Generation of Golomb Ruler Sequences and Optimization Using Genetic Algorithm

A Thesis

Submitted in the partial fulfillment of requirements for the award of the degree of

Master of Engineering
In
Electronics and Communication

by

Shobhika
Regn. No. 8024129

Under the guidance of

Dr. R. S. Kaler
Head of Department
Deptt. of Electronics & Commn. Engg.
TIET, Patiala

Dr. Shakti Kumar
Additional Director
Haryana Engineering College
Jagadhri

Department of Electronics & Communication Engineering
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
(Deemed University)
PATIALA-147004

(2005)

CERTIFICATE

This is to certify that thesis entitled "**Generation of Golomb ruler Sequences and Optimization using Genetic Algorithm**" submitted by **SHOBHIKA Regn. No. 8024129** to **Thapar Institute of Engineering and Technology (Deemed University), Patiala (Punjab)** in partial fulfillment of the requirements for the award of degree of **Master of Engineering in Electronics and Communication Engineering,** is a bonafide research work carried out by her under our supervision and guidance and no part of this thesis has been submitted to any other university or institute for award of any degree.

(Dr. R. S. Kaler)
Head of Deptt. & Guide
Department of Electronics and
Communication Engineering
T.I.E.T. Patiala

(Dr. Shakti Kumar)
Additional Director & Guide
Haryana Engineering College
Jagadhri

(Countersigned by)

(Dr. R.S.Kaler)
Professor & Head
Department of Electronics and
Communication. Engineering.
T.I.E.T. Patiala.

(Dr. D.S.Bawa)
Dean of Academic Affairs
T.I.E.T. Patiala.

ACKNOWLEDGEMENT

The real sprit of achieving a goal is through the way of excellence and austerous discipline. I would have never succeeded in completing my task without the cooperation, encouragement and help provided to me by various personalities.

With deep sense of gratitude I express my sincere thanks to my esteemed and worthy supervisors, **Dr. R. S. Kaler**, Professor and Head , Department of Electronics and Communication Engineering and **Dr. Shakti Kumar**, Additional Director, Haryana Engineering College, Jagadhri, for their valuable guidance in carrying out this work under their effective supervision, encouragement, enlightenment and cooperation. I am indebted to them for their moral support at every step. I am also thankful to all the staff members of the Electronics & Communication Engineering Department for their full cooperation and help.

I take pride of my self in being a part of a great family whose everlasting desire, sacrifice, and help made it possible to tide over the numerous problems which so ever came up during the study. I am grateful to my sister for her guidance, constant encouragement and affectionate blessings without which it would have not been possible for me to complete my studies.

Above all I render my gratitude to the ALMIGHTY who bestowed self-confidence, ability and strength in me to complete this work.

Shobhika

Place: T.I.E.T., Patiala (SHOBHIKA)

ABSTRACT

The attractiveness of lightwave communications is the ability of silica-optical fibers to carry large amounts of information over long repeaterless spans. To utilize the available bandwidth, numerous channels at different wavelengths can be multiplexed on the same fiber. To increase system margins, higher transmitter powers or lower fiber losses are required. All these attempts to fully utilize the capabilities of silica fibers will ultimately be limited by nonlinear interactions between the information bearing lightwaves and the transmission medium. These optical nonlinearities can lead to interference, distortion and excess attenuation of the optical signals, resulting in system degradations. One of these nonlinearities is the four-wave mixing whereby two or more optical waves at different wavelengths mix to produce new optical waves at other wavelengths and generate interfering signals for other channels. This will degrade both detection and heterodyne systems.

Four-wave mixing (FWM) is a nonlinear process that occurs in materials such as silica with symmetric molecular structures and leads to generation of power at new optical frequencies from two or more co-propagating waves. The most significant negative effects of FWM are parametric gain and crosstalk in long haul, multi-channel Wavelength Division Multiplexed (WDM) systems. FWM efficiency is maximized at the zero dispersion wavelengths and for closely spaced input waves. Of particular concern is the case of WDM channels that are equally spaced in frequency as any FWM products will fall on frequencies used by other channels. By using unequal channel separations, the products of FWM can be made not to fall on allocated frequencies.

In an attempt to reduce FWM effect in WDM systems, many unequally spaced channel allocation methods were proposed. However, they resulted in increase of bandwidth requirement compared to equally spaced channel allocation. This is due to the constraint of the minimum channel spacing between each channel and that the difference in the channel spacing between any two channels is assigned to be distinct. As the number of

channels increases, the bandwidth for the unequally spaced channel allocation methods increases proportionately.

In mathematics, the term 'Golomb Ruler' refers to a set of non-negative integers such that no distinct pairs of numbers from the set have the same difference. Since the difference between any two numbers is distinct, the new FWM frequencies generated would not fall into the one already assigned for the carrier channels. However, using conventional Golomb Rulers was not efficient as the operating bandwidth would be increased significantly as the existing methods. An Optimal Golomb Ruler (OGR) is the shortest Golomb Ruler possible for a given number of marks. Therefore, applying OGR to the channel allocation problem, it was possible to achieve the smallest distinct number to be used for the channel allocation. The OGRs were used to allocate the channels so as to restrict the expansion of the bandwidth.

The Golomb Rulers have to be optimized to form OGRs using an optimization technique. There are several optimization techniques, the most common for engineering problems being stimulated annealing, evolution strategies and genetic algorithms. Simulated annealing probabilistically generates a sequence of states based on a cooling schedule to ultimately converge to the global optimum. Evolution strategies use mutations as search mechanisms and selection to direct the search toward the prospective regions in the search space. Genetic algorithms generate a sequence of populations by using a selection mechanism, and use crossover and mutation as search mechanisms. Thus, the genetic algorithms provide an alternative to traditional optimization techniques by using directed random searches to locate optimal solutions in complex landscapes.

The GA is a stochastic global search method that mimics the metaphor of natural biological evolution. GAs operate on a population of potential solutions applying the principle of survival of the fittest to produce better and better approximations to a solution. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptation.

In the present study, first of all a parametric run on fiber dispersion is performed to show dependency of four wave mixing on dispersion value. For operation near the zero dispersion wavelength, the FWM efficiency is maximized and as the dispersion increases, the power of the FWM products falls, that is, FWM effect decreases with increasing dispersion and vice-versa. Thus, for WDM systems deploying dispersion-shifted fibers to compensate for the dispersion induced in the fiber, it is a must to reduce FWM effect. In this concern, two algorithms for the generation of Golomb Ruler sequences are described. In the first algorithm, the modified extended quadratic congruence (EQC) sequences are generated and the second is the algorithm based on searching. These algorithms produce Golomb Rulers for any number of marks (channels). For both the algorithms, various examples of rulers for different number of marks are provided and bandwidth expansion factors are calculated that gives the factor of the bandwidth expanded with the use of unequal channel allocation in comparison with that of equal channel allocation. The total slot bandwidth and time requirements are calculated for different number of marks and compared for the two algorithms. It is seen that the bandwidth requirements for the search algorithm are less and in addition, the time required for the generation of sequences in search algorithm remains constant irrespective of the number of marks. Both these algorithm needed large bandwidth requirements. As the number of marks increases, the bandwidth increases exponentially. Optimization of the rulers means to have the short rulers, that is, the length of the rulers is minimized so as to conserve the bandwidth. Any optimization technique could be used here but for the NP-complete problems, like the Golomb Ruler, genetic algorithms are the best optimization technique to be used. Thus here the genetic algorithms are used to produce Optimal Golomb Rulers. The length of the rulers is the objective function to be minimized. As the number of generations increases, the length of the ruler and thus the bandwidth occupied tends to decrease, thus the rulers become more and more optimized, until no further improvements are there. This is the point where we get the optimal, or the *shortest* ruler.

It is seen that by using genetic algorithms, bandwidth requirements become nearly one-third of that of the previous conventional algorithms, for the same number of channels.

The physical significance of this work is seen as the reduction of the four-wave mixing effect in WDM systems. As the difference between every pair of channels is unique, the additional frequencies produced by the FWM process do not tend to fall upon the already assigned frequencies. Hence the crosstalk, which is the main concern in WDM systems, is minimized enhancing the efficiency of the overall optical system.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DSF	Dispersion Shifted Fiber
EDFA	Erbium-doped Fiber Amplifier
EQC	Extended Quadratic Congruence
FWM	Four Wave Mixing
GA	Genetic Algorithm
OGR	Optimal Golomb Ruler
SBS	Stimulated Brillouin Scattering
SOA	Semiconductor Optical Amplifier
SPM	Self Phase Modulation
SRS	Stimulated Raman Scattering
WDM	Wavelength Division Multiplexing
XPM	Cross Phase Modulation

CHAPTER –1
INTRODUCTION

The concept of 'Golomb rulers' was first introduced by W. C. Babcock in 1952. Today the rulers are named after Solomon W. Golomb, a professor of Mathematics and Electrical Engineering at the University of Southern California.

A Golomb ruler is defined as a ruler that has marks at integer locations such that the distance between any two marks of the ruler is unique. Normally the first mark of the ruler is set on position 0 and the position of the right-most mark is called the length of the ruler. Most interesting are rulers having smallest possible length n for a given number m of marks. Such rulers are called 'optimum Golomb rulers' and they are used in a variety of real-world applications such as Communications and Radio astronomy, X-Ray Crystallography and Self-Orthogonal Codes. If each of the distances 1, 2, ..., n of a Golomb ruler of length n is measured exactly once the ruler is called perfect.

In optical communication systems deploying WDM to increase the system capacity, the Golomb Rulers are used for unequal channel allocation in order to reduce the four-wave mixing effects.

1.1 Effects of Four Wave Mixing on WDM systems

Very-high-capacity, long-haul optical communication systems can be designed by wavelength division multiplexing of high-bit-rate channels and by using erbium-doped fiber amplifier to periodically compensate the fiber loss. WDM, which allows information at various channels to be transmitted in different wavelengths, fully exploits the vast bandwidth provided by optical fiber. EDFAs are used to compensate signal attenuation in long-distance transmission without requiring bottleneck-prone optoelectronic/electro-optic conversions. Moreover, to allow high-data-rate transmission over a long distance, DSF is deployed to lessen the deleterious effects of chromatic dispersion. On the other hand, with all these attempts to exploit the capabilities of fiber and to maintain high-bit-rate and long-haul transmission, undesirable nonlinear interactions are created and accumulated as the multiplexed optical signals propagating via long length of fiber. These optical nonlinearities, such as stimulated Raman scattering, stimulated Brillouin scattering, cross-phase modulation, and four-wave mixing, can lead to interference, distortion, and excess attenuation of optical signals, resulting in system degradation and imposing limits on the achievable performance of the system.

In particular, four-wave mixing is analogous to inter-modulation distortion when optical signals with two or three frequencies (or wavelengths) interact through the third-order electric susceptibility of optical fiber. FWM generates new optical signals at other frequencies (or wavelengths) and creates signal depletion to pre-assigned channels in WDM systems. The efficiency of FWM depends on both channel spacing and fiber dispersion. Greater group velocity mismatch between the original and FWM signals is caused by fiber dispersion or larger channel spacing. As a result, with the use of DSF in high-capacity, long-haul, repeater-less WDM systems, the efficiency of FWM increases and FWM crosstalk becomes most likely the dominant nonlinear effect.

1.2 Role of Golomb Rulers

In conventional WDM systems, channels are usually assigned with center frequencies (or wavelengths) equally spaced from each other. Therefore, solely increasing channel spacing, which can only reduce the chance and magnitude of the spectral sidebands of unwanted FWM signals entering the pre-assigned channels, cannot prevent the FWM problem. There is still very high probability that FWM signals may fall into the WDM channels, resulting in severe crosstalk.

