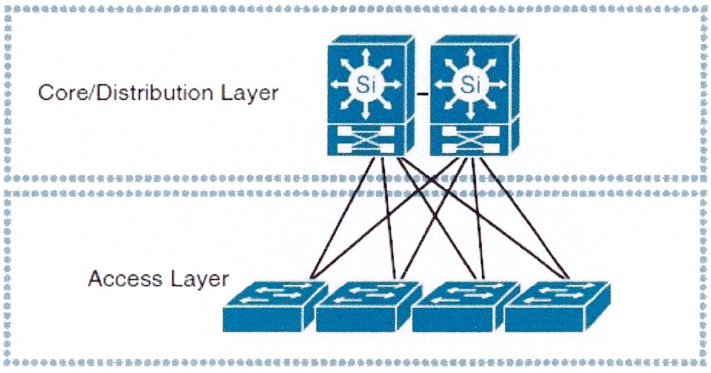


Figure (3.5)
Two-Tier Network Design Model

3.2 Literature Review

Network topology design (NTD) problem has received a notable attention in past years. Researchers try to develop novel techniques to find the best possible topology in order to solve this complex optimization problem (Estepa et al., 2011; Gobjuka & Breitbart, 2010; Gupta & Dally, 2006; S. A. Khan, 2013; Kumar & Banerjee, 2011; Silva, de Souza Britto Jr, Oliveira, & Koerich, 2013; Sung et al., 2011). They look at the problem from different perspectives and based on that goal they assign different number of objectives. (Silva et al., 2013) proposed a topology design of multilevel network with consideration of the number and the geographic location of devices as well as the link among them. The design built each network level separately. Hence, the ultimate goal was to minimize the cost of topology design. The proposed algorithm reduced the cost up to 40%. (Kumar & Banerjee, 2011) considered performance related objectives in their network topology design. They used two objectives which are the network cost and the network delay, where the aim was to minimize both objectives. The generated topologies were reliable for single link failure and guaranteed that the maximum load of a link will not exceed the link capacity. Estepa et al. (Estepa et al., 2011) proposed a network design that includes the network downtimes negative impact on organizational productivity. The design tries to minimize the overall expected cost, which is the capital expenditure (CAPEX) cost plus the operational expenditure (OPEX) cost, which is the cost of unproductiveness due to network downtimes over the expected period of network operation. The OPEX can be estimated through the network's reliability model and cost model of each network element. Hence, the design tries to minimize the CAPEX cost and OPEX cost. The optimized design is the trade-off between the implementation

cost CAPEX and reliability obtained by redundancy. The proposed design highlighted those nodes that have higher impact on organization productivity and provided a practical approach for network topology design for high-risk environments. Another example of NTD is what was proposed by Wang and Chang in 2008 (C.S. Wang & Chang, 2008). They established a model with multiple objectives and multiple criteria in order to solve the NTD problem. The proposed model can be used to solve both construct and extend network topology problems with shared risk link group constraints. The authors considered four optimizing objectives such as network capacity, linkage degree, cost, and reliability. Also, they assigned multiple criteria such as network capacity, reliability, and linkage degree for each objective. Kleeman and other authors in (Kleeman et al., 2012) proposed a design for real-world complex network communication problem called multicommodity capacitated network design problem (MCNDP). In order to generate the topology for this network, the authors use three objectives which are the total cost, delay, and average number of hops. The aim was to minimize the three objectives.

NTD problems are complex and have been classified as NP-hard problem (Gobjuka & Breitbart, 2010; Kumar & Banerjee, 2011). The complexity is further increased by involving multiple objectives that are conflicting in nature and required to be optimized simultaneously. In addition, NTD problems are complex with a huge search space. For a network of n nodes, there exist $2^{\left(\frac{n(n-1)}{2}-1\right)}$ different topologies (Salman A. Khan, 2009). For example, a network with $n = 10$ has 2^{44} distinct topologies. Hence, NTD problems required computational power and time in order to find the best topology. Apart from this, constraints that are related to the problem further increase the complexity of the problem.

Therefore, heuristic approximation methods are the most appropriate to solve NTD problems and to produce high quality solutions in acceptable time. Evolutionary algorithms have been proposed to address the NTD problems. For example, applying the Genetic Algorithms (GAs) have been reported in many studies (Coello, 1999; Elbaum & Sidi, 1996; Ersoy & Panwar, 1993; Estepa et al., 2011; Fonseca & Fleming, 1993; Pierre & Legault, 1998; C.-S. Wang & Chang, 2008). Kleeman *et al.* in 2012, applied the nondominated sorting genetic algorithm (NSGA-II) for MCNDP (Kleeman et al., 2012). Silva *et al.* in 2013, proposed a NTD by developing GA and Particle Swarm Optimization (PSO) and compared both results (Silva et al., 2013). Kumar *et al.* in 2011, presented two algorithms to solve the NTD problem using two objectives (Kumar & Banerjee, 2011): Pareto Converging Genetic Algorithm (PCGA) and Pareto Branch Exchange (PBE) heuristic. Aparicio-Pardo *et al.* in 2012 proposed a very efficient heuristic algorithm named as Greedy Approach with Reconfiguration Flattening to solve the virtual topology design problem in optical networks (Aparicio-Pardo, Skorin-Kapov, Pavon-Marino, & Garcia-Manrubia, 2012). Similarly, Sung et al. in 2011 developed a greedy heuristic to determine grouping of hosts into VLANs in EN (Sung et al., 2011). Gobjuka and Breitbart in 2010 designed several heuristic algorithms to discover the topology of Ethernet networks: One-Node algorithm, Half-Complete-RSs algorithm, and termed Connection algorithm (Gobjuka & Breitbart, 2010). Other metaheuristics such as Simulated Evolution (Salman A. Khan & Engelbrecht, 2007; Youssef, Sait, & Khan, 2002), Particle Swarm Optimization (PSO) (S. Khan & Engelbrecht, 2012), Ant Colony Optimization (ACO) (Salman A. Khan, 2009), and Simulated Annealing (S. Khan & Engelbrecht, 2009) have also been utilized.

3.3 Problem Definition

The EN topology design problem studied in this work was formulated by Khan & Engelbrecht (Salman A. Khan & Engelbrecht, 2007). The objective is to find a quality feasible topology of near optimum of EN that satisfies all design objectives and technical constraints related to the problem. With assumption that the LANs topology exist, the goal is to design backbone topology that connects all nodes without redundancy. All nodes should be connected in a tree structure without any loop.

The optimality is measured with respect to four objectives: network reliability, average network delay, monetary cost, and maximum number of hops between any source-distention pair. The search targets feasible topologies which maximizes the first objective and minimizes the rest three objectives In addition, the analysis is performed with respect to eight preference rules. The rules have been categorized into two levels.

3.4 Design Objectives

This section describes the design objectives applied in this thesis. Four objectives need to be optimized simultaneously. The description of each objective is as follows.

3.4.1 Network Reliability

Network reliability should be maximized. It can be defined as the probability of occurrence of an event in which each node communicates with every other node in the network. For the purpose of this thesis, the reliability of tree topology may be expressed by the product of the

reliabilities of all links present the topology. The reliability can be calculated using the Equation (3.1) where R_i is the reliability of link i and R_s is the reliability of the network.

$$R_s = \prod_{i=1}^{L} R_i \tag{3.1}$$

3.4.2 Average Network Delay

The aim of this objective is to minimize the network delay sustained on a packet while transmission from a source node to destination node. In order to calculate the average network delay, both link delay and network device delay should be considered, where the network average delay is the summation of delays over all links and network devices. If a given link connects two local sites i and j, Equation (3.2) is used to calculate to average network delay (Salman A. Khan, 2009).

$$D = \frac{1}{\Upsilon}\sum_{i=1}^{L}\frac{\lambda_i}{\lambda_{max,i} - \lambda_i} + \frac{1}{\Upsilon}\sum_{i=1}^{d}\sum_{j=1}^{d}\Upsilon_{ij}B_{ij} \tag{3.2}$$

Where d is the total number of network devices in the network, Υ is the total traffic in the network, L is the number of links in the topology, λ_i is the traffic on link i in the bits per second (bps), $\lambda_{max,i}$ is the capacity of link i in bps, delay per bit in a link ij is $B_{ij} = b_{ij}/\omega$ where b_{ij} is delay per packet, and ω is the average packet size in bits.

3.4.3 Monetary Cost

In order to build a quality network topology, redundancy of network devices (Estepa et al., 2011) seem to be the best choice. However, it adds extra monetary cost, which is not desirable from financial decision

makers. Hence, the goal is to design a network topology with minimum cost. Since the number of devices remains the same in any topology, then the only cost that affects is the cabling cost. Hence, the following equation is utilized to calculate the monetary cost where $Length$ represents the total length of used cable and C_{cable} refers to the cost per cable unit.

$$Cost = Length \times C_{cable} \quad (3.3)$$

3.4.4 Number of Hops

The fourth objective is the maximum number of hops between any pair of source-destination. The aim is to minimize number of hops. During the transmission, when a packet crosses a network, one hop is counted. Routing information protocol (RIP) uses hop count to measure the distance between a source to a destination, and maximum hops allowed in a path are 15. If the hop count is greater than 15, then the destination is unreachable.

3.5 Assumptions

The following are the assumptions in EN topology design problem:

- A tree topology is considered for network design.
- The root node is a switch node acting like a collapsed backbone with given required interfaces.
- A node represents a LAN. The node refers to network devices (switch, router or gateway) from layer 2 and above.
- All nodes are fault free. Only link may get failure.

- Distances between nodes are predefined. The location of a node can be represented by its Cartesian coordinates with respect to some reference point.
- Each link is bidirectional.
- The reliability of each link is known.
- The maximum utilization of any link should remain within a desired threshold 60%.
- The capacity of a network device is measured by the number of ports available on it. The ports are used to connect the users.

3.6 Constraints

The following are constraints are applied.

- C_1: The maximum number of nodes attached to any network device should not exceed its capacity.
- C_2: The traffic on any link must not exceed its maximum bandwidth.
- C_3: Routing information protocol (RIP) is used to count the number of hops.
- C_4: Maximum number of hops from any source to any destination should not exceed the maximum allowed number which is 15 in practical.
- C_5: Other constraints that are related to designers such as certain nodes must be leaf/terminal nodes, others should be

interior nodes of the tree, and some cannot be attached directly to the central backbone.

3.7 Einstein Operator based Fuzzy Logic Approach for EN Topology Design Problem

Topology design of EN is a complex problem which requires simultaneous optimization of a number of design objectives. Important objectives include monetary cost, network average delay, maximum number of hops between a source-destination pair, and network reliability. Some of these objectives are conflicting by nature, where optimizing one objective has negative impact on others. Thus, a trade-off between the conflicting objectives is required. Fuzzy logic is a convenient method of combining conflicting objectives and expert human knowledge.

The rest of this section describes how fuzzy logic is employed in combining the four conflicting objectives into a single overall objective. This overall objective estimates the quality of a solution in terms of membership of a given topology to the fuzzy set of quality topologies. For simplicity, the following discussion uses the terms cost, delay, hops, and reliability. The goal is to find a high quality solution, represented by the following fuzzy rule.

IF *cost is low* AND *delay is low* AND *hops is low* AND *reliability is high* THEN *the topology is good*

In order to calculate the quality of the overall solution using fuzzy logic, the quality of individual objectives needs to be evaluated first through membership functions. Thus, a membership function μ needs to

be defined for each objective. It is a value in the interval [0,1]. This process is described below.

To calculate the fuzzy value for the cost objective, the membership function of link cost in the topology respect to *near optimum monetary cost* is determined as follows: select the minimum and maximum costs from the cost matrix which shows the minimum and the maximum costs for each possible link for each test case alone. Consider them as the lower and upper bonds and termed as "CostMin" and "CostMax" respectively. Based on these values the membership function μ_c is calculated as follows:

$$\mu_c = \begin{cases} 1 & \text{if } Cost \leq CostMin \\ \dfrac{CostMax - Cost}{CostMax - CostMin} & \text{if } CostMin < Cost \leq CostMax \\ 0 & \text{if } Cost > CostMax \end{cases}$$

Similarly, the member function of a link μ_d with respect to *near optimum delay* can be found as follows: the lower limit called "DelayMin" was taken from the initial solution for each test. The upper bound called "DelayMax" which is calculated based on the maximum traffic of a possible link between a source and destination pair for each test case alone.

$$\mu_d = \begin{cases} 1 & \text{if } Delay \leq DelayMin \\ \dfrac{DelayMax - Delay}{DelayMax - DelayMin} & \text{if } DelayMin < Delay \leq DelayMax \\ 0 & \text{if } Delay > DelayMax \end{cases}$$

For the third objective, the member function of a link μ_h respect to *maximum number of hops* can defined as: the lower bound called "HopsMin" which is 1, and upper bound called "HopsMax" which is 15,

where the destination is not reachable after this number, μ_h is computed using below.

$$\mu_h = \begin{cases} 1 & \text{if } Hops \leq HopsMin \\ \dfrac{HopsMax - Hops}{HopsMax - HopsMin} & \text{if } HopsMin < Hops \leq HopsMax \\ 0 & \text{if } Hops > HopsMax \end{cases}$$

Finally, the membership function for reliability μ_h objective is calculated by: the lower bound "RelMin" is 0.001 and the upper bound "RelMax" is found based on previous experiments in (Salman A. Khan, 2009) as shown below.

$$\mu_h = \begin{cases} 1 & \text{if } Rel \geq RelMax \\ \dfrac{RelMax - Rel}{RelMax - RelMin} & \text{if } RelMin < Rel \leq RelMax \\ 0 & \text{if } Hops < RelMin \end{cases}$$

After obtaining the membership functions of the four objectives, the next phase is to combine these functions into an overall membership function of "topology is good" using fuzzy logic. A good topology is the one that is characterized by the above fuzzy rule. This fuzzy rule can be mathematically represented using the Einstein's operator. The Einstein's operator given in Equation (2.11) consists of numerator and denominator that takes two membership functions as an input and produces the solution membership value as an output. For the simultaneous optimization, the Einstein's operator should consider all four design objectives at the same time. This can be achieved by applying the operator recursively as shown below.

$$E(\mu_c, \mu_d, \mu_h, \mu_r) = E(E(E(\mu_c, \mu_d), \mu_h), \mu_r) \qquad (3.4)$$

The membership value of Equation (3.4) represents the membership values of the generated solution. It is a value in the interval [0,1], where 0 indicates that the solution is not valid and not acceptable and 1 shows that the solution is the global optimum solution. The solution which results in the maximum value for Equation (3.4) is reported as the best solution found.

Therefore, the EN topology design problem can be formulated mathematically as follows:

minimize Cost

minimize Delay

minimize Hops

maximize Reliability

subject to c_1, c_2, c_3, c_4, c_5

3.7.1Preference Rules

The 7-point preference scheme has been used in this thesis to define a number of preference rules. These rules are divided into three categories as described below.

The first category contains preference rules that involve all four objectives. The following are some example as follows:

- Cost is strongly preferred over the other three objectives.
- Delay is highly preferred over the other three objectives.
- Reliability is considerably preferred over the other three objectives.

