

# Network Engineering Using Multi-Objective Evolutionary Algorithms

Atumbe Jules Baruani



Thesis presented in partial fulfilment  
of the requirements for the degree of  
Master of Science  
at the University of Stellenbosch

Dr. Antoine B. Bagula

December 2007

# Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature: .....

Date: .....

Copyright ©2007 Stellenbosch University

All rights reserved

# Abstract

We use Evolutionary Multi-Objective Optimisation (EMOO) algorithms to optimise objective functions that reflect situations in communication networks. These include functions that optimise Network Engineering (NE) objective functions in core, metro and wireless sensor networks. The main contributions of this thesis are threefold.

## **Routing and Wavelength Assignment (RWA) for IP backbone networks.**

Routing and Wavelength Assignment (RWA) is a problem that has been widely addressed by the optical research community. A recent interest in this problem has been raised by the need to achieve routing optimisation in the emerging generation multilayer networks where data networks are layered above a Dense Wavelength Division Multiplexing (DWDM) network. We formulate the RWA as both a single and a multi-objective optimisation problem which are solved using a two-step solution where (1) a set of paths are found using genetic optimisation and (2) a graph coloring approach is implemented to assign wavelengths to these paths. The experimental results from both optimisation scenarios reveal the impact of (1) the cost metric used which equivalently defines the fitness function (2) the algorithmic solution adopted and (3) the topology of the network on the performance achieved by the RWA procedure in terms of path quality and wavelength assignment.

## **Optimisation of Arrayed Waveguide Grating (AWG) Metro Networks.**

An Arrayed Waveguide Grating (AWG) is a device that can be used as a multiplexer or demultiplexer in WDM systems. It can also be used as a drop-and-insert element or even a wavelength router. We take a closer look at how the hardware and software parameters of an AWG can be fine tuned in order to maximise throughput and minimise the delay. We adopt a multi-objective optimisation approach for multi-service AWG-based single hop

metro WDM networks. Using a previously proposed multi-objective optimisation model as a benchmark, we propose several EMOO solutions and compare their efficiency by evaluating their impact on the performance achieved by the AWG optimisation process. Simulation reveals that (1) different EMOO algorithms can exhibit different performance patterns and (2) good network planning and operation solutions for a wide range of traffic scenarios can result from a well selected EMOO algorithm.

### **Wireless Sensor Networks (WSNs) Topology (layout) Optimisation.**

WSNs have been used in a number of application areas to achieve vital functions in situations where humans cannot constantly be available for certain tasks such as in hostile areas like war zones, seismic sensing where continuous inspection and detection are needed, and many other applications such as environment monitoring, military operations and surveillance. Research and practice have shown that there is a need to optimise the topology (layout) of such sensors on the ground because the position on which they land may affect the sensing efficiency. We formulate the problem of layout optimisation as a multi-objective optimisation problem consisting of maximising both the coverage (area) and the lifetime of the wireless sensor network. We propose different algorithmic evolutionary multi-objective methods and compare their performance in terms of Pareto solutions. Simulations reveal that the Pareto solutions found lead to different performance patterns and types of layouts.

# Opsomming

Ons gebruik "Evolutionary Multi-Objective Optimisation (EMOO)" algoritmes om teiken funksies, wat egte situasies in kommunikasie netwerke voorstel, te optimiseer. Hierdie sluit funksies in wat "Network Engineering" teiken funksies in kern, metro en wireless sensor netwerke optimiseer. Die hoof doelwitte van hierdie tesis is dus drievuldig.

## **RWA vir IP backbone netwerke**

"Routing and Wavelength Assignment (RWA)" is 'n probleem wat al menigte kere in die optiese navorsings kringe aangespreek is. Belangstelling in hierdie veld het onlangs ontstaan a.g.v. die aanvraag na die optimisering van routing in die opkomende generasie van veelvuldige vlak netwerke waar data netwerke in 'n vlak hoër as 'n "Dense Wavelength Division Multiplexing (DWDM)" netwerk gele is. Ons formuleer die RWA as beide 'n enkele and veelvuldige teiken optimiserings probleem wat opgelos word deur 'n 2-stap oplossing waar (1) 'n stel roetes gevind word deur genetiese optimisering te gebruik en (2) 'n grafiek kleuring benadering geïmplementeer word om golflengtes aan hierdie roetes toe te ken. Die eksperimentele resultate van beide optimiserings gevalle vertoon die impak van (1) die koste on wat gebruik word wat die ekwivalente fitness funksie definieer, (2) die algoritmiese oplossing wat gebruik word en (3) die topologie van die netwerk op die werkverrigting van die RWA prosedure i.t.v. roete kwaliteit en golflengte toekenning.

## **Optimisering van AWG Metro netwerk**

'n "Arrayed Waveguide Grating (AWG)" is 'n toestel wat gebruik kan word as 'n multi-pleksor of demulti-pleksor in WDM sisteme. Dit kan ook gebruik word as 'n val-en-inplaas element of selfs 'n golflengte router. Kennis word ingestel na hoe die hardeware en sagte-ware parameters van 'n AWG ingestel kan word om die deurset tempo te maksimeer en ver-

tragings te minimiseer. Ons neem 'n multi-teiken optimiserings benadering vir multi diens, AWG gebaseerde, enkel skakel, metro WDM netwerke aan. Deur 'n vooraf voorgestelde multi teiken optimiserings model as "benchmark" te gebruik, stel ons 'n aantal EMOO oplossings voor en vergelyk ons hul effektiwiteit deur hul impak op die werkverrigting wat deur die AWG optimiserings proses bereik kan word, te vergelyk. Simulasie modelle wys dat (1) verskillende EMOO algoritmes verskillende werkverrigtings patrone kan vertoon en (2) dat goeie netwerk beplanning en werking oplossings vir 'n wye verskeidenheid van verkeer gevalle kan plaasvind a.g.v 'n EMOO algoritme wat reg gekies word.

### **"Wireless Sensor Network" Topologie Optimisering**

WSNs is al gebruik om belangrike funksies te verrig in 'n aantal toepassings waar menslike beheer nie konstant beskikbaar is nie, of kan wees nie. Voorbeelde van sulke gevalle is oorlog gebiede, seismiese metings waar aaneenlopende inspeksie en meting nodig is, omgewings meting, militêre operasies en bewaking. Navorsing en praktiese toepassing het getoon dat daar 'n aanvraag na die optimisering van die topologie van sulke sensors is, gebaseer op gronde van die feit dat die posisie waar die sensor beland, die effektiwiteit van die sensor kan affekteer. Ons formuleer die probleem van uitleg optimisering as 'n veelvuldige vlak optimiserings probleem wat bestaan uit die maksimering van beide die bedekkings area en die leeftyd van die wireless sensor netwerk. Ons stel verskillende algoritmiese, evolutionêre, veelvuldige vlak oplossings voor en vergelyk hul werkverrigting i.t.v Pareto oplossings. Simulasie modelle wys dat die Pareto oplossings wat gevind word lei na verskillende werkverrigtings patrone en uitleg tipes.

# Acknowledgments

First of all I would like to thank the Almighty God for giving strength and knowledge that have helped me to finish this thesis. Special thanks to my supervisor Dr. Antoine Bagula for his invaluable assistance and guidance throughout my studies. I would like to thank the head of Telkom Center of Excellence Prof AE Krzesinski for his insightful comments and corrections of my thesis. Many thanks to all my colleagues Lusilao Zodi, Gaston Kuzamunu Mazandu, Tayseer Fath Elrahman, Jean Andre Okito Lokake for their group discussions and constructive criticism of my work.

Special thanks to my mother Mlebinge Esholoke, my dad Baruani Ishibutunga for their support, encouragements and trust in me since my childhood. Thanks to my brothers Lwebula Baruani, Aeloheba Baruani, Ilangyi Baruani and Mlumbi Baruani for their financial and moral supports.

Last but not least, I would like to thank my wife to be Rosine for encouraging and supporting me throughout my studies.

## List of Publications

Atumbe Jules Baruani and Antoine B.Bagula, *On Routing IP Traffic Using Single- and Multi-objective Genetic Optimization*, Southern African Telecommunication Networks and Applications Conference (SATNAC), 2006



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Evolutionary Multi-Objective Optimisation . . . . .	1
1.3	Why are EMOOs Suitable? . . . . .	2
1.4	Literature Review . . . . .	3
1.5	Thesis Contributions . . . . .	5
1.6	Thesis Outline . . . . .	7
<b>2</b>	<b>Evolutionary Multi-Objective Optimisation</b>	<b>9</b>
2.1	Introduction . . . . .	9
2.2	Genetic Algorithms . . . . .	10
2.3	MOP Basic Concepts . . . . .	12
2.3.1	What is MOP . . . . .	12
2.3.2	Pareto Optimality . . . . .	13
2.4	MOP Solution Approaches . . . . .	14
2.4.1	Weighted Sum Approach . . . . .	15

---

2.4.2	Vector Evaluated Genetic Algorithm: VEGA . . . . .	16
2.4.3	Multi-Objective Genetic Algorithm: MOGA . . . . .	17
2.4.4	Non-dominated Sorting Genetic Algorithm: NSGA . . . . .	18
2.5	Population Diversity . . . . .	20
2.5.1	Fitness Sharing . . . . .	20
2.5.2	Crowding Distance . . . . .	22
2.5.3	Cell Based Density . . . . .	22
2.5.4	Elitism . . . . .	23
<b>3</b>	<b>Routing and Wavelength Assignment</b>	<b>25</b>
3.1	Introduction . . . . .	25
3.2	Routing and Wavelength Assignment . . . . .	26
3.2.1	Overview . . . . .	26
3.2.2	Formulation of the RWA Problem . . . . .	27
3.3	Routing Problem . . . . .	29
3.3.1	Problem Statement . . . . .	29
3.3.2	Single Objective Optimisation . . . . .	29
3.3.3	Multi-Objective optimisation . . . . .	33
3.3.4	Multi-Objective Solution . . . . .	34
3.3.5	Simulation Results . . . . .	35
3.3.6	Multi-Objective Optimisation . . . . .	40
3.4	Wavelength Assignment . . . . .	46

---

<b>4</b>	<b>Optimisation of an AWG-based Metro WDM Network</b>	<b>51</b>
4.1	Introduction . . . . .	51
4.2	Wavelength Division Multiplexing . . . . .	52
4.3	Arrayed Waveguide Grating . . . . .	54
4.3.1	Overview . . . . .	54
4.3.2	Architecture of AWG and Working Principles . . . . .	55
4.4	AWG-Based Network . . . . .	56
4.4.1	Network Architecture . . . . .	56
4.4.2	Medium Access Control (MAC) Protocol . . . . .	56
4.5	Formulation of The Multi-Objective Problem . . . . .	58
4.5.1	Network Overview . . . . .	58
4.5.2	Objective Function . . . . .	59
4.6	Experimental Results . . . . .	63
4.6.1	Parameter Settings . . . . .	64
4.6.2	Solution Approaches: EMOO Algorithms . . . . .	64
4.6.3	Comparison of the Algorithms . . . . .	67
<b>5</b>	<b>Layout Optimisation of a Wireless Sensor Network</b>	<b>75</b>
5.1	What are Sensor Networks ? . . . . .	75
5.2	Sensor Coverage . . . . .	76
5.3	Sensor Design Issues . . . . .	77
5.4	Sensor Energy Consumption . . . . .	77

Contents	xi
<hr/>	
5.5 The Optimisation Model . . . . .	78
5.5.1 Overview . . . . .	78
5.5.2 Assumptions . . . . .	79
5.5.3 Calculation of Objective Functions . . . . .	79
5.5.4 Problem Formulation . . . . .	81
5.5.5 Algorithmic Solution . . . . .	81
5.6 Experimental Results . . . . .	82
5.6.1 Parameter Settings . . . . .	82
5.6.2 Genetic Operators . . . . .	82
5.6.3 Pareto Points Analysis . . . . .	83
5.7 Layout Analysis . . . . .	85
5.8 Analysis of the Performance of the Algorithms . . . . .	87
<b>6 Conclusion and Future Work</b>	<b>91</b>
<b>Bibliography</b>	<b>94</b>

# List of Figures

2.1	GA flowchart . . . . .	12
2.2	Example of a Pareto front . . . . .	13
2.3	NSGA flowchart . . . . .	19
2.4	Example of cell density for two functions $f_1$ and $f_2$ . . . . .	23
3.1	An example of a simple network . . . . .	26
3.2	A typical chromosome: each gene encodes a path corresponding to one source destination pair . . . . .	32
3.3	MA algorithm . . . . .	33
3.4	Route usage for the African network . . . . .	37
3.5	Route usage for the USA network . . . . .	37
3.6	Route lengths for the African network . . . . .	39
3.7	Route lengths for the USA network . . . . .	39
3.8	Route multiplicity for the African network . . . . .	40
3.9	Route multiplicity for the USA network . . . . .	40
3.10	Route used by both algorithms for the African network (power function with $\alpha = 0.75$ ) . . . . .	40

---

3.11	Route used by both algorithms for the USA network (power function with $\alpha = 0.75$ ) . . . . .	41
3.12	The USA (a) and African (b) networks . . . . .	42
3.13	Pareto front of the African network . . . . .	46
3.14	Pareto front of the USA network . . . . .	47
4.1	An example of a simple WDM system . . . . .	54
4.2	An example of a AWG-based simple network . . . . .	57
4.3	A typical frame of a data and control packet . . . . .	58
4.4	Pareto front obtained for $\sigma = 0.9$ . . . . .	65
4.5	Pareto front for $\sigma = 0.1$ , $q = 0.3$ and $q = 0.8$ . . . . .	67
4.6	Pareto front of an $2 \times 2$ AWG for $\sigma = 0.9$ and $q = 0.9$ . . . . .	68
4.7	Pareto front of an $2 \times 2$ AWG for $\sigma = 0.9$ and $q = 0.4$ . . . . .	68
4.8	Pareto front of an $4 \times 4$ AWG for $\sigma = 0.9$ and $q = 0.9$ . . . . .	68
4.9	Pareto front of an $4 \times 4$ AWG for $\sigma = 0.9$ and $q = 0.4$ . . . . .	68
4.10	Generation counter vs number of Pareto points for NSGAI . . . . .	69
4.11	Generation counter vs number of Pareto points for MOGA . . . . .	69
4.12	Generation counter vs number of Pareto points for PAES . . . . .	70
4.13	Generation counter vs number of Pareto points for Micro-GA . . . . .	70
4.14	Pareto front for NSGAI . . . . .	72
4.15	Pareto front for MOGA . . . . .	72
4.16	Pareto front for PAES . . . . .	72

---

4.17 Pareto front for Micro-GA . . . . .	72
5.1 Area coverage . . . . .	77
5.2 Point coverage . . . . .	77
5.3 Barrier coverage . . . . .	77
5.4 Generation counter vs number of Pareto points for Micro-GA . . . . .	84
5.5 Generation counter vs number of Pareto points for NSGAI . . . . .	84
5.6 Pareto front for Micro-GA . . . . .	85
5.7 Pareto front for NSGAI . . . . .	85
5.8 Sensors packed together around the HECN . . . . .	86
5.9 Sensors organised in a hub-and-spoke pattern . . . . .	88
5.10 Sensors nodes in a linear array . . . . .	89

# List of Tables

3.1	Percentage of routes used by each algorithm for the USA network . . . . .	37
3.2	Percentage of routes used by each algorithm for the African network . . . . .	38
3.3	Statistics for the USA network, displaying the maximum and the average lengths and standard deviation . . . . .	38
3.4	Statistics for the African network, displaying the maximum and the average lengths and standard deviation . . . . .	39
3.5	Table showing wavelengths for the African network . . . . .	43
3.6	Table showing wavelengths for the USA network . . . . .	43
3.7	Pareto points for problem 1 for the African network . . . . .	44
3.8	Pareto points for problem 2 for the African network . . . . .	44
3.9	Pareto points for problem 1 for the USA network . . . . .	45
3.10	Pareto points for problem 2 for the USA network . . . . .	45
3.11	Pareto points with corresponding wavelengths for the African network . . . . .	49
3.12	Pareto points with corresponding wavelengths for the USA network . . . . .	50
4.1	Table displaying parameters of the optimisation model . . . . .	59
4.2	Pareto points for PAES . . . . .	73



---

4.3	Pareto points for Micro-GA . . . . .	73
4.4	Pareto points for NSGAI . . . . .	74
4.5	Pareto points for MOGA . . . . .	74
5.1	Sample of Pareto points found by NSGAI . . . . .	90
5.2	Sample of Pareto points found by Micro-GA . . . . .	90

# Chapter 1

## Introduction

### 1.1 Background

The potential of evolutionary algorithms was first hinted at by Rosenberg [1] when he noticed that genetic-based search could be used to solve multi-objective optimisation problems. However little attention was given to this field of study until the last two decades when interest has grown considerably as indicated in journals, conferences and interest groups on the Internet. This considerable interest can be explained by the fact that there are still many open questions in this field of study. The first work in genetic algorithms was produced by Holland [2]. Since then genetic algorithms have been used to solve many problems in different fields of study. Although existing Evolutionary Multi-Objective Optimisation (EMOO) techniques have proven to be applicable to a number of problems, not many algorithms have been developed when we compare with traditional methods. We take advantage of these algorithms and use them in this work to solve some of the problems that today's broadband communications are faced with.

### 1.2 Evolutionary Multi-Objective Optimisation

Evolutionary Multi-Objective Optimisation (EMOO) comprises heuristic search methods which mimic the evolution of natural selection. EMOOs use the principle of survival of the fittest, that is in nature individuals compete for scarce resources for their survival and

those that are fitter are more likely to survive and breed off-spring for the next generation.

Many multi-objective problems arise in nature . For example we may wish to minimise the cost of routing the traffic in a network to avoid flow competition on links (interference) while minimising the delay. These two objectives are in conflict and there is no single point that minimises both objectives simultaneously. Flow competition avoidance can lead to routing the traffic over longer routes with longer delays.

Unlike in the single objective optimisation case where one searches for an optimum solution, in multi-objective optimisation problems the concept of optimality becomes vague as there is usually no single point that can optimise all the objective functions in question at once. Methods such as weighted sum [3] combine the two objectives into a single objective to solve the problem as a single objective optimisation problem. However this method does not always guarantee an optimum solution because choosing the weight vector can prove to be a daunting task. Even if one is familiar with the problem, choosing appropriate weights is not an easy problem because small changes in the weight vector can yield very different solutions.

It is therefore desirable to find a set of points which present a trade-off between the objective functions and the user can then choose between those points depending on the situation at hand.

### 1.3 Why are EMOOs Suitable?

There are a number of analytical and numerical methods that handle multi-objective optimisation problems quite well but their ability to solve some real world problems is limited. Stochastic optimisation techniques like Tabu search, ant colony optimisation and simulated annealing have also been developed to solve mutli-objective optimisation problems. However these methods do not guarantee to find optimal trade-off points as the solutions obtained often get stuck at good approximations [4].

EMOOs use the concept of biological evolution to solve optimisation problems. Several solutions can be sought in one run through the use of a population. This is because the population is made of chromosomes each of which represents a possible solution. Although

they do not guarantee to find very good solution points as is the case in traditional optimisation problems, EMOOs have been widely used and preferred to their traditional methods counterpart because:

- EMOOs can find a set of trade-off points in a single run using the so-called population because each chromosome in the population represents a possible solution point from the search space.
- EMOOs are not vulnerable to the shape of the curve consisting of the solution points because genetic algorithms work on the encoding of the variables instead of manipulating the variables themselves.
- EMOOs can handle complex problems with discontinuities, noisy functions or even disjoint feasible spaces.

## 1.4 Literature Review

In recent years, Evolutionary Multi-Objective Optimisation (EMOO) algorithms have been used in a number of applications ranging from computer science, engineering, manufacturing, telecommunication etc. A comprehensive list of both Masters and PhD theses can be found at the Coello website [5], indicating a growing interest in this field.

Simple genetic algorithms have been used to solve the problem of routing in computer networks. Mitsuo Gen *et al.* [6] use a simple genetic algorithm to solve the bi-criteria network optimisation problem including the Maximum Flow problem (MXF) and the Minimum Cost Flow problem (MCF). They found a set of trade-off points that give a possible maximum flow with minimum cost in a network. A faster genetic algorithm for finding network paths is presented in [7]. The authors claim to have found some flaws in [6] and present their new algorithm for solving the shortest path problem. It is based on the methodology of dynamic coding of the priority of vertex and gene weights. Further genetic algorithms and their variants are used to solve different problems in networks as seen in [8, 9, 10].

Apart from finding a set of paths to route traffic, evolutionary algorithms have been used in designing networks such as WDM and mobile telecommunication networks. A genetic

algorithm was used to solve the routing and wavelength assignment problem in WDM networks [11]. The problem is formulated as a single and a multi-objective optimisation problem. A hybrid approach is used and it is based on  $k$ -shortest pre-computed paths from each source destination pair. A function that expresses the frequency of occurrence of a link in different source destination paths is used to evaluate the fitness of the chromosome. Another genetic algorithm based methodology for optimising multi-service convergence in metro WDM network is developed in [12]. An optimisation model is developed and EMOOs are used to solve the Multi-Objective Problem (MOP). An Arrayed Waveguide Grating (AWG) based network is considered with the aim of providing high throughput and low delay connectivity. Recent studies have shown that AWG has the potential for providing a good throughput-delay performance in metropolitan area networks. However several parameters need to be set, including software and hardware parameters. The study found that the solution to the MOP can be very useful in the planning and operation of a variety of traffic scenarios.

A multi-objective genetic algorithm for radio network optimisation is studied in [13]. In this study, the author formulates the problem of placing a Base Station (BS) in a radio network as a MOP. Placing a BS is a complex problem as it involves setting up different antennas at different pre-defined sites. However determining the number of antennas and their respective configurations may prove to be a complex exercise. Thanks to the powerful capabilities of EMOO, they have been used to solve this problem and the results found in real life situations are very encouraging [13].

Further network designs are studied in the case of Wireless Sensor Networks (WSN). A multi-objective genetic algorithm for the automated planning of a wireless sensor network to monitor a critical facility is investigated in [14]. The authors examine the placement of sensor nodes in an optimal way. Sensors are deployed to the ground from an aircraft. An EMOO is then used to design a WSN that provides clear assessments of movements in and out of the critical facility. At the same time the design should minimize both the likelihood of sensor nodes being discovered and the number of sensors to be dropped. Finally, the optimisation of the WSN is investigated in [15]. In this paper, the multi-objective GA is used to maximise both the network coverage and the lifetime in a WSN. The sensing and communication ranges of sensor nodes are kept the same and the results reveal two interesting types of layouts [15].

In this thesis we look at the design of core networks, WDM in metro networks and also in wireless sensor networks. We use some of the models described above and execute the models on different EMOO algorithms. We compare the results obtained from different algorithms and draw our conclusions based on the performance of each algorithm.

## 1.5 Thesis Contributions

Traffic Engineering (TE) and Network Engineering (NE) are two management methods which are commonly used in communication networks to achieve Quality of Service (QoS) agreements between the offered traffic and the available resources. TE moves traffic to where the resources are located in the network while NE moves the network resources to where traffic is offered to the network. In this thesis we use Evolutionary Multi-Objective Optimisation (EMOO) algorithms to optimise objective functions that reflect real situations in communications networks. Some of the functions are TE objective functions consisting of finding the paths to route the traffic offered to a network, and others are NE objective functions consisting of assigning the minimum number of wavelengths to the computed paths in Routing and Wavelength Assignment (RWA) settings. We also consider functions that express the optimal settings of a network by selecting the optimal working parameters for both hardware (equipment) and software (protocol) components of a network as these have a significant impact on the performance of a network. In Wireless Sensor Networks (WSNs), we consider objective functions which do not reflect hardware or NE requirements directly but seek to optimise the proper placement of nodes where they are deployed and also maximise their duration of use.

The main contributions of this thesis are

### **Routing and Wavelength Assignment (RWA) For Fixed IP Backbone Networks.**

Routing and Wavelength assignment (RWA) is a problem that has been widely addressed by the optical research community. A recent interest for this problem has been raised by the need to achieve routing optimisation in the emerging generation multilayer networks where

data networks are layered above a Dense Wavelength Division Multiplexing (DWDM) network. We formulate the RWA as both single and multi-objective optimisation problems which are solved using a two-step solution where (1) a set of paths is found using genetic optimisation and (2) a graph coloring approach is implemented to assign wavelengths to these paths. The experimental results from both optimisation scenarios reveal the impact of (1) the cost metric used which equivalently defines the fitness function (2) the algorithmic solution adopted and (3) the topology of the network on the performance achieved by the RWA procedure in terms of path quality and wavelength assignment.

### **Metro WDM Network optimisation Using Arrayed Waveguide Grating (AWG) Calibration.**

An Arrayed Waveguide Grating (AWG) is a device used as a multiplexer or demultiplexer in WDM systems. It can also be used as a drop-and-insert element or a wavelength router. Taking a closer look at how the hardware and software parameters of an AWG can be fine tuned in order to maximise throughput and minimise the delay, we formulate the AWG network optimisation as a multi-objective optimisation problem solved using evolutionary optimisation solutions based on genetic algorithms. Using a previously proposed multi-objective optimisation model as a benchmark, we propose several EMOO solutions and compare their efficiency by evaluating their impact on the performance achieved by the AWG optimisation process. Simulation reveals that (1) different EMOO algorithms can exhibit different performance patterns and (2) good network planning and operation solutions for a wide range of traffic scenario can result from a well selected EMOO algorithm.

### **Wireless Sensor Networks Layout optimisation Using Genetic optimisation.**

WSNs are vital in situations where humans cannot be available for certain tasks. For example in hostile areas such as war zones or in seismic sensing where continuous inspection and detection are needed, sensor networks become unavoidable. It is widely recognized that there is a need to optimise the layout of wireless sensor networks on the ground since the position on which they land may affect the sensing. Looking particularly at sensors

deployed in a military zone from an aircraft, we formulate the layout optimisation as a multi-objective problem that simultaneously maximises both the network coverage (area) and the lifetime of a WSN. We slightly change the way energy is depleted in a sensor node. In [15] the energy of a node is depleted by an arbitrary unit each time a node transmits information. In our case, for every transmission of information, the energy is depleted by an amount that is proportional to the distance to which the information is sent. We use several EMOO genetic algorithms to optimise these objective functions. The experimental results reveal that the Pareto solutions found lead to different performance patterns and types of layouts.

## 1.6 Thesis Outline

After presenting the problem at hand and the need to use evolutionary multi-objective optimisation, chapter 2 introduces evolutionary multi-objective optimisation giving a comprehensive background on genetic algorithms and in particular multi-objective genetic algorithms. We describe a few EMOO algorithms giving their working principles and their characteristics that differentiate them from other algorithms. We also present the main issue faced by the multi-objective optimisation problem namely, guiding the search towards the optimal region and also maintaining the diversity in the population. We explain the methods used by recent algorithms in order to tackle these problems. Such strategies include elitism, fitness sharing and scaling, cell based density and the crowding distance method.

In chapter 3, we solve the static Routing and Wavelength Assignment (RWA) problem. This problem is solved in two steps. First, we use both single and multi-objective optimisation in order to find plausible paths from each source destination pair in a network with the aim of routing different types of traffic on these paths and eventually introducing network load balancing. We use a cost function that takes into account link interferences in the case of single objective optimisation. In the case of multi-objective we add more objectives namely the weighted link delay as in the case of an  $M/M/1$  queuing system. The results show that we can find plausible paths that are suitable for different types of traffic. Second, we use the paths to form a logical network before we use a graph coloring algorithm to



---

assign wavelength in the network.

Chapter 4 presents a model which optimises the hardware and software components of an Arrayed Waveguide Grating (AWG) in a single metropolitan WDM network. AWGs have shown to provide high throughput and low delay connectivity in metropolitan and local area networks [12]. However the performance of an AWG largely depends on its software and hardware parameters. We make use of evolutionary genetic algorithms to solve this problem by finding suitable parameter combinations. The results are a set of points that represent throughput delay trade-off points that can also provide low delay for delay tolerant traffic.