In WDM systems with equally spaced channels all the product terms generated by FWM in the bandwidth of the system fall at the channel frequencies, giving rise to crosstalk. The crosstalk on the "1" bits is enhanced by the coherent interference between mixing waves and signal at the detector. All WDM systems for which FWM crosstalk is the main source of performance degradation can be substantially improved if FWM generation at the channel frequencies is avoided. It is therefore important to develop algorithms to allocate the channel frequencies in order to minimize the effect of FWM. If the frequency separation of any two channels of a WDM system is different from that of any other pair of channels, no FWM waves will be generated at any of the channel frequencies, thereby suppressing FWM crosstalk. A design methodology of channel spacings is presented to satisfy the above requirement. The method is a generalization of what had been proposed in the 1950's to reduce the effect of 3rd-order inter-modulation interference in radio systems. The use of proper unequal channel spacing keeps FWM waves from coherently interfering with the signals.

In mathematics, the term 'Golomb Ruler' refers to a set of non-negative integers such that no distinct pairs of numbers from the set have the same difference. Since the difference between any two numbers is distinct, the new FWM frequencies generated would not fall into the one already assigned for the carrier channels.

In an attempt to reduce FWM effect in WDM systems, many unequally spaced channel allocation methods were proposed. However, they resulted in increase of bandwidth requirement compared to equally spaced channel allocation. This is due to the constraint of the minimum channel spacing between each channel and that the difference in the channel spacing between any two channels is assigned to be distinct. As the number of channels increases, the bandwidth for the unequally spaced channel allocation methods increases in proportion. A fractional bandwidth channel allocation algorithm is designed taking into consideration the concept of Optimal Golomb Ruler. This is a novel method for channel allocation to reduce the FWM effect so as to improve WDM performance without inducing additional cost. This proposed technique allows the gradual computation of system performance to result in an optimal point where degradation caused by inter-channel interference and FWM is minimal.

1.3 Need of Genetic Algorithms

Golomb Rulers represents a class of problems known as NP-complete. The exhaustive search of such problems is impossible for higher order models. As another mark is added to the ruler, the time required to search the permutations and to test the ruler grows exponentially. The genetic algorithms find good solutions to such type of problems. For Golomb rulers, the genetic algorithms help in removing unnecessary permutations and provide *good* as well as *short* rulers such that both the time and cost parameters are reduced, thus providing the optimum solutions for channel allocation.

1.4 Objective of Thesis

The objectives of thesis are:

- To study the effect of dispersion on four wave mixing using Optsim.
- To generate the Golomb Ruler sequences for unequal channel allocation using EQC construction and search algorithm.
- To compare the performance of two algorithms in terms of bandwidth and time.
- To optimize the Golomb Ruler sequences using genetic algorithm and study the effects of variations of generations on total and average bandwidth.
- To compare the bandwidths achieved by the use of genetic algorithms with that achieved by the two previous conventional algorithms.

1.5 Organization of Thesis

After giving introduction in chapter 1, Chapter 2 provides the literature review of all the constituents of the succeeding chapters - nonlinear fiber optics, the predominating fiber nonlinearity called four-wave mixing, the Golomb rulers and the optimization algorithms namely the genetic algorithms.

Chapter 3 provides the simulation results of four-wave mixing using Optsim, the Optical Simulation software. The dispersion and phase matching effects are described and the effect of dispersion on the FWM is illustrated.

In chapter 4, the problem of unequal channel allocation is formulated and two algorithms are explained to generate the Golomb Ruler sequences, along with the constraints and bounds. The simulation sequences and results are given, along with the comparison of the two algorithms.

Chapter 5 describes the optimization of Golomb Ruler sequences using genetic algorithms. The steps for simulation are discussed and the simulation results are provided. An introduction to the Matlab GA toolbox is also given.

In chapter 6, the conclusions and the future scopes of the work are discussed.

Appendix A gives the list of optimum rulers.
Appendix B and C contain the MATLAB codes for simulating the Golomb Ruler algorithms and the genetic algorithm respectively.

CHAPTER –2
LITERATURE REVIEW

2.1 Nonlinear Fiber Optics

The response of any dielectric to light becomes nonlinear for intense electromagnetic fields, and optical fibers are no exception. Even though silica is intrinsically not a highly nonlinear material, the waveguide geometry that confines light to a small cross section over long fiber lengths makes nonlinear effects quite important in the design of modern lightwave systems.

2.1.1 Historical Perspective

The basic principle of total internal reflection, responsible for the guiding of light in optical fibers, is known from the nineteenth century. Although uncladded glass fibers were fabricated in the 1920s, the field of fiber optics was not born until the 1950s when the use of a cladding layer led to considerable improvement in the fiber characteristics. The field developed rapidly, mainly for the purpose of image transmission through a bundle of glass fibers. These early fibers were extremely lossy (typical loss about 1000 dB/km). However, the situation changed drastically in 1970 when the loss of silica fibers was reduced to about 20 dB/km. Further progress in the fabrication technology resulted by 1979 in a loss of about 0.2 dB/km near the 1.55 m wavelength, a loss level limited mainly by the fundamental process of Rayleigh scattering.

The availability of such low-loss fibers has led not only to a revolution in the field of optical fiber communications but also to the advent of a new field of nonlinear fiber optics. Stimulated Raman and Brillion scatterings in single-mode fibers were studied as early as 1972 both experimentally and theoretically. This work stimulated the study of other nonlinear phenomena such as four wave mixing and self phase modulation. An important contribution was made in 1973 when it was suggested that optical fibers can support soliton-like pulses as a result of interplay between the dispersive and nonlinear effects. Optical solitons were later observed and have led to a number of advances to the generation and control of ultra short optical pulses. An equally important development is related to the use of optical fibers for pulse compression, pulses as short as 6 fs have been generated using fiber-based nonlinear compression techniques. The phenomenon of cross-phase modulation, occurring when two pulses co-propagate simultaneously inside the fiber, has attracted considerable attenuation. The field of nonlinear fiber optics has progressed enormously over the last 15 years or so. It has led to a number of advances important from the fundamental as well as technological point of view.

2.1.2 Introduction

On a fundamental level, the origin of nonlinear response is related to a harmonic motion of bound electrons under the influence of an applied field. As a result, the induced polarization P from the electric dipoles is not linear in the electric field E, but satisfies the more general relation [1]

$$P = \varepsilon_0 \left(\chi^{(1)} \cdot E + \chi^{(2)} : EE + \chi^{(3)} \vdots EEE + \ldots \right) \qquad (2.1)$$

where ε_0 is the vacuum permittivity and $\chi^{(j)}$ (j = 1, 2, ……) is jth order susceptibility. To account for the light polarization effects, $\chi^{(j)}$ is a tensor of rank j + 1. The linear susceptibility $\chi^{(1)}$ represents the dominant contribution to P. Its effects are included through the refractive index n and the attenuation coefficient α. The second order susceptibility $\chi^{(2)}$ is responsible for such nonlinear effects as second-harmonic generation and sum-frequency generation. However, it is nonzero only for media that lacks inversion symmetry at the molecular level. Since SiO_2 is a symmetric molecule, $\chi^{(2)}$ vanishes for silica glasses. As a result, optical fibers do not normally exhibit second-order nonlinear effects. Nonetheless, the electric-quadrupole and magnetic-dipole moments can generate weak second-order nonlinear effects. The dopants inside the fiber core can also contribute to second-harmonic generation under certain conditions. The lowest-order nonlinear effects in optical fibers originate from the third-order susceptibility $\chi^{(3)}$ which is responsible for phenomena such as third-harmonic generation, four wave mixing, and nonlinear refraction. However, unless special efforts are made to achieve phase matching, the nonlinear processes which evolve the generation of new frequencies (e.g. third-harmonic generation or four-wave mixing) are not efficient in optical fibers. Most of the nonlinear effects in optical fibers therefore originate from nonlinear refraction, a phenomenon that refers to the intensity dependence of the refractive index resulting from the contribution of $\chi^{(3)}$. This intensity dependence of the refractive index leads to a large number of nonlinear effects, such as self-phase modulation and cross-phase modulation.

2.1.3 Stimulated Light Scattering

The Rayleigh scattering is an example of elastic scattering for which the frequency (or the photon energy) of scattered light remains unchanged. By contrast, the frequency of scattered light is shifted downward during inelastic scattering. Two examples of inelastic scattering are *Raman scattering* and *Brillouin scattering* [2]. Both of them can be understood as scattering of a photon to a lower energy photon such that the energy difference appears in the form of a photon. The main difference between the two is that optical photons participate in Raman scattering, whereas acoustic photons participate in Brillouin scattering. Both scattering processes result in a loss of power at the incident frequency. However, their scattering cross sections are sufficiently small that loss is negligible at low power levels. But at high power levels, the nonlinear phenomena of stimulated Raman scattering and stimulated Brillouin scattering become important. The intensity of the scattered light in both the cases grows exponentially once the incident power exceeds a threshold value. Even though SRS and SBS are quite similar in their origin, different dispersion relations for acoustic and optical phonons lead to the following differences between the two in single-mode fibers [1]: (1) SBS occurs only in the backward direction whereas SRS can occur in both directions. (2) The scattered light is shifted in frequency by about 10 GHz for SBS but by 13 THz for SRS (this shift is called the Stokes shift); and (3) The Brillouin gain spectrum is extremely narrow (bandwidth less than 100 MHz) compared with the Raman-gain spectrum that extends over 20-30 THz.

2.1.4 Nonlinear Phase Modulation

All materials behave nonlinearly at high intensities and their refractive index increases with intensity. The physical origin of this effect lies in harmonic response of electrons to optical fields, resulting in a nonlinear susceptibility [2]. The refractive index n of many optical materials have a weak dependence on optical intensity I (equal to the optical power per effective area in the fiber) given by [1]

$$n = n_0 + n_2 I = n_0 + n_2 \frac{P}{A_{eff}} \tag{2.2}$$

where n_0 is the ordinary refractive index of the material and n_2 is the nonlinear index coefficient, P is the optical power and A_{eff} is the effective mode area. The numerical value of n_2 is about 2.6×10^{-20} m^2/W for silica fibers and varies somewhat with dopants used inside the core. Because of this relatively small value, the nonlinear part of the refractive index is quite small ($<10^{-12}$ at a power level of 1mW). The nonlinearity in the refractive index is known as the *Kerr nonlinearity*. This nonlinearity produces a carrier-induced phase modulation of the propagating signal, which is called the *Kerr effect*. Nevertheless, it affects modern lightwave systems considerably because of long fiber lengths [3]. In particular, it leads to the phenomena of self and cross-phase modulations.

(a) Self-Phase Modulation

An interesting manifestation of the intensity dependence of the refractive index in nonlinear media occurs through self-phase modulation, a phenomenon that leads to spectral broadening of optical pulses. SPM is the temporal analog of self-focusing. If we use the first-order perturbation theory to see how fiber modes are affected by the nonlinear term in equation 2.2, we find that the mode shape does not change but the propagation constant becomes power dependent. It can be written as [1]

$$\beta' = \beta + k_0 n_2 \frac{P}{A_{eff}} \equiv \beta + \gamma P \tag{2.3}$$

where $\gamma = 2\pi n_2 / A_{eff} \lambda$ is an important nonlinear parameter with values ranging from 1 to 5 W^{-1}/km depending on the values of A_{eff} and the wavelength. Noting that the optical phase increases linearly with z, the γ term produces a nonlinear shift given by

$$\Phi_{NL} = \int_0^L (\beta' - \beta) dz = \int_0^L \gamma P(z) dz = \gamma P_{in} L_{eff} \tag{2.4}$$

where $P(z) = P_{in} \exp(-\alpha z)$ accounts for fiber losses and L_{eff} is the effective interaction length defined as

$$L_{eff} = [1 - \exp(-\alpha L)] / \alpha \tag{2.5}$$

In deriving equation 2.4, P_{in} was assumed to be constant. In practice, time dependence of P_{in} makes Φ_{NL} to vary with time. In fact, the optical phase changes with time in exactly the same fashion as the optical signal. Since this nonlinear phase modulation is self-induced, the nonlinear phenomenon responsible for it is called self-phase modulation . It is clear from equation 2.2 that SPM leads to frequency chirping of optical pulses. This frequency chirp is proportional to the derivative dP_{in} / dt and depends on the pulse shape. The SPM-induced chirp affects the pulse shape through GVD and often leads to additional pulse broadening. In general, spectral broadening of the pulse induced by SPM increases the signal bandwidth considerably and limits the performance of lightwave systems.