The second category consists of preference rules that involve three objectives. Some examples are listed below:

- Cost is mildly preferred over hops and delay.
- Reliability is slightly preferred over cost and delay.
- Hops is considerably preferred over cost and reliability.

The third category is of form where two objectives are involved. Some examples are listed below:

- Reliability is strongly preferred over cost.
- Cost is slightly preferred over delay.
- Delay is mildly preferred over hops.

3.7.2Preference Rules for EN Topology Design

The main fuzzy rule and the preference rules (PRs) have been combined. As a result, a huge number PRs can be generated. For the purpose of this thesis, eight PRs are generated in order to evaluate the 7-point preference scheme. These rules are categorized into two levels. The first level (level 1) consists of one rule. However, level 2 is the combination of two preference rules.

The following are the Level 1 preference rules:

1. PR1: Cost is strongly preferred over the other three objectives. Then the objective function can be calculated as:

$$\mu_s = E(0.916\mu_c, 0.084\mu_d, 0.084\mu_h, 0.084\mu_r)$$

2. PR2: Delay is highly preferred over the other three objectives. Then the objective function can be calculated as:

$$\mu_s = E(0.168\mu_c, 0.832\mu_d, 0.168\mu_h, 0.168\mu_r)$$

3. PR3: Reliability is considerably preferred over the other three objectives. Then the objective function can be calculated as:

$$\mu_s = E(0.2\mu_c, 0.2\mu_d, 0.2\mu_h, 0.8\mu_r)$$

4. PR4: Cost is moderately preferred over hops and reliability. Then the objective function can be calculated as:

$$\mu_s = E(0.7\mu_c, 0.3\mu_d, 0.5\mu_h, 0.3\mu_r)$$

On the other hand, the level 2 preference rules can be described as follows:

1. PR5: Delay is mildly preferred over reliability AND Hops is slightly preferred over cost. Then the objective function can be calculated as:

$$\mu_s = E(0.42\mu_c, 0.501\mu_d, 0.58\mu_h, 0.499\mu_r)$$

2. PR6: Cost is mildly preferred over hops AND Delay is moderately preferred over reliability. Then the objective function can be calculated as:

$$\mu_s = E(0.555\mu_c, 0.7\mu_d, 0.445\mu_h, 0.3\mu_r)$$

3. PR7: Cost is slightly preferred over reliability AND hops is considerably preferred over delay. Then the objective function can be calculated as:

$$\mu_s = E(0.663\mu_c, 0.2\mu_d, 0.8\mu_h, 0.337\mu_r)$$

4. PR8: Reliability is highly preferred over hops AND delay is slightly preferred over cost. Then the objective function can be calculated as:

$$\mu_s = E(0.4\mu_c, 0.6\mu_d, 0.17\mu_h, 0.83\mu_r)$$

3.8 Characteristics of Test Cases

The test cases used in this thesis have been used previously (Salman A. Khan, 2009). Theses test cases were used to evaluate the performance of the Einstein's operator and the 7-point preference scheme based on two algorithms (AFSE and FACO). The test cases represent networks consisting of local sites. Traffc generated by each local site for these test cases was collected from real sites, and costs of the network cables were collected from vendors. Other characteristics, such as the number of ports on a network device and its type are listed in Table (3.1).

Table (3.1)
Network characteristics assumed for experiments

Parameter	Characteristic
Cost of fiber optic cable	$ 5 per meter
Delay per bit due to networking device	250 $\mu sec.$
Maximum traffic on a link allowed	60
Average packet size	500 bytes
Type of Networking device	Router, switch, or hub
Number of ports on a networking device	4, 8, or 12

Five test cases were used for experimental work. For these test cases, the number of local sites ranged between 15 (denoted by n15) and 50 (denoted by n50). Table (3.2) summarizes the features of these test cases.

Table (3.2)
Characteristics of test cases used in experiments. Min Cost is in US$, Min Delay is in milliseconds, and traffic is in Mbps

Test Case	# of Local sites	Min Cost	Min Delay	Max Reliability	Traffic
n15	15	4640	2.14296	0.868746	24.63
n25	25	5120	2.14296	0.785678	74.12
n33	33	8158	2.15444	0.72498	117.81
n40	40	9646	2.08757	0.675729	144.76
n50	50	11616	2.08965	0.611117	164.12

3.9 Conclusion

This chapter provided a discussion on the problem structure adopted in this thesis. The underlying EN topology design problem consists of four design objectives which were discussed in this chapter, along with the problem constraints. The formulation of the Einstein operator with respect to the problem domain was also presented. Furthermore, various preference rules were developed that are used in this study.

Chapter Four
Archived Multi-objective Fuzzy Simulated Evolution Algorithm for EN Topology Design

4.1 Proposed Algorithm	59
4.1.1 Initialization	59
4.1.2 Fuzzy Evaluation	59
4.1.3 Selection	59
4.1.4 Fuzzy Allocation	59
4.2 Experimental Results	60
4.3 Analysis of Preference Rules	67
4.4 Conclusion	70

Chapter Four

Archived Multi-objective Fuzzy Simulated Evolution Algorithm for EN Topology Design

This chapter discusses the experiments conducted and the results obtained with regard to the application of Einstein's operator to SE. Initially, the chapter describes the design parameters of AFSE algorithm, and how fuzzy logic has been integrated. This is followed by empirical results to evaluate the performance of AFSE algorithm. Later on, the chapter analyses the eight preference rules and their effectiveness on the five test cases using AFSE algorithm.

4.1 Proposed Algorithm

In chapter 2, overall idea of SE was discussed. It consists of four major steps: initialization, evaluation, selection, and allocation. Two steps which are the evaluation and the allocation are considered to be the most important, since they involve in measuring the quality of the solution and effect on the overall performance of the algorithm. In the evaluation step, the quality of each individual element (a link that interconnects two network devices) is evaluated based on a goodness measure. Similarly, in the allocation phase, the current solution is perturbed in order to generate a new solution. Also, by using appropriate function, the quality of old and new solution is calculated and compared. Fuzzy logic plays an important role in both steps as discussed below for the AFSE algorithm.

4.1.1 Initialization

The seed to the algorithm which is the initial solution can be generated randomly. The solution should be a valid one which consist of all design principal and constraints. To compare both algorithms (SE and ACO), the initial solution is predefined.

4.1.2 Fuzzy Evaluation

In this phase, the membership value of each individual objective is computed as discussed in section 3.7. Also, the goodness value for each link in its current position is calculated.

4.1.3 Selection

For each link l_i in the current topology, where $i = 1,2,\ldots,n-1$, a random number $RANDON \in [0,1]$. If $RANDOM > g_{c_i} + B$, where B is the selection bias, then link l_i is selected for allocation and considered for removal from the topology. The bias is used to control the size of the set of links selected for removal, as discussed earlier in section 2.3.1.

4.1.4 Fuzzy Allocation

In the allocation phase, selected links are removed from the topology. For each removed link, a new link is introduced in the topology that do not violate the conditions and constraints of the problem. The procedure is repeated for all links that are presented in the selection set.

After generating the membership functions for all four objectives in fuzzy evaluation phase, the goodness value of the current solution in an iteration is calculated using the fuzzy rule as discussed in Section 2.2. The Einstein's operator for four membership functions as shown in

Equation (3.4) is used for this purpose. Furthermore, in order to generate Pareto front, the current solution is archived as discussed in Section 2.3.1.

4.2 Experimental Results

The experiments are implemented over five test cases. The traffic generated for each test case by each local site was collected from real sites. Other features were assumed such as the types of network devices, numbers of ports in each device, etc. However, the cost of network devices and links were obtained from previous studies (Salman A. Khan, 1999). The smallest test case consist of 15 local sites and the largest has 50 local sites. The bias value is used from previous studies (Salman A. Khan, 2009), where the value was found based on experiments in the same network topology design problem. The algorithm was executed for 30 times for each test case and each test instance was run for 3000 iterations (except $n33$ which was run for 2500 iterations to be comparable with ACO algorithm as discussed in the next chapter). The average and standard deviation of each objective as well as the overall goodness of best solutions were calculated.

Table (4.1)
Average and standard deviation for best solutions for AFSE algorithm.
Cost is in US$, Delay is in milliseconds, and Time is in seconds

Case	Cost	Delay	Hops	Reliability	Overall Goodness	Run Time
n15	10004 ±725.352	0.00290 ±0.0002	6.3 ±1.26	0.2862 ±0.044	0.1405 ±0.028	22.5
n25	13065 ±871.188	0.00383 ±0.0002	8.7 ±1.30	0.2607 ±0.064	0.1574 ±0.026	105.8
n33	21448 ±1659.471	0.00599 ±0.0028	10.4 ±2.47	0.2632 ±0.060	0.1225 ±0.017	343.9
n40	29718 ±2373.787	0.00750 ±0.0025	10.7 ±1.15	0.1436 ±0.060	0.1411 ±0.023	795.6
n50	43055 ±3370.805	0.00667 ±0.0015	11.3 ±1.51	0.0890 ±0.037	0.1459 ±0.021	2080.8

Table (4.1) shows the average values with standard deviation obtained for each objective as well as the overall goodness and execution time in different test cases. It is noticeable in general that as the test case size increases, the cost, the delay, and the number of hops increased, and the reliability decreased. In spite of using large number of test cases ($n40$ and $n50$), the number of hops stayed constant at ≈ 11. This consistency means that there are a good number of ports within the same number of hops (network devices) where the algorithm was able to connect the hosts together and find the solution to the problem.

In order to see the behavior of AFSE algorithm with Einstein's operator, we plot overall goodness for a typical run in each test case respect to the run time as below:

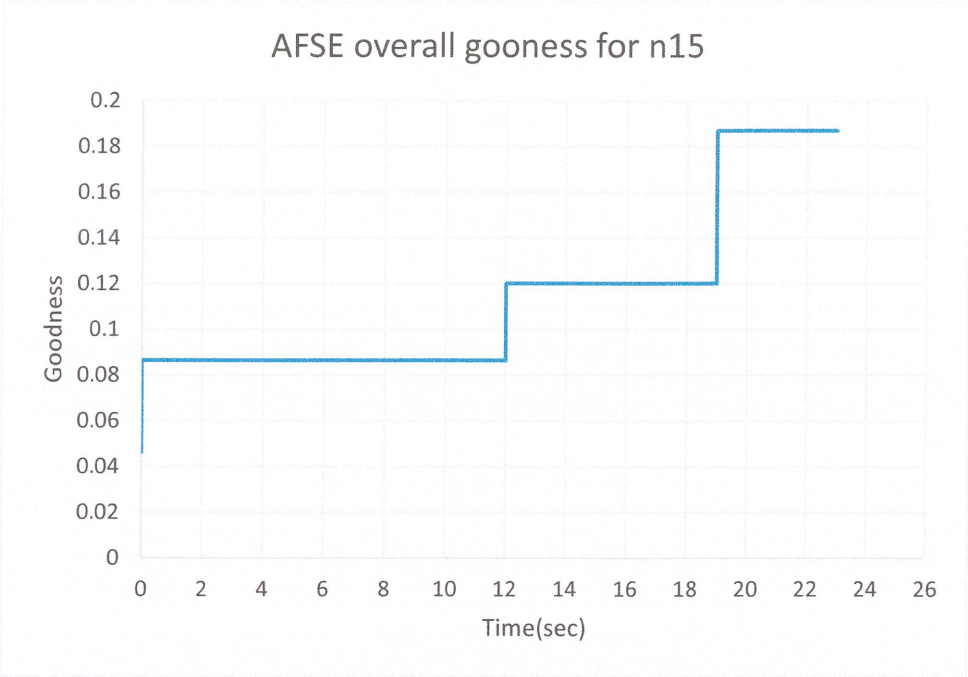


Figure (4.1)
AFSE overall goodness for a typical run for n15

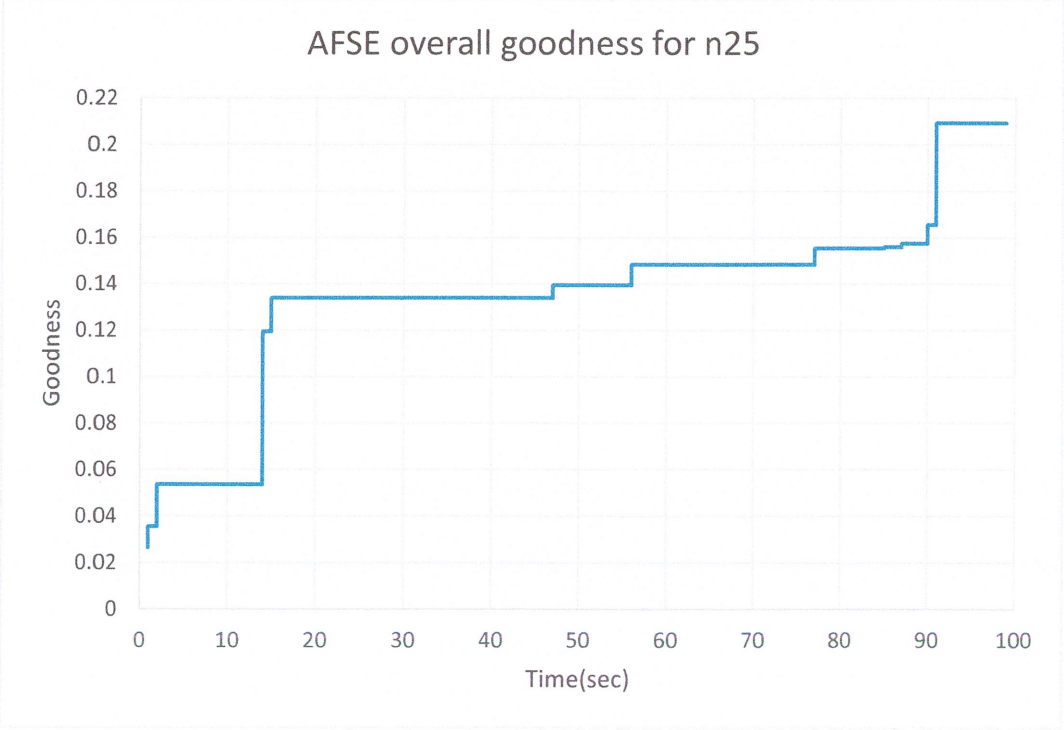


Figure (4.2)
AFSE overall goodness for a typical run for n25

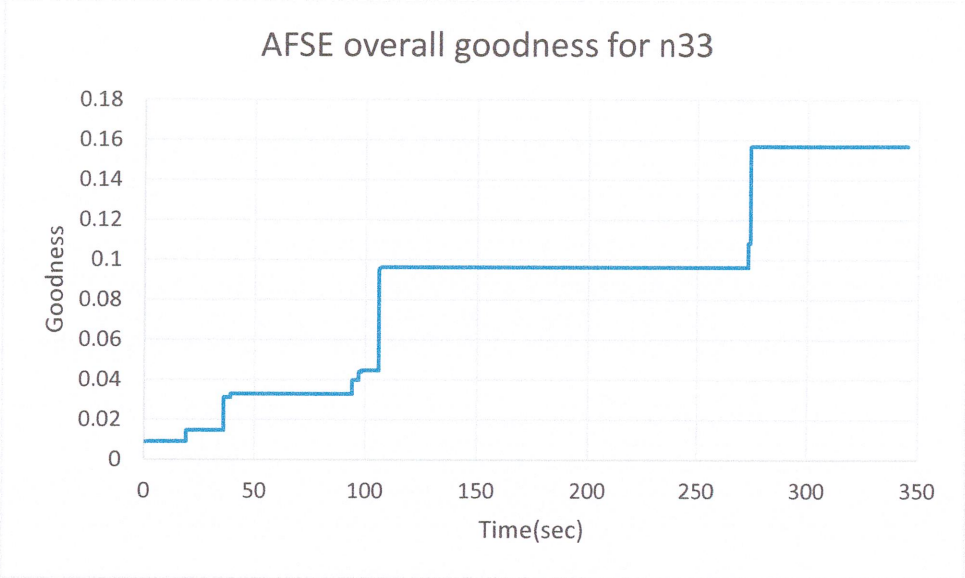


Figure (4.3)
AFSE overall goodness for a typical run for n33

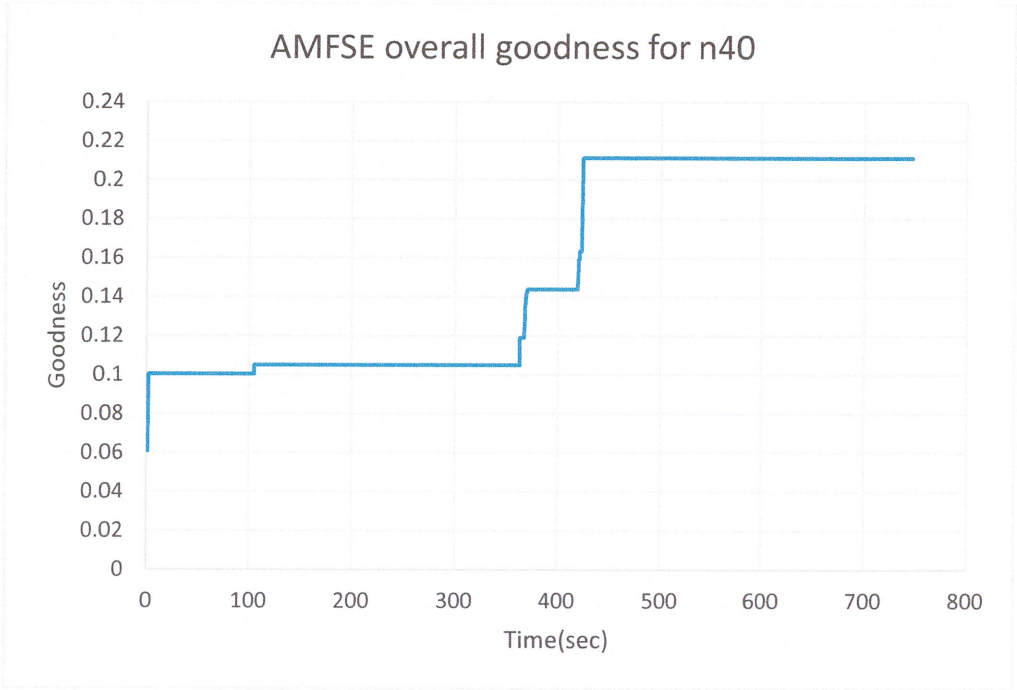


Figure (4.4)
AFSE overall goodness for a typical run for n40

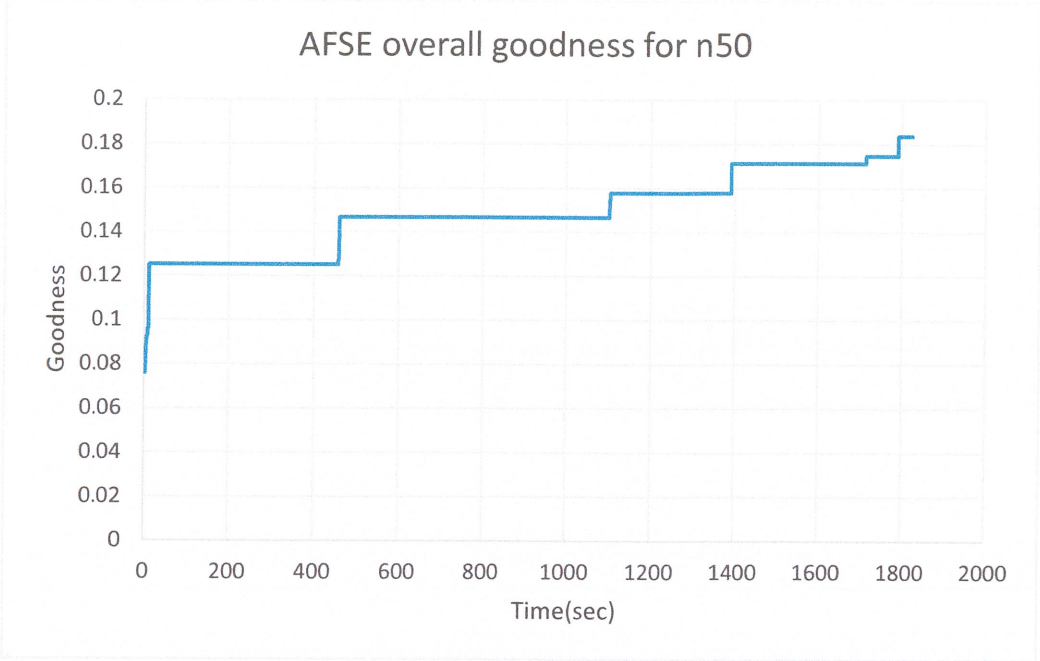


Figure (4.5)
AFSE overall goodness for a typical run for n50

The trends in Figure (4.1) to (4.5) indicate that in general, AFSE was able to improve solutions for all test cases in a typical run; the solution goodness increases throughout the execution of the algorithm which indicates that AFSE was capable of coming out of local optima even at the later stages of the algorithm execution. Figures from (4.6) to (4.10) show the all goodness values respect to time for a typical run for all for all test cases.

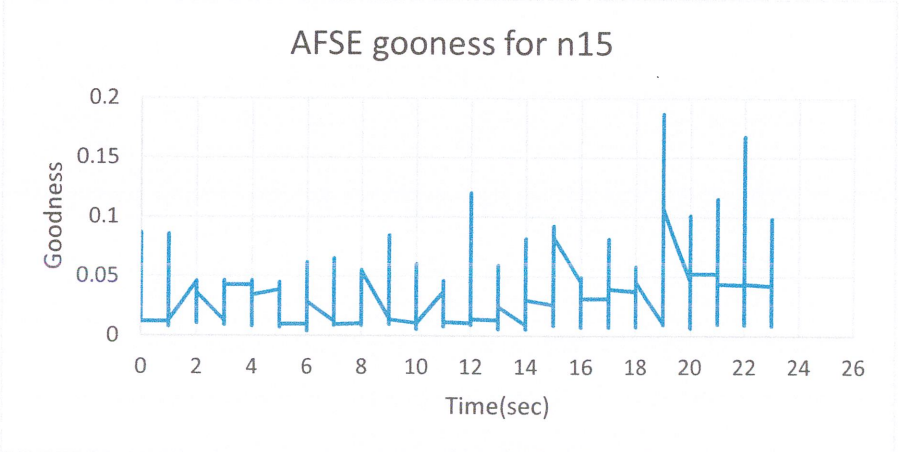


Figure (4.6)
AFSE goodness for a typical run for n15

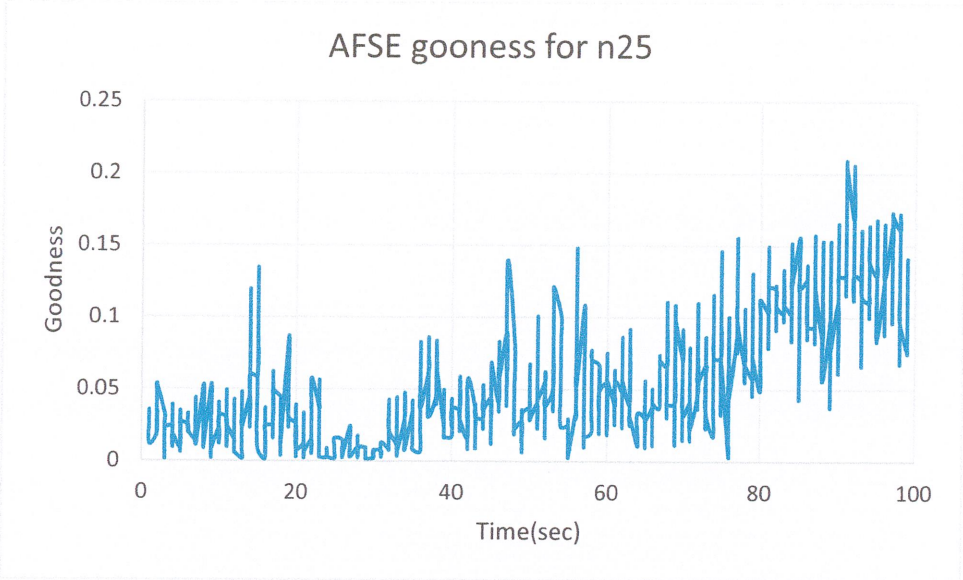


Figure (4.7)
AFSE goodness for a typical run for n25

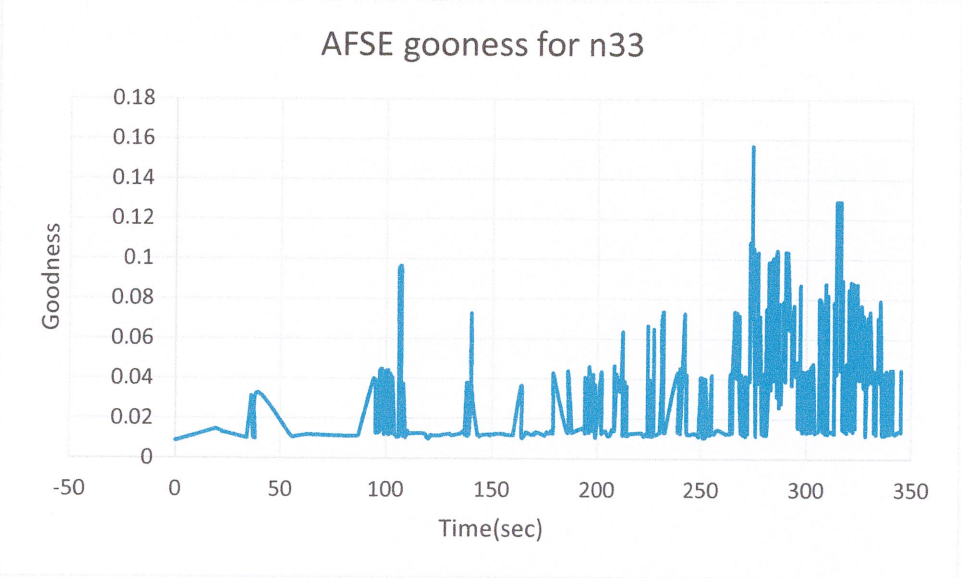


Figure (4.8)
AFSE goodness for a typical run for n33

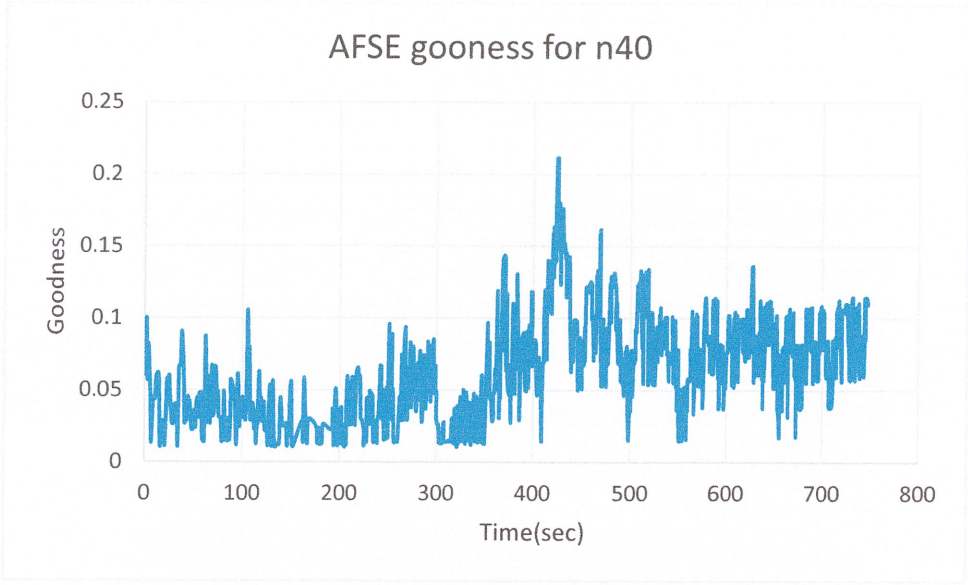


Figure (4.9)
AFSE goodness for a typical run for n40

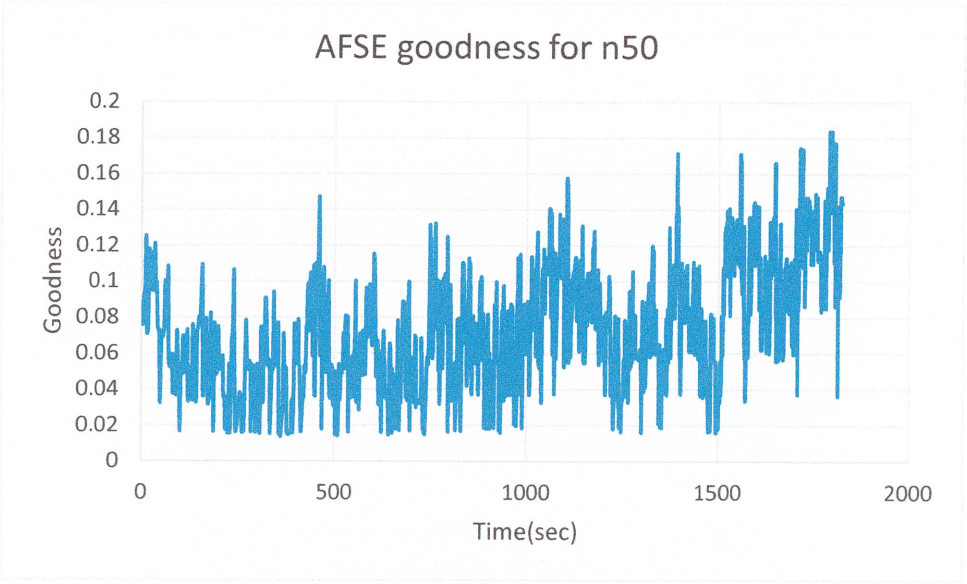


Figure (4.10)
AFSE goodness for a typical run for n50

4.3 Analysis of Preference Rules

This section discusses the validity and applicability of the eight preference rules described in Section 3.7.2 to the problem of topology design of EN. For each rule, thirty runs were executed for each test case. Linear regression analysis (see Appendix (1) for background) was utilized to validate the rules. It is a statistical process for estimating the relationships among variables. In order to verify the rules, relationships between the four different objectives and the overall goodness over each rule in every test case were tested. This strength of this relationship was measured through *regression coefficient*. The value of the coefficient is in the range [0,1], where a value near 0 represents least significance and value near 1 represents extremely strong relationship. Furthermore, a positive sign indicates direct proportionality while a negative sign suggests inverse relationship. The results for each rule are given in Tables (4.2) to (4.9).