Chapter 5 looks at a WSN where both its coverage and lifetime are maximised. To simplify the model, we consider sensors where the sensing radius is equal to the communication radius. This is not always the case in real situations. Designing an effective topology for a wireless network is never an easy problem especially in the case where the sensor nodes are dropped randomly. Once the sensors are deployed they must be positioned properly for their efficient operation. We look particularly at sensors deployed from an aircraft in a military zone. Their positions on the ground are not certain and the aim is to avoid them overlapping and being too close. On the other hand they must not be too distant as this may affect the lifetime of the sensors. This seems to be a difficult problem to solve and we make use of evolutionary multi-objective optimisation algorithms to solve the problem. The solutions are points that give us different type of topologies with different values of the lifetime which may be suitable for specific applications.

We conclude with chapter 6 where we briefly outline the main issues and findings of this work. We present our contributions and what we have learned from this work. We also mention challenges and future research work.

## Chapter 2

# Evolutionary Multi-Objective Optimisation

### 2.1 Introduction

The concept of optimisation in the case of single objective optimisation is straightforward in the sense that any optimisation technique will find the optimum value (if it exists) of the objective function. However, real world problems often have more than one objective function, and in this case we may wish to optimise these objectives simultaneously, giving rise to the Multi-Objective Problem (MOP). Multi-Objective optimisation, also known as Multi-Criteria optimisation consists of optimising a number of objective functions subject to some constraints. The concept of optimum becomes somewhat vague because the objectives to be optimised are always in conflict. For example we may want to optimise two objectives, to minimise the cost and improve the reliability of some item. It is clear that a highly reliable item will cost more and the item with less cost will have low reliability. It will therefore be difficult to find a solution that optimises both functions at the same time. The technique is then to find a set of trade-off points which give acceptable values with respect to each objective.

EMOO has attracted the interest of many researchers since early 1960 [16]. The pioneering work in the field of MOP was first carried out by Vilfredo Pareto whose name was attributed to the set of points that constitute an acceptable solution in MOP. There are a number of traditional methods (direct and gradient methods) to solve MOP problems. However

some problems have mathematical properties (such as discontinuities, noise) that makes it difficult for the traditional methods to solve. In the past two decades, genetic algorithms have emerged as a new and promising trend to solve MOP problems. They are able to eliminate the problems faced by the traditional methods with their successful application in different fields of study [17]. The category of optimisation techniques that uses GA to solve MOP problems is referred to as genetic multi-objective optimisation.

## 2.2 Genetic Algorithms

### Working principles

Genetic algorithms are heuristic search methods which are based on the idea of evolution. They mimic the process of survival of the fittest [18]. It is known that in nature, individual compete for scarce resources for their survival and those that are fitter are more likely to survive to the next generation and breed offspring to pass them onto the next generation while weak individuals die out and disappear.

Living creatures are organisms made of cells which contain chromosomes. Each chromosome contains genes each of which encodes a trait. Possible settings of a trait are called alleles. In order to reproduce, genes recombine and thereafter produce new chromosomes. However in optimisation terms, a chromosome is usually encoded as a string of zeros and ones, although there exist other types of encoding [19]. Genes are the bits that form the chromosome (a set of strings) and an allele would be the value of the gene which may differ depending on the gene.

Using the so-called population (the set of chromosomes), GAs can explore possible solutions simultaneously because each individual of the population represents a possible solution in the search space. Instead of working on the parameters of the problem itself, GAs work on the encoding of the parameters and then evaluate the goodness of each individual in the population using a fitness function.

The ability of GAs to perform better on a particular problem relies on how the algorithm manipulate the chromosomes, how it selects them and how it reproduces new chromosomes

for the next generation. There are more GA operators used depending on the problem at hand but the common ones are reproduction, crossover and mutation. During reproduction individuals are selected to participate in the breeding of new offspring for the next generation. Crossover involves two chromosomes which combine and exchange parts of their genes in order to breed new offspring. Mutation involves altering part of the chromosome by randomly changing part of its genes. Mutation is used in order to introduce diversity into the population. Crossover and mutation are each used depending on a probability which determines how often these operations should be carried out.

## Basic GA Algorithm

Below we describe the basic GA algorithm and show the corresponding flow chart in figure 2.1.

1. Generate a set of chromosomes (population) at random.
2. Evaluate the fitness of each chromosome in the population.
3. Reproduce a new population by repeatedly carrying out the following operations:
  - (a) Select two individuals in the population according to their respective fitnesses.
  - (b) Given the crossover probability, crossover the two selected individuals in order to breed new offspring.
  - (c) Mutate the new offspring according to the mutation probability.
4. Replace the old population with the new generated population.
5. Verify if the terminating condition is satisfied to end the algorithm otherwise return to the second step.

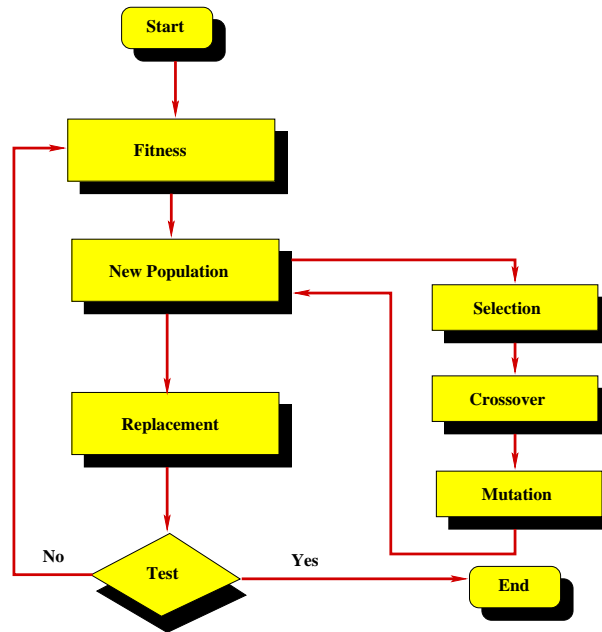


Figure 2.1: GA flowchart

## 2.3 MOP Basic Concepts

### 2.3.1 What is MOP

The problem of MOP consists of simultaneously optimising  $k$  objective functions subject to  $m$  constraints which may be inequalities and/or equality constraints. The solution to the problem is a vector of  $n$  decision variables, which presents the trade-off solution. Throughout this work, we will assume minimisation unless otherwise stated. Formally the MOP can be formulated as follows:

$$\begin{aligned} & \text{minimise} && F(\vec{x}) = (f_1(\vec{x}), \dots, f_k(\vec{x})) \\ & \text{subject to} && g_i(\vec{x}) \leq 0, \quad i = 1, \dots, m \end{aligned}$$

where  $\vec{x} \in \mathbb{R}^n$  is a vector of  $n$  decision variables,  $f_i(\vec{x})$  are the objective functions and  $g_i(\vec{x})$  are inequality constraints.

### 2.3.2 Pareto Optimality

**Definition:**

**Pareto dominance** A vector  $u = (u_1, \dots, u_k)$  is said to dominate a vector  $v = (v_1, \dots, v_n)$ , written as  $u \preceq v$ , if and only if every component  $u_i$  is less than or equal to  $v_i$ , and at least there is one component in  $u$  which is strictly less than the corresponding component in  $v$ . This can be formulated as:  $u \preceq v \iff \forall i \in 1, \dots, k, u_i \leq v_i$  and  $\exists i \in 1, \dots, k : u_i < v_i$ .

**Pareto optimality** A solution  $x \in \mathcal{U}$  is said to be Pareto optimal if there is not any other solution  $x' \in \mathcal{U}$  whose objective vector  $u' = F(x')$  dominates  $u = F(x)$ . In other words, a solution whose objectives can not be improved simultaneously by any other solution is Pareto optimum.

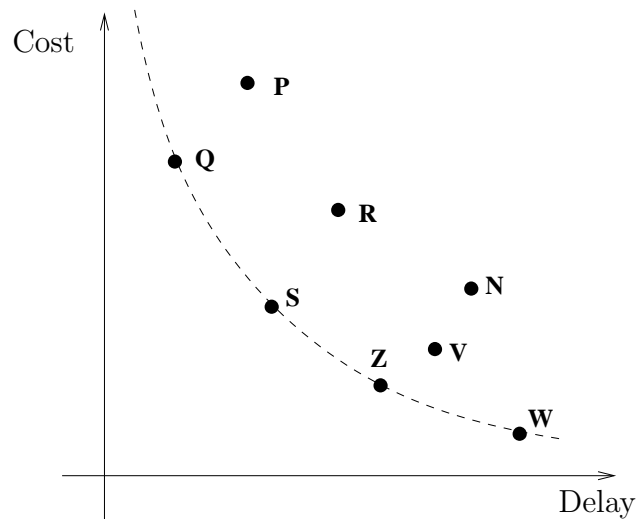


Figure 2.2: Example of a Pareto front

Figure 2.2 shows two axes namely cost and delay. Suppose we wish to minimise both cost and delay in a telecommunication network. Point  $Q$  has a small delay but a high cost and point  $W$  has a small cost with high delay. It is therefore not right to say that either point is better than the other. In addition, points  $S$  and  $Z$  have the same characteristics as points  $Q$ ,  $W$  and none of them can be classified, in any way, as better than the other points. In fact there are many such points which are known to be *non-dominated* or *non-inferior* points. The curve through these points determines the *Pareto front*. There are

other points in the space like the points  $P$ ,  $R$  and  $V$ . Taking a look at point  $V$  in particular and comparing it to  $S$ , we can see that  $V$  has a small value of cost and high value of delay while point  $S$  is the other way around. This raises the question as to whether  $V$  is also in the Pareto set. Since  $Z$  has values of both cost and delay which are smaller than  $V$ , the answer to our question is that  $V$  is not in the Pareto set. Points like  $V$  are termed *dominated* or *inferior* points.

One thing that is worth mentioning is the difference between the *Pareto-Optimal set* and the *non-dominated set*. When solving MOOP, it is usually difficult to find all non-dominated solutions due to the large size of the search space. In most cases a sample of the search space is considered. Points that are not dominated in the sample are known to be non-dominated solutions. Such points then form a non-dominated set. On the other hand, a Pareto-optimal set is a non-dominated set when the whole search space is considered [19].

When dealing with multi-objective optimisation we need to keep the following two aspects in mind:

- The search must be oriented towards the Pareto-optimal region. This can be accomplished by applying proper selection mechanisms and fitness assignments.
- Maintain a diverse population so as to avoid premature convergence with the aim of getting a well distributed Pareto-optimal front [20].

## 2.4 MOP Solution Approaches

There are several methods aimed at solving MOP problems. The following are some of the most well known evolutionary genetic algorithms: Vector Evaluated GA (VEGA) [21], Multi-objective GA (MOGA) [22], Niche Pareto GA (NPGA) [23], the Pareto Envelope-based Selection Algorithm for multi-objective optimisation (PESA) [24], the Pareto Archived Evolutionary Strategy (PAES) [25], the Non-dominated Sorting GA (NSGA) [26], the Strength Pareto Evolutionary Algorithm (SPEA) [27], the Dynamic Multi-Objective Evolutionary Algorithm (DMOEA) [28], the Rank-density-based Multi-objective GA (RDGA)

[29] and the Micro-Genetic Algorithm for Multi-objective Optimisation (Micro-GA) [30]. We can classify the above algorithms into three different groups [31].

- *Aggregating approaches*: this approach reduces the MOP problem to a single objective optimisation problem by combining (using summation) all objectives into one objective using weight coefficients.
- *Non-Pareto approaches*: the selection of individuals for the next generation is based on the special handling of the objectives and the way in which the population is manipulated.
- *Pareto based approaches*: the selection of the new population is based on the dominance of each individual in the population.

## Aggregating Approaches:

### 2.4.1 Weighted Sum Approach

These approaches eliminate the MOP problem and combine the objective functions into a single objective function in order to form a single objective optimisation problem. This involves adding the objectives together using weighting coefficients for each objective. The result is a scalar optimisation problem which can be formulated as follows

$$\min \sum_{i=1}^k w_i f_i(\vec{x}) \quad \text{where} \quad \sum_{i=1}^k w_i = 1 \quad (2.1)$$

and the weights  $w_i$  represent the relative importance of each objective.

This method of solution is straightforward because it is easy to implement. However, the solution to the problem largely depends on the weights  $w_i$ . Changing the weights by a small value can lead to a large change in the obtained solution. Usually the objective with a high preference is assigned a high coefficient, that is a high value of  $w$ , although such a weight does not reflect in a proportional fashion. In general these weights reflect how to locate the Pareto points when they are altered because every time we change the weight vector we also guide the search in a different direction [32].



The advantage of using this method is that it is easy to implement and it is guaranteed that the entire set of Pareto points can be found provided that the objective function is convex [33]. However in case of non-convex functions this claim is no longer true. The difficulty that arises from this method is to choose appropriate weights and this may prove to be difficult even if one is familiar with the optimisation problem.

Many variants of this method that can be found in the literature. Murata, Ishibuchi and Tanaka [34, 35] proposed an approach termed as the *random weight approach*. Instead of keeping the weights fixed throughout the run of the algorithm, weights are changed at each stage during the run of the algorithm. This helps the algorithm to conduct a search in multiple directions [3].

Gen and Cheng [3, 36] proposed another approach known as the adaptive approach. Weights are also changed during the run of the algorithm but this approach utilises some useful information from the population of the current generation in order to re-adjust weights so as to obtain a search pressure towards a positive ideal point. To find a point that maximises or minimises all objectives simultaneously is usually difficult. An *ideal point* or an *ideal positive point* is a point that optimises all objectives at the same time but in multi-objective optimisation such a point is usually not attainable.

## Non-Pareto Based Approaches

### 2.4.2 Vector Evaluated Genetic Algorithm: VEGA

The first algorithm implemented to solve MOP was the Vector Evaluated Genetic Algorithm (VEGA) by David Schaffer [21]. He slightly modified the simple genetic algorithm by changing the selection operator. Given  $k$  objective functions to be optimised, the population is first divided into  $k$  sub-populations. The selection operator is applied so that it selects individuals from each sub-population. The selected individuals are proportional to each objective function. Next, all  $k$  sub-populations are merged and shuffled to form a population of a fixed size  $N$  on which the normal crossover and mutation operations are carried out.

Although VEGA was proven to be successful in a number of problems, see for example [3],

Schaffer noticed two problems with his algorithm: first, the algorithm produced solutions that were non-dominated in local sense because of the limitation of their non-dominance in the current population [19]. Second, he noticed another problem known as speciation. Speciation causes bias towards certain regions or individuals in the population. It leads the algorithm to converge towards a particular optimum region after a number of generations. To solve the two problems above, Schaffer proposed two heuristics namely the non-dominated selection heuristic and the mate selection heuristic [19]. The first heuristic penalises dominated individuals in order to avoid their possible massive participation in the next generation. The second heuristic introduces another type of selection different from the traditional genetic algorithm where an individual is mated with another one which has the maximum Euclidean distance in performance space from it [33].

## Pareto Based Approaches

Pareto-optimality can be used at least in two ways to drive a rank-based selection mechanism within the genetic algorithm. One way is to rank individuals according to the front they belong to and the other one ranks individuals based on the number of individuals by which it is dominated.

### 2.4.3 Multi-Objective Genetic Algorithm: MOGA

MOGA is one of the multi-objective optimisation algorithm that uses the concept of Pareto optimality to rank individuals. Fonseca and Fleming [37] proposed a mechanism of ranking in this fashion. First, all non-dominated individuals are identified and assigned rank 1. For each of the rest of individuals, the total number of individuals that strictly dominates it is first found, then the rank of this individual is equal to the number of individuals that dominates it plus one. For example, an individual  $x_i$  dominated by 8 other individuals in the current population will have rank 9.

The fitness assignment is carried out in the following way [37].

1. Sort the entire population according to the rank of each individual.

2. Assign a fitness to each individual according to some function, usually linear but not necessarily by interpolating from the individual with rank 1 to the one with rank  $n \leq N$  as proposed by Goldberg [18].
3. Average the fitness of the individuals with the same rank so that all of them will be sampled at the same rate.

As noted by Goldberg and Deb [38], this type of blocked fitness assignment is most likely to produce a large selection pressure which might in turn cause premature convergence. To avoid this problem, MOGA uses a niche-formation method to distribute the population over the Pareto-optimal region. Instead of performing sharing on parameter values, Fleming and Fonseca [37] used sharing on objective function values. As a consequence of this sharing scheme, two different vectors with the same objective function values cannot exist at the same time. As a matter of fact this is undesirable for the decision maker (DM) [19, 32]. A good aspect of MOGA is that it is relatively easy to implement and quite efficient.

#### 2.4.4 Non-dominated Sorting Genetic Algorithm: NSGA

Srinivas and Deb [26] proposed another algorithm based on the concept of Pareto dominance. Individuals are classified into different layers before crossover and mutation operation are applied.

At first the population is ranked based on their dominance. All non-dominated individuals are identified and combined to form the first front. These individuals are then assigned a large fitness value which is proportional to the population size. The reason for assigning the same fitness value to these individuals is to give them an equal chance to reproduce. Later these individuals are shared with their dummy fitness value with the aim of keeping a diverse population. The fitness is dummy because it is not the fitness of the individual evaluated on the objective functions. For more information on *sharing*, the reader is referred to [39]. These individuals are then temporarily ignored and another layer of non-dominated individuals is considered and assigned a dummy fitness value smaller than that of the previous front.

NSGA only differs from the simple GA in the way selection is performed. In this case stochastic remainder proportionate selection is used. In addition, the classification of indi-

viduals into several fronts and the use of the sharing operation makes NSGA different from a simple GA. Individuals in the first front always get the maximum fitness value hence they always appear more often than the rest of the population in the next generation. This helps to search for non-dominated regions and results in quick convergence of the population towards non-dominated regions. Sharing is also used to help distribute the search towards the non-dominated regions. The efficiency of NSGA lies in the way multiple objectives are reduced to a dummy fitness function using a non-dominated sorting procedure (hence the name NSGA) [26]. Another good aspect of NSGA is its capability of handling any number of objectives as well as minimisation and maximisation. NSGA also handles both real and binary coded variables.

The first version of this algorithm was much criticised for its high computational complexity of non-dominated sorting, lack of elitism and the need for specifying the sharing parameter. However Deb *et al* [40] developed a new version of this algorithm which seems to have alleviated the three problems cited above. Figure 2.3 taken from [26] shows a flow chart of the algorithm.

## 2.5 Population Diversity

We mentioned in subsection 2.3.2 that the crucial point in EMOO is to keep a diverse population in order to obtain solutions that are uniformly distributed over the Pareto front. Failure to do so may lead to a phenomenon known as genetic drift. Several approaches have been proposed to tackle this problem. The following section describes some of those approaches that have been implemented in a number of evolutionary genetic algorithms and have proven to work well.

### 2.5.1 Fitness Sharing

The idea of fitness sharing was first proposed by Goldberg and Richardson [41] and later by Fonseca and Fleming [22] when they investigated the existence of multiple local optima for multimodal functions.

Fitness sharing drives the search in unexplored regions of a Pareto front by reducing the

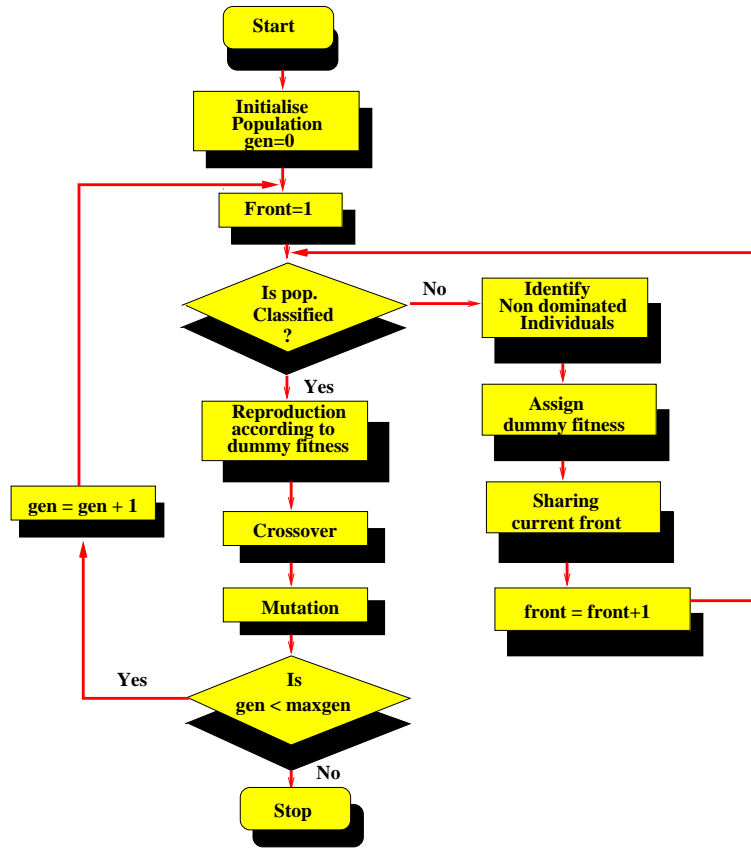


Figure 2.3: NSGA flowchart

payoff in densely populated areas. Individuals located in such areas are penalised using some penalty method [42]. The shared fitness  $f_{shared}$  of an individual  $i$  having fitness  $f_i$  is given by:

$$f_{shared} = f_i/n_i$$

where  $n_i$  is the niche count. A niche count is an estimate of the number of individuals with whom the fitness  $f_i$  is shared. The niche count is

$$n_i = \sum_{j=1}^N sh(d_{ij})$$

where  $N$  is the population size and  $d_{ij}$  is the distance between the individuals  $i$  and  $j$ .

The distance  $d_{ij}$  can either be genotypic or phenotypic based. For chromosomes encoded as binary strings, the Hamming distance is usually used. The Euclidean distance is used on the other hand when sharing is linked to real parameters of the search space. However, it was reported that sharing based on phenotypic distance usually performs better than its genotypic counterpart [43].

The most widely known sharing function is

$$sh(d_{ij}) = \begin{cases} 1 - (d_{ij}/\sigma_s)^\alpha & \text{if } d_{ij} < \sigma_s \\ 0 & \text{otherwise} \end{cases} \quad (2.2)$$

where  $\sigma_s$  denotes the threshold of dissimilarity (also known as the distance cutoff or the niche radius) and  $0 < \alpha \leq 1$  is a parameter that changes the shape of the function. If  $\alpha = 1$  then  $sh(d_{ij})$  becomes linear.

The sharing function describes the level of similarity between two individuals. It takes a value of one if the two individuals are equal and zero if the distance between them is bigger than a pre-set threshold of dissimilarity. It returns a value between zero and one if the two individuals have an acceptable level of dissimilarity.

The advantage of using sharing is that it encourages search in unexplored regions and it also favours formation of stable sub-populations [43]. However its disadvantage is that  $\sigma_s$  must be set beforehand and this requires us to know how far apart the optima are. It is also a computationally expensive exercise. Another advantage is that some methods have been developed to increase the effectiveness of sharing [22].

## Fitness Scaling

An alternative to improving the efficiency of fitness sharing is to use fitness scaling. A scaled function increases differentiation between optima and reduces deception [43]. A common method used to scale the fitness function is to use a power scaling. The scaled fitness of an individual  $i$  is given by:

$$f_i = f_i^\beta / n_i$$

where  $\beta$  is a parameter to which power  $f_i$  is raised and  $n_i$  is the niche count. The difficulty with a scaling function lies in the choice of  $\beta$ . If  $\beta$  is large we may end up with highly fit individuals in the population which can lead to premature convergence. On the other hand if the value of  $\beta$  is too low, differentiation between optima can be insufficient and this can prevent the sharing method from appropriately detecting optima. However in order to prevent premature convergence and increase the efficiency of the sharing method, annealing the scaling power during the search is recommended [29].

## 2.5.2 Crowding Distance

Crowding distance methods insert new elements in the population by replacing similar elements. This approach aims at obtaining a uniform spread of solutions along the Pareto front. NSGA, the EMOO that was extensively used in this thesis, uses crowding distance. In NSGA, the crowding distance is used to break the tie between two individuals during the tournament selection process. Individuals  $i$  and  $j$  are first selected randomly, and if they are both in the same front, the one with higher crowding distance is selected. If they are not in the same front, the one with lower rank is the winner.

There are variants of crowding distance methods such as standard crowding, deterministic crowding and restricted tournament selection [42].

## 2.5.3 Cell Based Density

Like the sharing method, cell-based density is also used to keep the population diverse. This approach divides the objective space into  $n$ -dimensional cells where  $n$  is the number of objectives. The density of a cell is defined as the number of the potential solutions in that cell. Figure 2.4 shows an example where the density of the top left cell is 3 because it contains 3 solutions while the density of the top right cell is zero because it does not contain any solution. The density of a particular solution is equal to the density of the cell in which it is found. Some of the evolutionary algorithms that use this method include

SPEAII [44] and DMOEA [28].

The main advantage of using the cell-based density approach is that a global density map of the objective function space is obtained as a result of the density calculation [42]. Another advantage of this approach is that it is computationally efficient when compared to niche or neighbourhood-based density techniques.

There exists yet another method for diversifying the population known as *clearing*. It is similar to fitness sharing but it is based on the concept of limited resources of the environment [45].

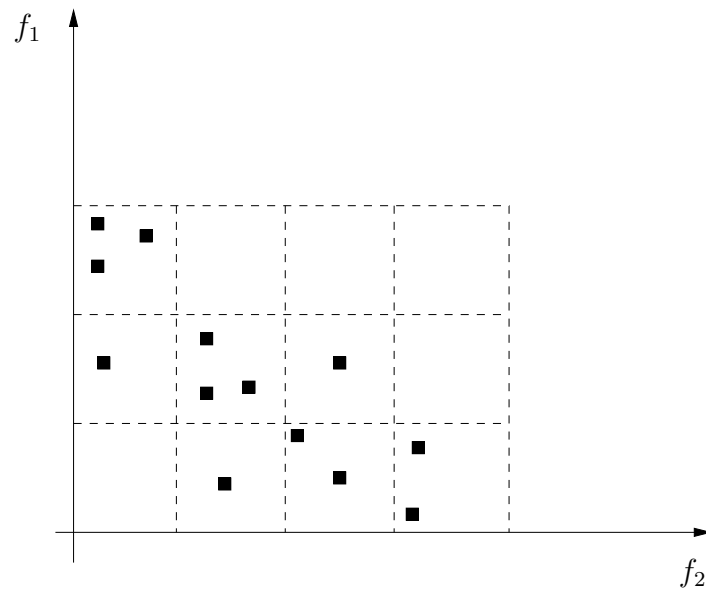


Figure 2.4: Example of cell density for two functions  $f_1$  and  $f_2$ .

#### 2.5.4 Elitism

Elitism is a mechanism that ensures that highly fit individuals are passed onto the next generation without being altered by genetic operators. It guarantees that the minimum fitness of the population can never decrease from generation  $t$  to generation  $t + 1$ .

In case of single objective GA, an elite is the best solution in a run, and therefore survives to the next generation. However in multi-objective GA, an elite is a non-dominated solution. To implement elitism in single objective GA is straightforward. However, due to the large



set of elites in multi-objective GA, this becomes difficult. Multi-objective GA use two strategies to implement elitism [42].

- i. Maintain elitism solutions in the population.
- ii. Storing elitist solutions in an external list.

NSGAI [40] uses the first approach of elitism in order to drive the search towards the Pareto front regions. The advantage of keeping elites in the populations is its ease of implementation. The second approach keeps elites in an external list and then reintroduces them in the population afterwards. Such elites are kept in a separate list which is updated each time a new non-dominated solution is found.

Using elitism usually results in a more rapid convergence of the population toward an optimum solution. However in some applications elitism may improve the chances of locating an optimal solution while in others it does not, but leads to a premature convergence.