(b) Cross-Phase Modulation

The intensity dependence of the refractive index in equation (2.2) can lead to another nonlinear phenomenon known as cross-phase modulation. It occurs when two or more optical channels are transmitted simultaneously inside an optical fiber using the WDM technique. In such systems, the nonlinear phase shift for a specific channel depends not only on the power of that channel but also on the power of other channels. The phase shift of the *j*th channel becomes [2]

$$\Phi_j^{NL} = \gamma L_{eff}\left(P_j + 2\sum_{m \neq j} P_m\right) \tag{2.6}$$

where the sum extends over the number of channels. The factor of 2 in equation (2.6) has its origin in the form of the nonlinear susceptibility and indicates that XPM is twice as effective as SPM for the same amount of power. The total phase shift depends on the powers in all channels and would vary from bit to bit depending on the bit pattern of the neighbouring channels. If we assume equal channel powers, the phase shift in the worst case in which all channels simultaneously carry 1 bit and all pulses overlap is given by

$$\Phi_j^{NL} = (\gamma / \alpha)(2M - 1)P_j \tag{2.7}$$

It is difficult to estimate the impact of XPM on the performance of multi-channel lightwave systems. The reason is that the preceding discussion has implicitly assumed that XPM occurs in isolation without dispersive effects and is valid only for CW optical beams. In practice, pulses in different channels travel at different speeds. The XPM-induced phase shift can occur only when two pulses overlap in time. For widely separated channels they overlap for such a short time that XPM effects are virtually negligible. On the other hand, pulses in neighbouring channels will overlap long enough for XPM effects to accumulate.

(c) Four-Wave Mixing

The power dependence of the refractive index in equation (2.2) has its origin in the third order nonlinear susceptibility denoted by $\chi^{(3)}$. The nonlinear phenomenon, known as four-wave mixing, also originates from $\chi^{(3)}$. If three optical fields with carrier frequencies ω_1, ω_2 and ω_3 co-propagate inside the fiber simultaneously, $\chi^{(3)}$ generates a fourth field whose frequency is related to the other frequencies by a relation $\omega_4 = \omega_1 \pm \omega_2 \pm \omega_3$. Several frequencies corresponding to different plus and minus sign combinations are possible in principle. In practice, most of these combinations do not build up because of a phase-matching requirement [1]. Frequency combinations of the form $\omega_4 = \omega_1 + \omega_2 - \omega_3$ are often troublesome for multi-channel communication systems since they can become nearly phase-matched when channel wavelengths lie close to the zero-dispersion wavelength. In fact, the degenerate FWM process for which $\omega_1 = \omega_2$ is often the dominant process and impacts the system performance most.

The four-wave mixing process results in transfer of power from each channel to its nearest neighbors. Such a power transfer not only results in the power loss for the channel but also induces inter-channel crosstalk that degrades the system performance severely. Modern WDM systems avoid FWM by using the technique of dispersion management in which GVD is kept locally high in each fiber section even though it is low on average [3]. Commercial dispersions shifted fibers are designed with a dispersion of 4 ps / (km-nm), a value found large enough to suppress FWM.

Four-wave mixing can also be useful in designing lightwave systems. It is often used for demultiplexing individual channels when time-division multiplexing is used in the optical domain. It can also be used for wavelength conversion. FWM in optical fibers is sometimes used for generating a spectrally inverted signal through the process of phase conjugation. This technique is useful for dispersion compensation.

2.1.5 Nonlinear Gain and System Parameters

Most nonlinear optical interactions involving two overlapping optical waves propagating in a medium can be characterized by [4]

$$P_1(L) = P_1(O)\exp(gP_2L / A) \tag{2.8}$$

Where P_1 (O) and P_1 (L) is the power of one wave entering and exiting, respectively, in a medium of length L. This amplified wave is commonly called the probe wave. P_2 is the injected power of the other wave, called the pump, which generates the gain for first wave. The cross-sectional area common to the light beams is A; the gain coefficient g (expressed in centimeter per Watt) is a direct measure of the strength of the nonlinearity. Equation (2.8) assumes that P_2 is constant throughout the nonlinear medium, that is, there is no pump depletion due to nonlinearity and no intrinsic loss. Furthermore, (2.8) assumes that the polarization states of the pump and probe waves are the same. Neither of these assumptions typically holds in fibers. Attenuation in long fibers in not negligible and the polarization states of the pump and probe waves can evolve differently in fiber. Consequently, (2.8) must be modified to be applicable to single-mode optical fibers [4]. The correct expression is

$$P_1(L) = P_1(O)\exp(gP_2L_e / bA_e) \tag{2.9}$$

where P_2, P_1 (O), P_1 (L) and g are defined as before. The effective area of the propagating waves A_e is evaluated by calculating the average modal overlap between the pump and probe waves. However, in general, if the pump and probe wavelengths are comparable and both are slightly longer than the fiber cutoff wavelength, then $A_e \approx A$, where A is the core area of the fiber.

The effective fiber length L_e replaces the actual length L in order to account for the exponential decay with length of the pump power due to fiber loss. L_e is given by equation (2.5) where α is the loss coefficient of the fiber. For $\alpha L \ll 1$, $L_e \approx L$ and for $\alpha L \gg 1$, $L \approx 1 / \alpha$. The factor b accounts for the relative polarizations of pump and probe waves and the polarization properties of the fiber. In a polarization-maintaining fiber, with identical pump and probe polarization states, b = 1. In a conventional fiber that does not maintain polarization, b = 2. Equation (2.9) describes the strength of optical nonlinearities as a function of system parameters.

2.1.6 Importance of Non Linear Effects

The measurements of the nonlinear-index coefficient n_2 in silica fibers yield a value of about 2.3×10^{-22} m^2 / V^2 or 3.2×10^{-16} cm^2 / W [1]. This value is small compared to most other nonlinear media by at least two orders of magnitude. In spite of the intrinsically small values of the nonlinearity coefficients in fused silica, the nonlinear effects in optical fibers can be observed at relatively low power levels. This is possible because of two important characteristics of single-mode fibers, a small spot size and extremely low loss. A figure of merit for the efficiency of a nonlinear process in bulk media is the product $I.L_{eff}$ where I is the optical intensity and L_{eff} is the effective length of interactive region. If light is focused to a spot of radius w_0, then $I = P / \pi w_0^2$, where P is the incident optical power. Clearly, I can be increased by focusing the light strongly to reduce w_0. However, this results in a smaller L_{eff} since the length of the focal region decreases with tight focusing. For a Gaussian beam, $L_{eff} \approx \pi w_0^2 / \lambda$, and the product

$$IL_{eff} = \frac{P}{\pi w_0^2} \frac{\pi w_0^2}{\lambda} = \frac{P}{\lambda} \tag{2.10}$$

is independent of the spot size w_0. In single mode fibers, the spot size w_0 is determined by the core radius a. Using $I(z) = I_0 \exp(-\alpha z)$, where $I_0 = P / \pi w_0^2$ and P is the optical power coupled into the fiber, the product IL_{eff} becomes

$$ILeff = \int_0^L \frac{P}{\pi w_0^2} \exp(-\alpha z) dz = \frac{P}{\pi w_0^2} \left(\frac{1 - \exp(-\alpha L)}{\alpha} \right) \tag{2.11}$$

A comparison of equations (2.8) and (2.9) shows that the efficiency of the nonlinear response in optical fibers can be improved by a factor [1]

$$\frac{(IL_{eff})fiber}{(IL_{eff})bulk} = \frac{\lambda}{\pi w_0^2 \alpha} \tag{2.12}$$

where it was assumed that $\alpha L >> 1$. In the visible region, typically $\lambda = 0.53$ µm, $w_0 = 2$ µm and $\alpha = 2.5 \times 10^{-5}$ cm^{-1} (10 dB/km) resulting in an enhancement by a factor -10^7. The enhancement can be ~10^9 at the minimum loss wavelength near 1.55 µm, where $\alpha = 5 \times 10^{-7}$ cm^{-1} (0.2 dB/km). It is this tremendous enhancement in the efficiency of the nonlinear processes that make optical fibers a suitable nonlinear medium for the observation of a wide variety of nonlinear effects at relatively low power levels.

2.2 Four Wave Mixing

Dense WDM transmission in which individual wavelength channels are modulated at rates of 10 Gb/s offers capacities of N × 10 Gb/s, where N is the number of wavelengths. To transmit such high capacities over long distances requires operation in the 1550-nm window of dispersion-shifted fiber. In addition, to preserve an adequate signal-to-noise ratio, a 10-Gb/s system operating over long distances and having nominal optical repeater spacings of 100 km needs optical launch powers of around 1 mW per channel. For such WDM systems, the simultaneous requirements of high launch power and low dispersion give rise to the generation of new frequencies due to four-wave mixing.

2.2.1 Introduction

When a high power optical signal is launched into the fiber, the linearity of the optical response is lost. One such nonlinear effect, which is due to the third order susceptibility, is called the Optical Kerr Effect [1]. Four-wave mixing is a type of optical Kerr effect and occurs when light of two or more different wavelengths is launched into the fiber. Generally speaking, FWM occurs when light of three different wavelengths is launched into the fiber, giving rise to a new wave, the wavelength of which does not coincide with any of the others. Thus, FWM is the interaction of two or more wavelengths (channels) which results in sidebands (or ghost channels) and is caused by non-linear effects. The sidebands can coincide with other channels (figure 2.1) resulting in distortion. The effect on DWDM channels can be similar to crosstalk if only few channels are used and similar to noise if a number of channels are used.

Four-wave mixing is a non-linear process in that it is a consequence of a non-linear relationship between the polarization induced within a dielectric medium and the amplitude of the incident electromagnetic field. The effect manifests itself in the appearance or amplification of radiation at a number of frequencies due to the incidence of radiation at two or more other frequencies onto an active material. The generation of a new frequency of radiation due to FWM has applications in the development of tunable sources and wavelength conversion in all-optical routing systems whereas the amplification of an existing signal can be used for optical amplification and demultiplexing at bit rates of up to 500 Gb/s. However, frequency generation and amplification can be disadvantageous in systems where many frequencies are co-propagating but each one must remain independent. The most important example of this situation is in WDM optical communication systems.