It is observed from Tables (4.2) to (4.6) that in general, many rules were not satisfied. On average, for each test case, three rules were satisfied. Furthermore, the level of satisfaction was higher for smaller test cases compared to bigger test cases. For example, for *n15* and *n25*, four rules were satisfied. However, for *n40* and *n50*, three rules were satisfied. One exception was *n33* where only two rules were satisfied. Furthermore, the highest satisfaction level was for PR1, PR2 PR4, and PR6 which were satisfied for three test cases, while PR 3 and PR 8 were not satisfied for any test case. Note that both PR3 and PR8 involve giving priority to reliability (which is not the case with other rules). This indicates that for the five test cases, reliability cannot be improved beyond a certain level, no matter how much preference is given to it. Another observation from

the results is that, overall, as the complexity of a rule increases, the satisfaction level decreases.

Table (4.2)
Regression analysis results for n15 using AFSE

Rule	Cost	Delay	Hops	Reliability	Status
PR1	-0.44	-0.36	-0.26	0.271	VALID
PR2	-0.39	-0.45	-0.22	-0.35	VALID
PR3	-0.44	-0.32	-0.24	-0.32	INVALID
PR4	-0.56	-0.1	-0.58	-0.37	VALID
PR5	-0.03	-0.53	-0.39	0.108	VALID
PR6	-0.32	-0.24	-0.47	-0.23	INVALID
PR7	-0.23	-0.33	-0.43	-0.19	INVALID
PR8	-0.14	-0.36	-0.29	-0.28	INVALID

Table (4.3)
Regression analysis results for n25 using AFSE

Rule	Cost	Delay	Hops	Reliability	Status
PR1	-0.66	-0.47	-0.39	0.075	VALID
PR2	-0.12	-0.37	-0.59	0.192	INVALID
PR3	-0.69	-0.25	-0.39	0.022	INVALID
PR4	-0.52	-0.37	-0.27	0.076	VALID
PR5	-0.62	-0.59	-0.54	-0.16	INVALID
PR6	-0.41	-0.55	-0.4	-0.06	VALID
PR7	-0.73	-0.2	-0.39	-0.04	VALID
PR8	-0.45	-0.5	-0.28	-0.05	INVALID

Table (4.4)
Regression analysis results for n33 using AFSE

Rule	Cost	Delay	Hops	Reliability	Status
PR1	-0.27	-0.18	-0.35	0.191	INVALID
PR2	0.122	0.422	-0.4	-0.11	VALID
PR3	-0.27	-0.2	-0.33	0.117	INVALID
PR4	-0.11	-0.02	-0.41	0.025	INVALID
PR5	-0.35	-0.05	-0.25	0.237	INVALID
PR6	-0.32	-0.381	-0.22	0.2	VALID
PR7	-0.08	-0.12	-0.43	-0.05	INVALID
PR8	-0.11	0.194	-0.47	-0.03	INVALID

Table (4.5)
Regression analysis results for n40 using AFSE

Rule	Cost	Delay	Hops	Reliability	Status
PR1	-0.71	0.066	-0.51	0.558	VALID
PR2	-0.64	0.193	-0.42	0.229	INVALID
PR3	-0.54	-0.46	-0.39	0.271	INVALID
PR4	-0.62	0.248	-0.55	0.386	VALID
PR5	-0.46	0.013	-0.42	0.057	INVALID
PR6	-0.66	-0.11	-0.31	0.124	INVALID
PR7	-0.6	-0.09	-0.57	0.298	VALID
PR8	-0.56	0.036	-0.16	0.211	INVALID

Table (4.6)
Regression analysis results for n50 using AFSE

Rule	Cost	Delay	Hops	Reliability	Status
PR1	-0.67	-0.15	-0.68	0.49	INVALID
PR2	-0.27	-0.35	-0.44	0.117	VALID
PR3	-0.21	-0.37	-0.68	0.393	INVALID
PR4	-0.56	-0.08	-0.7	0.672	INVALID
PR5	-0.56	0.06	-0.28	0.266	INVALID
PR6	-0.63	-0.13	-0.33	0.36	VALID
PR7	-0.36	0.255	-0.77	0.041	INVALID
PR8	-0.58	-0.03	-0.49	0.399	INVALID

4.4 Conclusion

In this chapter, we discussed the AFSE algorithm, a modified version of SE. The algorithm is required to solve our MOO problem of designing the topology of EN. Fuzzy logic with Einstein operator was used to aggregate the four objectives into a single objective function. AFSE was evaluated with five different test cases. It was observed that the AFSE algorithm was able to produce improved solutions throughout its search, which indicates that the algorithm was able to escape local minima. Furthermore, analysis of preferences with eight preference rules revealed that rate of satisfaction of rules is higher for small test cases as compared to large test cases.

Chapter Five
Einstein Operator based Fuzzy Ant Colony Optimization Algorithm

5.1 FACO Algorithms and its Design Parameters	72
5.2 Experimental Results	73
5.3 Analysis of Preference Rules	79
5.4 Conclusion	81

Chapter Five

Einstein Operator based Fuzzy Ant Colony Optimization Algorithm

This chapter presents application the Einstein's operator to the fuzzy ant colony optimization (FACO) algorithm for EN topology design problem. The chapter starts with the introduction of the FACO algorithm for EN topology design. This is followed by empirical results and discussion. Following this, the chapter analyses the eight preference rules and their effectiveness on the five test cases using the FACO algorithm.

5.1 FACO algorithm and its design parameters

The ACO algorithm works on population of ants where each ant is responsible for establishing a feasible network topology. An ant gradually builds a topology starting from the root node. Selecting a particular path is based on some factors such as evaporation, pheromone trail, and the heuristic value. A complete journey by an ant is considered to be a complete feasible network topology.

In initialization phase, a set of initial solutions are generated randomly. Each ant produces a feasible solution without any previous information or feedback about the paths. However in each step, the constraints are checked to ensure the feasibility of the solution. In addition, the algorithm parameters such as evaporation, pheromone trail, and the heuristic value are also initialized in this phase.

After the generation of the set of initial solutions, the next step is to update the pheromone value based on the value that is generated by elitist

ant (Salman A. Khan, 2009). Then the updated values of pheromone are passed to the following ants and the above cycle is repeated until the stopping condition is met. Furthermore, the fuzzy heuristic value in the context of EN topology design is used on the link between any two nodes. The details of calculation the fuzzy value for each objective have been discussed in section 3.7. Moreover, the Einstein's operator for the four objectives as shown in Equation (3.4) was integrated with the algorithm in solution construction phase.

5.2 Experimental Results

The fuzzy logic with the Einstein operator was embedded in ACO, and the resulting algorithm was applied to the five test cases. Parameters such as number of ants used for each test case, pheromone, evaporation, heuristic value are taken from (Salman A. Khan, 2009). The goal is to test the performance of the Einstein operator with ACO. The FACO algorithm was run 30 times for each test case and each test instance was run for 100 iterations.

Table (5.1)
Average and standard deviation for best of best solutions for FACO algorithm. Cost is in US$, Delay is in milliseconds, and Time in seconds

Case	Cost	Delay	Hops	Reliability	Overall Goodness	Run Time
n15	10294 ±1399.797	0.00314 ±0.0006	5.4 ±1.65	0.4426 ±0.056	0.1021 ±0.034	7.2
n25	13048 ±1338.857	0.00390 ±0.0003	6.7 ±0.74	0.0961 ±0.011	0.1460 ±0.030	24.8
n33	26748 ±3367.639	0.00537 ±0.0020	7.3 ±1.88	0.0494 ±0.014	0.1502 ±0.027	55.9
n40	36818 ±2198.259	0.00482 ±0.0003	7.7 ±0.88	0.0283 ±0.008	0.1504 ±0.035	637.4
n50	53270 ±5273.222	0.00523 ±0.0009	8.1 ±0.66	0.0137 ±0.004	0.1672 ±0.033	3337.9

Table (5.1) shows the average values and standard deviation for each objective as well as the overall goodness in different test cases. As an overall trend, as the test case size increases, the cost, the delay, and the number of hops are increases. Whereas on the other hand, the reliability of the network decreases.

In order to see the behavior of FACO algorithm with the Einstein's operator, we plot the "best of the best" overall goodness for a typical run in each test case respect to the run time as below:

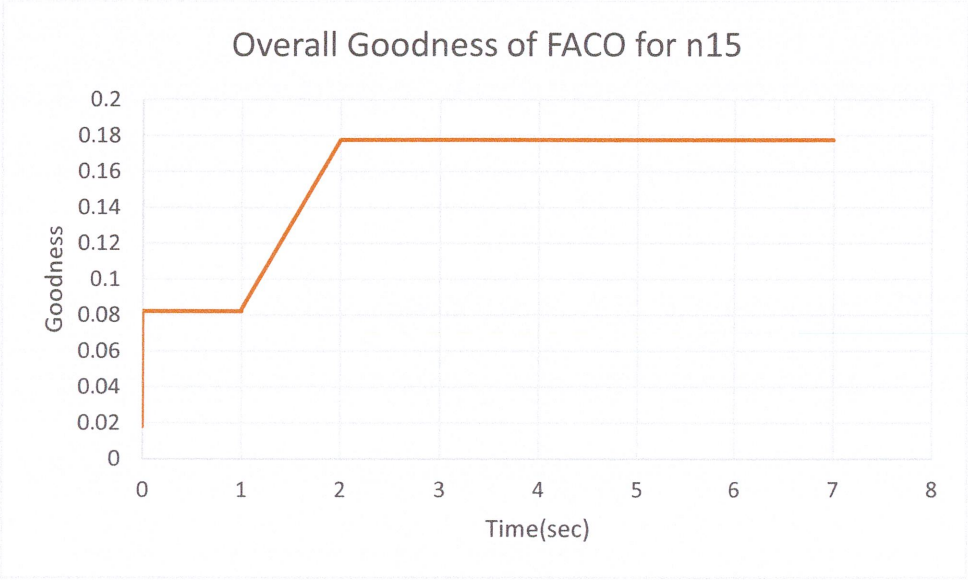


Figure (5.1)
Overall goodness value of FACO for a typical run for n15

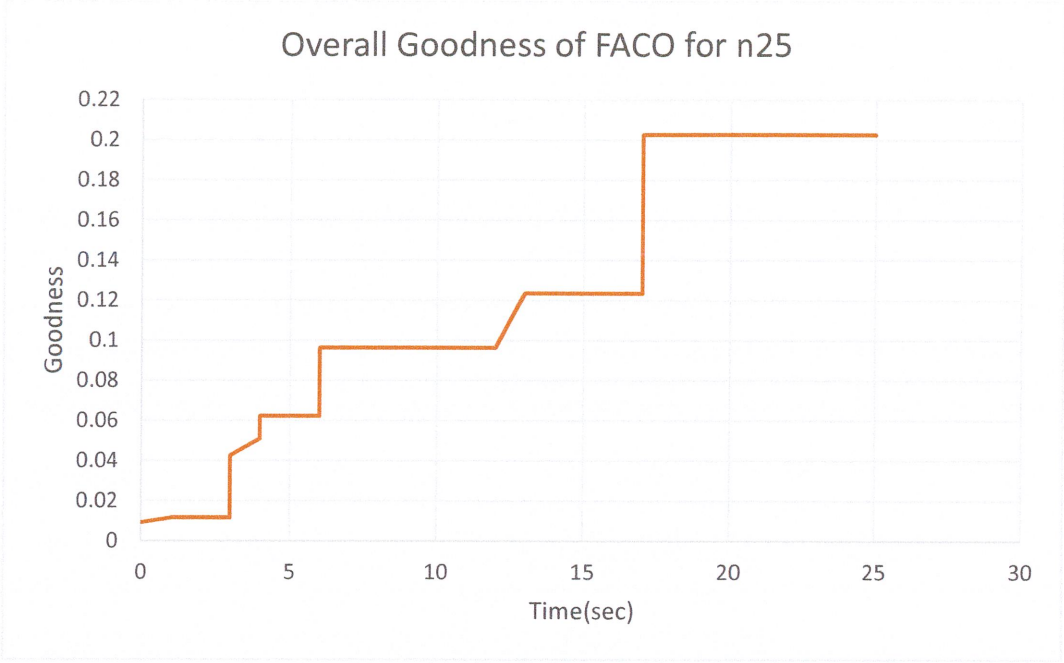


Figure (5.2)
Overall goodness value of FACO for a typical run for n25

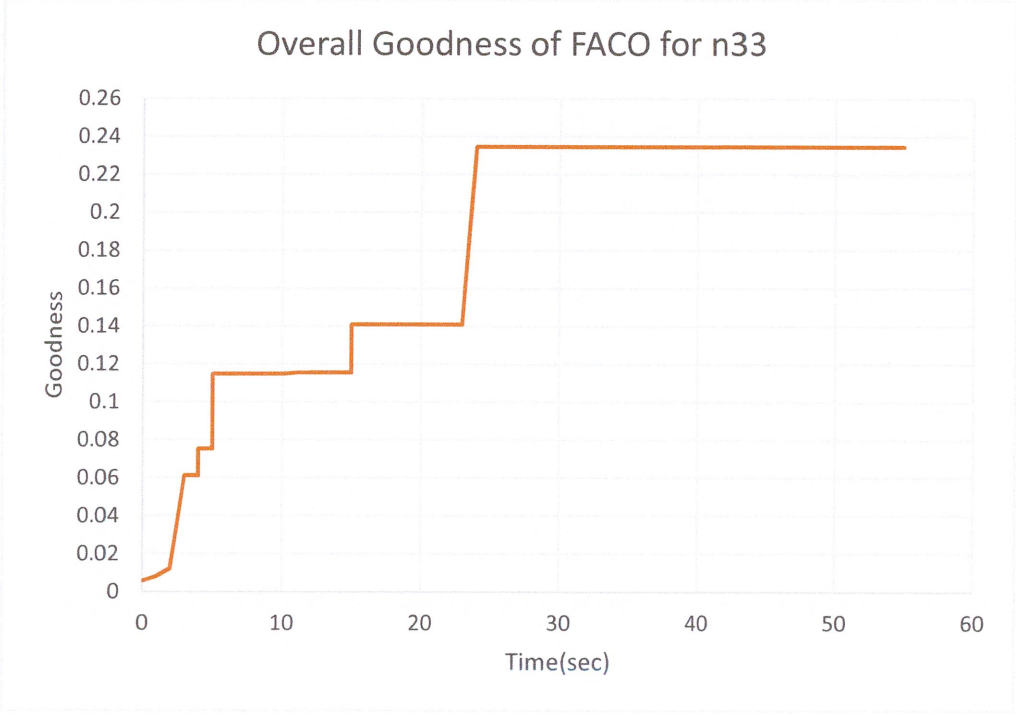


Figure (5.3)
Overall goodness value of FACO for a typical run for n33

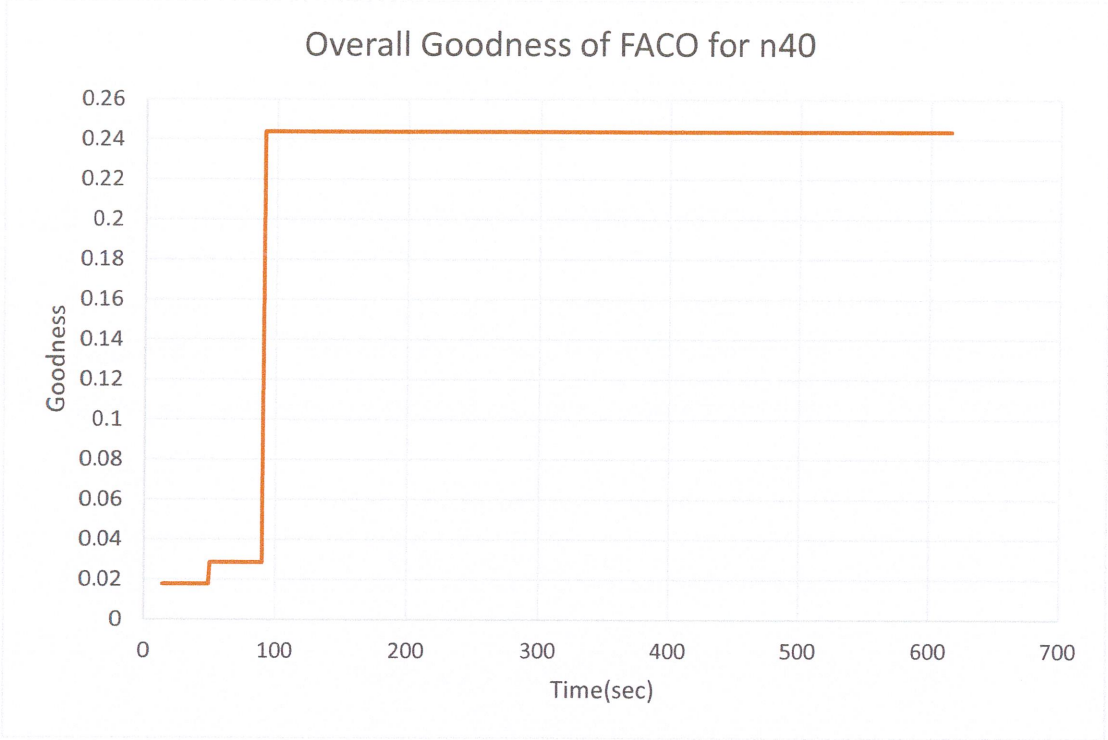


Figure (5.4)
Overall goodness value of FACO for a typical run for n40

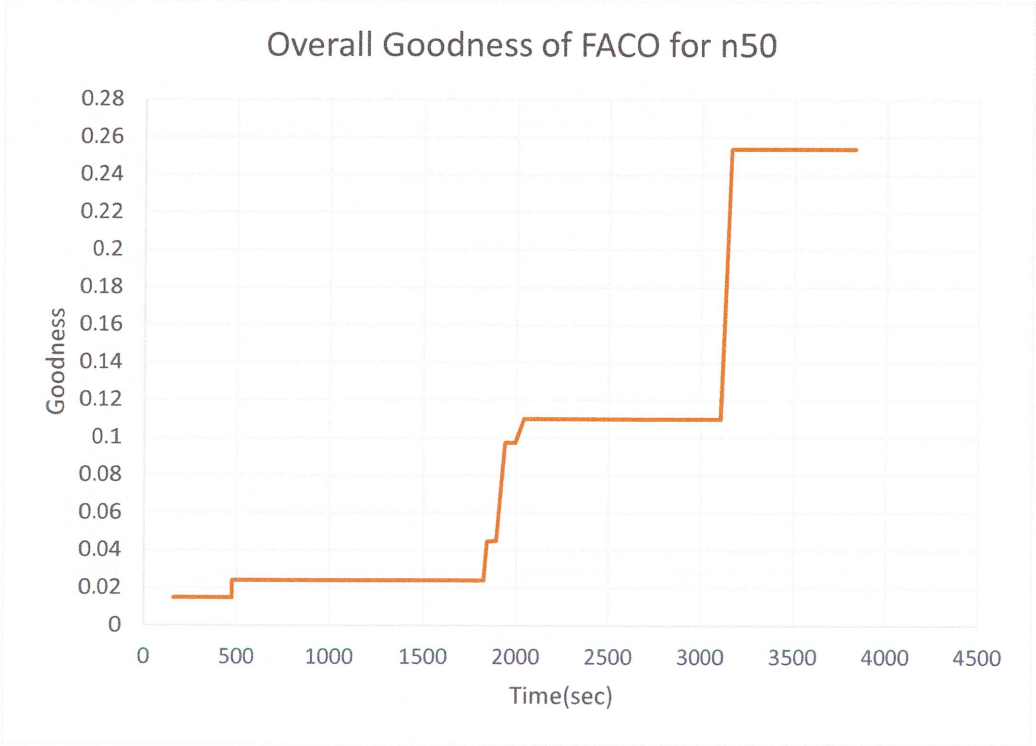


Figure (5.5)
Overall goodness value of FACO for a typical run for n50

As an overall trend from the plots Figure (5.1) to (5.5), the elitist ant was able to improve the quality of generated solutions over the time except in $n40$, and to some extent in $n15$, where the improvement was noticed in the early moments of execution only. Figures (5.6) to (5.10) show the all goodness values respect to time for a typical run for all for all test cases.

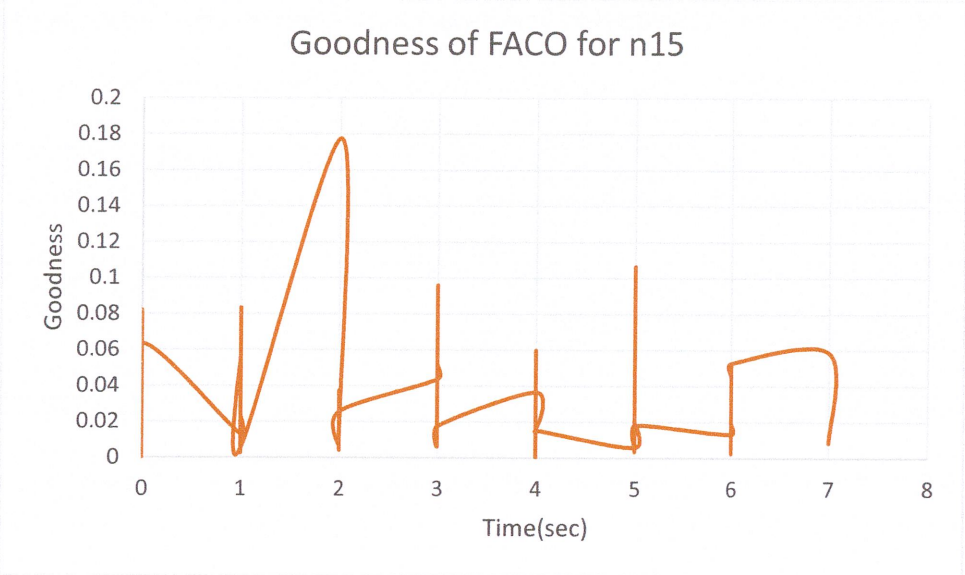


Figure (5.6)
Goodness value of FACO for a typical run for n15



Figure (5.7)
Goodness value of FACO for a typical run for n25



Figure (5.8)
Goodness value of FACO for a typical run for n33

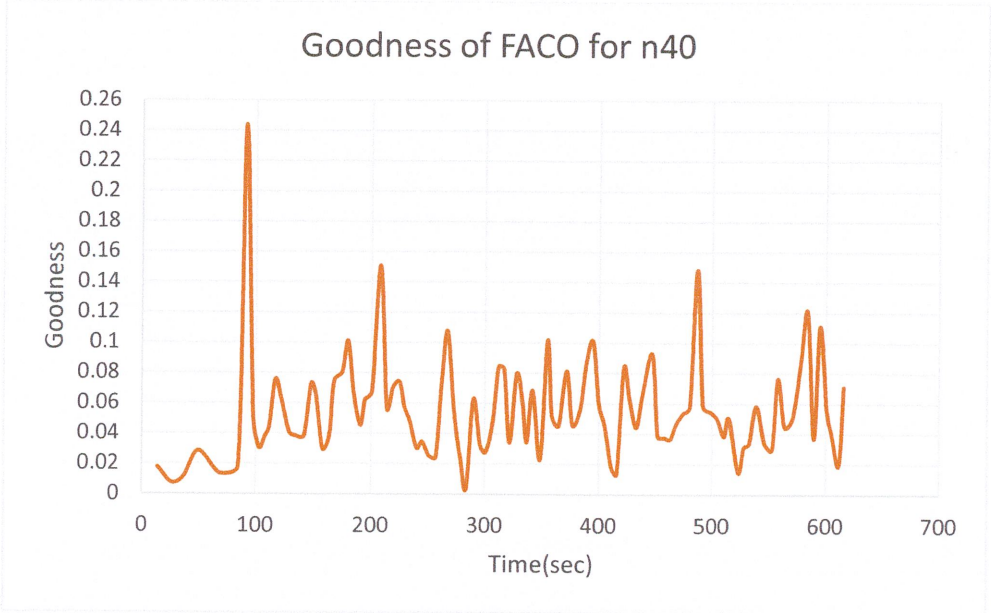


Figure (5.9)
Goodness value of FACO for a typical run for n40

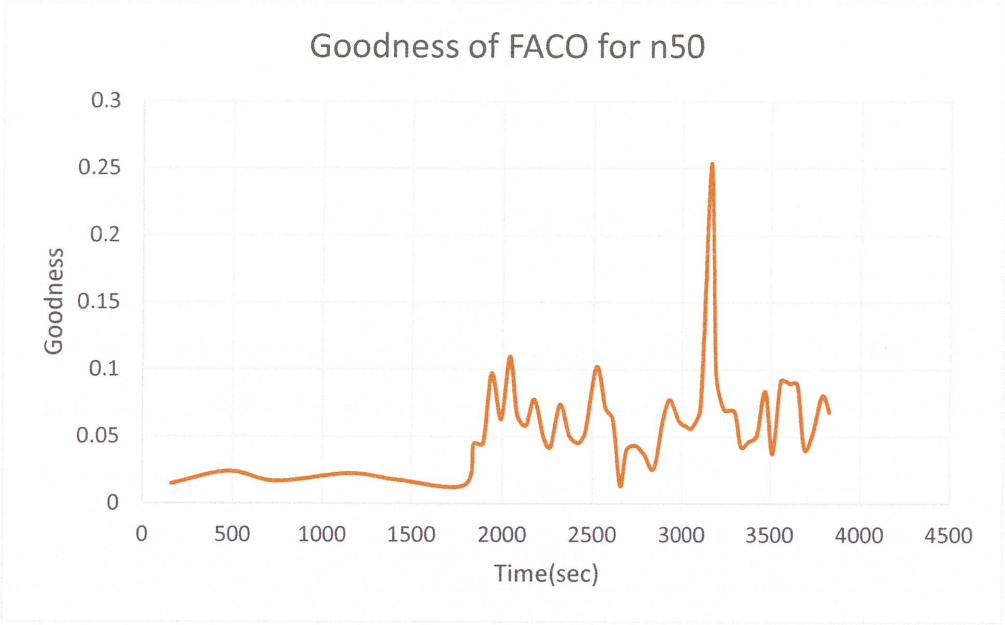


Figure (5.10)
Goodness value of FACO for a typical run for n50

5.3 Analysis of Preference Rules

This section discusses empirical results and analysis of applying the eight preference rules to FACO algorithm as described in Section 4.2. For each rule, thirty runs were executed for each test case. An approach similar to that used in Chapter 4 for AFSME was adopted here; the rules were validated using linear regression analysis.

It is observed from Tables (5.2) to (5.6) that, as in the case of AMSE, many rules were not satisfied. On average, for each test case, around four rules were satisfied. Furthermore, more rules were satisfied for smaller test cases as compared to bigger test cases. For example, for *n15*, five rules were satisfied, while for *n25* four rules were satisfied. For *n33*, *n40* and *n50*, three rules were satisfied for each test case. Moreover, the highest satisfaction level was for PR1 which was satisfied for four test cases. Following this were PR2, PR3, and PR4 which were satisfied for three test cases. PR5 and PR 7 were satisfied for two test cases, and PR6 was satisfied for one test case. PR8 was not satisfied for any test case.

Table (5.2)
Regression analysis results for n15 using FACO

Rule	Cost	Delay	Hops	Reliability	Status
PR1	-0.77	-0.32	-0.41	0.19	VALID
PR2	-0.139	-0.53	-0.18	0.15	VALID
PR3	0.131	-0.06	-0.12	0.31	VALID
PR4	-0.319	-0.28	0.236	0.09	VALID
PR5	-0.176	-0.25	-0.21	0.17	VALID
PR6	-0.08	-0.05	-0.02	0.105	INVALID
PR7	0.272	-0.48	-0.43	-0.1	INVALID
PR8	0.315	-0.21	-0.08	-0.22	INVALID

Table (5.3)
Regression analysis results for n25 using FACO

Rule	Cost	Delay	Hops	Reliability	Status
PR1	-0.66	-0.49	-0.15	0.14	VALID
PR2	-0.26	-0.44	-0.05	0.07	VALID
PR3	-0.24	-0.48	-0.48	0.15	INVALID
PR4	-0.34	-0.15	-0.23	-0.13	VALID
PR5	-0.276	-0.12	-0.47	0.062	VALID
PR6	-0.34	-0.2	0	-0.19	INVALID
PR7	-0.44	-0.32	-0.22	0.021	INVALID
PR8	-0.16	-0.14	-0.31	-0.28	INVALID

Table (5.4)
Regression analysis results for n33 using FACO

Rule	Cost	Delay	Hops	Reliability	Status
PR1	-0.39	-0.1	-0.19	0.08	VALID
PR2	-0.16	-0.17	0.144	0.2	INVALID
PR3	-0.14	-0.17	-0.04	0.3	VALID
PR4	-0.27	-0.06	-0.01	0.13	VALID
PR5	-0.05	-0.067	0.067	0.034	INVALID
PR6	-0.07	-0.254	0.193	-0.15	INVALID
PR7	-0.014	-0.09	-0.01	0.147	INVALID
PR8	-0.22	-0.11	-0.04	0.125	INVALID

Table (5.5)
Regression analysis results for n40 using FACO

Rule	Cost	Delay	Hops	Reliability	Status
PR1	-0.53	-0.296	-0.43	0.307	VALID
PR2	-0.231	-0.23	-0.06	0.144	INVALID
PR3	-0.226	-0.1	-0.29	-0.14	INVALID
PR4	-0.175	-0.185	-0.023	0.24	INVALID
PR5	-0.02	-0.08	-0.1	0.331	INVALID
PR6	-0.24	-0.324	-0.15	0.17	VALID
PR7	-0.19	-0.05	-0.19	0.117	VALID
PR8	-0.47	0.255	-0.04	0.336	INVALID

Table (5.6)
Regression analysis results for n50 using FACO

Rule	Cost	Delay	Hops	Reliability	Status
PR1	-0.241	-0.1	-0.19	0.19	INVALID
PR2	0.168	-0.39	-0.26	0.03	VALID
PR3	-0.1	-0.09	-0.06	0.48	VALID
PR4	0.331	-0.055	-0.17	0.341	INVALID
PR5	0.138	-0.062	-0.14	-0.06	INVALID
PR6	-0.35	-0.016	-0.3	0.392	INVALID
PR7	-0.35	-0.1	-0.27	-0.31	VALID
PR8	-0.08	-0.29	0.085	-0.05	INVALID

5.4 Conclusion

The chapter discussed the Einstein operator based fuzzy ACO (FACO) algorithm in the context of its application to EN topology design. In order to combine the four design objectives used in this work, fuzzy logic and Einstien operator were used. The performance of FACO was evaluated with five test cases. It was observed that, in general, the search capabilities of FACO were satisfactory since the algorithm was able to escape local minima in majority of cases. Furthermore, analysis of eight preference rules revealed that for small test cases, the level of satisfaction

of rules was better compared to large test cases. As the test case size increases, satisfaction of rules generally decreases.