## Chapter 3

# Routing and Wavelength Assignment

### 3.1 Introduction

The scalability of the Internet as seen in the last decade has led Internet Service Providers (ISPs) to deploy new technologies that improve the utilisation of the network resources. Most of these networks are Internet Protocol (IP) networks that run on top of the reliable Transmission Control Protocol (TCP). Some of the protocols that run on top of TCP (on the application layer) that are utilised by many users include FTP, TELNET, HTTP. The need to manage and efficiently utilise the available network resources is inevitable. Data packets are routed along the network paths according to a certain protocol. The most commonly used protocol in intra-domain routing is the Open Shortest Path First (OSPF) protocol. In OSPF, traffic is routed along the shortest path.

To a certain extent, we may say that IP networks manage themselves [46]. Every host that implements TCP adjusts its sending rate to the bandwidth available on the path to the destination. Routers in their turn periodically compute new paths in the network depending on changes in the network topology. This has made the Internet an extremely robust communication network despite problems that it is faced with like rapid growth and occasional failures. Despite the robustness of an IP network, the mechanisms that are in place do not ensure the efficient operation of the entire network. Different types of traffic traverse the same link. Each type of traffic may have its own routing specifications. For example some traffic may tolerate latency like http traffic while some other traffic

(like Voice over IP) may not. Some of the links may be overloaded while other links are lightly loaded or underutilised. For example in the sample network shown in figure 3.1, every packet from  $S_1$  to  $D_1$  will most likely take the path  $1 - 7 - 8 - 5$  while the route  $1 - 2 - 3 - 4 - 5$  may be underutilised. This shows the need to split traffic not only among shortest paths but also among other paths which may be underutilised.

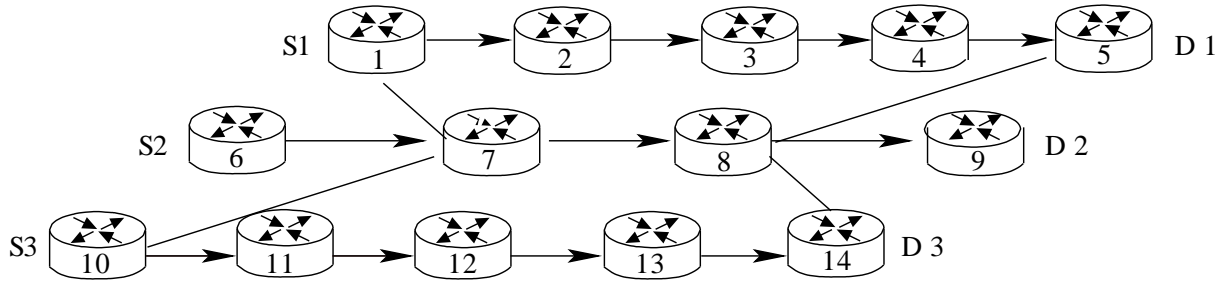


Figure 3.1: An example of a simple network

## 3.2 Routing and Wavelength Assignment

### 3.2.1 Overview

Optical networks that use Wavelength Division Multiplexing (WDM) offer solutions to the high bandwidth demands that many telecommunication companies experience today. Optic fiber can carry huge amounts of data where several communication channels are multiplexed together to transport information from one node to another. The Internet backbone is mainly made of optic fiber where wavelength routing is used to connect different access nodes. Data is mainly routed in the optical domain between these access nodes, that is to say no conversion from electronic to optical or optical to electronic takes place. Networks that only operate in the optical domain are called *all-optical* networks.

To be able to connect two nodes in the optical domain a connection similar to circuit switched network needs to be established. This is accomplished by first determining the path between two nodes and then allocating a free wavelength on all links between the source and the destination nodes. The similarity with circuit-switched network lies in the fact that all bandwidth on this path will be reserved for this connection until the connection is terminated. Once the connection is terminated, the associated wavelength

will be available on all links along the path. Such an all-optical path is referred to as a *lightpath* and it may traverse more than one fiber link without any intermediate electronic processing [47]. When a lightpath operates on the same wavelength across all fiber links that it traverses, the lightpath is said to satisfy the *wavelength continuity constraint*. The wavelength continuity constraint dictates that no two identical wavelengths (i.e. of the same color) should be routed across the same path. The problem that arises in wavelength routed networks is that, given a certain number of lightpaths, we need to route each lightpath and assign a single wavelength to it. This problem is known as the *Routing and Wavelength Assignment* (RWA) problem.

### 3.2.2 Formulation of the RWA Problem

The routing and wavelength assignment problem can be stated as follows. Given a set of lightpaths that need to be established on the network, and a given constraint on the number of wavelengths, determine the routes over which these lightpaths should be set up and also determine the wavelengths which should be assigned to these lightpaths so that the maximum number of lightpaths may be established [48].

We now define and formalise the problem in mathematical terms. We assume that there is at most one lightpath between every source destination pair. We also assume that no conversion from electronic to optical or optical to electronic takes place. Let  $\lambda_{sd}$  denote the traffic (in terms of a lightpath) from any source  $s$  to any destination  $d$ .  $\lambda_{sd} = 1$  if there is a lightpath from  $s$  to  $d$ ; otherwise  $\lambda_{sd} = 0$ . Let  $F_{ij}^{sd}$  denote the traffic (in terms of number of lightpaths) that is flowing from source  $s$  to destination  $d$  on link  $ij$ .  $F_{ij}^{sd}$  is one if there is a lightpath between  $s$  and  $d$  and  $ij$  is a link on that path. The problem can now be formulated as a linear programming problem

$$\text{Minimise: } F_{max} \tag{3.1}$$

such that:

$$F_{max} \geq \sum_{s,d} F_{ij}^{sd} \quad \forall ij \tag{3.2}$$

$$\sum_i F_{ij}^{sd} - \sum_k F_{jk}^{sd} = \begin{cases} -\lambda_{sd} & \text{if } s = j \\ \lambda_{sd} & \text{if } d = j \\ 0 & \text{otherwise} \end{cases} \quad (3.3)$$

$$\lambda_{sd} = 0, 1 \quad (3.4)$$

$$F_{ij}^{sd} = 0, 1 \quad (3.5)$$

The RWA problem is equivalent to the graph coloring problem, hence it is an NP-complete problem [49] but several heuristics have been developed to solve the problem [50, 51]. We do not attempt to solve the problem as described above, we instead solve the static RWA [52] that is, first we find the routes using genetic algorithms and then use the graph coloring algorithm [53] to solve the wavelength assignment problem. Research conducted on RWA is mainly based on assumptions concerning traffic patterns, availability of wavelengths converters and desired objectives. The traffic assumptions generally fall into two different categories, dynamic and static models.

In dynamic models, requests for lightpaths between source destination pairs are not known beforehand, they instead arrive at random and also terminate at random times. In this case the objective would be for example to minimise the call blocking probability [47]. In static models demands are fixed and assumed to be known beforehand. That is to say lightpaths that are to be set up between every source destination pair are known *a priori*. The objective in this case would be to accommodate these demands while minimising the number of wavelengths used on all links.

We use fictitious African and USA networks as our test networks and solve the static RWA. In the next section we describe the models used in order to find the routes before we present the number of wavelengths that each routing scheme was able to find.

## 3.3 Routing Problem

### 3.3.1 Problem Statement

We formulate the routing of traffic in emerging IP networks as both single and multi-objective optimisation problems which we solve using *classical* and *hybrid* genetic optimisations. We adopt the occurrence of links in the computed paths as a cost function used to achieve load balancing by reducing the interference among competing flows. We adapt the Non-dominated Sorting Genetic Algorithm (NSGA) [26] to find a *Pareto front* for the multi-objective problem. This *Pareto front* expresses a set of solutions which represent plausible set of paths that minimise both the interference among competing flows and the queuing delay in the network. Using both multiplicative and additive composition rules, we compare the quality of the paths achieved by the different evolutionary optimisation strategies. In the rest of this chapter the following words are used interchangeably: source and origin, vertex and node, link and edge.

### 3.3.2 Single Objective Optimisation

We consider a single objective optimisation model where a physical network is viewed as a graph  $\mathcal{G} = (\mathcal{N}, \mathcal{L})$  where  $\mathcal{N}$  is the set of nodes and  $\mathcal{L}$  is the set of links.  $S$  is a set of ordered pairs consisting of origin  $o$  and destination  $d$ , that is

$$S = \{(o, d) \mid o \text{ is the origin node and } d \text{ is the destination node.}\}$$

In most routing schemes traffic is routed along shortest paths. As a result these paths become overloaded thus causing a bottleneck in the network. It is therefore reasonable to select a set of paths which may not be the shortest but are lightly loaded. The selection of lightly loaded paths is achieved by assigning a high cost to all chromosomes that represent paths that are heavily loaded.

Our aim is to find a set of paths by minimising a penalty function expressing the frequency of occurrence of each link in the set of paths found between all source destination pairs. A similar goal consists of finding a set of link weights to be used as routing metrics (cost metrics) when finding the least cost paths.

## Problem Formulation

### Problem 1

Equations (3.6) and (3.7) define the cost functions of a network. We will refer to equations (3.6) and (3.7) as the power and the additive cost functions respectively.

$$\frac{N_l^\alpha}{(C_l - f_l)^{(1-\alpha)}} \quad (3.6)$$

where  $N_l$  is the number of flows traversing link  $l$ ,  $C_l$  is the capacity of link  $l$ ,  $f_l$  is the bandwidth used on link  $l$  and  $\alpha$  is an arbitrary value between zero and one ( $0 < \alpha < 1$ ). Equation (3.6) can be expressed in logarithmic form as

$$\alpha \ln(N_l) + (\alpha - 1) \ln(C_l - f_l) + (1 - \alpha) \times \ln(C^*) \quad (3.7)$$

where  $C^*$  is the maximum link capacity i.e  $C^* = \max_{l \in \mathcal{L}} \{C_l\}$  and the term  $(1 - \alpha) \times \ln(C^*)$  is added to avoid the expression being negative. For example if  $C_l = 100$ ,  $f_l = 30$ ,  $N_l = 3$  and  $\alpha = 0.5$  the first two terms yield  $-1.5749$ .

The main objective is to find a set of link weights that will result into finding optimal paths minimising the functions in equation (3.6) and (3.7).

### Problem 2

In a genetic optimisation setting, the optimum paths are obtained from a chromosome with the least cost. For example if a chromosome consists of paths whose links do not appear in any other paths, this will be the optimum chromosome. Let a chromosome be presented as  $H = P_{ij}, \dots, P_{od}$  where  $P_{od}$  represent a path chosen randomly from the set of all paths found between the source  $o$  and the destination  $d$ . We define  $E(P_{od})$  as the set of links in the path  $P_{od}$ . Now, let  $K^* = \bigcup_{P_{od} \in H} E(P_{od})$ . A cost  $c_l$  is assigned to each link  $l \in K^*$  where  $c_l$  is the number of paths in  $H$  having  $l$  as an edge.

The problem can now be formulated as an optimisation problem as follows:

$$\text{Minimise} \quad G(H) = \sum_{\forall l \in K^*} N^{c_l} \quad (3.8)$$

where  $N$  is the number of nodes in the network. As expressed by equation (3.8), the penalty function to be minimised is expressed as a power function which is computationally expensive. However this function (referred to as link occurrence or simply occurrence) has been selected to differentiate low and high cost paths by setting a large gap between them.

The mathematical formulation above is derived from [54] and illustrated by Figure 3.2. The objective function to be minimised is presented as follows. Each chromosome is associated with a cost which measures its fitness. A cost refers to a value that the function takes when evaluated in each chromosome and the fitness of a chromosome is a measure of this value. The smaller the value the higher the fitness will be and *vice versa*.

### **The Genetic Algorithm**

In this section we present the genetic algorithm that we have used in order to solve the problems (3.6), (3.7) and (3.8). We also present its features as well as issues pertaining to its implementation.

#### **Initialization Of The Population**

The daunting task in any genetic algorithm is the encoding of the chromosome to represent the relevant problem. In this case each chromosome is represented by genes (a set of one or more bits) which point to paths in the look-up table. Figure (3.2) displays an example of a typical chromosome. Genes of a chromosome are generated randomly with each decimal value of a gene representing the index of a path for a particular source destination pair. In each source destination pair a path is selected randomly and its index is converted to binary and so represents an actual path in the look-up table. A candidate chromosome therefore contains genes, each one of them pointing to a path in all source destination pairs. Therefore, from a single chromosome one gets plausible paths for all source destination pairs in the network.

#### **Crossover and Reproduction**

Crossover is performed according to a certain probability. Unlike other typical crossovers where two chromosomes exchange one or more bits at random, in this case the exchange of genes is also random but in addition the exchange is done in a way that the identity of



a path is not changed. This is to avoid a gene pointing to a non-existent path index. A roulette wheel selection is then applied to reproduce chromosomes for the next generation. This selection method is fitness proportional, that is to say fitter individuals (with lower cost) have a great chance of being selected and passed onto the next generation while weaker ones (with higher cost) are less likely to be selected but left out and discarded.

**Mutation**

The mutation operation is carried with a small probability of 0.01; this is in hope that it will strengthen weak individuals. This operation is also aimed at introducing diversity in the population by bringing in features that are not presently in the parent chromosomes. Offspring obtained from crossover are modified by randomly changing some parts of their chromosome bearing in mind that the gene should point to a candidate path in that respective source destination pair.

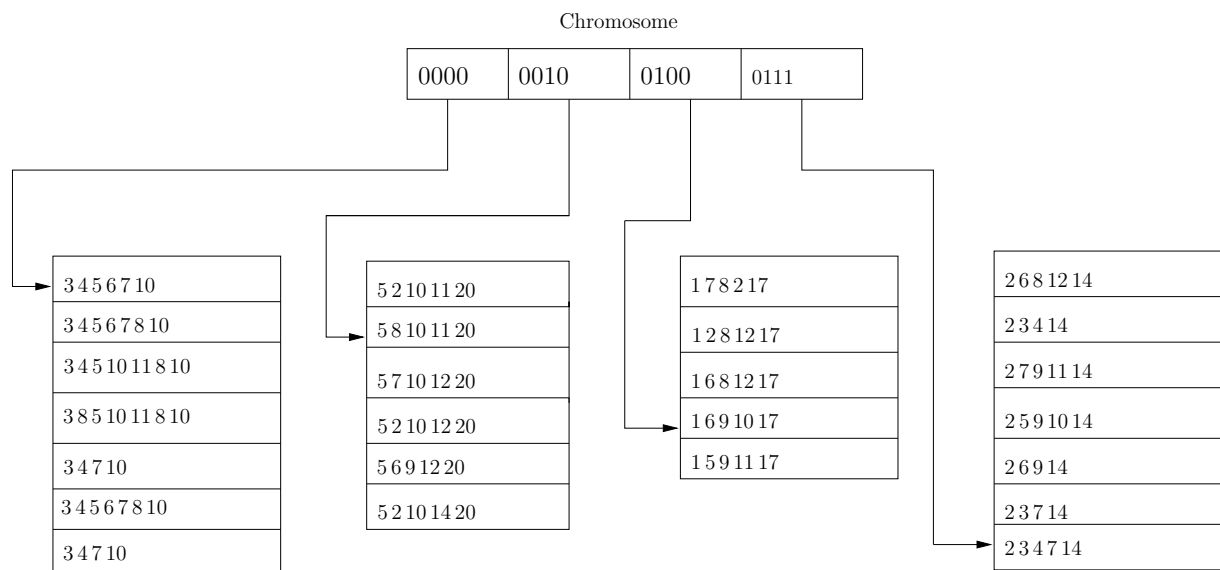


Figure 3.2: A typical chromosome: each gene encodes a path corresponding to one source destination pair

In figure 3.2, the first block from the left shows paths from source 3 to destination 10, the second one from 5 to 20, the third one from 1 to 17 and the last block on the right shows paths from source 2 to destination 14. In each block a single path is randomly selected, thus the chromosome in the figure represents possible paths from each source destination pair in the network.

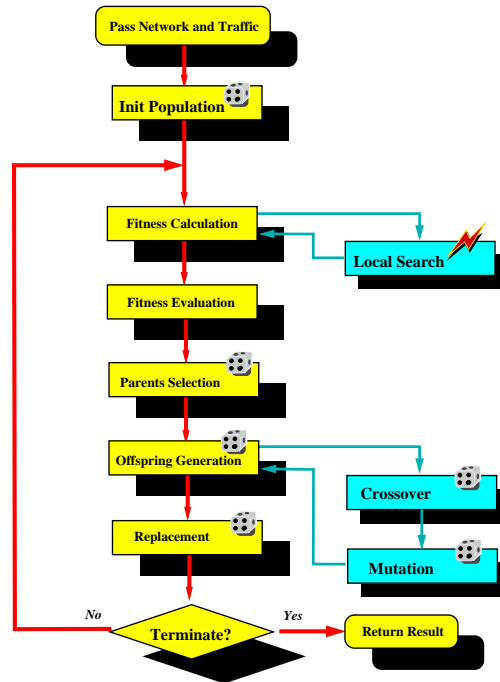


Figure 3.3: MA algorithm

### The Memetic Algorithm

*Memetic Algorithms (MAs)* belong to a class of evolutionary algorithms referred to in this thesis as **HybridGA**. They use a hybrid optimisation model where local search is used to complement the global search implemented by classical **GAs** to improve the genetic fitness of the individuals through hill-climbing. Local search is used to speed up the convergence process towards an optimal solution while at the same time avoiding premature convergence. We consider a memetic algorithm illustrated in Figure 3.3(a) where the fitness of each chromosome is evaluated using either of the functions in equations (3.6), (3.7) or (3.8) depending on whether a multiplicative, an additive composition rule or link occurrence is used.

### 3.3.3 Multi-Objective optimisation

In the multi-objective case, in addition to the objective in equation (3.8) we add the delay as another objective and try to minimise the two objectives simultaneously. The link delay

is taken as in the case of an  $M/M/1$  queueing system and it is given by  $f_l/(C_l - f_l)$  where  $f_l$  is the total used bandwidth used on link  $l$  and  $C_l$  the capacity of link  $l$ .

## Problem Formulation

### Problem 1

The multi-objective problem is :

$$\text{Minimise } \begin{cases} F(H) = \frac{1}{C_l - f_l} \\ G(H) = N_l \end{cases} \quad (3.9)$$

### Problem 2

The multi-objective problem is :

$$\text{Minimise } \begin{cases} F(H) = \frac{f_l}{C_l - f_l} \\ G(H) = \sum_{\forall l \in K^*} N^{c_l} \end{cases} \quad (3.10)$$

To solve the above multi-objective optimisation problems we have used a slight modification of the Non-dominated Sorting Genetic Algorithm (NSGA) proposed by Srinivas and Deb [26].

### 3.3.4 Multi-Objective Solution

We have used NSGA to solve our multi-objective problem. The working principles and features of NSGA are explained in section 2.4.4. To be able to use it to solve our problem we have added a few features that we mention below. Note that they are mostly based on the encoding of the chromosome.

- **Population encoding:** Although the original NSGA solves problem encoded in both real and binary variables, in our case we first encode all chromosomes as binary variables.

The length of a chromosome is not entered as an input since it is not known before hand because it depends on the size of the network in question.

- **The maximum and minimum values** of the functions to be optimised are also not known *a priori*, they are therefore not entered as inputs.
- **Objective functions** are not entered as inputs but these are calculated based on the occurrence of an edge in a set of paths to be found at run time.
- **More program features** have been embedded in NSGA. For example modules that read a network topology and traffic demands, those that find all paths leading from each source to destination and those that evaluate the fitness of each chromosome which is later evaluated as the objective function value.
- **Finally the best population** is represented in both real (representing Pareto points) and binary (representing chromosome with acceptable routing paths) encodings.

### 3.3.5 Simulation Results

#### Single Objective

We ran both the genetic and memetic algorithms with the following parameters:

Initial population : 100; Number of generations :100; Probability of crossover: 0.9; Probability of mutation :0.01.

To solve the problem we simulated these algorithms on two test networks. The African network (see Figure 3.12(b)) and the USA network (see Figure 3.12(a)). For the African network we considered 930 source destination pairs with 31 nodes and 128 links. In the USA network, we consider 506 source destination pairs and the network consists of 23 nodes with 76 links. The results found by our genetic algorithms were compared with those found by the HybridGA that uses the cost function described in equations (3.6), (3.7) and (3.8) and then route the traffic so that the above cost functions are minimised.

We looked at the similarity the paths found by each algorithm. We give as input to the program, two sets of paths from each algorithm and the number of nodes in the network.

Out of the total number of routes (routes used by HybridGA and GA) we wanted to know, from each source destination pair, the number of routes used by both algorithms, the number of routes exclusively used by each algorithm and the number of routes that share some links (at least one but not all) from each algorithm.

The statistics of paths found by both algorithms on the African and the USA network are displayed in tables 3.1, 3.2, 3.3 and 3.4. From these tables we see that the statistics of both networks change with parameter  $\alpha$  as expected. This means that as we vary the value of  $\alpha$  we get different routes and performance. The first two tables (3.1 and 3.2) show the percentage of routes used by each algorithm and the last two (3.3 and 3.4) show the route lengths.

In column 3 of table 3.1 we observe that 41.3% of routes in the USA network are used by both algorithms meaning that 41.3% routes of the two algorithms being compared are identical. What is worth mentioning is that the additive function finds a larger percentage of routes that are used by both algorithms when the value of  $\alpha$  is 0.5. For the African network (see table 3.2) the additive function still finds a larger percentage (31.94% ) but this time when the value of  $\alpha$  is 0.25. In terms of the overall number of routes that are used by both the algorithms when all three functions are compared, we notice that the additive function finds a larger percentage followed by the power function and lastly the occurrence function follows.

Column 4 of tables 3.1 and 3.2 reveals 0% of routes that share some links but not all. This means that from each source destination pair, routes found by each algorithms do not have any link in common. Lastly column 5 and 6 of the same tables show the percentage of routes that are exclusively used by each algorithm. For example in the last row, each algorithm uses 36.46% of the routes and 27.08% of routes are used by both algorithms.

Tables 3.3 and 3.4 show that GA mostly uses longer paths (at least one more node) than HybridGA. This tells us that GA will not only choose the shortest paths at times but mostly avoid congested links as a result of paths it chooses to route on. In all cases, the routes found by GA have a larger standard deviation, meaning that most of its paths have lengths that vary greatly from the average length.

Figures 3.4 and 3.5 show the usage of the most used routes by the two different algorithms

Table 3.1: Percentage of routes used by each algorithm for the USA network

Objective function	Parameter	Routes used by both algorithms	Routes that share some links	Routes used only by HybriGA	Routes used only by GA
Occurrence	-	24.31 %	0%	37.85%	37.85 %
Power	$\alpha = 0.25$	39.13 %	0%	30.43%	30.43 %
	$\alpha = 0.5$	38.74%	0 %	30.63%	30.63%
	$\alpha = 0.75$	25.49%	0%	37.25%	37.25%
Additive	$\alpha = 0.25$	40.12 %	0%	29.94%	29.94 %
	$\alpha = 0.5$	41.3%	0 %	29.35%	29.35%
	$\alpha = 0.75$	27.08%	0%	36.46%	36.46%

for both networks. We can see that the percentage is 100% meaning that all routes found by the algorithms are used. Figures 3.6 and 3.7 show the percentage of route length (number of hops) for both algorithms. Figures 3.8 and 3.9 show route multiplicity. We see from the graph that one route is mostly used from each source destination pair and in a few cases two routes are used. The percentage of source destination pairs using one route is more than 90% for both algorithms. Finally, figures 3.10 and 3.11 show the routes that are used by both the algorithms as explained above.

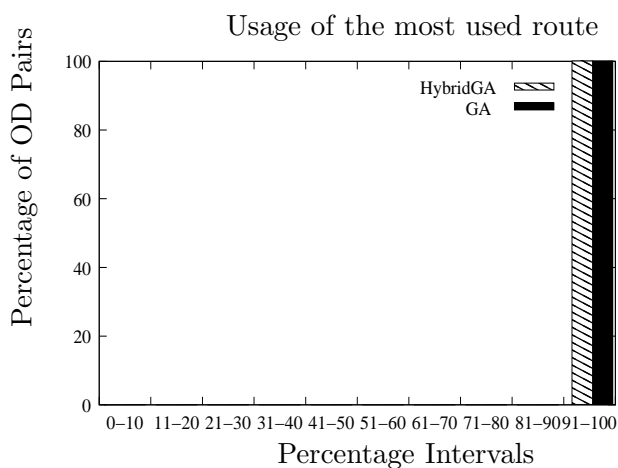


Figure 3.4: Route usage for the African network

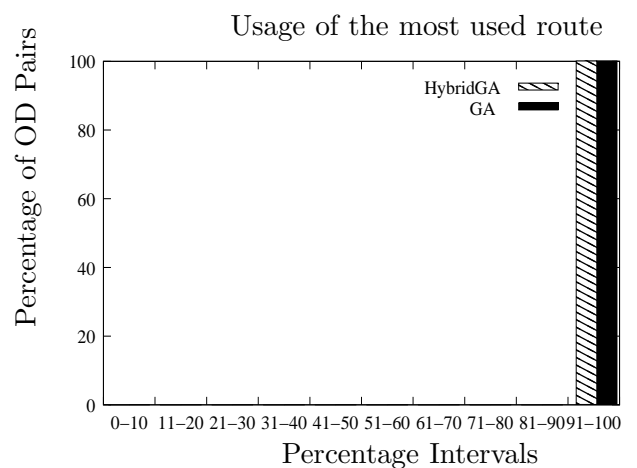


Figure 3.5: Route usage for the USA network

Table 3.2: Percentage of routes used by each algorithm for the African network

Objective function	Parameter	Routes used by both algorithms	Routes that share some links	Routes used only by HybridGA	Routes used only by GA
Occurrence	-	14.41 %	0%	42.8%	42.8 %
Power	$\alpha = 0.25$	26.88 %	0%	36.56%	36.56 %
	$\alpha = 0.5$	22.47%	0 %	38.76%	38.76%
	$\alpha = 0.75$	17.31%	0%	41.34%	41.34%
Additive	$\alpha = 0.25$	31.94 %	0%	34.03%	34.03 %
	$\alpha = 0.5$	30.85%	0 %	34.62%	34.62%
	$\alpha = 0.75$	14.3%	0%	42.85%	42.85%

Table 3.3: Statistics for the USA network, displaying the maximum and the average lengths and standard deviation

Functions & algorithm	parameter	Max. length	avg. length	stdDev
Hybrid(Occurrence)	-	8	3	1.66
Hybrid (Power)	$\alpha = 0.25$	7	3	1.38
	$\alpha = 0.50$	7	3	1.38
	$\alpha = 0.75$	10	3	1.84
Hybrid (Additive)	$\alpha = 0.25$	6	2	1.68
	$\alpha = 0.50$	6	2	1.68
	$\alpha = 0.75$	8	3	1.53
GA (Occurrence)	-	12	4	2.18
GA (Power)	$\alpha = 0.25$	12	4	2.03
	$\alpha = 0.50$	12	4	1.97
	$\alpha = 0.75$	12	4	1.94
GA (Additive)	$\alpha = 0.25$	12	4	2.07
	$\alpha = 0.50$	12	4	2.05
	$\alpha = 0.75$	8	3	1.98

Table 3.4: Statistics for the African network, displaying the maximum and the average lengths and standard deviation

Functions & algorithm	parameter	Max. length	avg. length	stdDev
Hybrid(Occurrence)	-	11	3	2.36
Hybrid (Power)	$\alpha = 0.25$	9	3	1.79
	$\alpha = 0.50$	3	3	1.79
	$\alpha = 0.75$	11	3	2.23
Hybrid (Additive)	$\alpha = 0.25$	8	3	1.77
	$\alpha = 0.50$	8	3	1.77
	$\alpha = 0.75$	12	4	2.26
GA (Occurrence)	-	14	5	2.43
GA (Power)	$\alpha = 0.25$	15	5	2.61
	$\alpha = 0.50$	16	5	2.49
	$\alpha = 0.75$	15	4	2.63
GA (Additive)	$\alpha = 0.25$	15	4	2.61
	$\alpha = 0.50$	14	4	2.58
	$\alpha = 0.75$	14	4	2.68