Figure 2.1 illustrates how four-wave mixing can lead to generation of a number of extra frequencies from the interaction between light at two or three incident frequencies [5]. The relation that gives the frequency of the generated wave, ω_{ijk} is

$$\omega_{ijk} = \omega_i + \omega_j - \omega_k \tag{2.13}$$

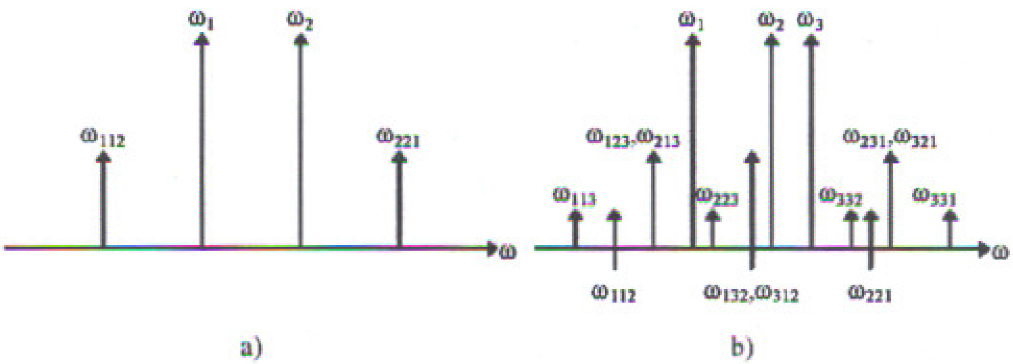


Figure 2.1: Additional Frequencies Generated Through FWM in the (a) partially degenerate and (b) non-degenerate case

If two of the three waves have the same frequency and therefore there are only two distinct frequencies initially in the fiber (figure 2.1a), then the effect is known as partially degenerate four-wave mixing (PDFWM). In this case, one of the incident waves assumes the role of both ω_i and ω_j in equation 2.1 and only two new frequencies are generated - one for $\omega_i = \omega_j = \omega_1$ and one for $\omega_i = \omega_j = \omega_2$. The non-degenerate case (NDFWM), in which three unique frequencies are initially present, is illustrated in figure 2.1(b). It can be appreciated from this diagram that a large number of generated frequencies are possible through different permutations of ω_i, ω_j and ω_k..

2.2.2 Theory of Four Wave Mixing

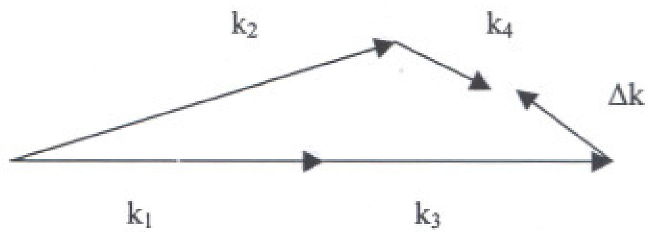
The concept of three electromagnetic fields interacting to produce a fourth field is central to the description of all four-wave mixing processes. Physically, we may understand this process by considering the individual interactions of the fields within a dielectric medium. The first input field causes an oscillating polarization in the dielectric which re-radiates with some phase shift determined by the damping of the individual dipoles; this is just traditional Rayleigh scattering described by linear optics. The application of a second field will also drive the polarization of the dielectric, and the interference of the two waves will cause harmonics in the polarization at the sum and difference frequencies. Now, application of a third field will also drive the polarization, and this will beat with both the other input fields as well as the sum and difference frequencies. This beating with the sum and difference frequencies is what gives rise to the fourth field in four-wave mixing.

The traditional method of modeling an optical material's nonlinear response is to expand the induced polarization as a power series in the electric field strength

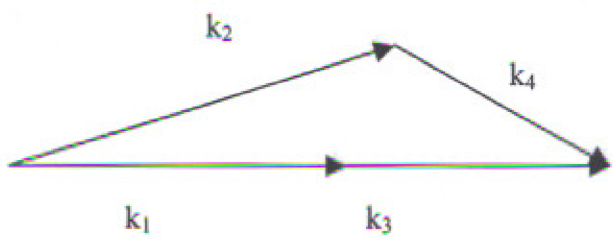
$$\vec{P} = \chi^{(1)} \cdot \vec{E} + \chi^{(2)} \cdot \vec{E}\vec{E} + \chi^{(3)} \cdot \vec{E}\vec{E}\vec{E} + \ldots \quad (2.14)$$

The expansion coefficients are known as susceptibilities. The third order nonlinear susceptibility is responsible for four-wave mixing processes. Efficient coupling between the four waves may only occur when energy and momentum are both conserved [6]: $\omega_4 = \omega_1 - \omega_2 + \omega_3$ and $k_4 = k_1 - k_2 + k_3$. Another equivalent way of understanding these conditions is by realizing that since the energy transfer is a coherent process; all four waves must maintain a constant phase relative to the others in order to avoid any destructive interference.

Phase matching is the process of choosing the directions, and polarizations in birefringent media, in order to eliminate the wave vector mismatch. Figure 2.2 is a pictorial representation of the wave vector mismatch: situation (a) shows a finite wave vector mismatch, while (b) demonstrates the corresponding phase matched case. The constraints imposed by phase matching are responsible for the highly directional nature of the signals produced by four wave mixing and the ease of spatially separating the output fields. Phase matching may also be used as a spectral filter since only a very narrow frequency band may be phase matched along a particular direction.



(a)



(b)

Figure 2.2: Phase Matching Diagram

(a) Wave vector mismatch of Δ k (b) perfectly phase matched case.

2.2.3 General Requirements for Four-Wave Mixing

Four-wave mixing is a process in which optical waves at four frequencies interact via the non-linear response of a medium to the electric fields of the waves. Assuming for simplicity that four optical waves incident upon a dielectric medium are linearly polarized parallel to the x-axis, the combined electric field can be expressed as [5]

$$E_{TOT} = \hat{x}\sum_{j=1}^{4} E_j \cos\left(\beta_j z - \omega_j t\right) \qquad (2.15)$$

where E_j is the amplitude of the jth electric field and β_j is the propagation constant of the wave at frequency ω_j.

The propagation constant is given by the relation

$$\beta_j = n_j(\omega)\omega_j / c \tag{2.16}$$

where $n_j(\omega)$ is the frequency-dependent refractive index and c is the speed of light in vacuum. The second and third order expansion terms when equation 2.15 is substituted into 2.14 will include a number of terms involving products of the four waves. By collecting terms at each of the four incident frequencies, the non-linear polarizations in the material, P_{NL} , can be resolved into components at each frequency, *j*, such that

$$P = P_L + P_{NL} = P_L + \hat{x}\sum_{j=1}^{4} E_j \cos(\beta_j z - \omega_j t) \tag{2.17}$$

where

$$P_j = \frac{3\varepsilon_0}{4}\chi_3\left[SPM_j + XPM_j + f_j(i\theta +) + g_j(i\theta -) + \ldots\right] \tag{2.18}$$

and *f* and g are functions of the electric field amplitudes and also the parameters θ+ and θ- discussed below.

The polarization of the material at each of the incident frequencies can be seen from equation 2.18. To include terms due to self-phase modulation, cross-phase modulation and two functions of the parameters θ+ and θ- [5] that represent the relative phase between the electric field and polarizations, P_j at the *j*th frequency. Four-wave mixing is most efficient when θ+ or θ- approaches zero, where

$$\theta + = (\beta_1 + \beta_2 + \beta_3 - \beta_4)z - (\omega_1 + \omega_2 + \omega_3 - \omega_4)\,t \tag{2.19}$$

$$\theta - = (\beta_1 + \beta_2 - \beta_3 - \beta_4)z - (\omega_1 + \omega_2 - \omega_3 - \omega_4)\,t \tag{2.20}$$

These two conditions give rise to different four-wave mixing phenomena. To minimize either θ + or θ - requires that certain conditions are satisfied by both the frequencies of the signals and their propagation constants. These conditions effectively amount to conservation of energy and momentum before and after the FWM interaction. The satisfaction of the latter condition is referred to as phase matching. The frequency condition for equation 2.19 is that $\omega_4 = \omega_1 + \omega_2 + \omega_3$, which permits *third harmonic generation* in the fully degenerate case when $\omega_1 = \omega_2 = \omega_3$ and also generation of waves at $2\omega_1 + \omega_2$ if $\omega_1 = \omega_3 \neq \omega_3$, i.e. the partially degenerate case. Phase matching of propagation constants for this condition ($\Delta\beta = (\beta_1 + \beta_2 - \beta_3 - \beta_4) = 0$) is difficult to satisfy and in practice it is the second of the two FWM mechanisms that is normally observed in fibers.

Minimization of equation 2.20 occurs when the frequencies of the waves satisfy the relation $\omega_1 + \omega_2 = \omega_3 + \omega_4$ and the associated phase matching requirements are met, i.e. $\Delta\beta = (\beta_1 + \beta_2 - \beta_3 - \beta_4) = 0$. This can occur for the non-degenerate case, where $\omega_1 \neq \omega_2 \neq \omega_3 \neq \omega_4$, or the partially degenerate case, in which $\omega_1 = \omega_2 \neq \omega_3 \neq \omega_4$. Both of these situations are illustrated in figure 2.1, in which incident frequencies ω_1, ω_2 and ω_3 generate extra frequencies at $\omega_{ijk} = \omega_i - \omega_j + \omega_k$ through four-wave mixing.

2.2.4 Efficiency of Four Wave Mixing

The time-averaged optical power generated at frequency ω_4 for co-polarized input signal at ω_1, ω_2 and ω_3 is given by [6]

$$p_4(L) = \eta\left(\frac{1024\pi^6}{n^4\lambda^2c^2}\right)\left(\frac{L_{eff}}{A_{eff}}\right)^2(D\chi_3)^2 p_1(0)p_2(0)p_3(0)\exp(-\alpha L) \tag{2.21}$$

where η is the *four-wave mixing efficiency*, *D* in this case is the degeneracy factor and L_{eff} is the *effective length* of the fibre. The degeneracy factor takes a value of 1, 3 or 6 depending on whether, respectively, all, two or none of the incident frequencies are the same. Effective length is a parameter commonly used in the study of non-linear effects in fibres and is defined as the length of lossless fibre that would generate the same amount of non-linearity as the lossy fibre under consideration. The use of effective length permits simple inter-comparisons of non-linearity in fibre with different attenuation characteristics. It is defined by equation (2.5)

The four-wave mixing efficiency parameter, η, is a function of phase mismatch, $\Delta\beta$, and is the most useful measurable quantity for describing four-wave mixing in optical fibres. The efficiency is normalized to 1 at zero phase mismatch and is given by

$$\eta = \frac{p_4(L,k)}{p_4(L,k=0)} = \frac{\alpha^2}{\alpha^2 + k^2}\left[1 + \frac{4\exp(-\alpha L)\sin^2(kL)}{\{1-\exp(-\alpha L)\}^2}\right] \qquad (2.22)$$

where k is defined in equation 3.14 as being the net phase mismatch between the four signals due to material and waveguide dispersion and non-linear effects.

2.2.5 Polarization Dependence

Four-wave mixing is dependent on both the absolute and relative polarizations of the interacting waves [7], with maximum efficiency for parallel linearly polarized waves. In general, input waves are not linearly but elliptically polarized and the states of polarisation evolve along the fibre due to residual birefringence. The exception to this in standard, non-polarization-maintaining, fiber is when the light is launched with its polarization axis parallel to one of the principle states of polarization. Under these circumstances, the state of polarization still evolves along the fibre but waves with closely spaced frequencies evolve together and maintain parallel, if not linear, polarization. Experimental investigations into FWM in fibres are usually performed using polarization controllers with each optical source to allow adjustment of the relative and absolute states of input polarization. Typically, the polarization controllers are simply adjusted until FWM is maximized, without attempting to ensure linear polarization [6, 7].

2.2.6 Applications of Four Wave Mixing

(1) High-Speed Optical Multiplexing/Demultiplexing

The four-wave mixing all-optical demultiplexer [5] uses short, intense, co-propagating pump pulses synchronized to the time slot of the required data. Either the data or the pump pulses must be at the zero dispersion wavelength of the fibre so that the pump interacts with the data through partially degenerate FWM to generate an identical data stream at a complimentary frequency. This frequency-converted signal is then separated from the main data stream using a band pass filter. The advantage of converting the optical frequency of the demultiplexed data before detecting it is that the band pass filter can remove noise due to the main data stream. This gives a better SNR than other methods where the demultiplexed data remains at the original optical frequency of the channel.