Chapter Six
Comparison of Techniques

6.1 Results and Discussion	84
6.1.1 Comparison Based on the Quality of Solutions	85
6.1.2 Comparison Based on Pareto-Optimal Solutions	90
6.2 Conclusion	91

Chapter Six

Comparison of Techniques

This chapter presents an overall comparison of the applied techniques in solving the problem of designing the topology of EN. Previously, chapter 4 and chapter 5 discussed the results obtained for SE and ACO respectively alone. In this chapter, the comparison is done based on the overall performance of the both algorithms (SE and ACO) using the Einstein's operator. To make a fair comparison, both algorithms were produced the same number of solutions in the *Optimal Solution Set (S)*. Finally, charts are prepared and comparisons are done for the five test cases herein.

6.1 Results and Discussion

Although AFSE and FACO are from different sub-families, where SE is from metaheuristics based on single solution and ACO is from population based metaheuristics , the rightful comparison is possible by running both algorithms to produce the same number of solutions and comparing the results.

For the best performance of FACO algorithm in all test cases, all the setup parameters were taken from the previous work as discussed in chapter 5. Accordingly, all test cases were run for 100 iteration and the best number of ants was taken, as 30 ants for $n15$, $n25$, $n40$, and $n50$ and 25 ants for $n33$. Therefore, the total number of generated solutions are 3000 solutions in $n15$, $n25$, $n40$, and $n50$ and 2500 solutions in $n33$. Since SE generates a single solution in an iteration, $n15$, $n25$, $n40$, and $n50$ were run for 3000 iterations and $n33$ was run for 2500 iterations to generate the same number of solutions as that with FACO.

The comparison of the algorithms was done based on two aspects: based on the quality of produced solutions and based on the number of Pareto-optimal solutions as below.

6.1.1 Comparison Based on the Quality of Solutions

In order to evaluate the performance of the Einstein's operator based AFSE and FACO, the overall goodness was compared as below.

Table (6.1)
Comparision of AFSE and FACO

Case	AFSE		ACO		% imp AFSE vs FACO
	Overall Goodness	Time	Overall Goodness	Time	
n15	0.1405 ±0.028	22.5	0.1021 ±0.034	7.2	27.33
n25	0.1574 ±0.026	105.8	0.146 ±0.030	24.8	7.24
n33	0.1225 ±0.017	343.9	0.1502 ±0.027	55.9	-22.61
n40	0.1411 ±0.023	795.6	0.1504 ±0.035	637.4	-6.59
n50	0.1459 ±0.021	2080.8	0.1672 ±0.033	3337.9	-14.60

From Table (6.1), it is clear that the Einstein's operator performed better with SE algorithm in small test cases of $n15$ and $n25$. On the other hand, the operator performed better with FACO algorithm in large test cases of $n33$, $n40$, and $n50$. Note that the results were also subject to statistical testing with Wilcoxon ranked-sum test, which indicated that that all improvements were statistically significant. The results could be justified by the following.

By considering the fundamental characteristics of SE and ACO algorithms separately, each algorithm has totally different approach to

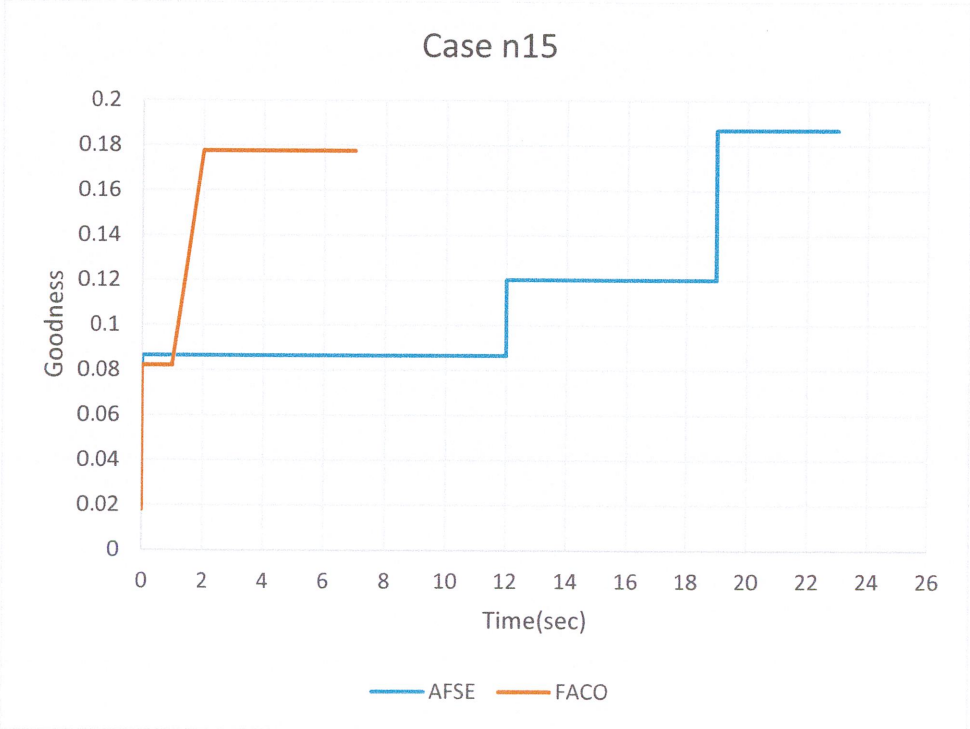
generate and process solutions. SE deploys a modification process to the current solution, in order to generate a neighbor solution by selecting and changing one or more links based on the goodness value of each element in its current location. This scheme provides SE algorithm with a strong local search ability and fine tuning. On the other hand, ACO applies a diversification process that implies the generation of new solutions (based on number of ants) from the scratch every time in each iteration. This process rewards ACO algorithm with a powerful global search capability.

Consequently and from controlling parameters perspective in each algorithm, the $bias\ 'B'$ in SE controls the selection of individual elements which are unfit (in bad position) in the current solution. A high positive value of B decreases the probability of selecting the elements for removal. As a result, the selection set is small which speed-up the algorithm, but reduces the quality of solution and increase the risk of being trapped in local minima and vice versa. The $bias$ value is problem specific and the best suitable value is found based on trial and error. While in ACO, new feasible solutions are always generated from the scratch that encourages the exploration process in search space. This process is controlled by the pheromone information and heuristic values which are problem dependent and can be found heuristically. The higher the pheromone and heuristic information for a given edge, the higher the probability of that particular edge to be selected in a graph.

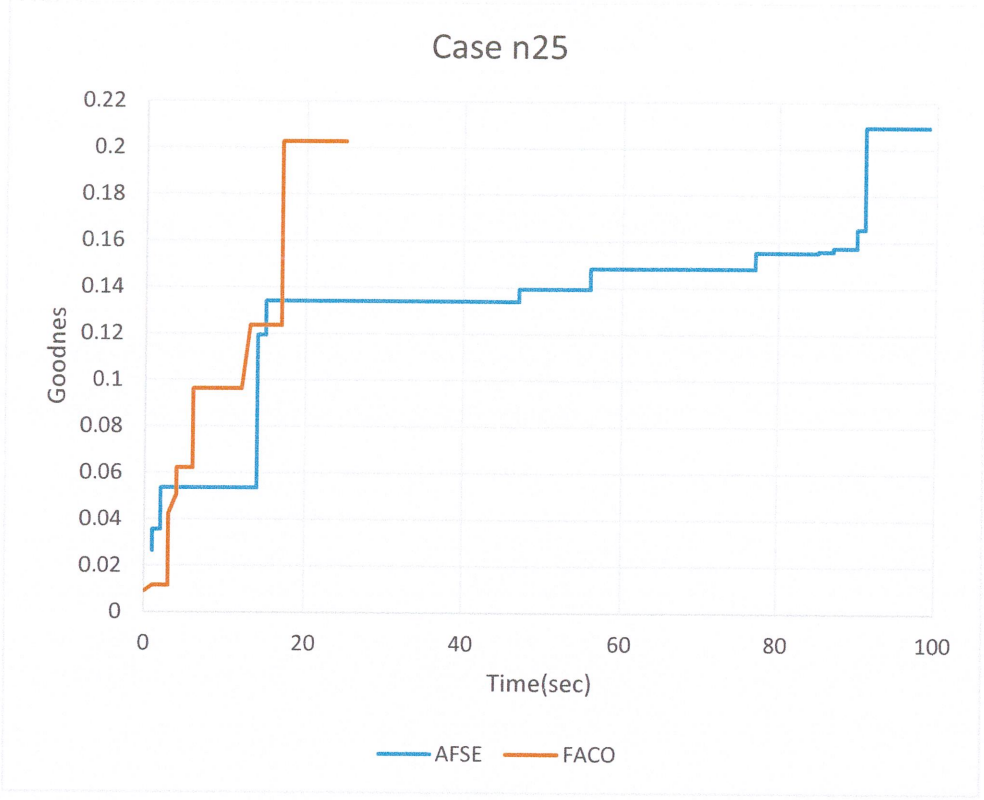
Since SE produces a mutant solution from its original version by selecting and modifying a few links, the process of generating a mutant solution is simple and easier and less expensive especially for smaller cases than creating a whole new solution as the case in ACO. Although ACO consumed less time in the first four cases, the quality of solutions obtained by SE was better than ACO in smaller test cases. Furthermore, in SE, the best solution found so far is kept in the population and used

again for producing a new modified solution in each iteration until it reaches the termination condition. While in ACO, the best solution discovered so far is not kept in the population as a candidate solution since it is replaced with a newly produced solution by the corresponding ant in the next iteration. The pheromone trail used for memorizing the past experience by storing the characteristics of good solutions and pass them to the next iteration. The previous solution does not contribute very effectively in the creation of new candidate solutions; it is some attribute of good links which are passed to the next generation, rather than the whole complete solution as in SE.

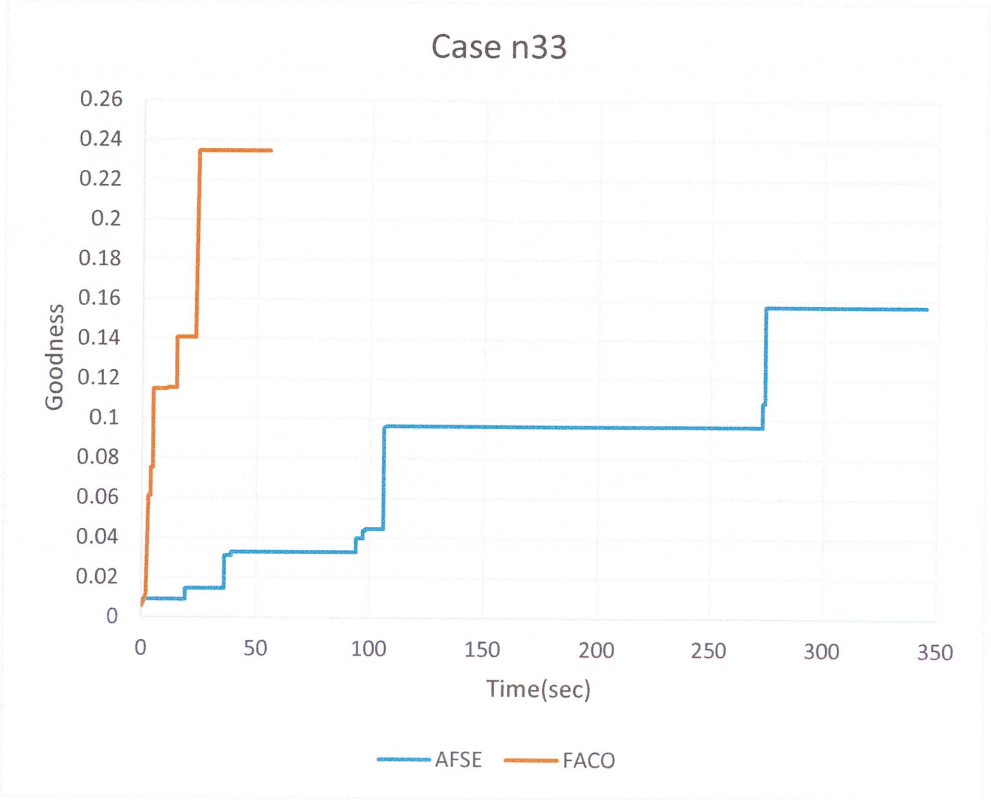
From the above discussion, AFSE applies small changes trying to improve the solution. In contrast, FACO adopts large step wise process and generates an entire new solution from scratch every time in each iteration. Therefore, AFSE performed better in smaller test cases and FACO performed better in medium and larger test cases.