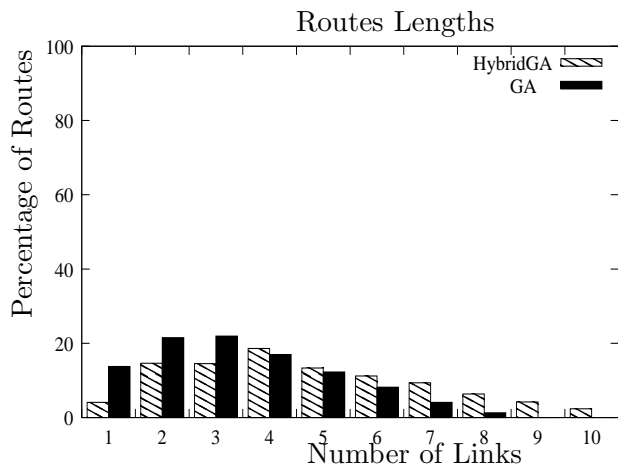


Figure 3.6: Route lengths for the African network

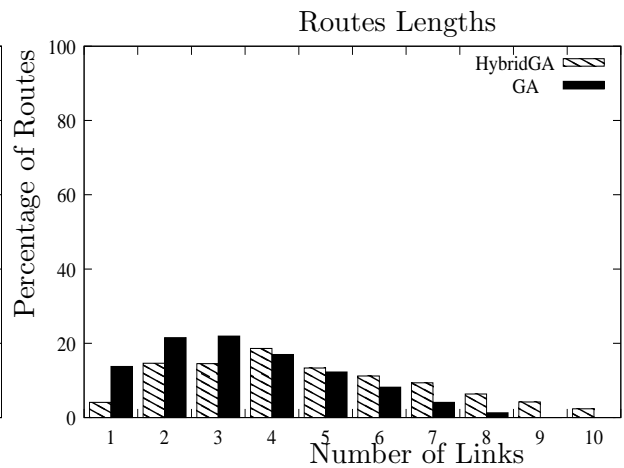


Figure 3.7: Route lengths for the USA network



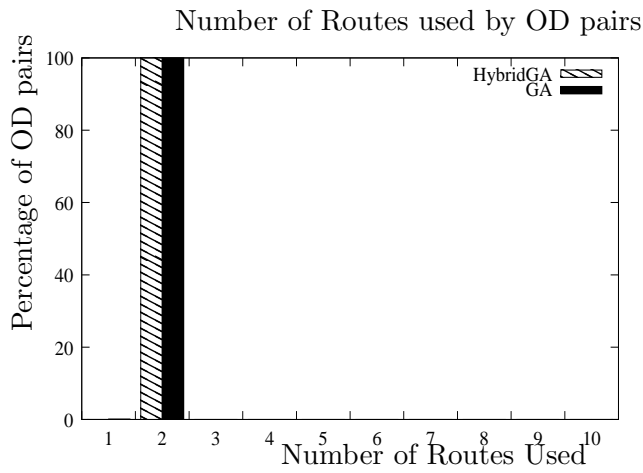


Figure 3.8: Route multiplicity for the African network

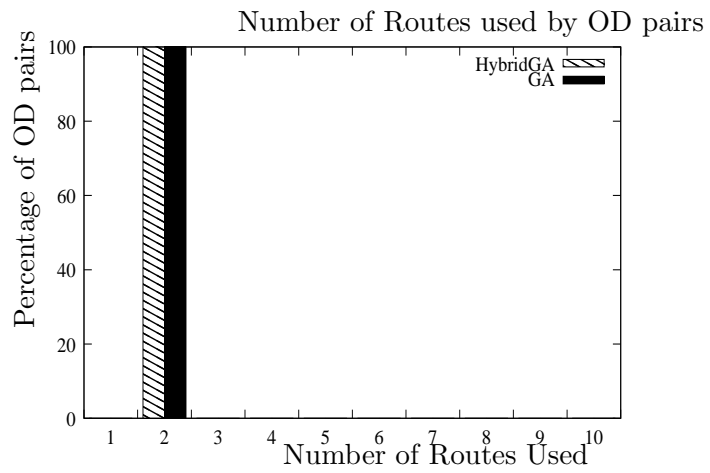


Figure 3.9: Route multiplicity for the USA network

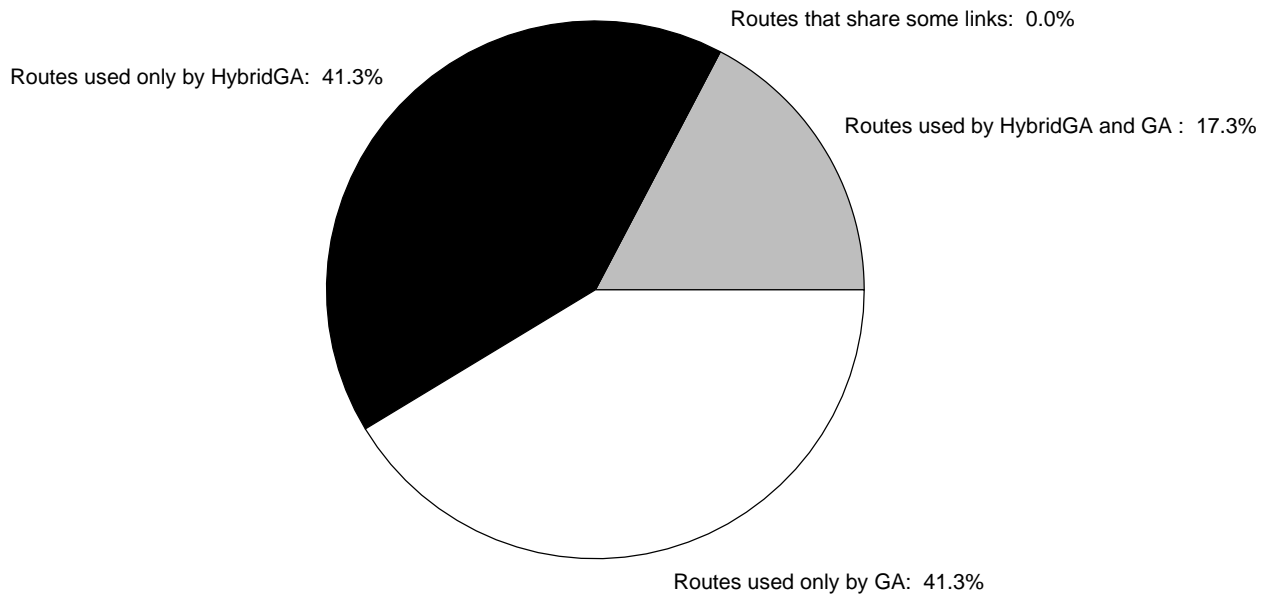


Figure 3.10: Route used by both algorithms for the African network (power function with  $\alpha = 0.75$ )

### 3.3.6 Multi-Objective Optimisation

Like other evolutionary multi-objective optimisation algorithms, there is no optimum solution. The term optimum means finding a set of point(s) which will give all objective functions acceptable values to the decision maker. In case of NSGA we have run the sim-

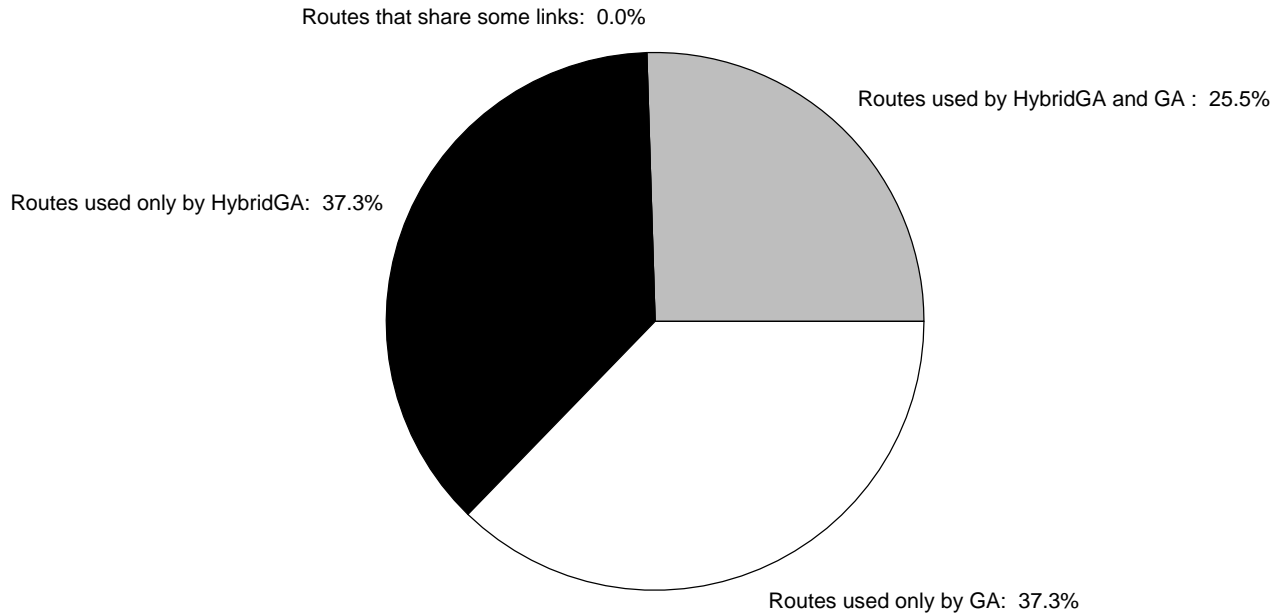


Figure 3.11: Route used by both algorithms for the USA network (power function with  $\alpha = 0.75$ )

ulation on the African network where we minimised the functions in equations (3.9) and (3.10). We ran the simulation with the following parameters:

Initial population : 200, number of generations :100, probability of crossover: 0.8, probability of mutation :0.01, distribution index for real variable SBX crossover: 5, distribution index for real variable polynomial mutation: 10.

The optimum paths can now be chosen by hand from the set of Pareto points which represent a chromosome fitness. Having a chromosome at hand we can obtain the paths from its genes that point to the paths of each source destination pair.

The simulation was run for several different parameters and it showed that more Pareto points are obtained as the number of generations start reaching 100. However, increasing the number of generations beyond 100 yields no further improvement.

Figures 3.13(a), 3.13(b), 3.14(a) and 3.14(b) show the Pareto fronts obtained from both the African and the USA networks respectively using two different problems. From the figures, we observe that the USA network has relatively small delay and cost when compared to

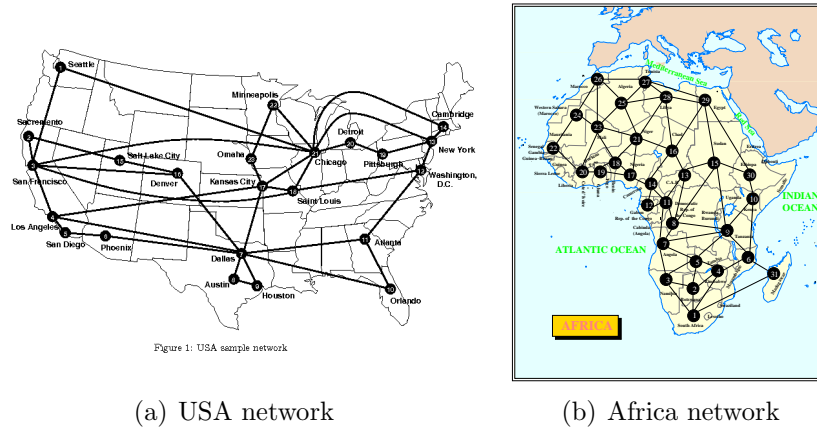


Figure 3.12: The USA (a) and African (b) networks

the African network. This is because the USA network has fewer links (a smaller network) compared to the African network.

The Pareto points obtained from both problems on both networks are displayed in tables 3.7, 3.8, 3.9 and 3.10.

What explains the larger objective function values for African network is its larger size compared to the USA network. This leads us to a reasonable deduction that when the network is large the cost and delay also increase. When we have a large network, we have more paths because of more source destination pairs. This will increase the probability of a link occurring in those paths thus increasing the cost. In case of delay, more occurrence ( a link occurs in many paths) implies that the link has a high number of flows because it is traversed more frequently. The number of flows is proportional to the bandwidth used on that link and finally increases the delay because the link will be heavily loaded.

The Pareto points provide us with sets of paths on which we can route different types of traffic depending on their delay sensitivity. For delay sensitive traffic like VoIP, a Pareto point with a delay value of 0.276 and cost 135852 (in case of the USA network problem 1) may be a good choice. On the other hand, email traffic for instance may take any other point with relatively high delay. In conclusion, achieving different routes with different characteristics may be of practical importance. Instead of all traffic taking the shortest paths and in turn overloading the network, we may use the Pareto points to give us suitable paths for different traffic scenarios.

Table 3.5: Table showing wavelengths for the African network

Obj. function	$\alpha$	$\lambda$	Algorithm
Occurrence	–	157	Hybrid
	–	78	GA
Power	$\alpha = 0.25$	85	Hybrid
	$\alpha = 0.25$	82	GA
	$\alpha = 0.50$	84	Hybrid
	$\alpha = 0.50$	81	GA
	$\alpha = 0.75$	162	Hybrid
	$\alpha = 0.75$	79	GA
Additive	$\alpha = 0.25$	85	Hybrid
	$\alpha = 0.25$	80	GA
	$\alpha = 0.50$	85	Hybrid
	$\alpha = 0.50$	79	GA
	$\alpha = 0.75$	126	Hybrid
	$\alpha = 0.75$	79	GA

Table 3.6: Table showing wavelengths for the USA network

Obj. function	$\alpha$	$\lambda$	Algorithm
Occurrence	–	74	Hybrid
	–	56	GA
Power	$\alpha = 0.25$	55	Hybrid
	$\alpha = 0.25$	53	GA
	$\alpha = 0.50$	55	Hybrid
	$\alpha = 0.50$	54	GA
	$\alpha = 0.75$	97	Hybrid
	$\alpha = 0.75$	61	GA
Additive	$\alpha = 0.25$	55	Hybrid
	$\alpha = 0.25$	54	GA
	$\alpha = 0.50$	55	Hybrid
	$\alpha = 0.50$	52	GA
	$\alpha = 0.75$	64	Hybrid
	$\alpha = 0.75$	53	GA

Table 3.7: Pareto points for problem 1 for the African network

$F(H)$	$G(H)$
553345	0.5489377
533021	1.238070
545926	0.9415441
536114	1.178173
535335	1.187310
534877	1.189187
544771	0.9435672
545964	0.9341992
537567	1.049727
546876	0.8583523
551092	0.5553029
529557	1.256382
551862	0.5534490
550125	0.5617487
541084	0.9512489
536371	1.173479
540738	1.001054
538564	1.002640

Table 3.8: Pareto points for problem 2 for the African network

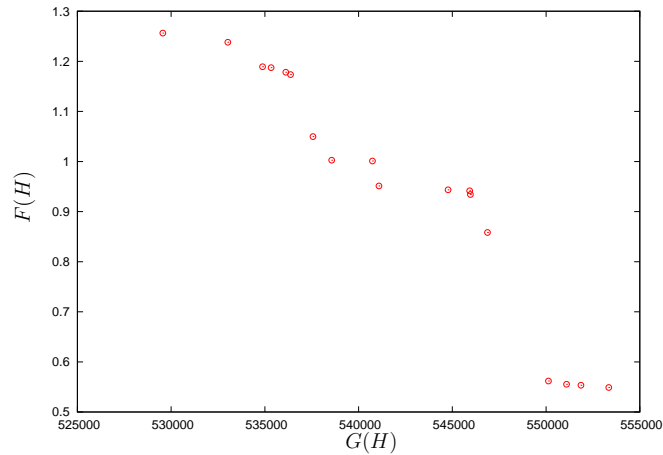
$F(H)$	$G(H)$
184000	0.1815765
183811	0.1826236
185323	0.1797125
185546	0.1796613
182209	0.1832482
185591	0.1795608
184316	0.1803357
184607	0.1797683
184399	0.1798227

Table 3.9: Pareto points for problem 1 for the USA network

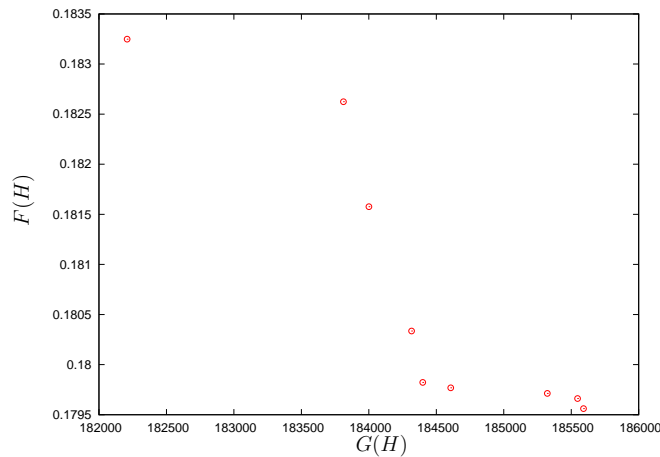
$F(H)$	$G(H)$
130648	0.2915859
135517	0.2797399
132302	0.2806475
131852	0.2847039
128143	0.2923844
131317	0.2851849
133681	0.2797650
132168	0.2813254
135852	0.2764559
133678	0.2806319
131011	0.2893453
131230	0.2861803
132330	0.2806472
130527	0.2920086

Table 3.10: Pareto points for problem 2 for the USA network

$F(H)$	$G(H)$
416368.6	0.6752853
408520.4	0.7565617
404133.8	0.7748202
409470.5	0.7102838
413599.9	0.6821854
406588.9	0.7596590
408310.3	0.7576518
408526.7	0.7409281
406065.3	0.7745344



(a) Pareto front of problem 1

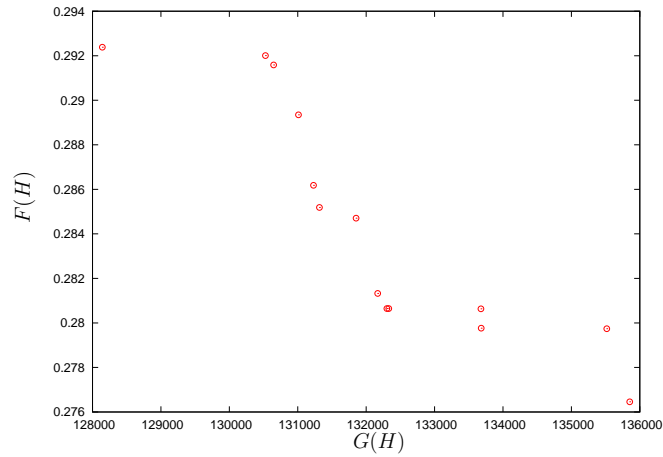


(b) Pareto front of problem 2

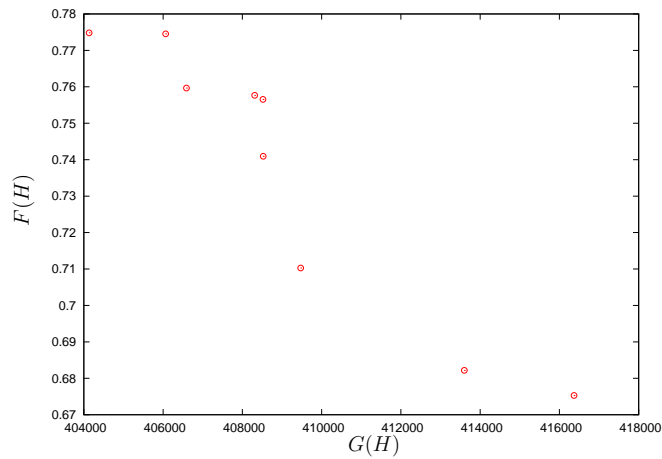
Figure 3.13: Pareto front of the African network

### 3.4 Wavelength Assignment

After finding a set of paths for each of the algorithms that we have described in the previous sections we are now in the position to assign wavelengths to each set of paths. We find the number of wavelengths but we do not route packet traffic between source and destination nodes over the logical topology. We assume that the wavelength continuity constraint is not violated, that means no conversion from electronic to optical or optical to electronic takes place in the network. Once the lightpath routes are fixed the remaining wavelength assignment problem can be represented as a graph coloring problem. Each lightpath corresponds to a node in the graph, and two nodes are set as neighbours only if



(a) Pareto front of problem 1



(b) Pareto front of problem 2

Figure 3.14: Pareto front of the USA network

the respective lightpaths share at least one common link. The objective is to configure a lightpath between each node pair. The graph coloring algorithm [53] assigns the smallest feasible wavelength channel to each lightpath in the descending order of node degree in the graph. There are many graph coloring algorithms and each algorithm may yield different results. The graph coloring algorithm used finds the number of wavelengths each of the routing scheme can achieve but this number is not necessarily the optimum.



## Discussion of Results

### Single Objective

The number of wavelengths depends on the size of the network. The larger the network the more wavelengths will be assigned to it. With our two networks this was not an exception as the African network was assigned more wavelengths compared to the USA network. Tables 3.5 and 3.6 show the number of wavelengths obtained by both algorithms using three different objective functions. Link occurrence is the worst as it obtains the highest number of wavelengths in both networks. This explains the problem of a link appearing in several paths at the same time. It not only has the bad effect of congesting the network but also results in assigning a large number of wavelengths. In terms of the performance of all objective functions the additive function is again the best having a minimum of 80 and 52 wavelengths for the African and the USA networks respectively. However what is interesting is the relatively small number of wavelengths that GA was able to find compared to hybridGA. Shortest paths are usually preferable but this choice may have to be sacrificed at times to allow more lightpaths to be set up. GA found a small number of wavelengths as most of its paths are also not shortest paths.

### Multi-Objective

The solution to the MOP is a set of trade-off points which represent a routing scheme (set of paths). Tables 3.11 and 3.12 show the number of wavelengths obtained in both networks using the objective functions in equations (3.9) and (3.10). For problem 1 in the African network, the best Pareto point in terms of the smallest number of wavelengths is the points with values  $G(H) = 535335$  and  $F(H) = 1.18$  with 73 wavelengths. However the delay is greater than 1 which is not a desirable situation in a network. For the USA network the maximum and minimum number of wavelengths are 48 and 57 respectively regardless of the objective function used. For problem 2 in the African network, the maximum and minimum number of wavelengths are 71 and 82 respectively. We conclude that the small number of wavelengths is obtained with a Pareto point having relatively high values of both objective functions and in particular  $G(H)$ .

Table 3.11: Pareto points with corresponding wavelengths for the African network

Problem 1			Problem 2		
$G(H)$	$F(H)$	No. of $\lambda$	$G(H)$	$F(H)$	No. of $\lambda$
553345	0.5489377	80	184000	0.1815765	78
533021	1.238070	79	183811	0.1826236	77
545926	0.9415441	76	185323	0.1797125	76
536114	1.178173	80	185546	0.1796613	82
535335	1.187310	73	182209	0.1832482	79
534877	1.189187	81	185591	0.1795608	71
544771	0.9435672	78	184316	0.1803357	76
545964	0.9341992	88	184607	0.1797683	80
537567	1.049727	83	184399	0.1798227	80
546876	0.8583523	76	-	-	-
551092	0.5553029	84	-	-	-
529557	1.256382	80	-	-	-
551862	0.5534490	80	-	-	-
550125	0.5617487	74	-	-	-
541084	0.9512489	79	-	-	-
536371	1.173479	74	-	-	-
540738	1.001054	81	-	-	-
538564	1.002640	75	-	-	-

Table 3.12: Pareto points with corresponding wavelengths for the USA network

Problem 1			Problem 2		
$G(H)$	$F(H)$	No. of $\lambda$	$G(H)$	$F(H)$	No. of $\lambda$
130648	0.2915859	48	416368	0.6752853	48
135517	0.2797399	51	408520	0.7565617	51
132302	0.2806475	48	404133	0.7748202	48
131852	0.2847039	53	409470	0.7102838	53
128143	0.2923844	55	413599	0.6821854	55
131317	0.2851849	57	406588	0.7596590	57
133681	0.2797650	48	408310	0.7576518	48
132168	0.2813254	54	408526	0.7409281	54
135852	0.2764559	53	406065	0.7745344	53
133678	0.2806319	47	-	-	-
131011	0.2893453	50	-	-	-
131230	0.2861803	48	-	-	-
132330	0.2806472	50	-	-	-
130527	0.2920086	51	-	-	-

## Chapter 4

# Optimisation of an AWG-based Metro WDM Network

### 4.1 Introduction

One of the key issues in today's telecommunications networks is the need to communicate faster and to be able to provide information when we need it, how we need it and where we need it. The number of Internet users has increased dramatically in recent years as reported by most service providers [10, 55, 56]. It has also been observed that many Internet users are using more and more bandwidth-intensive networking applications such as voice and video. This gives rise to the need for having network infrastructures in place that meet such high bandwidth demands. Such infrastructures should provide more bandwidth than current high speed networks such as ATM.

Recent studies [57, 58] have indicated that optical single hop Wavelength Division Multiplexing (WDM) networks have the potential to provide high network throughput and low delay connectivity in metropolitan and local area settings [12]. This is the case for a WDM network using an Arrayed Waveguide Grating (AWG). However, the best performance of such networks highly depends on the parameter settings of the AWG. In order to get a better throughput-delay performance, hardware parameters such as the degree of the underlying AWG and the degree of the combiners and splitters must be set properly. In addition the software parameters, Medium Access Control (MAC), such as node back-off probability or the number of control slots must also be set correctly.

This chapter addresses the issue of optimising an AWG-based metro WDM network with the objective of improving performance by simultaneously maximising the throughput and minimising the delay. This optimisation problem involves setting up the network parameters consistently. These two objectives are in conflict and choosing appropriate parameters is a difficult task due to the large search space. We make use of genetic algorithms to set up the parameters and the results are Pareto points with either high throughput and moderate delay or low delay but with moderate throughput. In real life such points are of practical importance depending on the traffic type we may wish to route. For example, HTTP or FTP traffic are not very delay-sensitive and points which yield high throughput and moderate delay may be well suited here. On the other hand delay sensitive traffic such as Voice over IP (VoIP) may be well suited for such points where the delay is low with a moderate throughput. All such points constitute the Pareto front which is important in the provisioning and planning of new networks in order to determine the best architecture. It is also crucial in controlling the efficient operation of installed network hardware [12].

We make use of EMOO algorithms to solve the problem of maximising throughput while minimising the delay in an AWG-based metro network. To achieve this, we use four different EMOO algorithms. The results obtained by all the algorithms are very similar. However the performance of each algorithm differs from one another. The same problem is developed and solved in [12] using the Multi-Objective Genetic Algorithm (MOGA) [22]. Our contribution is that we have solved the problem with more than one EMOO and compared the results obtained by each as well as the performance of each algorithm. Before we describe the network and model used, we first devote the section below to describing WDM system.