(2) Parametric Amplification

Parametric amplification is made possible by four-wave mixing [8]. In low-birefringence fibers the birefringence axes and strength vary randomly with distance. In recent years there has been a resurgence of interest in the parametric amplification (PA) of optical signals. Because of recent improvements in highly-nonlinear fibers, it is now a routine matter to produce FWM gains higher than 40 dB over bandwidths broader than 20 nm. Such performance makes possible wavelength conversion and impairment reduction by phase conjugation in wavelength-division-multiplexed communication systems. PA is driven most strongly when the pumps have parallel polarization vectors. However, the signal gain depends sensitively on the input signal polarization: It is maximal when the signal is polarized parallel to the pumps and is minimal when the signal is polarized perpendicular to the pumps.

(3) Wavelength Conversion

Future all-optical networks based on WDM and optical routing will need wavelength converters to allow maximum flexibility when multiplexing and allocating channels. Wavelength conversion based on four-wave mixing has the advantage over other methods such as cross-gain modulation and optoelectronic methods that conversion is coherent and maintains the phase of the wave. FWM-based wavelength converters are therefore suitable for coherent systems based on phase-shift keying as well as intensity modulation schemes.

Wavelength conversion by FWM in SOA's [9] is the only method for wavelength conversion that provides both strict transparency and arbitrary wavelength mapping. Fig.2.3 is a schematic for a typical FWM experimental configuration. Two co-polarized waves are coupled into the SOA. One of the waves, called the pump wave (Ep) with frequency ω_p, is typically stronger than the other wave (i.e., the input signal to be converted) called the probe wave (E_q) with frequency ω_q. Inside the SOA, the co propagating pump and probe waves mix, forming dynamic gain and index gratings in the

SOA through the mechanisms of carrier density modulation, carrier heating and spectral hole burning. The pump wave scattering from these gratings generates two waves, one at the probe frequency and one at a new frequency, $\omega_{cs} = 2\omega_p - \omega_q$. Probe scattering also generates two much weaker waves, one at the pump frequency and one at $\omega_{cs'} = 2\omega_q - \omega_p$ The wave with frequency is easily shown to be the phase conjugate replica of the original input signal. As such, it provides the wavelength converted signal.

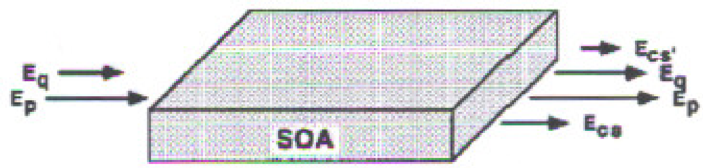


Figure 2.3: Wavelength Conversion Using FWM.

(4) Mobile Communications

Four-wave mixing in dispersion shifted fiber has recently been demonstrated to be an elegant method of generating a 60 GHz mm-wave suitable for broadband wireless systems. The CW output from a laser diode source with optical frequency f_0 was intensity modulated at frequency 20 GHz to produce two sidebands at $f_o + \Delta f$ and $f_o - \Delta f$. The three signals were then amplified using an EDFA and fed through a 35 km length of dispersion-shifted fiber. Efficient non-degenerate FWM in the fiber generated extra sidebands at frequencies $f_o \pm n\Delta f$ where n is an integer. All signals were optically filtered out except for the main carrier at and one of the sidebands at $f_o + 3\Delta f$. The sideband at f_o + 60 GHz was intensity modulated with a 156 MHz sinusoidal wave to simulate a data signal. The carrier and sideband were then transmitted along 108 km of standard single mode fiber and combined at a photodiode using a self-heterodyne technique. The recovered signals, consisting of a carrier wave at 60 GHz and sidebands at 60 GHz ± 156 MHz, showed excellent amplitude stability and narrow spectral widths. The amplitude and spectral stability was significantly better than that achieved when the modulated sideband was combined with a carrier signal taken directly from laser diode.

(5) Fiber Nonlinear Coefficient Measurement

Many schemes have been previously proposed to make measurement of non-linear coefficient (g). Four-Wave mixing is a reliable technique for determining g [10]. The well-known measuring setup based on FWM is made up of two DFB lasers, where the wavelength spacing is adjusted by temperature tuning. It uses more than one optical EDFA to raise the total optical power. One of the biggest disadvantages of this method is that the polarization controllers must be adjusted all the measurement time in order to obtain maximum FWM efficiency. The determination of g employing such FWM method has a limited measurement capability due to the polarization adjustment and insufficient amplification. One externally modulated laser source only can also be used as a measurement scheme based on FWM method. The use of such source eliminates polarization dependence and making measurement scheme simple and highly sensitive, what leads to higher accuracy.

(6) Phase Conjugation

DFWM can yield phase conjugation and is useful, for example, for correcting aberrations by using a phase conjugate mirror. To understand what phase conjugation is, consider an electric field

$$E_1(r,t) = \mathrm{Re}\{\wp(r)\exp[i(kz - \omega t)]\}$$

If we can produce in the material another electric field $E_2(r,t) = \mathrm{Re}\{\wp^*(r)\exp[i(-kz - \omega t)]\}$ at any z, we call this field E the phase *conjugate* of *E* [6]. The most frequently used DFWM configuration is the phase-conjugate geometry (fig.2.4 (a)) [6]. In this configuration, two counter propagating pump beams, called backward (B) and forward (F), and a probe beam (P) are incident on the nonlinear material. Since beams B and F are collinear, $k_F = - k_B$. The probe beam is incident at a small angle to the direction of the forward beam. Due to the third order nonlinear polarization, a fourth beam is created and is referred to as the conjugate beam. This fourth beam is counter propagating to the probe beam, $k_C = - k_P$.The process can be understood using the grating diagram (Figures 2.4 (b) and (c)). The forward and the probe beams interfere to form a grating from which the backward beam scatters, generating the conjugate beam.

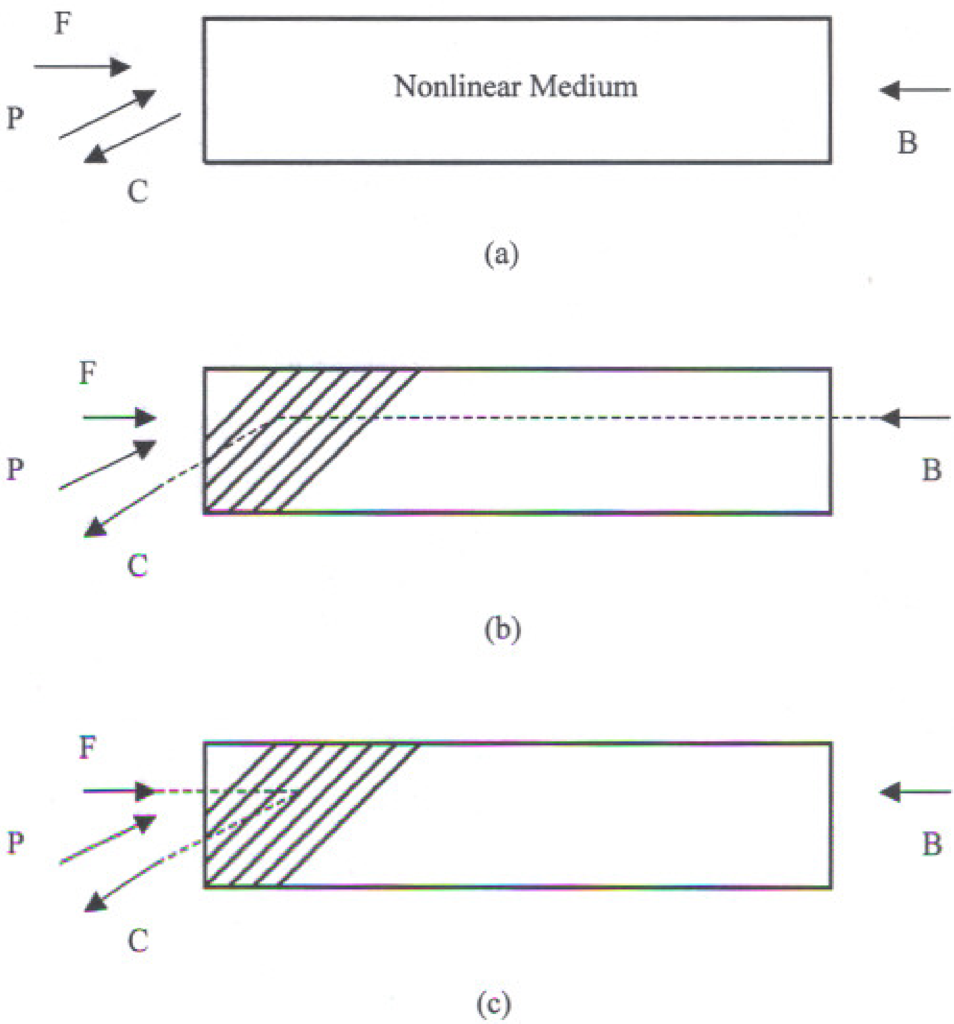


Figure 2.4: Phase Conjugation

a) Geometry and grating interpretation for phase-conjugation.

b) Backward beam scattering off grating created by forward and probe beams.

c) Forward beam scattering off grating formed by backward and probe beams.

2.2.7 Implications of FWM for WDM systems

Wavelength Division Multiplexed systems greatly increase the total bandwidth of each optical fibre by using a number of closely spaced channels at wavelengths within the typical 1530 nm to 1565 nm gain spectrum of erbium doped fibre amplifiers. Any interaction between these channels will lead to a degradation of the bit-error rate (BER) of the system for two reasons. Firstly, the pump channels will experience signal depletion as optical power is transferred to a different wavelength. Secondly, if the frequency of a FWM product coincides with one of the allocated system channels then this channel will suffer from noise. This is a particular problem for channels that are equally spaced in frequency. For a WDM system with N channels, the number of four-wave mixing products, M, will be [5]

$$M = \frac{1}{2}\left[N^3 - N^2\right] \tag{2.23}$$

Clearly, for a typical 8-channel WDM system, unless great care is taken to allocate the channel frequencies carefully, some crosstalk will occur due to FWM. The most detrimental cross modulation between channels in a WDM communication system occurs for closely-spaced frequencies with uniform frequency separation near the fibre dispersion zero, λ_0. This occurs for two reasons:

- The phase matching between the signals is more easily achieved in this region.
- The reduced chromatic dispersion slows the walk-off between neighbouring signals and reduces any averaging of the cross-modulation that would otherwise occur.

2.2.8 Reducing the Effects of Four Wave Mixing

(1) Reduction Using Sub Channel Multiplexing

The effect of intra-channel four-wave mixing can be reduced by replacing each 40 Gb/s wavelength-division multiplexed channel by two 20 Gb/s sub channels that are slightly separated in frequency [11]. These two sub channels are not separate channels, which would require them to be separately demultiplexed and detected. Instead, at the receiver the pulses in the two sub channels must not overlap in the time domain. This method works best for systems in which there is substantial pulse overlap during transmission, such as one would encounter when upgrading an installed system from 10 to 40 Gb/s. The method relies on trading of decreased spectral efficiency for increased propagation distance. The sub channel-multiplexed signal is obtained by replacing each wavelength-division multiplexed channel by a pair of sub channels that are created by shifting the central frequencies of the pulses in the even numbered bit slots by $+\Omega/2$ and those in the odd bit slots by $-\Omega/2$. For a 40 Gb/s signal, this procedure produces two 20 Gb/s signals whose pulse widths are appropriate for a 40 Gb/s signal and that are spaced $\Delta\Omega$ apart in frequency with a time offset of 25 ps. At the receiver the two sub channels are treated as if they were a single 40 Gb/s channel. In particular, each pair of sub channels is demultiplexed from the full wavelength-division multiplexed signal using a single optical filter whose central frequency is the average of the central frequencies of the two sub channels.