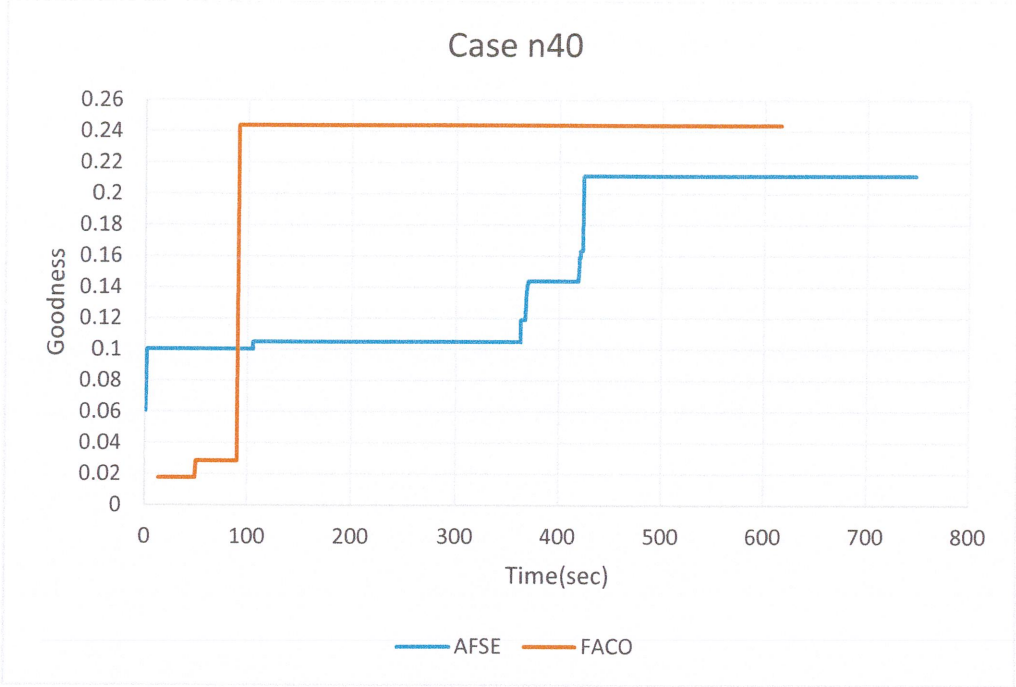
(a)


Case n25
0.22
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Goodnes
0
20
40
60
80
100
Time(sec)
AFSE
FACO

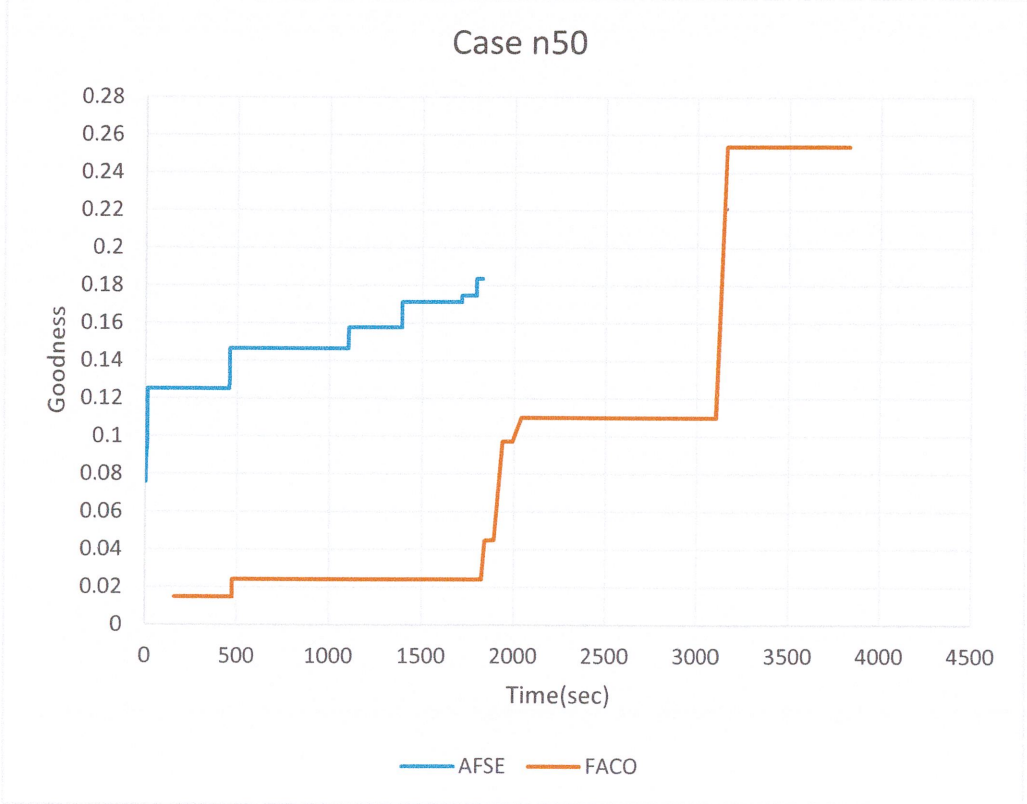
(b)


Case n33
0.26
0.24
0.22
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Goodness
0
50
100
150
200
250
300
350
Time(sec)
AFSE
FACO

(c)



(d)



(e)

Figure (6.1)
Plots of a typical run for AFSE and FACO for (a) n15, (b) n25, (c) n33, (d) n40, and (e) n50, Time in second

Figure (6.1) shows the performance of AFSE versus FACO in terms of overall goodness for a typical run. It confirms the above outcome such that the Einstein's operator performed better with AFSE algorithm in small test cases of $n15$ and $n25$. On the other hand, the operator performed better with FACO algorithm in cases of $n33$, $n40$, and $n50$.

6.1.2 Comparison Based on Pareto-Optimal Solutions

The two algorithms were also compared with respect to the Pareto – optimal solution they produced. For each algorithm, the run that produced the best fitness values of solution (among thirty runs) was studied. It can be seen from Table (6.2) that, in general, FACO generated more Pareto-optimal solutions than AFSE, with the exception of n50. Another finding is that, generally, the number of Pareto-optimal solutions decreased with increase in number of nodes. A reason behind it could be that as the number of nodes increases, the search space of the problem become highly restricted and it is very difficult to find a good number of solutions that satisfy the conditions and constraints of the problem.

Table (6.2)
PO solutions for a typical run for AFSE and FACO

Case	AFSE	FACO
n15	8	9
n25	4	7
n33	3	6
n40	3	4
n50	5	5

6.2 Conclusion

In this chapter, a mutual comparison of AFSE and FACO algorithms was performed and discussed. The performance of both algorithms was investigated in the context of usage of Einstein's operator. Two aspects of comparisons were used. The first one is based on the quality of solutions, where AFSE performed better in small test cases, while FACO performed better in large test cases. The second test was based on number of Pareto-optimal solutions produced by each algorithm. The results showed that FACO was able to produce better results than AFSE. Therefore, the Einstein's operator performed better with FACO in most test cases.

Chapter Seven
Conclusions and Future Work

7.1 Summary and Conclusion	94
7.2 Future Work	95

Chapter Seven

Conclusions and Future work

In this thesis, the problem of investigating the application and effectiveness of Einstein's operator in fuzzy logic with a preference scheme for topology design of Enterprise network was addressed. The problem was classified as a MOO problem considering four design objectives. They were monetary cost, network delay, maximum of hops between any source-destination pair, and network reliability. The main objective of this thesis was to utilize the Einstein's operator for the EN topology design problem using SE and ACO algorithms. This objective was accomplished by adapting the Einstein's operator to suit the problem using SE and ACO algorithms. The second objective was to propose a multi-objective SE algorithm based on the archiving approach. This objective was accomplished by applying capacity metric to the SE algorithm. The third objective was to analyze the 7-point preference scheme using SE and ACO algorithms. This objective was accomplished through empirical results by generating eight preference rules and then applying them to SE as well as to ACO. The final objective was to compare the approaches used in order to evaluate the performance of the Einstein's operator to the EN topology design problem. This objective was accomplished and results were discussed.

In the following section, the main findings and contributions of this thesis are summarized.

7.1 Summary and Conclusion

This thesis discussed the problem of EN topology design in Chapter 3. A detailed information was given including the literature review, a formal description of the problem, all design objectives, constraints, and assumptions. In addition, the fuzzy Einstein's operator was discussed as a method to combine the multiple design objectives into a single overall objective function. Also, the use of preferences in MOO problems was explained with the focus on the 7-point preference scheme. In order to evaluate the scheme, eight preference rules were generated.

In Chapter 4, the applicability and effectiveness of the Einstein's operator to the EN topology design problem was assessed through the Archived Multi-objective Fuzzy Simulated Evolution Algorithm (AFSE). The experiments were done on five test cases. The experimental results showed that as the test case size increases, the cost, the delay, and the number of hops are increases, while the reliability of the network decreases. The second important part in this chapter was to analyze the 7-point preference scheme based on the Einstein's operator. The scheme was evaluated through eight preference rules. Analysis of preferences with eight preference rules revealed that rate of satisfaction of rules is higher for small test cases as compared to large test cases.

The main goal of Chapter 5 was to analyze the Einstein's operator with fuzzy ant colony optimization (FACO) algorithm. The performance and the effectiveness of the Einstein's operator was evaluated. The same finding was found as in Chapter 4. Also, the 7-point preference scheme was analyzed using the eight preference rules. Analysis of eight preference rules revealed that for small test cases, the level of satisfaction of rules was better compared to large test cases. As the test case size increases, satisfaction of rules generally decreases.

Finally in Chapter 6, an overall comparison of AFSE and FACO was presented. The comparison revealed that, the Einstein's operator performed better with AFSE in small test cases and better with FACO in large test cases.

7.2 Future Work

Following are some directions for the future work.

- Application of the Einstein operator can be studied with the respect to its application to other iterative heuristics such as particle swarm optimization, cuckoo search, biogeography based optimization, genetic algorithms, differential evolution, simulated annealing, and many others.
- A more comprehensive study is required to evaluate under which conditions certain preference rules would work or fail. This can partially be accomplished through landscape analysis of the EN topology design to see how the search is structured in the search space.
- A comparative study of Einstein operator with other operators mentioned in this thesis can be done to evaluate as which operator would work the best for the underlying problem.
- A different formulation of the EN topology design could be done, thereby adding more design objectives (such as availability, utilization etc.), and then optimize that model with the different iterative heuristics.
- Application of other multi-objective decision techniques such as goal programming, weighted sum, lexicographic ordering, and others presented in this thesis.

References

Ahmadi, S., & Bathaee, S. M. T. (2015). Multi-objective genetic optimization of the fuel cell hybrid vehicle supervisory system: Fuzzy logic and operating mode control strategies. *International Journal of Hydrogen Energy, 40*(36), 12512-12521.

Al-Betar, M. A., Khader, A. T., & Zaman, M. (2012). University Course Timetabling Using a Hybrid Harmony Search Metaheuristic Algorithm. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 42*(5), 664-681.

Al-Taani, A. T., & Al-Awad, N. A. K. (2009). An Empirical Analysis of Arabic WebPages Classification using Fuzzy Operators. *International Journal of Engineering and Mathematical Sciences, 5*(1).

Aparicio-Pardo, R., Skorin-Kapov, N., Pavon-Marino, P., & Garcia-Manrubia, B. (2012). (Non-)Reconfigurable Virtual Topology Design Under Multihour Traffic in Optical Networks. *Networking, IEEE/ACM Transactions on, 20*(5), 1567-1580.

Arora, J. (2004). *Introduction to Optimum Design*. Burlington, MA, USA: Academic Press.

Arora, J. S. (2012). Chapter 17 - Multi-objective Optimum Design Concepts and Methods *Introduction to Optimum Design - Third Edition,* pp. 657 - 679.

Bandyopadhyay, S., Saha, S., Maulik, U., & Deb, K. (2008). A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA. *IEEE Transactions on Evolutionary Computation, 12*(3), 269-283.

Birattari, M., Pellegrini, P., & Dorigo, M. (2007). On the Invariance of Ant Colony Optimization. *Evolutionary Computation, IEEE Transactions on, 11*(6), 732-742.

Cara, A. B., Wagner, C., Hagras, H., Pomares, H., & Rojas, I. (2013). Multiobjective Optimization and Comparison of Nonsingleton Type-1 and Singleton Interval Type-2 Fuzzy Logic Systems. *Fuzzy Systems, IEEE Transactions on, 21*(3), 459-476.

Chen, C. H., Liu, T. K., & Chou, J. H. (2013). Integrated Short-Haul Airline Crew Scheduling Using Multiobjective Optimization Genetic Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 43*(5), 1077-1090.

Chun-Wei, T., & Rodrigues, J. J. P. C. (2014). Metaheuristic Scheduling for Cloud: A Survey. *Systems Journal, IEEE, 8*(1), 279-291.

Coello, C. A. C. (1999). A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems, 1*(3), 269 - 308.

Colorni, A., Dorigo, M., Maniezzo, V., Varela, F. J., & Bourgine, P. (1992). Distributed optimization by ant colonies. *European Conference on Artificial Life* (pp. 134-142). Dipartimento di Elettronica, Politecnico di Milano, Italy.

Cordón, O., Herrera, F., & Peregrín, A. (1997). Applicability of the fuzzy operators in the design of fuzzy logic controllers. *Fuzzy Sets and Systems, 86*(1), 15-41.

Cvetkovic, D., & Parmee, I. C. (2002). Preferences and their application in evolutionary multiobjective optimization. *Evolutionary Computation, IEEE Transactions on, 6*(1), 42-57.

da Silva, H. A., de Souza Britto Jr., A., de Oliveira, L. E. S., & Koerich, A. L. (2013). Network infrastructure design with a multilevel algorithm *Expert Systems with Applications, 40*(9), 3471 - 3480.

De Leon-Aldaco, S. E., Calleja, H., & Aguayo Alquicira, J. (2015). Metaheuristic Optimization Methods Applied to Power Converters: A Review. *Power Electronics, IEEE Transactions on, 30*(12), 6791-6803.

de Salles, D. C., Gonçalves Neto, A. C., & Marujo, L. G. (2016). Using fuzzy logic to implement decision policies in system dynamics models. *Expert Systems with Applications, 55*, 172-183.

Dombi, J. (1982). A general class of fuzzy operators, the demorgan class of fuzzy operators and fuzziness measures induced by fuzzy operators *Fuzzy Sets and Systems, 8*(2), 149 - 163.

Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *Computational Intelligence Magazine, IEEE, 1*(4), 28-39.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 26*(1), 29-41.

Dorigo, M., & Stützle, T. (2004). Ant Colony Optimization for NP-Hard Problems *Ant Colony Optimization* (pp. 153-222): MIT Press.

Dubois, D., & Prade, H. (1979). Operations in Fuzzy-valued Logic. *Information and Control, 43*, 224-240.

Dubois, D., & Prade, H. (1982). A Class of Fuzzy Measures Based on Triangular Norms. *International Journal of General Systems, 8*, 105-116.

Dubois, D., Prade, H., & Sedes, F. (2001). Fuzzy logic techniques in multimedia database querying: a preliminary investigation of the potentials. *IEEE Transactions on Knowledge and Data Engineering, 13*(3), 383-392.

El-Ghazali, T. (2009). *Metaheuristics : From Design to Implementation.* Hoboken, NJ, USA: John Wiley and Sons.

Elbaum, R., & Sidi, M. (1996). Topological design of local-area networks using genetic algorithms. *Networking, IEEE/ACM Transactions on, 4*(5), 766-778.

Ersoy, C., & Panwar, S. S. (1993). Topological design of interconnected LAN/MAN networks. *Selected Areas in Communications, IEEE Journal on, 11*(8), 1172-1182.

Estepa, R., Estepa, A., Cupertino, T., Vozmediano, J. M., & Madinabeitia, G. (2011). A Productivity-Based Approach to LAN Topology Design. *Communications Letters, IEEE, 15*(3), 349-351.