## 4.2 Wavelength Division Multiplexing

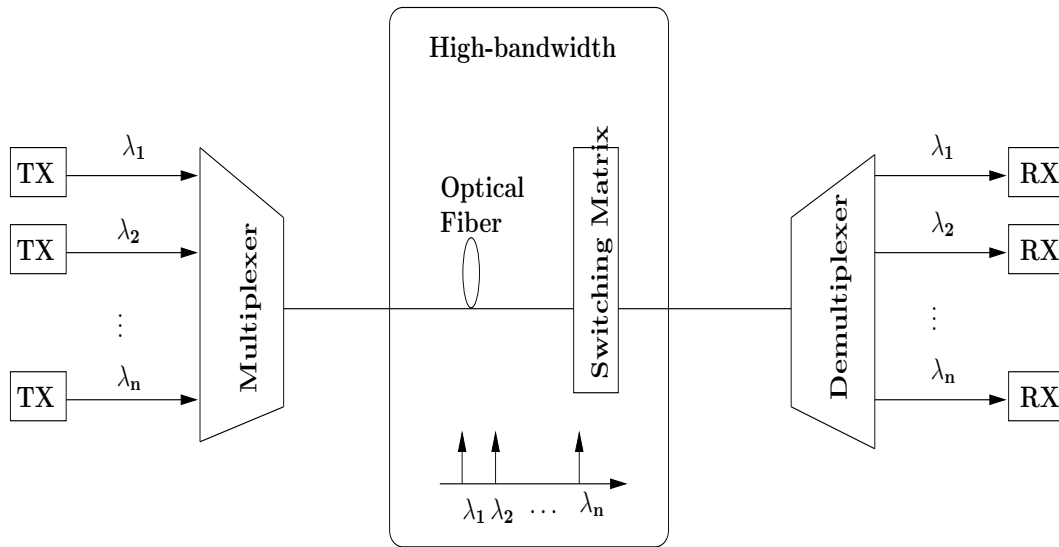
A single fiber can transport several multiplexed channels to form a higher speed channel. Several technologies are in place to multiplex channels but most of them do not use a large portion of the available fiber bandwidth. Time Division Multiplexing (TDM) can be used in either the optical or the electrical domain. In TDM, each channel is allocated a time

slot to transmit information and then waits for its turn after other channels have had the opportunity to transmit. Other methods exist such as Sub-Carrier Multiplexing (SCM), Space Division Multiplexing (SDM) and Code Division Multiplexing (CDM). All these methods do not utilise a large portion of fiber bandwidth either because of limitations in sub-carrier frequencies and data rates by the available electrical and optical components or they become too complex and costly to utilise [59]. The solution to utilising a large portion of the bandwidth available in fiber is to use wavelength division multiplexing.

With WDM using fiber optic technology, we can achieve enormous bandwidth (up to 50 Tera bits per second), low signal attenuation (up to  $0.2\text{dB}/\text{km}$ ), low signal distortion, low power requirements and low cost [60].

Different signals can be independently transmitted on a single fiber with each of them located at a different wavelength. Data can be routed on these wavelengths where each wavelength acts as a communication path by acting as the signature address of the origin, destination or routing. To be able to route traffic on these wavelengths we need equipment that is wavelength selective so as to allow recovery, routing and transmission of specific wavelengths. Some of the devices that operate in WDM systems include optical filters, multiplexers and demultiplexers, splitters, star couplers, wavelength routers, optical cross-connects, wavelength converters, transmitters and receivers. The reader is referred to [61, 62] for details concerning the operational principal and physical properties of all the above components.

In this chapter we will focus on a particular component known as the Arrayed Waveguide Grating (AWG) also known as the Optical Phased Arrayed (PHASAR), Phase Arrayed Waveguide Grating (PAWG) or Waveguide Grating Router (WGR). Figure 4.1 shows an example of a simple WDM system. Optical signals at  $n$  different wavelengths from separate transmitters are combined in a multiplexer as shown on the left of the diagram. After being transmitted through a high-bandwidth optical fiber, the combined optical signals are demultiplexed at the receiving end to each receiver. Each receiver selectively recovers only one wavelength by using a tunable optical filter.



Tx: Transmitter

Rx: Receiver

$\lambda_i$ : Wavelength

Figure 4.1: An example of a simple WDM system

## 4.3 Arrayed Waveguide Grating

### 4.3.1 Overview

Arrayed Waveguide Grating is a device that allows multiple wavelengths to be combined and separated in a wavelength division multiplexing system. It can be used as a dispersion slope compensator, wavelength router and drop-and-insert element [62]. Its popularity can be attributed to its capabilities of multiplexing/demultiplexing a high number of optical signal at a relatively low loss [63]. Several networks ranging from Local Area Networks (LANs), Passive Optical Networks (PONs) and national scale networks have been increasingly using AWG as a passive wavelength router. AWG is efficient due to its passive wavelength reuse. Being passive implies that it inherently requires no electric power to perform its task.

### 4.3.2 Architecture of AWG and Working Principles

An Arrayed Waveguide Grating can achieve three functions. These include periodic wavelength routing, spectrum slicing and wavelength reuse. Wavelength reuse is crucial in increasing the throughput-delay performance, which is the subject of this chapter. An  $N \times N$  AWG can simultaneously accept  $N$  input signals (wavelengths) at each of its input ports and route each signal to a particular output using one Free Spectral Range (FSR). If we have a  $4 \times 4$  AWG and we launch 8 wavelengths in one of its input ports, the FSR will be 4. This means that every fourth wavelength will be routed to the same output port. No collisions occur in a AWG because it is a non-blocking device [64].

The physical degree of an AWG, denoted by  $D$ , is the number of wavelengths per FSR. In the example above the degree is four and it is generally referred to as an  $4 \times 4$  arrayed waveguide grating. Using  $R$  fixed FSRs,  $R \in \mathbb{N}$ , we can achieve  $R$  simultaneous transmissions between a particular AWG input port and an arbitrary AWG output port.

AWG can also achieve spectrum slicing. Spectrum slicing means implementing broadcasting in WDM networks that are based on wavelength-sensitive devices [65]. When  $N$  broadband Light Emitting Diodes (LED) are launched in an input port, an AWG slices the broadband spectrum in such a way that in each FSR, one slice is routed to either AWG output port. Therefore with  $R$  FSRs, there are exactly  $R$  slices at each FSR. While both wavelengths and broadband signals can be simultaneously used at the same AWG input port, their use differs. Wavelengths serve as channels to carry data whereas broadband signals are well suited for carrying control information. Since each wavelength and broadband signal can be applied in all AWG input ports at the same time and no channel collision can occur at an AWG input port, a  $D \times D$  AWG can therefore spatially reuse each wavelength  $D$  times. Also, listening to a slice restricts the receiver to wavelengths that originate from the same AWG input port as a slice [65].



## 4.4 AWG-Based Network

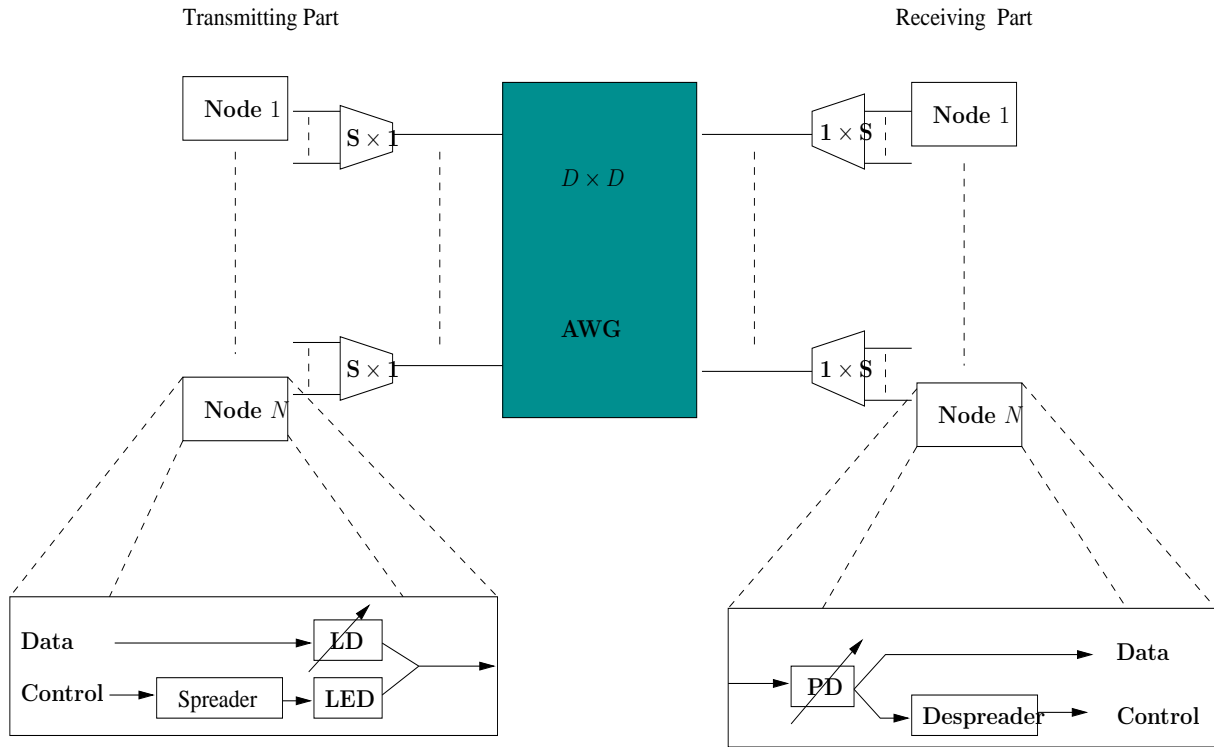
### 4.4.1 Network Architecture

Figure 4.2 shows an example of a  $D \times D$  AWG-based simple network. It consists of  $N$  different nodes each equipped with a single fiber for receiving information on one side and another fiber on the opposite side for sending information. Each transmitting fiber is connected to a wavelength-insensitive  $S \times 1$  combiner. A combiner is a device that combines several input optical signals (usually from fibers) into one or several output fibers. Each receiving part is equivalently connected to a wavelength-insensitive  $1 \times S$  splitter. A splitter is a device that splits multiple optical signals from a single optical signal. In this considered network a combiner collects data from  $S$  attached nodes and a splitter distributes signals to  $S$  nodes at each output port. Each node can transmit on different wavelengths at the same time, thus increasing the degree of concurrency. Since every node is connected to a combiner on its receiving part and also to a splitter on its transmitting part, opposite combiners and splitters must have the same number of ports. However this should not be regarded as a restriction on combiners and splitters. They may have different degrees where we wish to expand the network. In this thesis, only combiners/splitters of the same degree are considered in order to ensure uniformity of power losses throughout all nodes [65].

### 4.4.2 Medium Access Control (MAC) Protocol

Figure 4.3 shows an example of a data packet handled by a MAC protocol. There are three reasons why data handling is performed by the MAC protocol [65].

1. Since we are considering a single-hop network, there are no intermediate nodes or alternative routes in our network. As a result packet switching is handled by the MAC sublayer because of the absence of the network layer.
2. Fast transceivers with a limited tuning range are used in order to avoid latency. As a consequence there are fewer channels than nodes. This results in nodes sharing channels, calling for the MAC protocol for the control of these channels.



LD: Laser Diode  
 LED: Light Emitting Diode  
 PD: Photodiode  
 $S \times 1$  : Combiner,  $1 \times S$  : Splitter

Figure 4.2: An example of a AWG-based simple network

3. In order to provide full connectivity, each transceiver has to be tuned over at least one FSR. This is because of the routing characteristics of an AWG. Hence all nodes share all wavelengths raising the need of a MAC protocol.

A frame is divided into  $F$  slots,  $F \in \mathbb{N}$ , where the length of the slot equals the transmission time. The frame is divided into two major slots namely  $M$  slots with  $1 \leq M < F$  and  $F - M$  slots. The first  $M$  slots are mainly for control information processing. The coordination of control messages takes place in  $M$  slots prior to transmission. Data packets are sent in the remaining  $F - M$  slots allowing receivers to be tuned to any arbitrary wavelength. Now we describe how the MAC protocol works from both the transmitting and receiver end. From the transmitting part, if a node has no data to send, the light emitting diode and the Laser Diode (LD) remain idle. When a node has data to send, say node  $j$  with

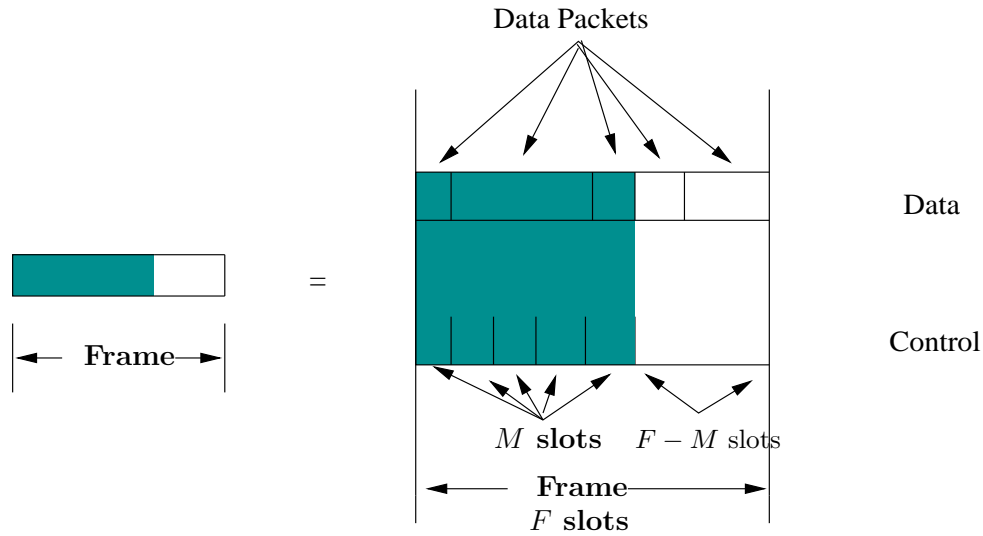


Figure 4.3: A typical frame of a data and control packet

$1 \leq j \leq N$ , and this packet arrives at node  $i \neq j$ ,  $1 \leq i \leq N$ , node  $i$ 's LED broadcasts a control packet in one of the  $M$  slots of the frame allocated to the AWG input port where node  $i$  is attached to. The data packet can be of variable length  $L$  where  $1 \leq L \leq F$  and the units are described in slots.

On the receiving part, every node learns about all activities of the other nodes and the success or failure of its own control packets. This is achieved by each node tuning its own receiver to one of the corresponding channels during the first  $M$  slots of each frame. In frame  $k$ ,  $1 \leq k \leq D$ , each receiver collects the control packets originating from nodes that are attached to AWG input port  $k$  [58].

## 4.5 Formulation of The Multi-Objective Problem

### 4.5.1 Network Overview

In this section we formulate the multi-objective optimisation problem by first describing the network based on an AWG [66]. The following model is based on the one developed in [67]. We consider an arrayed waveguide grating with degree  $D$ . Each node in the network is connected to  $S \times 1$  combiners on one side and to  $1 \times S$  splitters on the opposite side. In total the network connects  $N = DS$  nodes. There are  $R$  adjacent FSRs at each input

Table 4.1: Table displaying parameters of the optimisation model

Hardware parameters	$N$ : Number of nodes in the network. $\Lambda$ : Number of usable wavelengths at each AWG port. $D$ : Degree of AWG. $R$ : Number of FSRs. $S$ : Degree of combiner and splitter.
Software parameters	$F$ : Number of slots in a frame. $M$ : Number of reservation slots in a frame. $K$ : Length of short packets in slots. $p$ : Re-transmission probability of a control packet in ALOHA contention.
Traffic parameters	$\sigma$ : Packet generation probability. $q$ : Probability that a given packet is long.
Objective functions	$TH$ : Average network throughput in transmitting nodes. $Delay$ : Average packet delay in slots.

port, each consisting of  $D$  wavelengths. Therefore the total number of wavelengths at each AWG port is  $\Lambda = DR$ . The MAC protocol runs an attempt-and-defer type of protocol, that is to say a data packet is only transmitted after the corresponding control packet has been successfully transmitted [12].

The transmission time of a packet is divided into cycles. Each cycle consists of  $D$  frames which in turn consists of  $F$  slots. Each frame is divided into control and data slots. The length of the slot is equal to the transmission time of the control packet. At the beginning of transmission cycle, a node that is not backlogged generates a new packet with probability  $\sigma$ . It generates a long packet of size  $F$  slots with probability  $q$ . The probability of generating a short packet of size  $K = F - M$  is  $1 - q$ . The transmission protocol of the control packets in slots  $M$  is based on slotted ALOHA contention. The protocol requires all nodes to be tuned to one of the LED slices carrying the control information [12].

### 4.5.2 Objective Function

We now describe the model derived in [67]. We consider the case where  $M \leq F$ . Let  $E[\mathcal{L}]$  be the expected number of successfully scheduled long packets that originate from a given fixed AWG input port to a given fixed AWG output port per cycle. Similarly, let  $E[\mathcal{S}]$  be

the number of such short packets. Let  $v$  be the fraction of idle (not backlogged) nodes in steady state. The arrival rate of control packets to a given control slot is given by

$$\beta = \frac{S}{M}[\sigma v + p(1 - v)]. \quad (4.1)$$

Let  $Z$  denote the number of successful control packets destined to a given output port in a given frame. The probability that  $Z$  is equal to  $k$  is given by:

$$P(Z = k) = \binom{M}{k} \left(\frac{\beta e^{-\beta}}{D}\right)^k \left(1 - \frac{\beta e^{-\beta}}{D}\right)^{M-k}, \quad k = 0, \dots, M. \quad (4.2)$$

Let  $\tilde{q}$  be the probability that a given packet corresponds to a long data packet. Since long packets are more difficult to schedule and thus generally need more transmission time than short packets, we have  $\tilde{q} > q$ . Finally the expected number of scheduled long packets is

$$\begin{aligned} E[\mathcal{L}] &= \sum_{k=0}^M E[\mathcal{L}|Z = k]P(Z = k) \\ &= \sum_{k=0}^M \min(k, R)\tilde{q} \binom{M}{k} \left(\frac{\beta e^{-\beta}}{D}\right)^k \left(1 - \frac{\beta e^{-\beta}}{D}\right)^{M-k} \\ &= \tilde{q} \left[ R - \sum_{k=0}^{\min(R, M)} \binom{M}{k} \left(\frac{\beta e^{-\beta}}{D}\right)^k \left(1 - \frac{\beta e^{-\beta}}{D}\right)^{M-k} (R - k) \right] \\ &= \tilde{q} \left[ R - \sum_{k=0}^{\min(R, M)} P(Z = k)(R - k) \right] \\ &= \tilde{q}\varphi(\beta). \end{aligned} \quad (4.3)$$

where  $\varphi(\beta) = R - \sum_{k=0}^{\min(R, M)} P(Z = k)(R - k)$ .

Similarly the expected number of scheduled short packets is

$$\begin{aligned}
 E[\mathcal{S}] &= \sum_{k=0}^R (1 - \tilde{q})kP(Z = k) + (1 - \tilde{q})R \sum_{k=R+1}^M P(Z = k) \\
 &+ \sum_{k=R+1}^M \left[ \sum_{j=1}^{k-R} \gamma_j \sum_{m=j}^{k-R} \binom{k-R}{m} (1 - \tilde{q})^m \tilde{q}^{k-R-m} \right] P(Z = k) \\
 &= (1 - \tilde{q}) \left[ R - \sum_{k=0}^R (R - k)P(Z = k) \right] \\
 &+ \sum_{j=1}^{M-R} \gamma_j \sum_{m=j}^{M-R} \sum_{k=m+R}^M \binom{k-R}{m} (1 - \tilde{q})^m \tilde{q}^{k-R-m} P(Z = k) \\
 &=: h(\tilde{q}, \beta),
 \end{aligned} \tag{4.4}$$

where  $\gamma_j$  accounts for the packing of the short packets into the schedule and is given depending on these two cases:

Case 1. If  $\lfloor F/K \rfloor - 1 > 0$

$$\gamma_j = \sum_{m:m \leq v_j} \binom{R}{m} \tilde{q} (1 - \tilde{q})^{R-m}$$

Case 2. If  $\lfloor F/K \rfloor = 1$

$$\gamma_j = \begin{cases} 1, & \text{if } j < (D - 1)R \lfloor \frac{F-M}{k} \rfloor \\ 0, & \text{otherwise} \end{cases}$$

The average throughput of the network is therefore given by:

$$TH = D^2 \frac{FE[\mathcal{L}] + KE[\mathcal{S}]}{FD} \tag{4.5}$$

Equation (4.5) is defined as the average number of transmitting nodes in a slot or the average number of transmitted packets data packets per frame.

In equilibrium, the number of serviced long packets and short packets is equal to the number of newly generated long and short packets [12]. The probability that a given control packet corresponds to a long data packet is:

$$\tilde{q} = q \frac{S\sigma v}{D\varphi(\beta)} \tag{4.6}$$

and

$$(1 - q) \frac{S\sigma}{D} v = h(\tilde{q}, \beta). \quad (4.7)$$

We used code from Numerical Recipes in C++ [68] to obtain the values of  $E[\mathcal{S}]$  and  $E[\mathcal{L}]$  respectively. To solve the following problem we first find the values of  $E[\mathcal{L}]$  and  $E[\mathcal{S}]$ . But first we numerically solve for  $v$  from equation (4.7). We then introduce the obtained value of  $v$  into equation (4.1) to obtain  $\beta$  which is in turn inserted into equation (4.3) to obtain  $\varphi(\beta)$ .

The values of  $\varphi(\beta)$  and  $v$  are inserted in equation (4.6) to obtain the value of  $\tilde{q}$ . Finally we solve for  $E[\mathcal{L}]$  and  $E[\mathcal{S}]$  from equations (4.3) and (4.4) respectively using EMOO algorithms.

Since  $v$  is the fraction of idle nodes in steady state, it takes on values between zero and one. We have set the number of nodes to 320, if there are 60 nodes in steady state then  $v$  will be 0.1875.  $\beta$  is the arrival rate of control packets and it therefore takes positive values. However the system is assumed to have nodes which can only hold one data packet and one control packet in the buffer at a time [67]. At last,  $\tilde{q}$  is a probability and therefore takes its values between zero and one.

The mean packet delay is defined as the average time period in slots from the generation of the control packet corresponding to a data packet until the transmission of the data packet. The average delay is

$$Delay = \left[ \frac{S}{D(E[\mathcal{L}] + E[\mathcal{S}])} - \frac{1 - \sigma}{\sigma} \right] DF. \quad (4.8)$$

We now determine the decision variables and the constraints imposed on them. Almost all parameters (see Table 4.1) take integer values except for the packet re-transmission probability  $p$  which takes real values in the interval  $[0, 1]$ . The hardware decision variable  $D$  is greater than two,  $D > 2$  and  $D \leq \Lambda$  where  $\Lambda$  is the maximum number of wavelengths channels. We restrict  $D$  to values which are powers of two because the number of ports in commercially available photonic equipments is a power of two.

$R$  is also an integer such that  $R = \Lambda/D$ . The larger  $R$ , the more parallel channels we

will have and hence the larger the output. Combiners and splitters are arranged in such a way that minimises the required degree  $S$ ; which in turn minimises the splitting loss in the combiners/splitters. Hence we set  $S = \lceil N/D \rceil$ . The variable  $F$  can take any values greater than 1, while the variable  $M$  is restricted between 1 and  $F$  inclusive. However no particular values are imposed on  $M$  and  $F$ , instead we let the algorithms find the appropriate ones.

## 4.6 Experimental Results

We present our experiment results based on the simulations we performed using different EMOO algorithms. These include the Non-dominated Sorting GAI [40], the Multi-Objective GA [22], the Pareto Archived Evolutionary Strategy [25] and the Micro-Genetic Algorithm for multi-objective optimisation [30]. Our objective functions, *throughput* and *delay*, have  $F$  and  $M$  as decision variables; the rest of the parameters are given as inputs. The algorithms then find suitable combinations of such variables which give the best acceptable solutions.

Each of the four algorithms solves the problem in a slightly different way depending on the selection method used and the way the population is kept diverse. However they all have certain elements in common. First a set of chromosomes is randomly generated. These chromosomes represent candidate solutions to the problem. Each chromosome is then evaluated against the objective functions in order to find its fitness. Since we are solving a MOP problem, fitter individuals are identified based on their dominance. Mutation is applied to chromosomes so as to introduce diversity and strengthen weaker individuals whereas crossover breeds new individuals using a combination of two individuals. In the case of a binary mutation a bit in a chromosome is flipped and for a real binary mutation a small value is either added to the chromosome or subtracted from it. Chromosomes that are non-dominated are given a bigger chance of being selected and participate in the next generation. After maximum generation only non-dominated individuals will make part of the solution. For more details on how genetic algorithms are used in order to optimise functions, the reader is referred to [18].



### 4.6.1 Parameter Settings

In all our numerical experiments we have set the number of nodes to 320 and the maximum number of wavelengths at each AWG port to 8. That is  $N = 320$  and  $\Lambda = 8$ . We allow  $\sigma$ ,  $p$ , and  $q$  to take values between zero and one inclusive. We restrict the values of  $F$  and  $M$  in  $[1, 100]$  such that  $1 \leq M \leq F$ . We only consider a  $2 \times 2$  and a  $4 \times 4$  AWG that is to say  $D$  can only take two values namely 2 and 4.

For genetic algorithm parameters we set the population size of 100, the maximum generation to 100, for NSGAI and MOGA, and for PAES and Micro-GA the maximum population was set to 5000. Further parameters such as the probability of crossover and the probability of mutation probabilities were set to 0.9 and 0.01 respectively. We discuss the results from the obtained Pareto points using different combinations of parameters and then evaluate the performance of each EMOO genetic algorithm making a comparison between them.

### 4.6.2 Solution Approaches: EMOO Algorithms

We used the four algorithms which all solve multi-objective optimisation problems. NSGAI [40] and MOGA [22] were described in chapter 2 sections 2.4.4 and 2.4.3 respectively. We now briefly describe the working principles of the other two algorithms namely PAES and Micro-GA. For more details, the reader is referred to [25] and [30] respectively.

Micro-GA uses traditional genetic algorithm operators such as tournament selection, two point crossover and mutation with a small population to reinitialise the process of generating the Pareto front. To maintain diversity, it uses an adaptive grid similar to one described in section 2.5.3. This cell based density method together with elitism provide a good technique to solve multi-objective problems in a way that is computationally efficient. Micro-GA uses three different forms of elitism with only one non-dominated vector being arbitrarily selected in each generation and copied to the next generation. In addition, it uses an archive, a place where identified non-dominated solutions are temporarily kept. These non-dominated individuals are introduced in the population afterwards and some of them are used in the initial population in order to start a new evolutionary cycle.

PAES also uses elitism and cell based density as a method to keep the population diverse.

Starting with one randomly generated chromosome, PAES finds good Pareto points in the following way. First a copy of this chromosome is made and then mutated in order to produce a new chromosome. If the new candidate solution is found to be non-dominated it is archived. The archive is a way for not only storing non-dominated individuals but also used as a pool for parent selection.

We begin by comparing the performance of an AWG having degree two and four respectively. We observe from figure 4.4 that for the same parameters an AWG with degree four performs much better as it produces not only considerably higher throughput but reduces the mean delay by half. This is as expected because as we increase the degree of AWG we are able to use more wavelengths which increases the network throughput and decreases the delay. However we notice the exponential behaviour of the two scenarios. For a degree two AWG, the delay explodes just before it reaches 16. This behaviour is observed in all our experiments and it tells us that there is a certain limit to the traffic that the network can handle, anything else will lead to long queues thus causing large delays.