(2) Reduction by Wavelength Shift Keying

In WDM wavelength shifted keying, each user is assigned two specific wavelengths that are symmetric with respect to the zero dispersion wavelength [12]. One wavelength is used to transmit symbol "1", while the other is used to transmit symbol "0". The modulated signals of all users are combined, propagate along the long-haul dispersion-shifted fiber, and experience attenuation and spectrum deformation due to FWM. At the receiver, narrow band filters select the desired user's two wavelengths and they are detected by a balanced receiver. The received signal is positive for symbol "1"and negative for symbol "0". While wavelength shift keying requires twice as many wavelengths to support a given number of data channels, it has a number of advantages. Complementary keying has a 3 dB signal to noise advantage over on-off keying; dispersion shifted fiber permits higher transmission rates; and detection of both data symbols involves detection of energy at a nonzero threshold, reducing sensitivity to noise of all types. The balanced detection cancels all noise having a uniform spectral distribution and symmetric assignment of symbol wavelengths around the zero dispersion wavelengths cancels FWM interference to first order.

(3) Channel Allocation Algorithm

Channel allocation methods, attempting to reduce FWM effect in WDM systems, resulted in increase of bandwidth requirement compared to equally spaced channel allocation. A fractional bandwidth allocation algorithm is designed taking into consideration the concept of OGR so as to improve WDM system performance without inducing additional cost, in terms of bandwidth [13]. This technique allows the computation of a channel allocation set to result in an optimal point where degradation caused by inter-channel interference and FWM is minimal. In this method, reduction in FWM effect with the WDM system using the same operating bandwidth as for equally spaced channel allocation could be achieved by allocating some channels nearer to other channels. However, with the reduction of frequency spacing between some channels, the inter-channel interference (ICI) would be increased. Thus, this method will attempt to find an optimal point where the distortion caused by both the ICI and FWM will be minimal. Incorporating this algorithm with OGR, a systematic approach to the channel allocation could be achieved whereby no FWM signals fall exactly on the carrier channel and the operating bandwidth would not need to be expanded [13].

For N channels, the OGR for N marks is used. The first element of all OGRs is 0 and in this scheme, only the rest of the elements are utilized in the algorithm. The operating bandwidth is split into "Pre-allocate" and "Post-allocate" section. The "Pre-allocate" bandwidth section is divided by the total number of channels, so as to obtain the initial channel spacing. The "Post-allocate" bandwidth section is then divided into parts determined by the modified OGR. The OGR vector is formed by removing the first element of 0. The initial modified OGR vector is then formed by rearranging the elements so that the channel spacing of channels nearer to the center frequency will be wider. Subsequent modified OGR vectors are formed by incrementing each element with an Incrementation Factor. The notion here is that since the difference for any two elements in the OGR is distinct, incrementing each element of the OGR by a same value would still result in distinct difference. The Incrementation Factor will also have an impact on the allocation in that it will be in increasing order of significance from the outer to the inner elements. After which, a near equally spaced channel allocation situation would be reached following multiple iterations of the algorithm. The performance of the WDM system is then observed for each iteration to locate the allocation where the performance is optimal.

2.3 Optimal Golomb Ruler

The concept of a "Golomb Ruler" came form the work of Professor Solomon W. Golomb of the University of Southern California. In mathematics, the term "Golomb Ruler" refers to a set of non-negative integers such that no two distinct pairs of numbers from the set have the same difference. Conceptually, this is similar to a ruler constructed in such a way that no two pairs of marks measure the same difference. Golomb rulers are credited as being 'discovered' by W. Babcock in 1953 [14]. He was investigating the inter-modulation distortion, which appeared in third and fifth order consecutive radio bands. He discovered that by placing each pair of channels inside the frequency spectrum at a distinct distance the third order distortion was eliminated and the fifth order distortion was lessened greatly.

The earliest set of rulers was published in 1952 by W.C. Babcock [14]. He published rulers up to 8 marks in length. It is not very clear as to who first published the ruler with 9 marks. John P. Robinson in 1979 published rulers with 10, 11 and 12 marks. Douglas S. Robertson was responsible for discovering the ruler with 13 marks. Finally, in 1990, J. B. Shearer published rulers with 14, 15 and 16 marks [15]. Sibert discovered rulers of 17 and 18 marks. In 1997, W. C. Kyong [16] gave algorithms for finding out the golomb sequences for any prime number p. Three algebraic algorithms on finding the frequency locations of unequal spaced channels had been introduced based on finding optical CDMA code words with a predetermined pulse separation and aperiodic autocorrelation side lobes no greater than one. To show that the results are optimal, the bounds on the total occupied optical bandwidth of the unequal spaced channels had been provided. But the application of the algorithms was limited to prime powers.

William T. Rankin [17] provided a parallel distributed algorithm for Golomb Ruler namely shift algorithm in 1998. Using this algorithm, the optimality for 17, 18 and 19 mark rulers was proven computationally. The search for optimum Golomb rules by exhaustive search was in run and all optimum Golomb rules up to 23 marks were known. In July 2000 the website distributed.net [18] started the web search for the 24 and 25 OGRs. By the time, the idea of optimizing the Golomb rules using genetic algorithms came into light and the research is still going on.

2.3.1 What is a Golomb Ruler?

A normal ruler is used to measure distances. It has marks at equidistant points and the distances are measured from one point to another. We consider rulers with marks at integer locations. Let us assume that the ruler has m marks and the length of the ruler is n.

For a normal ruler, n = m - 1. Such a ruler is shown in figure 2.5. The ruler in the figure has four marks at 0, 1, 2 and 3. Thus, it can measure distances up to three units in length. The distances are marked on the ruler.

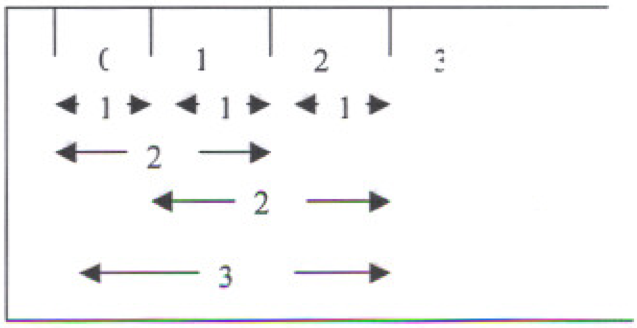


Figure 2.5: A Normal Ruler

Consider a ruler in which one distance can be measured only in one way. In other words, the distance between any two points on the ruler is unique. Such a ruler with four marks is shown in figure 2.6. The distance between each pair of marks is also shown. As can be seen, they are all distinct. It is necessary to define some basic terms used to describe the characteristics of Golomb Rulers.

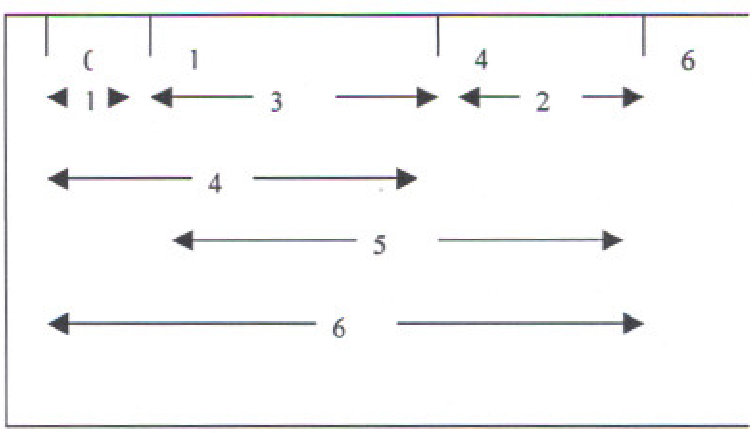


Figure 2.6: A Golomb Ruler

- **Marks and Length**

A Golomb Ruler consists of ordered series of integer numbers. These numbers are referred to as *marks*, and correspond to positions on a linear scale. The difference between the values of any two marks is called the distance between those marks. The difference between the largest and smallest number is referred to as the *length* of the ruler, and corresponds to the largest distance for that ruler. The first mark of the series is by convention at position zero. The number of marks on a ruler is sometimes referred to as the *size* of the ruler.

- **Difference Triangle**

It is very helpful to establish a data structure to contain all the distances measured by that ruler. A difference triangle is a grid of numbers where each number represents distance between a specific pair of marks. For a ruler of m marks, there will be [m (m -1)] / 2 entries in the difference triangle. Let the marks of ruler be at M_1, M_2, M_3 and M_4. Then the difference triangle is constructed as shown in figure 2.7 where $d1_x$, $d2_x$ and $d3_x$ are the first order, second order and third order differences respectively. The relationships are as follows:-

$$d1_x = M_{x+1} - M_x$$

$$d2_x = d1_x + d1_{x+1}$$

$$d3_x = d2_x + d2_{x+1} - d1_{x+1}$$

$d1_1$ $\quad$ $d1_2$ $\quad$ $d1_3$

$d21$ $\quad$ $d2_2$

$d3_1$

Figure 2.7: A Difference Triangle

- **Perfect Golomb Ruler**

A Perfect Golomb ruler measures all integer distances from zero to L where L is the length of the ruler as in figure 2.8. In other words, the difference triangle of a perfect Golomb ruler contains all numbers between one and the length of the ruler. The length of an N mark perfect Golomb ruler in N (N – 1) / 2.

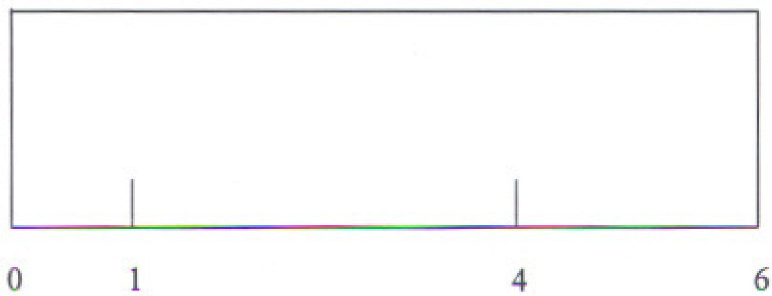


Figure 2.8: A Perfect Golomb Ruler

- **Optimum Golomb Rulers**

Since Perfect Rulers beyond four marks cannot exist, longer rulers are described in terms of whether or not they are optimum. An Optimum Golomb Ruler is defined as being the shortest length for a given number of marks. As will be shown, there may exist multiple different Optimum Golomb Rulers for a given number of marks.

2.3.2 Applications of Golomb Rulers

Golomb Rulers are not very useful in measuring distances since the distance would already have to be known to predict the marks which would measure the distance. However, the fact that all distances are unique can be used in many diverse applications. Optimal rulers are important because in many applications, the rule is - the shorter, the better. Listed below are some important applications of Golomb Rulers.

- **Radio Communications**

Perhaps the earliest reference to Golomb Rulers, although not by that name, was in the 1953 article by W. Babcock [14] regarding third and fifth order intermodulation distortion among consecutive channels in a radio band. Babcock explores methods of channel placement to reduce or eliminate third and fifth order distortion. After reviewing several algebraic and random channel distribution methods, Babcock presents the concept of placing each channel within the frequency spectrum on intervals corresponding to the marks on a Golomb Ruler.