Fonseca, C., & Fleming, P. (1993). Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion, and Generalization *5th International Conference on Genetic Algorithms* (pp. 416-423).

Frank, M. (1979). On the simultaneous associativity of F(x, y) and x+ y− F(x, y). *Aequationes Mathematica, 19*(1), 194.

Garg, H. (2016). Generalized intuitionistic fuzzy interactive geometric interaction operators using Einstein t-norm and t-conorm and their application to decision making. *Computers & Industrial Engineering, 101*, 53-69.

Gholamian, N., Mahdavi, I., Tavakkoli-Moghaddam, R., & Mahdavi-Amiri, N. (2015). Comprehensive fuzzy multi-objective multi-

product multi-site aggregate production planning decisions in a supply chain under uncertainty. *Applied Soft Computing, 37*, 585-607.

Gobjuka, H., & Breitbart, Y. J. (2010). Ethernet Topology Discovery for Networks With Incomplete Information. *Networking, IEEE/ACM Transactions on, 18*(4), 1220-1233.

Greenwood, G. W., Hu, X., & D'Ambrosio, J. G. (1997). Fitness Functions for Multiple Objective Optimization Problems: Combining Preferences with Pareto Rankings *FOGA* (Vol. 4, pp. 437-455): Morgan Kaufmann.

Guiwu, W., & Xiaofei, Z. (2013). Induced hesitant interval-valued fuzzy Einstein aggregation operators and their application to multiple attribute decision making. *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology, 24*(4), 789-803.

Gupta, A. K., & Dally, W. J. (2006). Topology optimization of interconnection networks. *Computer Architecture Letters, 5*(1), 10-13.

Hamacher., H. (1978). Ueber logische Verknupfungen Unschalfer Aussagen und deren Zugehoerige Bewertungs-funktione. *Progress in Cybernetics and Systems Research, 3*, 276-288.

Hsu-Chih, H. (2015). A Taguchi-Based Heterogeneous Parallel Metaheuristic ACO-PSO and Its FPGA Realization to Optimal Polar-Space Locomotion Control of Four-Wheeled Redundant Mobile Robots. *Industrial Informatics, IEEE Transactions on, 11*(4), 915-922.

Jiang, S., Ong, Y. S., Zhang, J., & Feng, L. (2014). Consistencies and Contradictions of Performance Metrics in Multiobjective

Optimization. *IEEE Transactions on Cybernetics, 44*(12), 2391-2404.

Jyothi, S. A., Singla, A., Godfrey, B., & Kolla, A. (2014). Measuring and Understanding Throughput of Network Topologies. *CoRR, abs/1402.2531*.

Khan, S., & Engelbrecht, A. (2009). Fuzzy hybrid simulated annealing algorithms for topology design of switched local area networks. *Soft Computing, 13*(1), 45-61.

Khan, S., & Engelbrecht, A. (2012). A fuzzy particle swarm optimization algorithm for computer communication network topology design. *Applied Intelligence, 36*(1), 161-177.

Khan, S. A. (1999). *Topology Design of Enterprise Network.* (Master Dessertation), King Fahd University of Petroleum and Minerals.

Khan, S. A. (2009). *Design and Analysis of Evolutionary and Swarm Intelligence Techniques for Topology Design of Distributed Local Area Networks.* (Ph.D. dessertation), University of Pretoria.

Khan, S. A. (2013, 19-21 Nov. 2013). *Application of extremal optimization algorithm to multi-objective topology design of enterprise networks.* Paper presented at the Computer, Control, Informatics and Its Applications (IC3INA), 2013 International Conference on.

Khan, S. A., & Engelbrecht, A. P. (2007). A new fuzzy operator and its application to topology design of distributed local area networks *Information Sciences, 177*(13), 2692 - 2711.

Kleeman, M. P., Seibert, B. A., Lamont, G. B., Hopkinson, K. M., & Graham, S. R. (2012). Solving Multicommodity Capacitated Network Design Problems Using Multiobjective Evolutionary

Algorithms. *Evolutionary Computation, IEEE Transactions on, 16*(4), 449-471.

Kling, R.-M., & Banerjee, P. (1990). Optimization by Simulated Evolution with Applications to Standard Cell Placement *Proceedings of the 27th ACM/IEEE Design Automation Conference* (pp. 20-25). Orlando, Florida, USA: ACM.

Kling, R. M., & Banerjee, P. (1989). ESp: Placement by simulated evolution. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 8*(3), 245-256.

Kumar, R., & Banerjee, N. (2011). Multiobjective network topology design *Applied Soft Computing, 11*(8), 5120 - 5128.

Lam, A. Y. S., & Li, V. O. K. (2010). Chemical-Reaction-Inspired Metaheuristic for Optimization. *Evolutionary Computation, IEEE Transactions on, 14*(3), 381-399.

Li, H.-X., & Yen, V. (1995). *Fuzzy sets and fuzzy decision making.*: CRC Press.

Majhi, B., & Anish, C. M. (2015). Multiobjective optimization based adaptive models with fuzzy decision making for stock market forecasting. *Neurocomputing, 167*, 502-511.

McMillan, T. (2011). Cisco Networking Essentials (1). Hoboken, US: Sybex.

Miettinen, K., Zitler, E., Deb, K., Thiele, L., Coello Coello, C. A., & Corne, D. (2001). Some methods for nonlinear multi-objective optimization. *First International Conference on Evolutionary Multi-Criterion Optimization*. Dept. of Math. Inf. Technol., Jyvaskyla Univ., Finland.

Mutingi, M., & Mbohwa, C. (2013, 10-13 Dec.). *A fuzzy simulated evolution algorithm for multi-objective homecare worker scheduling.* Paper presented at the 2013 IEEE International Conference on Industrial Engineering and Engineering Management.

Onieva, E., Hernandez-Jayo, U., Osaba, E., Perallos, A., & Zhang, X. (2015). A multi-objective evolutionary algorithm for the tuning of fuzzy rule bases for uncoordinated intersections in autonomous driving. *Information Sciences, 321*, 14-30.

Pierre, S., & Legault, G. (1998). A genetic algorithm for designing distributed computer network topologies. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 28*(2), 249-258.

Premprayoon, P., & Wardkein, P. (2005). Topological Communication Network Design using Ant Colony Optimization *The 7th International Conference on Advanced Communication Technology* (pp. 1147-1151).

Sait, S. M., El-Maleh, A. H., & Al-Abaji, R. H. (2006). Evolutionary algorithms for VLSI multi-objective netlist partitioning. *Engineering Applications of Artificial Intelligence, 19*(3), 257-268.

Sait, S. M., & Khan, J. A. (2003). Simulated evolution for timing and low power VLSI standard cell placement. *Engineering Applications of Artificial Intelligence, 16*(5–6), 407-423.

Schweizer, B., & Sklar, A. (1963). Associative Functions and Abstract Semigroups. *Publ. Math. Debrecen, 10*, 69-81.

Silva, H. A. d., de Souza Britto Jr, A., Oliveira, L. E. S. d., & Koerich, A. L. (2013). Network infrastructure design with a multilevel algorithm. *Expert Systems with Applications, 40*(9), 3471-3480.

Sung, Y. E., Xin, S., Rao, S. G., Xie, G. G., & Maltz, D. A. (2011). Towards Systematic Design of Enterprise Networks. *Networking, IEEE/ACM Transactions on, 19*(3), 695-708.

Thabti, B., Youssef, H., Mahjoub, A. R., & Meddeb, A. (2014, 19-21 Aug.). *Simulated evolution based algorithm versus exact method for virtual private network design.* Paper presented at the 2014 10th International Conference on Natural Computation (ICNC).

Wang, C.-S., & Chang, C.-T. (2008). Integrated Genetic Algorithm and Goal Programming for Network Topology Design Problem With Multiple Objectives and Multiple Criteria. *Networking, IEEE/ACM Transactions on, 16*(3), 680-690.

Wang, W., & Liu, X. (2013). The multi-attribute decision making method based on interval-valued intuitionistic fuzzy Einstein hybrid weighted geometric operator. *Computers & Mathematics with Applications, 66*(10), 1845-1856.

Wang, Y.-J. (2014). A fuzzy multi-criteria decision-making model by associating technique for order preference by similarity to ideal solution with relative preference relation *Information Sciences, 268*(0), 169 - 184.

Weber, S. (1983). A general concept of fuzzy connectives, negations and implications based on t-norms and t-conorms. *Fuzzy Sets and Systems, 11*(2), 115 - 134.

Weize, W., & Xinwang, L. (2013). Some Operations over Atanassov's Intuitionistic Fuzzy sets Based on Einstein T-norm and T-conorm.

International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 21(2), 263-276.

Xiaoning, S., Yu, G., Qingwei, C., & Weili, H. (2010). A multi-objective optimization evolutionary algorithm incorporating preference information based on fuzzy logic. *Computational Optimization and Applications, 46*(1), 159 - 188.

Xu, Z. (2007). Intuitionistic Fuzzy Aggregation Operators. *IEEE Transactions on Fuzzy Systems, 15*(6), 1179-1187.

Yager, R. R. (1977). Multiple objective decision-making using fuzzy sets. *International Journal of Man-Machine Studies, 9*(4), 375-382.

Yizhong, W., Jun, C., Lili, W., & Qingxin, M. (2012). Parameter Optimization of Classroom Energy-saving Controller Based on Fuzzy Multi-objective Optimization Algorithm. *Advanced Materials Research* (Vol. 393-395, pp. 867 - 870). Tianjin University of Science and Technology, School of Electronic Information and Automation, Tianjin, 300222, China, 66345.

Youssef, H., Sait, S., & Khan, S. (2001). Fuzzy Evolutionary Hybrid Metaheuristic for Network Topology Design. *Evolutionary Multi-Criterion Optimization,* pp. 400-415.

Youssef, H., Sait, S. M., & Ali, H. (2003). Fuzzy simulated evolution algorithm for VLSI cell placement. *Computers & Industrial Engineering, 44*(2), 227-247.

Youssef, H., Sait, S. M., & Khan, S. A. (2000). Fuzzy simulated evolution algorithm for topology design of campus networks *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on* (Vol. 1, pp. 180-187 vol.181).

Youssef, H., Sait, S. M., & Khan, S. A. (2002). Topology design of switched enterprise networks using a fuzzy simulated evolution algorithm *Engineering Applications of Artificial Intelligence, 15*(3–4), 327 - 340.

Zadeh, L. A. (1965). Fuzzy sets *Information and Control, 8*(3), 338 - 353.

Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning. III. *Information Sciences, 9*(1), 43-80.

Zhou, L., He, Y., Chen, H., & Liu, J. (2014). On compatibility of uncertain multiplicative linguistic preference relations based on the linguistic COWGA. *Applied Intelligence, 40*(2), 229-243.

Appendix (1)

Linear regression analysis

In many problems there are two or more variables that are related, and it is important to model and explore this relationship. Regression analysis is frequently used in this type of situation. In regression analysis data is analyzed from both designed and undesigned experiments. In *simple regression analysis*, the relationship between a single regressor variable x and a response variable y needs to be determined. The regressor variable x is usually assumed to be variable controllable by the experimenter. When the experiment is designed, the experimenter chooses that values of x and observes the corresponding value of y. The expected value of y for each value of x is given by the following mathematical model:

$$E(y|x) = w_0 + w_1 x + \epsilon$$

Where w_0 is the intercept, w_1 is called the regression coefficient associated with variable x, and ϵ is a random error with mean zero and variance σ^2. In the above equation, w_1 is of special interest. It signies the rate at which y changes if x is varied. A high value of w_1 will cause y to change at a faster rate when x is varied, while a low w_1 will have a slow effect on y when x is varied.

In the experiments conducted in this thesis, the following model was used:

$$E(\text{Objective}) = w_0 + w_1(\text{overall goodness}) + \epsilon$$

Here, the regressor variable is the overall goodness of solution, while the response variable is the membership value of design objective (cost, delay, hops, and reliability). Using the above model, many regression equations were developed to validate the eight preference rules using the Einstein's operator.

ملخص الأطروحة

في شبكات الحاسوب، يتم ربط عناصر الشبكة على صورة الهيكل الهرمي من أجل تصميم بنية تلك الشبكة. تتضمن هذه العناصر: الشبكات المحلية من المواقع الجغرافية المختلفة (LANs) والموجهات (Routers) ومحولات الطبقة الثالثة (Layer 3 switches). يعتبر تصميم بنية مثل هذه الشبكة من المسائل الصعبة (NP-hard problem). يتألف تصميم هذه الشبكة من عدة معايير التصميم والتي تتطلب أن تكون جميعها هي الأمثل في وقت واحد (simultaneous optimization). تتكون هذه المعايير من التكلفة النقدية (monetary cost) وزمن التأخير في الشبكة (network delay) وعدد النقلات من عقدتين (number of hops between two nodes) وفعالية الشبكة (network reliability). وبما أن بعض هذه المعايير تتعارض مع بعضها، يمكن اعتبار المسألة مشكلة الأمثل متعدد الأهداف (multi-objective optimization problem). في هذه الأطروحة، قد تم توظيف المنطق الضبابي (fuzzy logic) لمعالجة الطابع المتعدد الموضوعية للمسألة. وقد تم استخدام نوع معين من المشغلات الضبابية تعرف ب الآينشتاين (Einstein's operator) لتجميع معايير التصميم المشار إليها أعلاه في دالة الأمثل العددية الواحدة. يتم الجمع بين المشغل مع اثنين من خوارزميات التحسين (optimization algorithms) التي تعرف ب تطور المحاكاة (Simulated Evolution) ومستعمرة النمل الأمثل (Ant Colony Optimization). تتطلع الدراسة إلى أربعة أهداف رئيسية: الأول هو اقتراح خوارزمي تطور المحاكاة للتعامل مع مشكلة الأمثل متعدد الأهداف. والثاني هو لتطبيق ولتقييم أداء المشغل آينشتاين في مسألة تصميم متعدد الأهداف في سياق الخوارزميتين المذكورتين. المحور الثالث من هذه الدراسة هو تقييم فعالية وحساسية المشغل آينشتاين في وجود تفضيلات بين المعايير المذكورة والرابع هو لمقارنة اداء الخوارزميتين في مسألة تصميم متعدد الأهداف. تتلخص نتيجة الدراسة إلى أن الخوارزمية مستعمرة النمل الأمثل تمكنت من تحقيق نتائج أفضل في الاختبارات ذات الحجم الأكبر من حيث نوعية وجودة الحل. وعلاوة على ذلك، وفيما يتعلق تحليل التفضيلات بين المعايير، أنتجت الخوارزمية مستعمرة النمل الأمثل أيضا نتائج أفضل من تطور المحاكاة.

جامعة البحرين

كلية: تقنية المعلومات



التصميم التوبولوجي لشبكات الحاسوب باستخدام المنطق الضبابي والتفضيلات

مخطط بحث لاستكمال متطلبات الحصول على درجة الماجستير في تقنية المعلومات

إعـــداد

إبراهيم أحمد بامسعود

الرقم الجامعي: 20032199

إشـــراف

د. سلمان خان

(أستاذ مساعد)

جامعة البحرين

مـملكة البـحرين

ديسمبر 2016