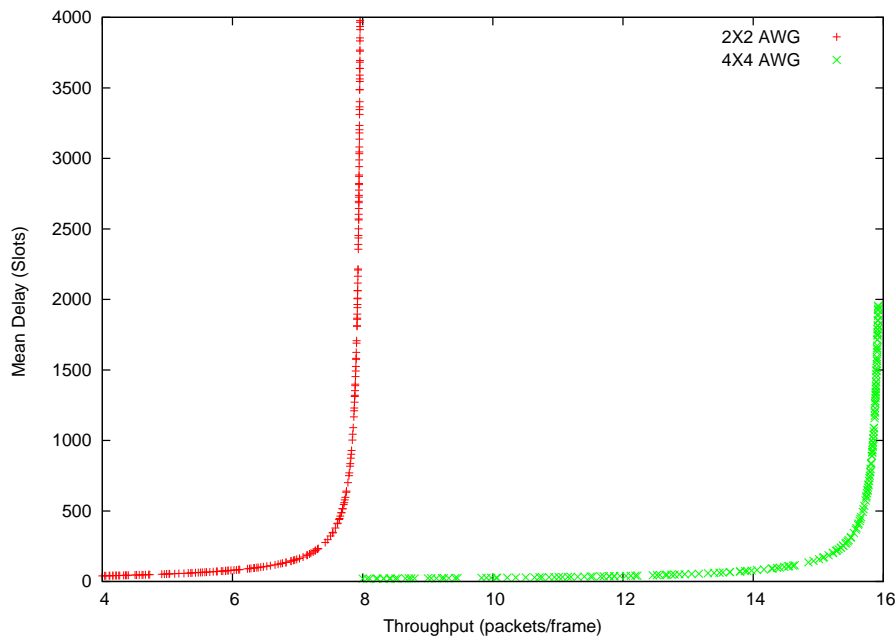


Figure 4.4: Pareto front obtained for  $\sigma = 0.9$

### Traffic Analysis

We observe from figure 4.5 that low traffic (small value of  $\sigma$ ) will provide relatively low throughput depending on the value of  $q$ . When we have a large transmission probability of long packets we are more likely to have a large number of short packets to transmit thus resulting in a large throughput and a small delay. For  $q = 0.8$  we have a large delay of up to more than 2500 slots whereas for  $q = 0.3$  the delay is as low as 1000 slots. The reason behind this is that there are exactly  $(D - 1)R$  retransmission slots for short packets in each cycle of the AWG input ports. Furthermore, having a large fraction of long packets results in more failures of packets scheduling. This will in turn require more and more retransmissions of the corresponding control packets. As a consequence larger delays will be observed.

A small value of  $\sigma$  does not only achieve low delay but also a low throughput. This can be explained by the fact that a small number  $M$  of control slots is sufficient to ensure a reasonably large success probability in the control packets contention. This happens when an idle node generates new packets at the beginning of a cycle with a small probability  $\sigma$  [12]. A small value of  $M$  will yield a small frame length  $F$  and in turn will yield short cycle length  $DF$  resulting in small delays.

Figures 4.6 and 4.9 on the other hand show the Pareto front obtained from relatively high traffic having a probability of packet regeneration of  $\sigma = 0.9$ . The two figures show that for a low value of  $q$  we have low delays whereas for a high value of  $q$  we have high values of the mean delay. We also observe an increase in the traffic regeneration probability which results in an increase in the throughput of the network. What we know is that we can only schedule at most  $R$  long packets and  $(D - 1)R$  short packets at each  $D$  AWG input port in a single cycle. We can at most use a total of  $\Lambda$  scheduled long packets and  $\Lambda(D - 1)$  short packets per cycle in the entire network. However short packets are allowed to take up the transmission slots of long packets. Therefore when  $D$  is large, the network will allow a large number of short packets to be scheduled which results in a larger throughput. This is because of the capability of an AWG to spatially reuse all  $\Lambda$  at all  $D$  ports of an AWG. However a larger  $D$  may increase the delay in the network if we keep the frame length constant. This is a result of a larger cycle length  $DF$ , which increases the delay incurred by the control packet pre-transmission coordination and re-transmissions, which operate

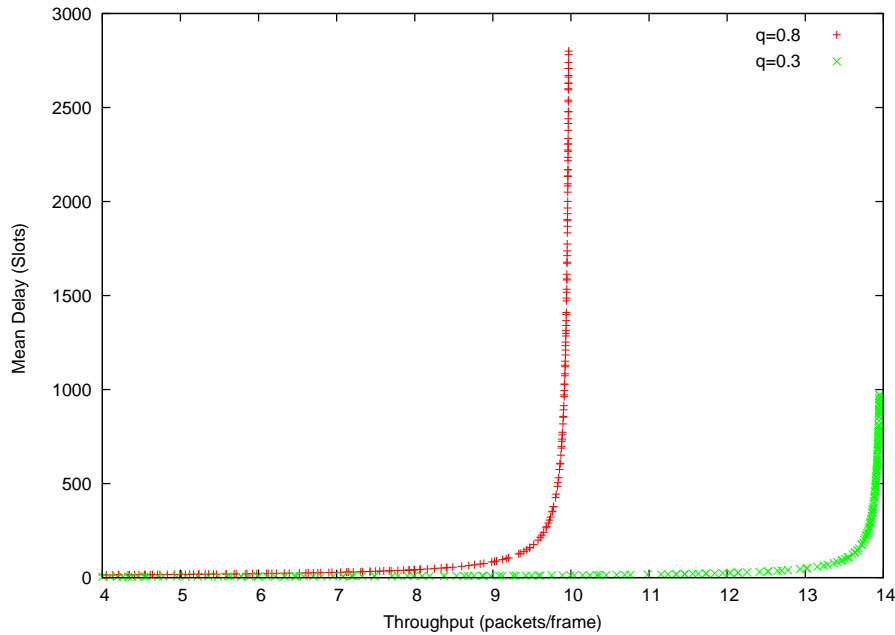


Figure 4.5: Pareto front for  $\sigma = 0.1$ ,  $q = 0.3$  and  $q = 0.8$

on a cycle basis [12].

### 4.6.3 Comparison of the Algorithms

All of these algorithms use Pareto dominance to evaluate the goodness of a solution. They also use elitism in order to provide convergence while keeping the population diverse. While NSGAI and MOGA use fitness sharing to diversify the population, PAES and Micro-GA on the other hand use cell based density.

We begin with our comparison by looking at the number of generations against the number of Pareto points found by NSGAI and MOGA. Figures 4.10 and 4.11 show the number of Pareto points found by the two algorithms as the evolution cycle evolves. The maximum generation is set to 100 and we observe that NSGAI finds 200 as early as from the 20<sup>th</sup> generation. On the other hand MOGA takes longer to stabilise as the number of Pareto points steadily increase before remaining constant at the 65<sup>th</sup> generation. NSGAI thus finds more Pareto points more quickly than its MOGA counterpart. However we cannot judge on this basis that NSGAI is a better algorithm than MOGA. A lesson to learn from

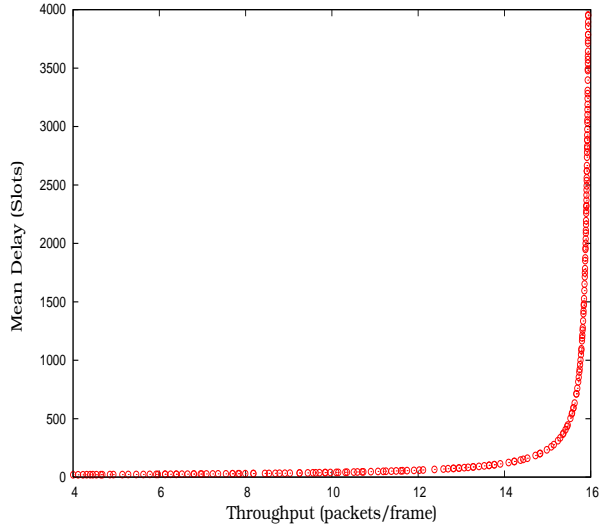


Figure 4.6: Pareto front of a  $2 \times 2$  AWG for  $\sigma = 0.9$  and  $q = 0.9$

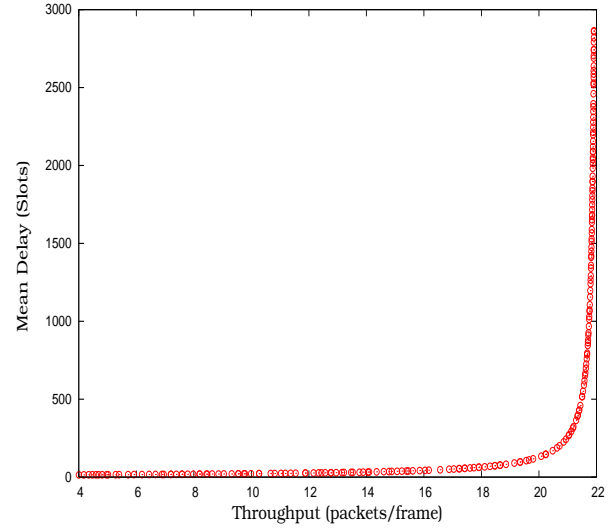


Figure 4.7: Pareto front of a  $2 \times 2$  AWG for  $\sigma = 0.9$  and  $q = 0.4$

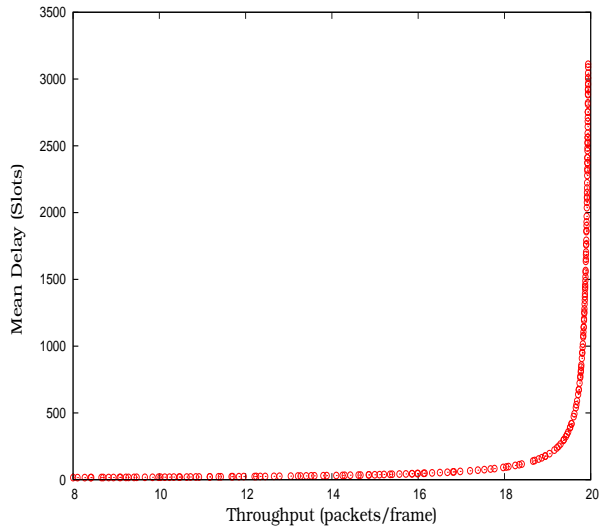


Figure 4.8: Pareto front of a  $4 \times 4$  AWG for  $\sigma = 0.9$  and  $q = 0.9$

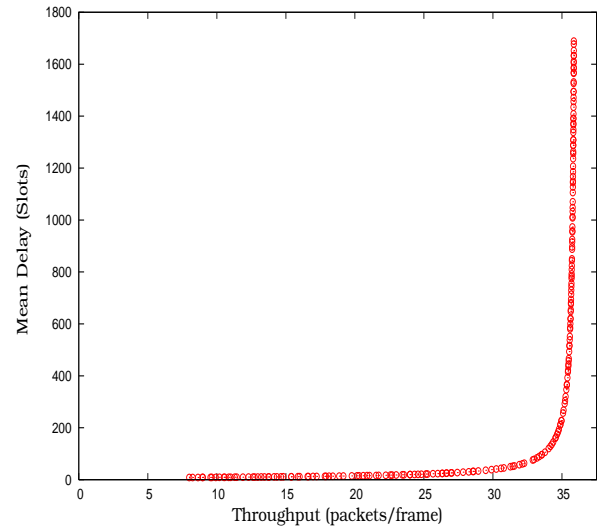


Figure 4.9: Pareto front of a  $4 \times 4$  AWG for  $\sigma = 0.9$  and  $q = 0.4$

here is that to increase the number of generations may in some instances leads to a large number of Pareto points. On the other hand a high number of maximum generations will not necessarily improve the number of Pareto points as is observed in case of NSGAI and MOGA after 20 and 65 generations respectively.

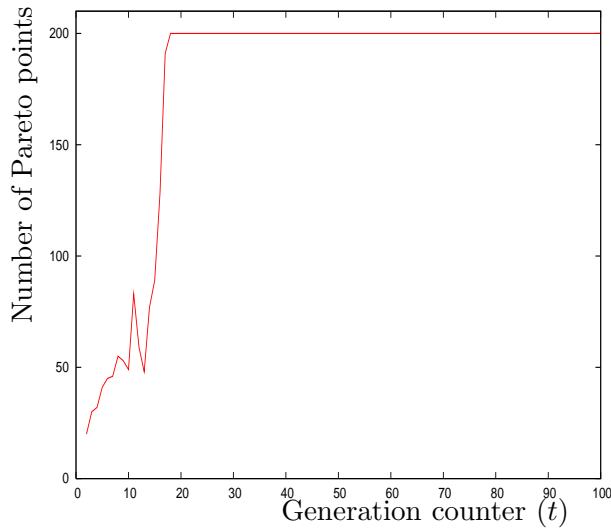


Figure 4.10: Generation counter vs number of Pareto points for NSGAI

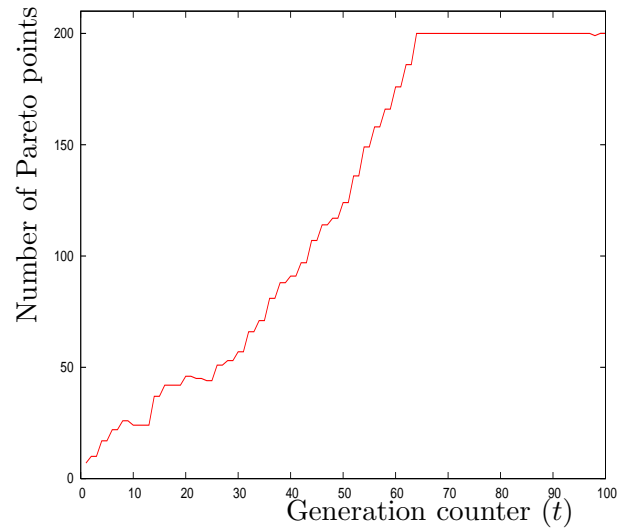


Figure 4.11: Generation counter vs number of Pareto points for MOGA

Similarly figures 4.12 and 4.13 show the number of Pareto points obtained by both PAES and Micro-GA respectively. One thing that is noticeable is that the maximum generations of both PAES and Micro-GA are greater than those of NSGAI and MOGA. Micro-GA finds more Pareto points before it changes the behaviour around generation 300 to 550 then continues finding more Pareto points before it stabilises just after the 1000<sup>th</sup> generation. From there on the number of points remains constant but a few down peaks are observed indicating a slight change in the number of points. On the other hand PAES finds points but does not stabilise until generation 2000. However a few fluctuations are observed indicating the change in the number of Pareto points obtained just after 2000<sup>th</sup> generation. This behaviour can be explained by the nature of genetic algorithms themselves which are sensitive to parameters settings and operators used.

Next we show the Pareto front obtained by all the four algorithms. They all display the same behaviour except, for the number of Pareto points found at each generation, they

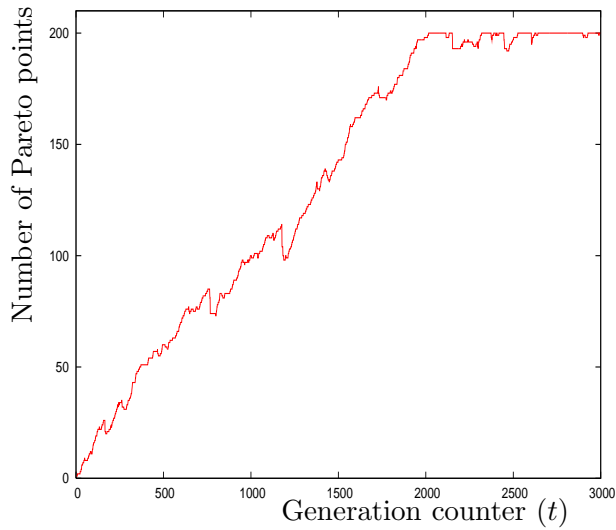


Figure 4.12: Generation counter vs number of Pareto points for PAES

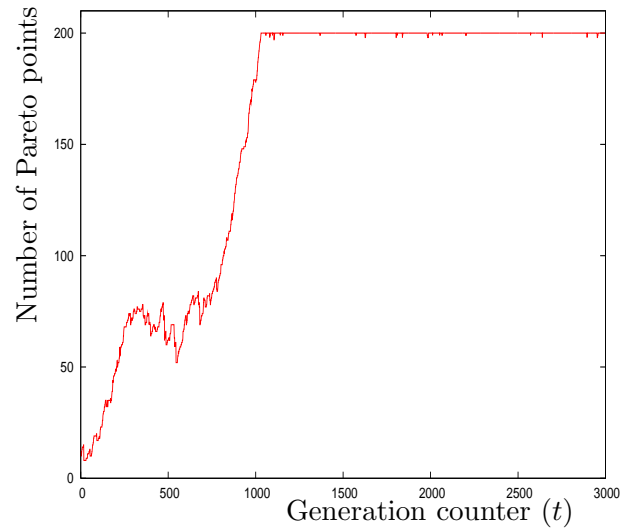


Figure 4.13: Generation counter vs number of Pareto points for Micro-GA

are almost the same. Figures 4.14, 4.15, 4.16 and 4.17 show different Pareto fronts. What is noticeable from all the graphs is the shape of the Pareto front which is the same for all algorithms as expected. However the difference also lies in the way the points are located along the curve. NSGAII seems to allocate these points evenly thus forming a smooth curve along the two axes. Both PAES and MOGA have points concentrated between 7 and 8 while Micro-GA also tries to spread the points but not as much as NSGAII.

There might be many reasons why the different algorithms behave differently in finding trade-off points. First of all each problem may have specific parameter settings such as crossover probability, population size, mutation probability and the number of generations. For the sake of a fair comparison we have kept the same population size for both NSGAII and MOGA. In addition MOGA uses fitness sharing on objective function values whereas NSGAII uses sharing on the dummy fitness of each individual and this fitness is proportional to the population size. To rank individuals, NSGAII keeps them in different fronts and then assigns them different ranks according to their respective fronts before the stochastic selection process is applied. On the other hand MOGA first identifies all non-dominated individuals and assigns them rank one. For the rest of individuals each one is assigned a rank that is proportional to the number by which it is dominated. The ranking scheme can affect the final solution because selection is carried out based on these

ranks.

In terms of execution speed we have run all our simulations on a 1.73 GHz Pentium IV machine with 512 RAM. It took 2.8 seconds for NSGAI to run and 0.47 seconds for MOGA. Apart from a good spread of Pareto points that NSGAI finds, it is also worth mentioning that MOGA found the lowest points with delay 2.9694 slots and a corresponding throughput of 40.0148 packets per frame and also the lowest run time.

We also compare trade-off points found by PAES to those found by Micro-GA. They both use archives to keep their non-dominated individuals and use cell density as a way to keep diversity in the population. Examining figures 4.16 and 4.17 we see that Micro-GA finds points which are well spread out between delay values ranging from 4 to 8 while most delay values for PAES are concentrated between 7 and 8. However we also see that PAES finds delay values slightly below one as shown in table 4.2. Both PAES and Micro-GA use archives and divide the solution space into grids depending on the number of objectives. For example if we have two objectives the solution space is divided into squares and if the number of objectives is three the solution space is divided into cubes. For PAES the size of the archive indicates the number of desired solutions and the number of subdivisions of the solution space in the grid is used to encourage diversity. On the other hand Micro-GA uses the archive to keep non-dominated individuals [30]. It also uses an adaptive grid similar to the one used in PAES in order to encourage diversity in the population. However this adaptive grid requires two parameters namely the expected size of the Pareto front, that is the number of desired solutions points, and the number of positions in which we will divide the solution space for each objective. This number of positions is similar to the number of subdivisions in PAES and in these experimental results we set it to 15. The archive size for both algorithms was set to 200. Despite similarities in the two algorithms, their respective working principals are quite different thus explaining the slight differences in the obtained results. The way the population is initialised and the selection process is different. On the running time issue it took PAES 1.0 second while Micro-GA took 0.53 seconds.

Although these algorithms have shown a slight difference in the Pareto points obtained the general results are almost the same as they all find Pareto fronts having the same shape. The issue of time also depends on the implementation of each algorithm and based on these differences one cannot judge that one algorithm is better than the other.



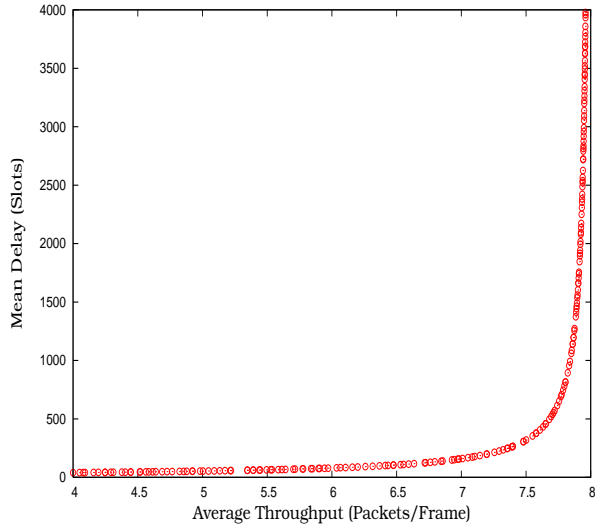


Figure 4.14: Pareto front for NS-GAII

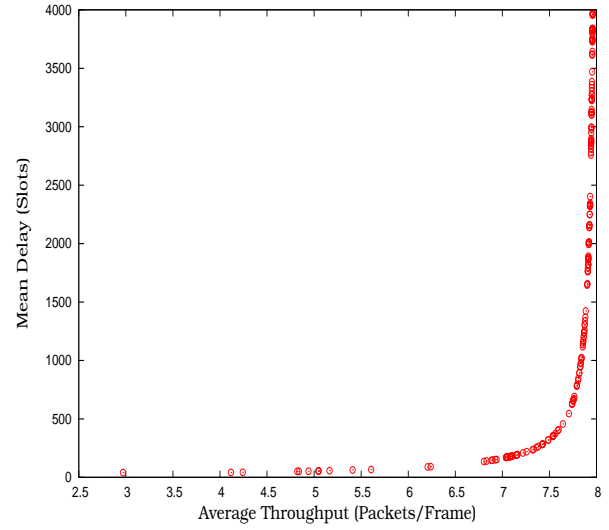


Figure 4.15: Pareto front for MOGA

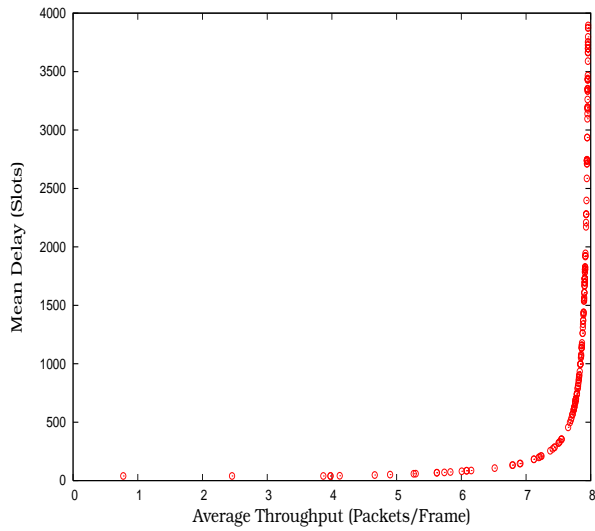


Figure 4.16: Pareto front for PAES

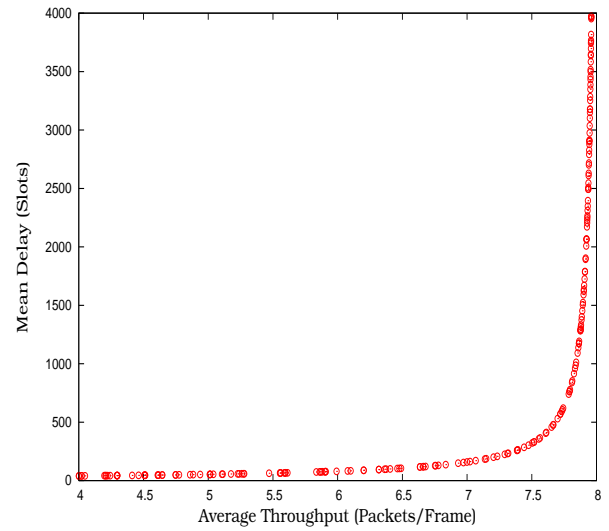


Figure 4.17: Pareto front for Micro-GA

Table 4.2: Pareto points for PAES

Mean Delay	Throughput
0.77	39.79
2.45	39.84
3.86	39.79
3.97	39.84
4.12	41.34
5.26	58.76
6.00	79.87
7.12	182.73
7.95	3895.04

Table 4.3: Pareto points for Micro-GA

Mean Delay	Throughput
4.00	39.7778
4.24	42.33
5.01	53.25
5.55	65.09
6.07	82.75
6.49	105.41
7.02	162.83
7.50	322.84
7.95	3974.58

Having performed simulations on four different algorithms we can now conclude by highlighting what we have observed. To this end a performance measure is needed in order to classify which of the algorithms has produced better results. Assessing the performance measure of EMOO is a multi-objective optimisation problem on its own. However the three most important aspects that can be used to assess the performance are [69]:

1. Maximise the number of Pareto points found.
2. Assuming we know the true Pareto front, we must minimise the distance of the Pareto front found by the algorithm with respect to the true Pareto front.
3. The Pareto front must be smooth and as uniformly distributed as possible.

Table 4.4: Pareto points for NSGAI

Mean Delay	Throughput
4.000005	39.77
4.515496	45.66
5.003341	53.09
5.495291	63.52
6.036289	81.02
6.530263	108.25
7.009696	160.66
7.500816	318.74
7.960000	3977.75

Table 4.5: Pareto points for MOGA

Mean Delay	Throughput
2.9694	40.0148
4.11628	40.983
5.04577	53.9384
5.60567	66.5515
6.20554	88.9328
6.80361	136.013
7.03839	169.491
7.53625	351.372
7.95992	3974.28

Knowing the shape of the true Pareto is not possible [69] and our case is not an exception due to the nature of the problem we are solving. However points 1 and 3 were taken into account. In terms of the number of Pareto points all algorithms found the same number of Pareto but at different maximum number of generations with NSGAI finding the total number of Pareto at the lowest maximum generation. In terms of finding Pareto points with fewer generations NSGAI was the first, followed by MOGA then Micro-GA and finally PAES. In terms of the smoothness of Pareto front NSGAI found the smoothest as seen from figure 4.14. MOGA also found one that is relatively smooth with PAES and Micro-GA trailing. Based on these observations one would rank NSGAI as an algorithm with the best performance on this particular problem.

## Chapter 5

# Layout Optimisation of a Wireless Sensor Network

### 5.1 What are Sensor Networks ?

Wireless Sensor Networks (WSN) consist of a large number of nodes. Usually when a large number of devices collaborate wirelessly and asymmetrically in a many-to-one data flow, they also constitute a sensor network [70]. These devices may be cameras, tiny computers or microphones. Sensor nodes (herein referred to as sensors) may perform different tasks such as sensing, communicating and processing. They are used to monitor environments, sensing may differ depending on the application. It may be seismic, acoustic, chemical, or physiological. Depending on the sensing, sensors are applied to provide home security, machine failure diagnosis, surveillance, animal or plant monitoring, military operations and chemical or biological detection. Sensors with different capabilities and processing powers may be deployed in the same field. However in a WSN three types of nodes exist: these include *sensor nodes*, *relay nodes* and *sink nodes*. The sensor node performs the task of reporting any activities happening in its surroundings. A relay node acts as a transit node and thus receives information and passes it to other nodes near it. A sink node acts as a Base Station (BS) as in cellular networks. All other nodes send their information to the sink node and the sink sends all information via satellite to a home base station where the information is processed. Every sensor can communicate or sense within a certain distance. A unit disk is usually considered for this matter and the distance of

communication/sensing is referred to as the communication/sensing radius. Sensing or communication may be achieved either by radio waves or light [71].

The area of deployment of WSN may depend on the application. Such areas may be wooded, mountainous or urban. The major problem with sensors is their limited source of energy. They operate on battery power and once the node is deployed, its battery cannot be replaced. The first important aspect of WSN is therefore to maximise the available energy in each sensor. The deployment of sensors may be random or deterministic. In the deterministic case, each sensor is placed in a pre-defined location with the objective of using the least number of sensors. In random deployment, for example in an airdrop, a large number of sensors is usually deployed because of the uncertainty of their landing position on the ground. The second important aspect of a WSN is coverage. When sensors are deployed, they must cover an area that is as large as possible. Coverage is fundamental and crucial as it reflects on how well the target field will be monitored or observed by the sensors.