- **X-Ray Crystallography**

Crystallography X-ray analysis of crystal structures can produce ambiguities when two different crystal lattice structures produce identical diffraction patterns. Golomb Ruler sequences were used to model the lattice structure and resolve these ambiguities, based upon the empirical observation that no two equal length. Golomb Rulers will have the same difference set. Although this observation has been disproved for a single pair of six mark rulers [19], no counter example for rulers of greater than six marks has been found, and this approach is still being used.

- **Coding Theory**

Golomb Rulers have been used extensively in the field of coding theory for error detection and correction. Sets of rulers are used to generate self-orthogonal codes that play an important role in communications [19]. These codes use what is known as a difference triangle set (DTS). A DTS consists of a group of rulers all having the same number of marks. A DTS has the additional quality that the difference triangle for each of the members of the set contains values unique to the set. An example of a DTS is shown in Figure 4.6 for the ruler set {{0, 6, 11, 13}; {0, 8, 17, 18}; {0, 3, 5, 19}}.

6 11 13	8 17 18	3 15 19
5 7	9 10	12 16
2	1	4

Figure 2.9: A Sample Difference Triangle Set

- **Linear Arrays**

Often, many elements are arranged in a linear fashion to obtain an antenna. These arrays may be used to generate beams (in order to obtain the desired shape and frequency distribution) or to receive information (such as radio telescopes). Though the requirements and specifications for both applications are distinct, Golomb rulers are effective in increasing the efficiency of both. This is achieved by placing elements at point represented by marks on a Golomb Ruler. . In the first case, Perfect Golomb Rulers provide the highest possible resolution for a given aperture length. However, larger arrays (with more than four elements) do not have access to perfect Golomb rulers. Thus, `Minimum Redundancy Linear Arrays' are used for deciding the placement. These arrays are a variation on Optimal Golomb Rulers in that they may have distances repeated in them along with the absence of certain distances. The redundancy is kept at a minimum balanced with a preference for an array with a short total length. In the second case, the placement helps maximize the information collected for a fixed cost. Since the information in such a device is based mostly on the phase difference of the incoming signal between each pair of elements, placement at Golomb marks ensures that no piece of data will be duplicated.

- **Computer Communication Network**

Computer communication networks today specify both source and destination addresses so that routers may successfully deliver messages across the network. If the nodes on a network were allotted addresses according to the marks on a Golomb Ruler, only the difference between the addresses of the origin and destination will have to be transmitted. Since the differences are unique, together with the direction of arrival, will immediately identify both the origin and the destination node [19]. This application will probably become useful and will gain more importance once rulers with many more marks have been discovered.

- **PPM Communications**

The growing use of optical communications has produced an increased interest in pulse phase modulation (PPM) as a form of communications. In PPM communications, a serial input sequence is divided up into a series of equal length frames. Each frame is subdivided into a number of slots. Within each frame a single pulse may be placed within one of the slots. The position of the pulse within the frame communicates the information. As we can see, proper frame synchronization is essential to assure error free communications. Frame synchronization is achieved through comparing the input stream with a stored sequence and generating a correlation factor based upon the summation of the compared inputs. A desirable PPM sequence maximizes this correlation factor when the frames are synchronized and minimizes the correlation for any off-shifted sequence. The maximum correlation can be achieved if no spacing between any pair of incoming pulses is repeated for any other pair. This is first requirement of a Golomb Ruler. The correlation of serial pulse trains also finds application in radar coding [19].

2.4 GENETIC ALGORITHMS

Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics [20-23]. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search. In every generation, a new set of artificial creatures (strings) is created using bits and pieces of the fittest of the old; an occasional new part is tried for good measure. While randomized, genetic algorithms are no simple random walk. They efficiently exploit historical information to speculate on new search points with expected improved performance.

Genetic algorithms have been developed by John Holland, his colleagues and his students at the University of Michigan [21]. The goal of their research has been twofold: (1) to abstract and rigorously explain the adaptive processes of natural systems and (2) to design artificial system software that retains the important mechanisms of natural systems. This has led to important discoveries in both natural and artificial systems science. The central theme of research on genetic algorithms has been robustness, the balance between efficiency and efficacy necessary for survival in many different environments.

In the 1960s, Rosenberg introduced "evolution strategies", a method he used to optimize real-valued parameters for devices such as airfoils. This idea was further developed by Schwefel. The field of evolution strategies has remained an active area of research, mostly developing independently from the field of genetic algorithms. Fogel, Owens and Walsh developed "evolutionary programming," a technique in which candidate solutions to given tasks were represented as finite-state machines, which were evolved by randomly mutating their state-transition diagrams and selecting the fittest. A somewhat broader formulation of evolutionary programming also remains an area of active research. Together, evolution strategies, evolutionary programming, and genetic algorithms form the backbone of the field of evolutionary computation.

Genetic algorithms were invented by John Holland in the 1960s and were developed by Holland and his students and colleagues at the University of Michigan in the 1960s and the 1970s. In contrast with evolution strategies and evolutionary programming, Holland's original goal was not to design algorithms to solve specific problems, but rather to formally study the phenomenon of adaptation as it occurs in nature and to develop ways in which the mechanism of natural adaptation might be imported into computer system. Holland's 1975 book Adaptation in Natural and Artificial Systems presented the genetic algorithm as an abstraction of biological evolution and gave a theoretical framework for adaptation under the GA. Holland's GAs is a method for moving from one population of "chromosomes" (e.g. strings of ones and zeros, or "bits") to a new population by using a kind of "natural selection" together with the genetics-inspired operators of crossover, mutation, and inversion. Holland was the first to attempt to put computational evolution on a firm theoretical footing. Until recently this theoretical foundation, based on the notion of "schemas", was the basis of almost all-subsequent theoretical work on genetic algorithms.

In the last several years there has been widespread interaction among researchers studying various evolutionary computation methods, and the boundaries approaches have broken down to some extent. Today, researchers often use the term "genetic algorithm" to describe something very far from Holland's original conception.

2.4.1 Goals of Optimization

We must be clearer about our goals when we say we want to optimize a function or process. What we are trying to achieve when we optimize? The conventional view is presented as follows [22]: Man's looking for perfection finds expression in the theory of optimization. It studies how to describe and attain what is best, once one knows how to measure and alter what is good or bad. Optimization theory encompasses the quantitative study of optima and methods for finding them. Thus optimization seeks to improve performance toward some optimal point or points.

2.4.2 How are Genetic Algorithms Different from Traditional Methods

In order for genetic algorithms to surpass their traditional cousins in the quest for robustness, GAs differ from more normal optimization and search procedures in four ways [23]

1. GAs search with a coding of the parameter set, not the parameters themselves.
2. GAs search from a population of points, not a single point.
3. GAs use payoff (objective function) information, not derivatives or other auxiliary knowledge.
4. GAs use probabilistic transition rules, not deterministic rules.

2.4.3 Biological Background

All living organisms consist of cells. In each cell, there is a set of chromosomes which are strings of DNA and serve as a model for the whole organism. A chromosome consists of genes on blocks of DNA. Each gene encodes a particular pattern. Basically, it can be said that each gene encodes a trait, e.g. color of eyes. Possible settings of traits are called *alleles*. Each gene has its own position in the chromosome search space. The position is called *locus*. Complete set of genetic material is called *genome* and a particular set of genes in genome is called *genotype*. The genotype is based on organism's phenotype (development after birth), its physical and mental characteristics such as eye color, intelligence and so on.

In genetic algorithms, the term chromosome typically refers to a candidate solution to a problem, often encoded as a bit string [20]. The genes are either single bits or short blocks of adjacent bits that encode a particular element of the candidate solution. An allele in a bit string is either 0 or 1; for larger alphabets more alleles are possible at each locus.

2.4.4 A Simple Genetic Algorithm

In the literature, Holland's genetic algorithm is commonly called the Simple Genetic Algorithm (SGA). Essential to the SGA's working is a population of binary strings. Each string of 0s and 1s is the encoded version of a solution to the optimization problem. Using genetic operators - crossover and mutation – the algorithm creates the subsequent generation from the strings of the current population. This generational cycle is repeated until a desired termination criterion is reached (for example. a predefined number of generations are processed).

Figure 2.10 summarizes the working of the SGA [24], in the form of a flowchart, which has the following components:

- a population of binary strings,
- control parameters.
- a fitness function,
- genetic operators (crossover and mutation)
- a selection mechanism, and
- a mechanism to encode the solutions as binary strings.

(1) Search Space

If we are solving some problems, we work towards some solution which is the best among others. The space for all possible feasible solutions is called *search space*. Each solution can be marked by its value of the fitness of the problem. 'Looking for a solution' means looking for extrema (either maximum or minimum) in search space. The se\arch space can be known by the time of solving a problem and we generate other points as the process of finding the solution continues.

The problem is that, search space is complicated and one does not know where to look for the solution or where to start from and this is where genetic algorithm is useful. GAs are inspired by the *Darwinian theory of the survival of the fittest* [20]. Algorithm is started with a set of solutions (represented by chromosomes) called populations. Solutions for one population are taken and used to form a new population. This is motivated by a hope that new population will be better than the old one. Solutions, which are selected to form new population (offspring), are selected according to their fitness.

The more suitable they are, the more chances they have to reproduce. This is repeated until some conditions for improvement of best solution are satisfied.