## 5.2 Sensor Coverage

When we deploy sensors we must keep in mind how they should be placed in order to cover a larger area. Sensors must cover a certain area or monitor certain points in a terrain. Different types of coverage have been studied in WSN [72]. They can be classified into three categories [73]: *area* coverage, *point* or *target* coverage and *barrier* coverage. In area coverage, the objective is to cover the whole monitored field. Point or target coverage addresses the issue of covering a set of discrete points or targets whose locations are known before hand. Finally, barrier coverage consists of finding a path in a sensor field that every point of this path satisfies a certain application-specified property.

The coverage of an area of targets must be complete [72]. It is such an important factor that Meguerdichian *et al.* [74] considered it as a measure of the Quality of Service (QoS) of the sensing function.

Figure 5.1 shows area coverage where a square must be covered by four circles. Figure 5.2 shows point coverage where the targets are the three dashed squares inside the big square which must all be covered. Lastly, figure 5.3 shows the path which must be fully monitored

by sensors.

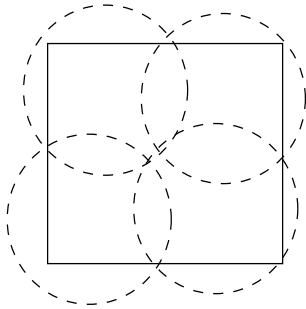


Figure 5.1: Area coverage

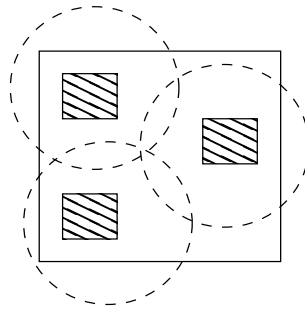


Figure 5.2: Point coverage

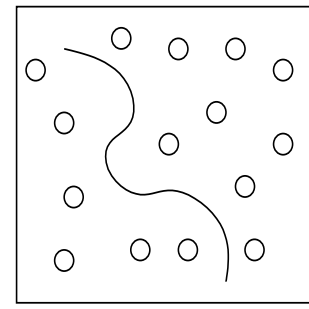


Figure 5.3: Barrier coverage

### 5.3 Sensor Design Issues

Sensors come in different types with various sizes, weights and cost restrictions which impact resource availability. They are not only limited in power but also in processing and communication capabilities. They can be organised and collaborate together in order to perform larger sensing tasks. Each sensor may have its own specifications and may be designed to perform a specific task but sensors of the same type are usually deployed together. For example, the temperature and relative humidity sensor MEP510 by Crossbow Technology Inc. is of size  $6.35\text{ cm} \times 4.13\text{ cm} \times 3.81\text{ cm}$  and of weight 89.8 grams [72].

Generally, a sensor can be in one of the following four modes or states: *transmit*, *receive*, *sleep* and *idle*. In the sleep state, the radio is turned off and in idle mode the sensor is neither receiving nor transmitting any information. The power usage for this WINS Rockwell seismic sensor for transmit, receive, idle and sleep operational modes are:  $0.7W$ ,  $0.36W$ ,  $0.34W$  and  $0.03W$  respectively. The power consumed during sensing is  $0.02W$  [73].

### 5.4 Sensor Energy Consumption

Once sensors are deployed in the field, the objective is to have them operate for as long as possible so that they can collect as much information as needed. However we can only limit the energy consumption up to a certain level using different mechanisms. Every activity

in the network consumes energy. When a node transmits, receives or is idle, its energy is depleted. Once the energy is exhausted in a node, such a node cannot do anything anymore and therefore no longer forms part of the network. Several studies have been conducted in order to seek strategies that help reduce the energy consumption in sensors [75, 76, 77, 78]. The time that a WSN takes in operation before it ceases to function is referred to as the *network lifetime*. More precisely, the network lifetime is the period from the network setup to the time that the deployed network cannot provide adequate coverage. While sensor energy is only consumed when the network is up, the consumption of such energy also depends on other factors. Therefore, to maximise the network lifetime, aspects such as the circuits of the sensor, the architecture, algorithms and protocols must be energy efficient [75]. The way sensors are connected may also reduce/increase energy consumption. Activity scheduling is used and it helps reduce the consumption of energy in sensors [73, 72]. In sensor activity scheduling, decisions are made on which sensors need to be in which states and for how long in order to attain certain network coverage requirement and a long network lifetime.

A survey on energy consumption has been written by Raghunathan *et al.* [79] provides information pertaining to energy consumption in Rockwell's WINS node and MEDUSA-II. For WINS, tuning the radio receiver increases the power consumption from 383 *mW* to 752 *mW* and 10 *mW* to 22 *mW* for MEDUSA-II. Using the transmitter increases the power consumption from 771 *mW* to 1081 *mW* for WINS and from 19 *mW* to 27 *mW* for MEDUSA-II [71].

Since the power limitation is such a constraint on WSN, it is acceptable to perform significant amounts of data processing and computation in order to reduce the amount of radio communication.

## 5.5 The Optimisation Model

### 5.5.1 Overview

Sections 5.2 and 5.4 show that it is important to maximise both coverage and lifetime in a WSN. Some optimisation models also include finding the optimum number of sensors to be deployed in a field [14]. In this thesis we present a mathematical model derived from [15] to

maximise both the coverage and lifetime. Recent operations in hostile environments such as war zones and disaster areas have demonstrated the limitations of surveillance performed by high altitude platforms such as satellites, Unmanned Aerial Vehicles (UAVs) and U2 surveillance aircraft. To be able to monitor an environment from a long distance may prove to be difficult. Thanks to technological advancements, WSNs provide an efficient way for monitoring areas remotely.

In this model we consider an hostile area where sensors cannot be deployed on the ground by humans. The deployment will be accomplished by an aircraft that will drop sensors from the air. The position of such sensors on the ground is uncertain, therefore a large number of them may be deployed but the aircraft has a limited capacity and can therefore only deploy a fixed number.

### 5.5.2 Assumptions

We assume that sensors are of the same type and the communication and sensing radii are the same and denoted by  $R_c$  and  $R_s$  respectively. All sensors can send or receive their own data or act as relay nodes. At the center of the area, we have a sink node that we refer to as the High Energy Communicating Node (HECN). All sensors send their data to the HECN which in turn sends all information via satellite to a home base station for final reporting. Only sensors which can communicate with the HECN either directly or indirectly are considered in the calculation of the objective functions. For every transmission of information, the energy is depleted by an amount that is proportional to the distance to which the information is sent. That is to say, a node consumes more energy when it is far from the node to which it is sending data. Sensors are deployed in an area that is assumed to be square and flat. A sensor can sense any activity happening within its radius  $R_s$  and it can communicate up to a maximum distance  $R_c$ .

### 5.5.3 Calculation of Objective Functions

Our aim is to maximise the network coverage and lifetime. These two objectives are in conflict. If we want to cover a large area, sensors must be far apart from each other in such a way that their individual sensing areas do not overlap. On the other hand if we want to



have maximum lifetime, we must not keep sensors too far apart. Each sensor must be close to the HECN so that it does not act as relay node but only transmits its own data. This is clearly against the idea of having a larger coverage thus showing that the two objectives are in conflict. As in any multi-objective problem, finding a solution to this problem will mean to find a set of Pareto points. Such points will give us a choice representing layouts with high coverage and moderate lifetime or *vice versa*.

We define the coverage as the union of the disks of radii  $R_s$  centered at each connected sensor normalised by the total area. Let  $C$  denote the coverage and  $A$  the total area of the disks. The network coverage is therefore given by

$$C = \left\{ \bigcup_{i=1}^n \pi R_s^2(x_i, y_i) \right\} / A \quad (5.1)$$

To find the total area covered by circles (*covered area*) we have repeatedly calculated the intersection of two circles then subtract it from their total area. There exist methods to find the intersection between two circles but finding intersection of more than two circles has to be done in a repetitive way. First of all we find the total area of all the circles and denote it by *total area*. Then, for each circle we find its intersection with any other circle with which it overlaps. We call this area *overlap area*. We add all these overlapping areas together to form what we call *total overlap area*. The covered area is then the difference between the total area and the total overlap area as expressed by

$$\text{coverage area} = \text{total area} - \sum \text{overlap area}$$

Similarly, let  $L$  denote the network lifetime and  $T_{max}$  the maximum sensor lifetime obtained by assuming that each sensor is directly connected to the HECN. We define the lifetime as the ratio of the time to first sensor failure (no more energy) and the maximum lifetime of the sensor. It is given by:

$$L = \min_{i=1 \dots n} (T_{failure,i}) / T_{max} \quad (5.2)$$

As mentioned in our assumptions, all sensors gather information at the same time and then send this information to the HECN. We call this a *sensing cycle*. During each sensing cycle, sensors can send data to the HECN either directly or through multiple hops in the network.

In the second case some sensors devote their energy to acting as relay nodes. In order to find routes from each node to the HECN, we use Dijkstra's shortest path algorithm. The cost of an outgoing edge is weighted as the inverse of the remaining energy in the node [15]. We repeat the number of sensing cycles until the energy of at least one sensor is completely drained. This number will give us the maximum sensing cycle a particular layout can perform.

#### 5.5.4 Problem Formulation

The multi-objective problem of maximising both the coverage and the lifetime can be formulated as

$$\text{Maximise } \begin{cases} C = \left\{ \bigcup_{i=1}^n \pi R_s^2(x_i, y_i) \right\} / A \\ L = \min_{i=1 \dots n} (T_{failure,i}) / T_{max} \end{cases} \quad (5.3)$$

#### 5.5.5 Algorithmic Solution

One problem with genetic algorithms is to find a proper problem encoding. Failure to do so may result in obtaining wrong answers or in solving a different problem. In this case problem encoding was not a problem. The design variables are the  $x$  and  $y$  coordinates. A chromosome is represented by the points that indicate the position of each sensor in the  $XY$  plane. In this example we consider a network consisting of ten nodes. This is not a restriction because this number can be changed as wished. However the computational complexity must be taken into account as it may increase with a large number of nodes. A typical chromosome will be

$$\text{Chromosome } [x_1, y_1, x_2, y_2, \dots, x_{10}, y_{10}]$$

For genetic algorithms, we have encoded each chromosome as a binary string consisting of 80 bits. Each set of four bits encodes an  $x$  or a  $y$  coordinate. Each eight bits therefore encodes the  $x$  and the  $y$  coordinates of one sensor. There are a fixed number of ten nodes thus making a total of eighty bits. To obtain a real value out of the binary string, we first

convert it to its decimal equivalent. If  $m$  is its binary equivalent, the corresponding real number  $r$  is given by:  $r = a + m \frac{b-a}{2^n-1}$  where  $n$  is the length of the sub-string and  $[a, b]$  is the interval in which  $r$  lies. In our case  $n = 4$ ,  $a = 0$  and  $b = 16$  because we encode each coordinate with four bits and the area considered is of length 16. For example,  $r = 0.0$  for the string 0000 and 16 for the string 1111. The real value corresponding to the string 0010 is 2.133 while 10.667 corresponds to the binary string 1010.

## 5.6 Experimental Results

### 5.6.1 Parameter Settings

We have solved the coverage problem using two algorithms, namely NSGAI [40] and Micro-GA [30]. The solutions obtained by these two algorithms are trade-off points. The two algorithms are different in their working principles and thus require different parameter settings. However we kept the same probability of mutation and crossover for both algorithms. The mutation probability was set to 0.1 while the crossover probability was set to 0.9. The number of generations was set to 125 and 300 for NSGAI and Micro-GA respectively.

To be able to use both algorithms we have made some modifications by incorporating some more features. These include the Dijkstra shortest path algorithm, programs that read the topology of the network while keeping track of all routes and sensing cycles. Each chromosome encodes a set of centres that represents a particular layout and subsequently the lifetime. Once we know the centers, we know the position of each sensor and we use Dijkstra's algorithm to find the path for each source destination pair. We repeat this for all nodes until we get the sensing cycle. The radii of communication and sensing is set to two arbitrary units.

### 5.6.2 Genetic Operators

The performance of genetic algorithms relies on its operators namely crossover, mutation and reproduction. As the evolution cycle evolves each chromosome undergoes changes

through mutation and new chromosomes are bred using crossover. Chromosomes are passed to the next generation using reproduction which mainly selects individuals based on their dominance (see section 2.3).

Mutation is used with the aim of introducing diversity and strengthening weaker individuals. This operation is performed with a small probability as shown in the sub-section above. Sensors are moved from one position to another when a chromosome is mutated. A change in one bit of a chromosome corresponds to a different position of a sensor whose bit was changed. Such a small change in a chromosome may yield a large change in the value of the objective function. For example a sensor with coordinates whose bits are represented by the two sub-strings **0100** and **1011** correspond to the real values 4.2667 and 11.733. After mutation the bits indicated in bold in two sub-strings are changed and the new sub-strings become **0101** and **1001** with corresponding real values 5.33 and 9.60. This sensor that was located at (4.2667, 11.733) will move after mutation to another position (5.33, 9.60).

Once a sensor has moved, it may or may not be part of the shortest path and the sensing cycle will also be affected. In some cases it may be moved far away and become disconnected.

The crossover operation combines two chromosomes to produce a single chromosome. The idea of combining two chromosomes is to get a new chromosome which is fitter and in our case we adjust the positions of the sensors so as to get a layout with a larger coverage and lifetime. Layouts that do not produce a larger lifetime and coverage are eliminated by the reproduction operator. Since we are solving a multi-objective problem, chromosome dominance is evaluated and dominated chromosomes are discarded according to the algorithm working principles. The evolution cycle will continue until the maximum number of generation is attained. Non-dominated chromosomes will constitute the solution to the problem and they will be evaluated using the two objectives in order to obtain the corresponding coverage and lifetime.

### 5.6.3 Pareto Points Analysis

Figures 5.4 and 5.5 show the evolution cycles as a function of the number of Pareto points of Micro-GA and NSGAI respectively. The two graphs show that both algorithms could hardly find a good number of Pareto points. Micro-GA could only find 16 points whereas NSGAI managed to find only 32 as seen from the highest points on both graphs. Both graphs fluctuate, with the number of Pareto points found at each generation decreasing or increasing. The general behaviour of the graphs is however increasing although they decrease at certain generations with NSGAI strictly increasing between 40<sup>th</sup> and 50<sup>th</sup> generations. Micro-GA shows a similar increasing behaviour but the number of Pareto points found remains the same between generations before changing thus having like a stepwise continuous function.

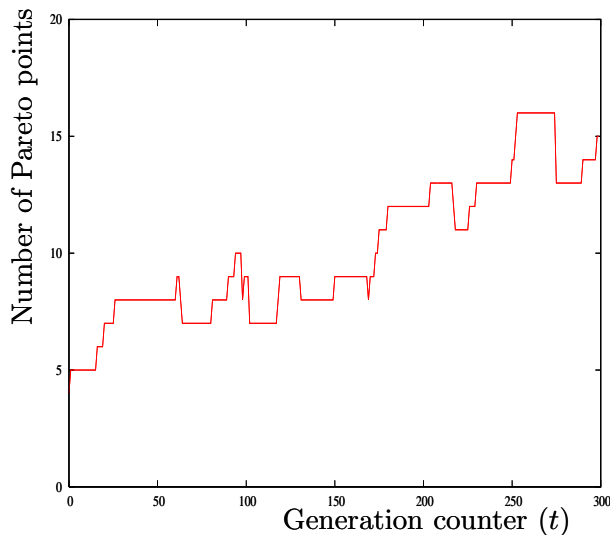


Figure 5.4: Generation counter vs number of Pareto points for Micro-GA

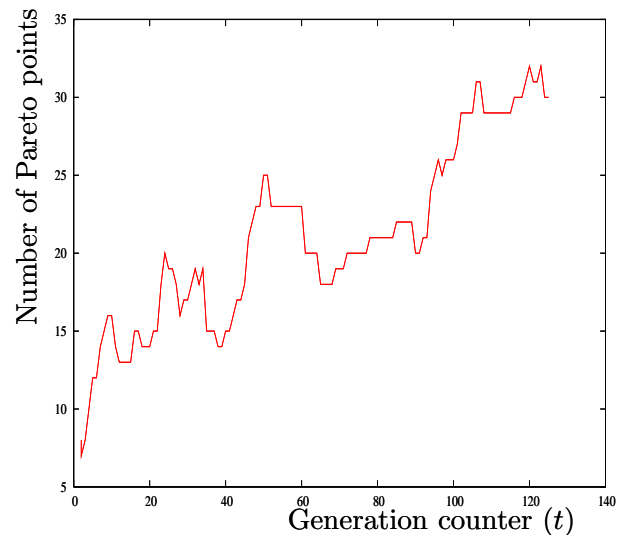


Figure 5.5: Generation counter vs number of Pareto points for NSGAI

Figure 5.6 and 5.7 show the Pareto fronts for both algorithms. Every solution point found represents a certain layout with its corresponding lifetime. The larger the coverage the smaller the corresponding lifetime will be and *vice versa*. That is to say, if we want to maximise the coverage we have to reduce the lifetime and that is why such points are called trade-off points. Two fundamental types of layouts were found. Sensors are either packed together around the HECN or slightly away from it (figure 5.8) or they form clusters (figure

5.9). Another interesting layout is observed where sensors are aligned to form a linear array as seen in figures 5.10(a) and 5.10(b).

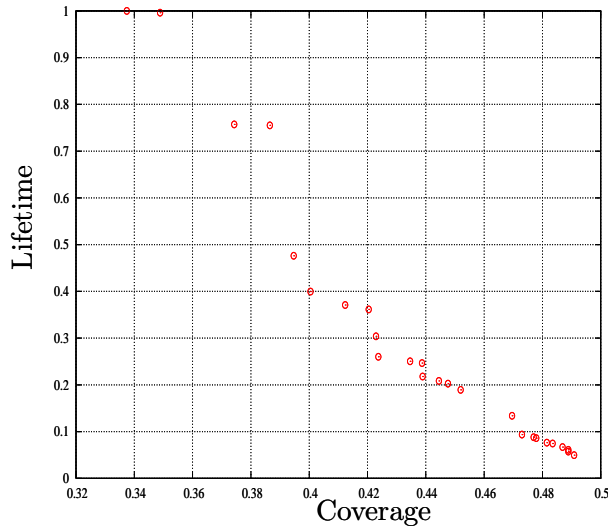


Figure 5.6: Pareto front for Micro-GA

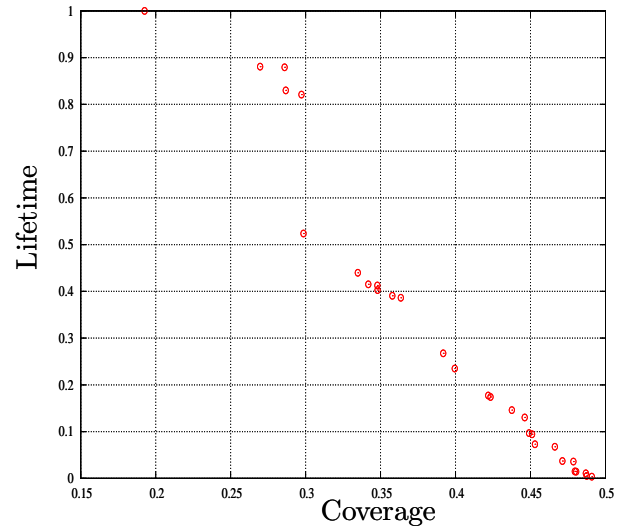


Figure 5.7: Pareto front for NS-GAII

## 5.7 Layout Analysis

The solution to an optimisation problem gives us the Pareto points which represent different combinations of coverage and lifetime. Such combinations may be suitable to different users depending on the application domain or what they want to achieve. In our experimental results, three different layouts have been found while analysing the layout of sensors. These layouts include sensors being packed together or organised into cluster groups and finally sensors being aligned into a linear array. We will briefly discuss each of the layouts below.

### Sensors Packed Together

Figure 5.8 shows sensors packed together around the HECN. This type of layout is good when we want to have a larger lifetime. It is ideal for applications that do not require a large area coverage. However when sensors are packed together, interference becomes a problem as radio signal interference may hinder proper communication between sensors.

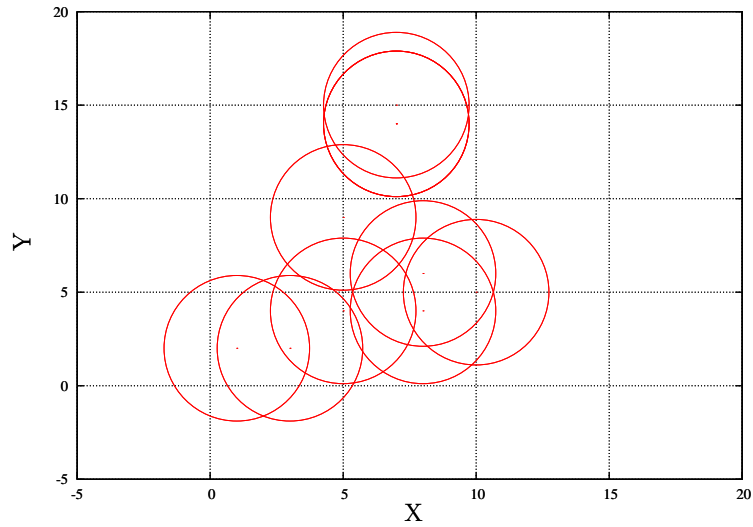
Results obtained by NSGAI as shown in figure 5.8(a), show sensors being concentrated around the HECN as opposed to results obtained by Micro-GA (see figure 5.8(b)). In figure 5.8(a), each sensor will most likely use its energy to only transmit its own data thus maximising the lifetime of the network. However the penalty we get from this is that sensors overlap and therefore cover a smaller area. Figure 5.8(b) also shows sensor packed together but covering a slightly larger area. In this case some of the sensors are not close to the HECN, hence they send their data through at least one transit node. The energy can still be conserved and at the same time a relatively larger area can be covered.

### Sensors Organised Into Groups

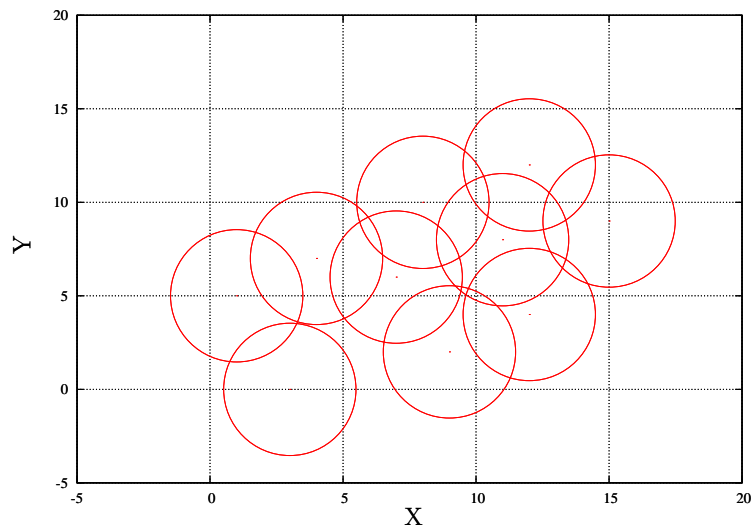
Figure 5.9 shows the sensors arranged into groups to form clusters. For Micro-GA (figure 5.9(a)) sensors are grouped into three different clusters. One group consists of four sensors and the other two consist of three sensors each. In this scenario, the sensors are somewhat far apart from each other thus trying to maximise the coverage. While trying to remain distant from each other and avoid overlap they risk leaving the area they are intended to monitor. The main problem with this situation is the energy that will be consumed for each transmission of information. Recall that energy is depleted by an amount proportional to the distance which the node is sending data. The further apart sensors are the more energy will be consumed thus shortening the network lifetime. Another undesirable effect of this is that, instead of sensors communicating directly with the HECN most of them will be relaying data. However for applications that may require a large area coverage this layout may be useful.

### Sensors Organised into a Linear Array

Figure 5.10 shows another interesting layout that we observed where the sensors are aligned into a linear array. Clearly this type of layout is bad as it does not cover a larger area although it gives a network lifetime that is relatively good. The coverage is minimised because the sensors overlap. However this layout may be of practical importance in barrier coverage where we want to monitor the whole path covered by the linear array.



(a) NSGAI algorithm for  $Coverage = 0.29$  and  $Lifetime = 0.82$



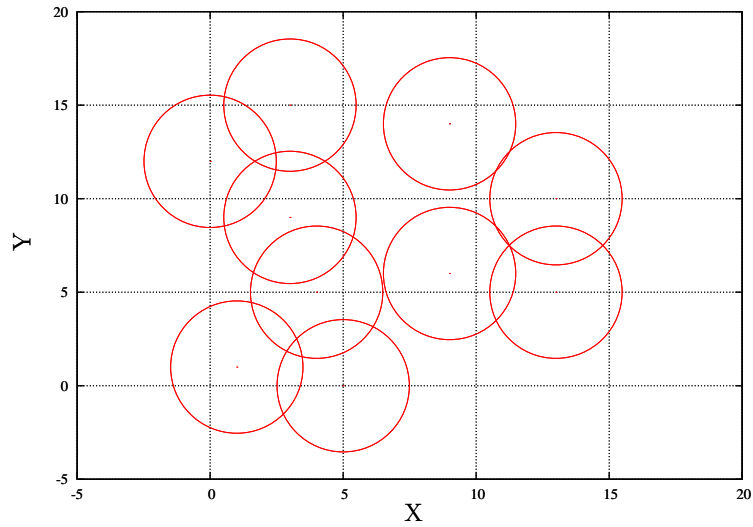
(b) Micro-GA algorithm for  $Coverage = 0.37$  and  $Lifetime = 0.55$

Figure 5.8: Sensors packed together around the HECN

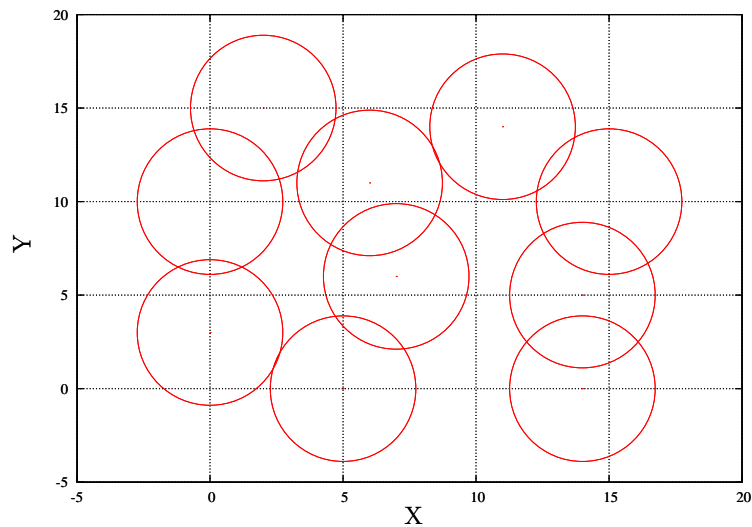
## 5.8 Analysis of the Performance of the Algorithms

The two algorithms used are fundamentally different and it is therefore reasonable that different results were obtained. As mentioned earlier, the number of Pareto points obtained was relatively small. Both algorithms found the same type of layouts except for NSGAI which found one more layout (linear array). However most of the Pareto points found for





(a) **Micro-GA** algorithm for  $Coverage = 0.43$  and  $Lifetime = 0.25$

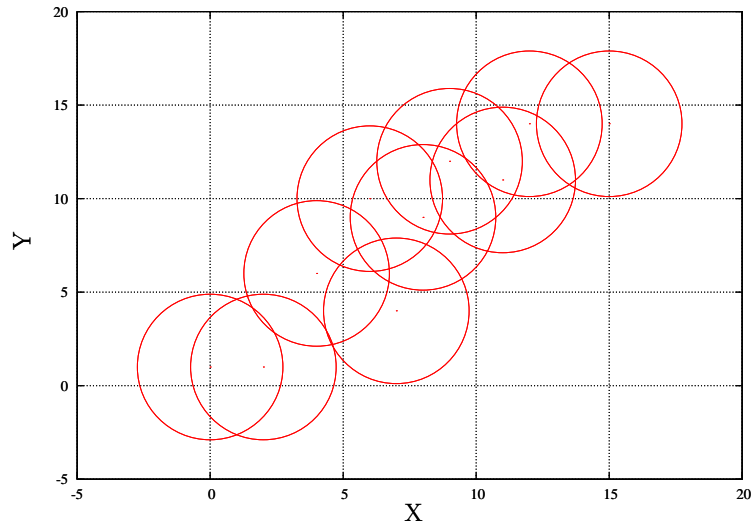


(b) **NSGAI** algorithm for  $Coverage = 0.39$  and  $Lifetime = 0.23$

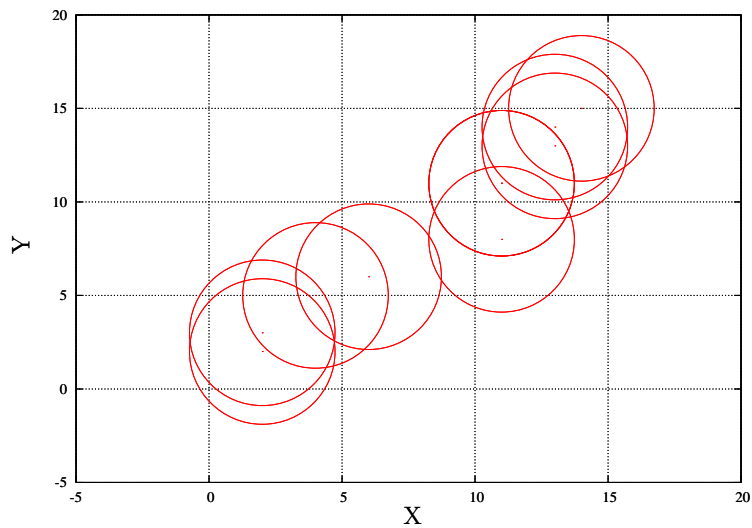
Figure 5.9: Sensors organised in a hub-and-spoke pattern

both algorithms showed two fundamental layouts: sensors packed together or organised into clusters. Damien *et al.* [15] used MOGA [22] to solve this problem and found the same two fundamental layouts as mentioned above. They also investigated the impact of sensing and communication radii. They concluded that the ratio of the sensing to the communication range can be a discrimination factor [15]. We also found one more layout besides the two fundamental ones and have used more than one multi-objective algorithm

to solve the problem thus looking at the problem from a different perspective. The Pareto points found by Micro-GA are relatively better than those found by its NSGAI counter part as shown in tables 5.1 and 5.2. For a lifetime of 1, Micro-GA found 33% coverage where NSGAI could only find 19% coverage. But one aspect of NSGAI that is worth to mention is that it found more Pareto points than Micro-GA at a lower maximum generation.



(a) linear array for  $Coverage = 0.29$  and  $Lifetime = 0.82$



(b) linear array for  $Coverage = 0.21$  and  $Lifetime = 0.83$

Figure 5.10: Sensors nodes in a linear array

Coverage	Lifetime
0.192563	1
0.26965	0.880817
0.285955	0.879347
0.286744	0.829867
0.297204	0.820937
0.298533	0.523847
0.334819	0.4397
0.341789	0.415012
0.34786	0.412543
0.347954	0.402826
0.357849	0.390535
0.363465	0.38628
0.391799	0.267623
0.39945	0.234951
0.421788	0.17733
0.423315	0.174125
0.437523	0.145971
0.4461	0.130213

Table 5.1: Sample of Pareto points found by NSGAI

Coverage	Lifetime
0.337433	1
0.348848	0.996176
0.374284	0.75717
0.386498	0.755258
0.394668	0.476099
0.400449	0.399618
0.412344	0.370937
0.420402	0.361377
0.422902	0.304015
0.423707	0.260038
0.434613	0.250478
0.43868	0.246654
0.438906	0.217973
0.444488	0.208413
0.447609	0.202677
0.451937	0.189293
0.469578	0.133843
0.472973	0.0936902

Table 5.2: Sample of Pareto points found by Micro-GA

## Chapter 6

# Conclusion and Future Work

We applied Evolutionary Multi-Objective Optimisation (EMOO) algorithms to three different types of network problems to achieve different routing optimisations: (1) efficient Routing and Wavelength Assignment (RWA) in IP backbone networks (2) optimisation of Arrayed Wave-guided Grating (AWG) Metro networks and (3) Wireless Sensor networks (WSNs) topology (layout) optimisation.

### **Efficient Routing and Wavelength Assignment (RWA) in IP backbone networks.**

Genetic optimisation algorithms were used in this thesis to achieve Routing and Wavelength Assignment (RWA) for IP backbone networks using a two-step process where (1) a set of optimal paths is found using genetic optimisation and (2) graph coloring techniques are used to assign wavelengths to these paths. The path comparison made raises the issue of the interplay between single- and multi-objective optimisation to evaluate the similarity between the set of paths found by a single objective optimisation model and the set of paths represented by some of the Pareto points. The mechanism used to extract the path sets from a Pareto front was explained in this thesis but the comparison between paths has been reserved for future research work.

## Optimisation of Arrayed Wave-guided Grating (AWG) Metro networks.

We examined the problem of maximising *throughput* and minimising *delay* in a AWG-based metropolitan WDM network using four different EMOO algorithms. These include the Non-dominated Sorting GAI [40], the Multi-Objective GA [22], the Pareto Archived Evolutionary Strategy [25] and the Micro-Genetic Algorithm for multi-objective optimisation [30]. Future work will look at various degrees of the AWG and their respective performance on the basis of throughput-delay trade-off points. Instead of looking at a single-hop AWG-based network, it would be interesting to study the performance of AWG in a multi-hop network. Hagen *et al.* [64] compared a single-hop to a multi-hop AWG-based WDM network in terms of mean hop distance and aggregate capacity. They also analysed which of the two networks type provides higher capacity for a given number of nodes. They concluded that the capacity depends on the cost ratio of fixed-tuned and tunable transceivers and on the tuning latency of the tunable transceivers. More evolutionary algorithms may also be used to solve the problem and compare their respective performance.

## Wireless Sensor networks (WSNs) Topology (layout) Optimisation.

We examined the problem of maximising the coverage and lifetime in a wireless sensor network. We used two different EMOO algorithms namely NSGAI [40] and Micro-GA [30]. We found three types of layouts two of which were also found in [15]. The model considered in this thesis does not take signal attenuation into account when sensors send data to distant nodes. It would be interesting to consider how the signal strength decreases as the distance increases. The model did not also take radio interference into account. The network lifetime is defined as the lifetime of a single sensor in the network, the network may still function even if a single node fails. However if the node's position was critical it can lead the whole network to fail. There is therefore a need to redefine the network lifetime and see how it will affect both the coverage and lifetime. The coverage might also

depend on the number of sensors deployed. However, deploying more sensors might not necessarily increase the coverage especially in the case where sensors are dropped randomly. To incorporate the number of sensors as a third objective in the model can be interesting in order to analyse its impact on lifetime and coverage.

# Bibliography

- [1] R.S Rosenberg. *Simulations of Genetic Populations With Biochemical Properties*. PhD thesis, University of Michigan, Ann Harbor, Michigan, 1967.
- [2] J.H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, 1975.
- [3] M. Gen and R. Cheng. *Genetic Algorithms & Engineering Optimization*. John Wiley & Sons, Inc., ISBN: 0-741-31531-1, first edition, 2000.
- [4] A. Abraham, L. C. Jain and R. Goldberg. *Evolutionary Multi-Objective Optimization*. Technical report, 2005. Springer-Verlag, ISBN: 1852337877.
- [5] C. A. C. Coello. Online MOOP repositories. <http://www.lania.mx/~ccoello/EMOO>.
- [6] M. Gen, L. Lin and R. Cheng. Bicriteria Network Optimization Problem Using Priority-based Genetic Algorithms. *IEEE Transactions on Electronic, Information & Systems*, 124(10):1972–1978, October 2004.
- [7] L. Yinzheng, H. Ruichun and G. Yaohuang. Faster Genetic Algorithm for Network Paths. *The Sixth International Symposium on Operations Research and its Applications (ISORA '06)*, pages 380–389, August 2006.
- [8] N. Selvanathan and W.J. Tee. A Genetic Algorithm Solution to Solve The Shortest Path Problem in OSPF and MPLS. *Malaysian Journal of Computer Science*, 16(1):58–67, 2003.
- [9] L.S Buriol, M.G.C. Resende, C.C Ribeiro and M. Thorup. A Hybrid Genetic Algorithm for the Weight Setting Problem in OSPF/IS-IS Routing. Technical Report TD-5NTN5G, Systems Research Center at AT&T Labs Research, 24 June 2003.

- 
- [10] L. Buriol, M. Resende, C.C Ribeiro and M. Thorup. A Memetic Algorithm for OSPF Routing. In *Proceedings of the 6th INFORMS Telecommunications*, pages 187–188, 2002.
- [11] N. Benerjee, V. Mehta and S. Pandey. A Genetic Algorithm Approach for Solving The Routing Wavelength Assignment Problem in WDM Networks. *International Conference on Networks (ICN'04), French Caribbean,, 2004.*
- [12] Y. Hyo-Sik, M. Maier, M. Reisslein and W.C. M. Carlyle. A Genetic Algorithm-Based Methodology for Optimizing Multiservice Convergence in a Metro WDM Network. *Journal of lightwave technology*, 21(5):1114–1133, 2003.
- [13] H. Meunier, T. El-Ghazali and P. Reininger. A Multiobjective Genetic Algorithm for Radio Network Optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation CEC00*, pages 317–324, La Jolla Marriott Hotel La Jolla, California, USA, 6-9 2000. IEEE Press.
- [14] J.B. Damien and O.L. de Weck. Multi-Objective Genetic Algorithm For the Automated Planning of A Wireless Sensor Network To Monitor a Critical Facility. In *Proceedings of the SPIE*, pages 565–575, September 2004.
- [15] B. Damien and O.L. de Weck. Layout Optimization for a Wireless Sensor Network Using a Multi-Objective Genetic Algorithm. *IEEE Trans., Vehicular Technology Conference*, 5:2466– 2470, Spring 2004.
- [16] W-H. Steeb. *The Nonlinear Workbook*. World Scientific Publishing, ISBN: 981-238-212-7, USA, second edition, 2002.
- [17] E. Mezura-Montes and C. A. C. Coello. A Numerical Comparison of Some Multiobjective-Based Techniques to Handle Constraints in Genetic Algorithms. Technical Report EVOCINV-03-2002, Evolutionary Computation Group at CINVESTAV-IPN, Mexico, D.F. 07300, September 2002.
- [18] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, ISBN: 0201157675, MA, 1989.



- 
- [19] K. Deb. Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design. In *Evolutionary Algorithms in Engineering and Computer Science*, pages 135–161. John Wiley & Sons, Ltd, Chichester, UK, 1999.
- [20] F. Ivo, Sbalzarini, S. Müller and K. Petros. Multi-Objective Optimization Using Evolutionary Algorithms. In *Proceedings of the summer Program*, 2000.
- [21] J.D Schaffer. Mutli-Objective Optimisation with Vector Evaluated Genetic Algorithms. In *Proceedings of the first international conference on genetic algorithms*, pages 93–100, 1985.
- [22] C.M. Fonseca and P.J. Fleming. Multiobjective Genetic Algorithms. *IEEE Colloquim on Genetic Algorithms for Control Systems Engineering*, (1993/130), 28 May 1993. London, UK.
- [23] J. Horn, N. Nafpliotis and D.E. Goldberg. A Nicheed Pareto Genetic Algorithm for Multiobjective Optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, volume 1, pages 82–87, Piscataway, New Jersey, 1994. IEEE Service Center.
- [24] D. Corne, J.D. Knowles and M.J. Oates. The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, J. J. Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 839–848, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.
- [25] J.D. Knowles and D. Corne. The Pareto Archived Evolution Strategy: A New Baseline Algorithm For Pareto Multi-Objective Optimisation. In Peter J. Angeline, Zbyszek Michalewicz, Marc Schoenauer, Xin Yao, and Ali Zalzal, editors, *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 98–105, Mayflower Hotel, Washington D.C., USA, 6-9 1999. IEEE Press.
- [26] N. Srinivas and K. Deb. Multiobjective Optimisation Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.

- 
- [27] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.
- [28] G.G Yen and H. Lu. Dynamic Multi-Objective Evolutionary Algorithm: Adaptive Cell-Based Rank And Density Estimation. *Evolutionary Computation, IEEE Transactions*, 7(3):253–274, June 2003.
- [29] H. Lu and G.G. Yen. Rank-Density-Based Multi-Objective Genetic Algorithm And Benchmark Test Function Study. *Evolutionary Computation, IEEE Transactions*, 7(4):325–343, August 2003.
- [30] C. A. C. Coello and T.P. Gregorio. A Micro-Genetic Algorithm for Multi-Objective Optimization. *First International Conference on Evolutionary Multi-Criterion Optimisation, Springer-Verlag. Lecture Notes in Computer Science*, (1993):126–140, 2001.
- [31] T. Bäck, D. B. Fogel and Z. Michalewicz. *Evolutionary Computation 2, Advanced Algorithms and Operators*. Institute of Physics, ISBN: 0750306653, Bristol, 2000.
- [32] C. A. C. Coello. Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems*, 1(3):129–156, 1999.
- [33] C. A. C. Coello. An updated Survey of Evolutionary Multi-Objective Optimisation Techniques: State of the Art and Future Trends. In *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 3–13, Mayflower Hotel, Washington D.C., USA, June-September 1999. IEEE Press.
- [34] H. Ishibuchi and T. Murata. A Multi-Objective Genetic Local Search Algorithm and its Application to Flowshop Scheduling. *IEEE Transactions on Systems, Man and Cybernetics*, 28(3):392–403, 1998.
- [35] H. Ishibuchi, T. Murata and H. Tanaka. A Multi-Objective Genetic Algorithm and its Application to Flowshop Scheduling. *Computers and Industrial Engineering*, 30(4):957 – 968, 1996.
- [36] R. Cheng and M. Gen. An Adaptive Superplane Approach for Multiple Objective Optimisation Problems. Technical report, Ashikaga Institute of Technology, 1998.

- 
- [37] C. M. Fonseca and P. J. Fleming. Genetic Algorithms for Multi-Objective Optimization: Formulation, Discussion and Generalisation. In *Genetic Algorithms: Proceedings of the Fifth International Conference*, pages 416–423. Morgan Kaufmann, 1993.
- [38] D. E. Goldberg and K. Deb. A Comparison Of Selection Schemes In Genetic Algorithms. *Foundation of genetic algorithms*, Morgan Kaufmann, San Mateo, California, pages 69–93, 1991.
- [39] David E. Goldberg and Jon Richardson. Genetic Algorithms With Sharing For Multimodal Function Optimization. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pages 41–49, Mahwah, NJ, USA, 1987. Lawrence Erlbaum Associates, Inc.
- [40] K. Deb, S. Agrawal, A. Pratab and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, J. J. Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.
- [41] D.E. Goldberg and J. Richardson. Genetic Algorithms With Sharing For Multimodal Function Optimization. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pages 41–49, Mahwah, NJ, USA, 1987. Lawrence Erlbaum Associates, Inc.
- [42] A. Konak, D. Coit and A. Smith. Multi-Objective Optimization Using Genetic Algorithms: A Tutorial. *Reliability Engineering & System Safety*, 91(9), September 2006.
- [43] B. Sareni and L. Krähenbühl. Fitness Sharing And Niching Methods Revisited. *Evolutionary Computation, IEEE Transactions*, 2(3), September 1998.
- [44] E. Zitzler, M. Laumanns, and L. Thiele. SPEAII: Improving The Strength Pareto Evolutionary Algorithm. Technical Report 103, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 2001.

- 
- [45] W. D. Corne, N. R. Jerram, J. D. Knowles and M. J. Oates. PESA-II: Region-Based Selection in Evolutionary Multi-Objective Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 283–290, San Francisco, California, USA, 7-11 2001. Morgan Kaufmann.
- [46] J. Rexford, B. Fortz and M. Thorup. Traffic Engineering With Traditional IP Routing Protocols. *IEEE Communications Magazine*, 40(10), October 2002.
- [47] A. E. Ozdaglar and D.P. Bertsekas. Routing and Wavelength Assignment in Optical Networks. *IEEE/ACM Transactions on Networking (TON)*, 11(2):259 – 272, April 2003.
- [48] D. Banerjee and B. Mukherjee. A practical Approach for Routing and Wavelength Assignment in Large Wavelength-Routed Optical Networks. *IEEE Journal of Selected Areas in Communications*, 14(5):903–908, 1996.
- [49] C. Law and K. Y. Siu. On-line Routing and Wavelength Assignment in Single-Hub WDM Rings. *IEEE Journal on Selected Areas in Communications*, 18(10):2111–2122, October 2000.
- [50] J. Kuri, N. Puech, M. Gagnaire and E. Dotaro. Routing and Wavelength Assignment of Scheduled Lightpath Demands in a WDM Optical Transport Network. In *Proceedings of ICOCN*, 2002. [citeseer.ist.psu.edu/article/kuri02routing.html](http://citeseer.ist.psu.edu/article/kuri02routing.html).
- [51] O. Gerstel, G. Sasaki, S. Kutten and R. Ramaswami. Worst-Case Analysis of Dynamic Wavelength Allocation in Optical Networks. *IEEE/ACM Transactions on Networking*, 7(6):833–846, 1999.
- [52] G. Rouskas. Routing and Wavelength Assignment in Optical WDM Networks. Wiley Encyclopedia of Telecommunications. John Wiley & Sons, 2001.
- [53] A. Mehrotra and M. A. Trick. A Column Generation Approach for Graph Coloring. *INFORMS Journal on Computing*, 8:344–354, 1996.
- [54] G. A. Birkan, L. J. Kennington, E.V. Olinick, A. Ortynski and G. Spiride. Practical Integrated Design Strategies For Opaque And All-Optical DWDM Networks: Optimization Models And Solutions Procedures. *Telecommunication Systems*, 31(1), January 2006.

- 
- [55] K.G. Coffman and A.M. Odlyzko. Internet Growth: Is There a Moore's Law For Data Traffic? Handbook of Massive Data Sets. Kluwer Academic Publishers, 2002.
- [56] M. Ericsson, M. G. C. Resende and P. M. Pardalos. A Genetic Algorithm For The Weight Setting Problem in OSPF routing. Technical report, AT&T Shannon Laboratory 180 Park Avenue Florham Park, NJ 07932, 2001.
- [57] M. Maier and A. Wolisz. Demonstrating The Potential off Arrayed-Waveguide Grating Based Single-Hop WDM Networks. *Optical Networks Magazine*, 5(2):75–85, September 2001.
- [58] M. Maier, M. Reisslein and A. Wolisz. Towards Efficient Packet Switching Metro WDM Networks. *Optical Network Magazine*, 3(6):44–62, November/December 2002.
- [59] S. Adar, S. Ronen and T. Aviad. WDM - Wavelength Division Multiplexing. <http://www2.rad.com/networks/1999/wdm/wdm.htm>.
- [60] B. Mukherjee. WDM Optical Communication Networks: Progress and Challenges. *IEEE Journal on Selected Areas in Communication*, 18(10):1810–1824, October 2000.
- [61] G.P. Agrawal. *Fiber-Optic Communication Systems*. John Wiley & Sons, Inc., New York, ISBN: 0-471-21571-6, third edition, 2002.
- [62] G. Keiser. *Optical Fiber Communications*. Electrical Engineering series. McGraw-hill int., ISBN: 0-07-232101-6, third edition, 2000.
- [63] Photon Technologies. Array Waveguide Gratings. [www.photeon.com/html/Press/Application\\_notes\\_AWG.pdf](http://www.photeon.com/html/Press/Application_notes_AWG.pdf).
- [64] W. Hagen, M. Maier and A. Wolisz. Comparison of a Single-Hop and Multihop AWG-based WDM Networks. pages 51–65. Kluwer Academic Publishers, 2003.
- [65] M. Maier, M. Reisslein and A. Wolisz. A Hybrid MAC Protocol for a Metro WDM Network Using Multiple Free Spectral Ranges Of An Arrayed Waveguide Grating. *Computer Networks*, 41:407–433, march 2003.
- [66] M. Maier, M. Reisslein and A. Wolisz. High-Performance Switchless WDM Network Using Free Spectral Ranges of an Arrayed Waveguide Grating. In *Terabit Optical*

- Networking: Architecture, Control and Management Issues*, volume 4213, pages 101–112, October 2000.
- [67] M. Scheutzow, M. Maier, M. Reisslein and A. Wolisz. Wavelength Reuse for Efficient Packet-Switched Transport in an AWG-Based Metro WDM Network. *OSA Journal of Lightwave Technology*, 21(6):1435–1455, June 2003.
- [68] William H. Press. *Numerical Recipes In C++ : The Art of Scientific Computing*. Cambridge University Press, ISBN: 0521750334, second edition, 2002.
- [69] C. A. C. Coello. 20 Years of Evolutionary Multi-Objective: What Has been Done and What Remains to be Done. *IEEE World Congress on Computational Intelligence*, July 2006. Vancouver, Canada.
- [70] J. Carle and D. Simplot-Ryl. Energy-Efficient Area Monitoring For Sensor Networks. *IEEE Computer networks*, 37(2):40–46, February 2004.
- [71] J. Suomela. Relay Placement In Sensor Networks. Master’s thesis, MSc Thesis, University of Helsinki, Department of Computer Science, October 2005. <http://ethesis.helsinki.fi/julkaisut/mat/tieto/pg/suomela/>.
- [72] W. Bang. A Survey on Coverage Problems in Wireless Sensor Networks. Technical report, Nanyang Technological University, Singapore, 2004.
- [73] J. Wu and C. Mihaela. Coverage in Wireless Sensor Networks. *CRC Press*, 2004. M. Ilyas and I. Maghgoub (editors), ISBN:0-8493-1968-4.
- [74] S. Megeurdichian, F. Koushafar, M. Potkonjak and N. Srivastava. Coverage Problems In Wireless Ad-Hoc sensor Networks. In *INFOCOM*, volume 3, pages 1380–1387, 2001.
- [75] A. Sinha and A. Chandrakasan. Dynamic Power Management in Wireless Sensor Networks. *IEEE Design & Test of Computers*, 18(2):62–74, March-April 2001.
- [76] P.K. Sungha, V. Ingrid, C.L Bo-Cheng and D.W David . Reducing Radio Energy Consumption of Key Management Protocols for Wireless Sensor Networks. In *ISLPED '04: Proceedings of the 2004 international symposium on Low power electronics and design*, pages 351–356, New York, NY, USA, 2004.

- 
- [77] S. Baek, G. de Veciana and X. Su. Minimizing Energy Consumption in Large-Scale Sensor Networks Through Distributed Data Compression and Hierarchical Aggregation. Technical report, Wireless Networking and Communication Group, University of Texas, Austin, TX, 2004.
- [78] K. Schwieger, H. Nuzskowski and G. Fettweis. Analysis of Node Energy Consumption in Sensor Networks. *European Workshop on Sensor Networks*, January 2004.
- [79] V. Raghunathan, C. Schurgers, S. Park and M. Srivastava. Energy Aware Wireless Microsensor Networks. *IEEE Signal Processing Magazine*, 19:40–50, March 2002.
- [80] B. Fortz and M. Thorup. Internet Traffic Engineering by Optimizing OSPF Weights. *IEEE INFOCOM*, pages 519–528, March 2000.
- [81] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlin, ISBN: 3-540-60676-9, third edition.
- [82] W. Banzhaf, P. Nordin, E. R. Keller and F.D. Francone. *Genetic Programming An Introduction*. Morgan Kaufman Publishers, ISBN: 3920993586.
- [83] W. B. Langdon and R. Poli. *Foundations of Genetic Programming*. Springer-Verlag, ISBN: 3540424512, Berlin, Germany, 2002.
- [84] D. A. Coley. *An Introduction to Genetic Algorithms for Scientists and Engineers*. World Scientific, Singapore, ISBN: 9810236026, 2005.
- [85] W. Kocay and D. L. Kreher. *Graphs, Algorithms and Optimization*. Chapman and Hall/CRC, ISBN: 1584883960, USA, 2005.
- [86] C. A. C. Coello. A Short Tutorial on Evolutionary Multi-Objective Optimization. *First International Conference on Evolutionary Multi-Criterion Optimization, Springer-Verlag. Lecture Notes in Computer Science*, 1993, 2003.
- [87] C. M. Fonseca and P.J. Fleming. An Overview Of evolutionary Algorithms In Multi-Objective Optimization. *Evolutionary Computation*, 3(1):1–16, 1995.
- [88] K. Deb. Multi-Objective Evolutionary Algorithms: Introducing Bias Among Pareto-Optimal Solutions. pages 263–292. Springer-Verlag, Berlin, ISBN: 3540433309, 2003.



- 
- [89] J. Lohn, W. Kraus and G. Haith. Comparing A CoEvolutionary Genetic Algorithm For Multi-Objective Optimisation. In *Proceedings of 2002 Congress on Evolutionary Computation (CEC'02)*, pages 1157–1162, Piscataway, NJ, 2002. IEEE Press.
- [90] E. Zitzler, K. Deb, and L. Thiele. Comparison Of Multi-Objective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, 2000.
- [91] E. Zitzler. *Evolutionary Algorithms For Multi-Objective Optimization: Methods And Applications*. PhD thesis, Zurich, Switzerland, December 1999.
- [92] F. Jiménez and J.L Verdegay. Constrained Multi-Objective Optimization By Evolutionary Algorithms. <http://citeseer.ist.psu.edu/368326.html>.
- [93] T. Murata and H. Ishibuchi. MOGA: Multi-Objective Genetic Algorithms. In *Proc. of the 1995 IEEE international conference on evolutionary computation*, Perth, WA. Australia, 29 Nov.-1 Dec. 1996.
- [94] J. Horn and N. Nafpliotis. Multi-Objective Optimization Using The Niched Pareto Genetic Algorithm. Technical Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1993.
- [95] P. Hajela and C. Lin. Genetic Search Strategies In Multicriterion Optimal Design. *Struct Optimization*, 4(2):99–107, June 1992.
- [96] C.A. Wook and R.S. Ramakrishna. Elitism-Based Compact Genetic Algorithms. *Evolutionary Computation, IEEE Transactions*, 7(4), August 2003.
- [97] H. Takahashi, K. Oda, H. Toba and Y. Inoue. Transmission Characteristics Of Arrayed Waveguide N x N Wavelength Multiplexer. *Journal of Lightwave Technology*, 13(3):447–455, march 1995.
- [98] E. Rigoni and S. Poles. NBI and MOGA-II, Two Complementary Algorithms For Multi-Objective Optimizations. In Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Ralph E. Steuer, editors, *Practical Approaches to Multi-Objective Optimization*, Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2005.



- 
- [99] C.A Brackett. Dense Wavelength Division Multiplexing Networks: Principles And Applications. *IEE Journal on Selected Areas in Communications*, 8(6):948–964, August 1990.
- [100] B. Mukherjee. *Optical Communication Networks*. McGraw-Hill, ISBN: 0-07-044435-8, July 1997.
- [101] B. Ramamuethy and B. Mukherjee. Wavelength Conversion in WDM Networks. *IEE Journal of Selected Areas in Communications*, 16:1061–1073, September 1996.
- [102] H. Karl and A. Willig. A Short Survey Of Wireless Sensor Networks. Technical report, Telecommunication Networks Group, Technische Universität, Berlin, 2003.
- [103] R. Ramaswami and K. N. Sivarajan. Routing and Wavelength Assignment in All-Optical Networks. *IEEE/ACM Transactions on Networking*, 3(5):489–500, 1995.
- [104] O. Gerstel and S.Kutten. Dynamic Wavelength Allocation in All-Optical Ring Networks. In *ICC (1)*, pages 432–436, 1997.