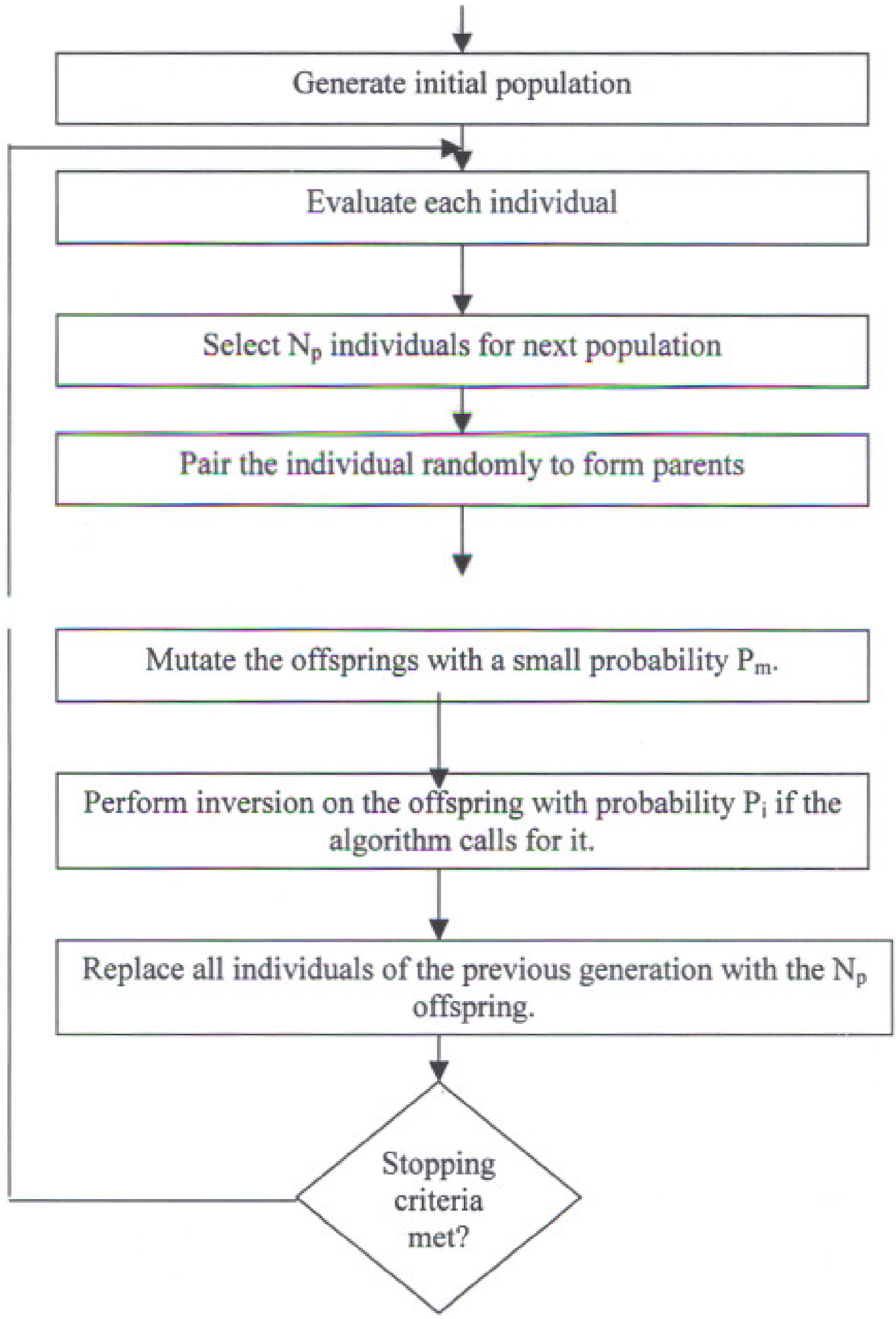


Figure2.10: Flowchart of the Simple Genetic Algorithm Structure

(2) Encoding Mechanisms

Fundamental to the GA structure is the encoding mechanism for representing the optimization problem's variables. The encoding mechanism depends on the nature of the problem variables. A large number of optimization problems have real-valued continuous variables. A common method of encoding them uses their integer representation. Each variable is first linearly mapped to an integer defined in a specified range, and the integer is encoded using a fixed number of binary bits. The binary codes of all the variables are then concatenated to obtain a binary string. For example, consider a continuous variable defined in a range from -1.28 to 1.28. We could encode this continuous variable with an accuracy of two decimal places by multiplying its real value by 100 and then discarding the decimal portion of the product. Thus the value that the variable attains is linearly mapped to integers in the range [-128, 128]. The binary code corresponding to each integer can be easily computed.

(3) Fitness Function

The objective function, the function to be optimized, provides the mechanism for evaluating each string. However, its range of values varies from problem to problem. To maintain uniformity over various problem domains, we use the fitness function to normalize the objective function to a convenient range of 0 to 1. The normalized value of the objective function is the fitness of the string, which the selection mechanism uses to evaluate the strings of the population.

As pointed out earlier GAs mimic the Darwinian theory of survival of the fittest and principle of nature to make a search process [23]. Therefore, GAs are usually suitable for solving maximization problems. Minimization problems are usually transformed into maximization problems by some suitable transformation. In general, fitness function F(X) is first derived from the objective function and used in successive genetic operations. Certain genetic operators require that fitness function be non-negative, although certain operators do not have this requirement. Consider the following transformations

$$F(X) = f(X) \text{ for maximization problem}$$

$$F(X) = 1/f(X) \text{ for minimization problem, if } f(X) \neq 0$$

$$F(X) = 1/(1 + f(X)), \text{ if } f(X) = 0$$

A number of such transformations are possible. The fitness function value of the string is known as string's fitness.

(4) Selection Mechanisms

Selection models nature's survival-of-the-fittest mechanism. Fitter solutions survive while weaker ones perish. In the SGA, a fitter string receives a higher number of offspring and thus has a higher chance of surviving in the subsequent generation. The various methods of selecting chromosomes [20] are:

(i) Roulette-Wheel Selection

The commonly used selection operator is the proportionate reproductive operator where a string is selected from the mating pool with a probability proportional to the fitness. Thus, i^{th} string in the population is selected with a probability proportional to F_i where F_i is the fitness value for that string. Since the population size is usually kept fixed in a simple GA, the sum of the probabilities of each string being selected for the mating pool must be one. The probability of the i^{th} selected string is

$$p_i = F_i \Big/ \sum_{j=1}^{n} F_j \tag{2.24}$$

where n is the population size.

One way to implement this selection scheme is to imagine a Roulette-wheel with its circumference for each string marked proportionate to string's fitness. An example is shown in figure 2.11. The fitness of the population is calculated as Roulette-wheel is spun 'n' times, each time selecting an instance of the string chosen by the Roulette-wheel pointer. Since the circumference of the wheel is marked according to a string's fitness, the Roulette wheel mechanism is expected to make F_i/F copies of the ith string of the mating pool. The average fitness

$$\overline{F} = \sum_{j=1}^{n} F_j \Big/ n \tag{2.25}$$

Although the above Roulette-wheel selection is easier to implement, it is noisy. A better stable version of the selection operator is sometimes used. After the expected count for each individual string is calculated, the strings are first assigned value exactly equal to the mantissa of the expected count. Thereafter, the regular Roulette-wheel selection is implemented using decimal part of the expected count of the probability distribution. This selection method is less noisy and is known as *stochastic remainder selection.*

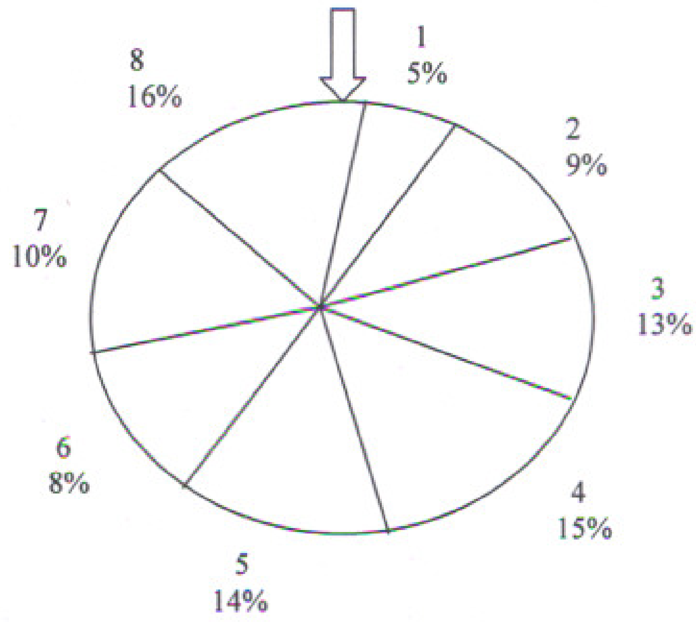


Figure 2.11: Roulette-Wheel Selection

(ii) Tournament Selection

GA uses strategy to select he individuals from population and insert them into a mating pool. Individuals from the mating pool are used to generate new offspring, which are the basis for the next generation. As the individuals in the mating pool the ones whose genes will be inherited by the next generation, it is desirable that the mating pool consists of good individuals. A selection strategy in GA is simply a process that favors the selection of better individuals in the population for mating pool. There are two important issues in the evolution process of genetic search:

- Population diversity means that the genes from the already discovered good individuals are exploited while promising the new areas of the search space continue to be explored.

- Selective pressure is the degree to which the better individuals are favored.

The Roulette-wheel selection is likely to lead to two problems, namely

1. Stagnation of search because it lacks selective pressure, and
2. Premature convergence of the search because it causes the search to narrow down too quickly.

Unlike the Roulette wheel selection, the tournament selection strategy provides selective pressure by holding a tournament competition among N_U individuals (Frequency of $N_U = 2$). The best individual (the winner) from the tournament is the one with highest fitness. Tournament competitors and the winner are then inserted into the mating pool. The tournament competition is repeated until the mating pool for generating new offspring is filled. The mating pool comprising of tournament winner has higher average population fitness. The fitness difference provides the selection pressure, which drives GA to improve the fitness of succeeding genes.

(iii) Rank Selection

The Roulette wheel will have problem when the fitness values differ very much. For example, if the best chromosome fitness is 90%, its circumference occupies 90% of Roulette wheel, then other chromosomes will have very few chances to be selected. Rank selection first ranks the population and taken every chromosome, receives fitness from the ranking. The worst will have fitness 1, the next 2, ..., and the best will have fitness N (N is the number of chromosomes in the population). This method can lead to slow convergence because the best chromosome does not differ so much from the other.

(5) Crossover

The selection mechanisms make clones of good strings, but do not create new ones. Crossover operator is applied to the mating pool with a hope that it would create a better string. The aim of the crossover operator is to search the parameter space. In addition, search is to be made in a way that the information stored in the present string is maximally preserved because these parent strings are instances of good strings during reproduction. There exist many types of crossover operations [20] in genetic algorithm such as:

(i) Single-Point Crossover

In single point crossover, a cross-site is selected randomly along the length of the mated strings and bits next to the cross-sites are exchanged as shown in figure 2.12.

(ii) Two-Point Crossover

In a two point crossover, two random sites are chosen and the contents bracketed by these sites are exchanged between two mated parents as shown in figure 2.13.

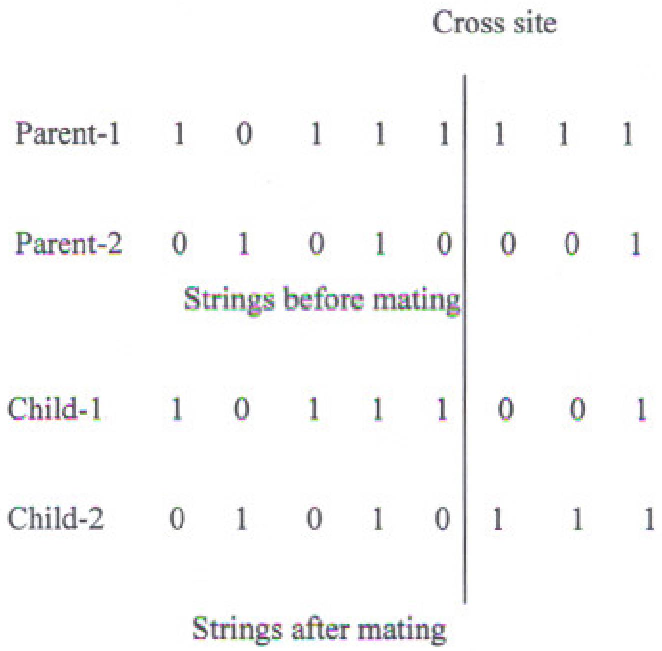


Figure 2.12: Single-Point Crossover

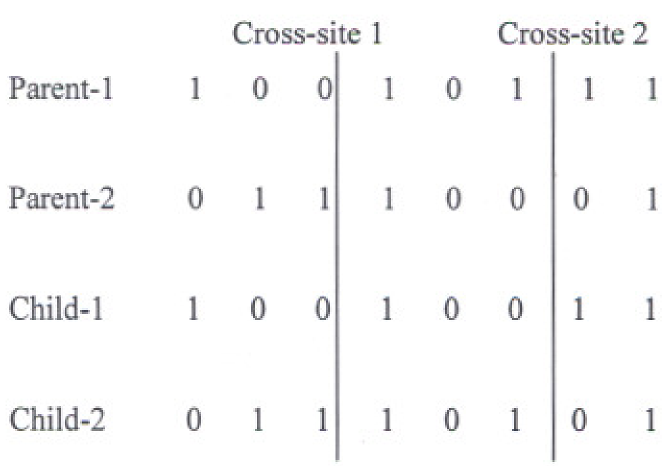


Figure 2.13: Two-Point Crossover

(iii) Multi-Point Crossover

In a multi-point crossover, there are two cases. One is even number of cross-sites and second one is the odd number of cross-sites. In case of even numbered cross-sites, the string is treated as a ring with no beginning or end. The cross-sites are selected around the circle uniformly at random. Now the information between alternate pairs of sites is interchanged. If the number of cross-sites is odd, then a different cross-point is always assumed at the string beginning. The information (genes) between alternate pairs is exchanged.

(iv) Uniform Crossover

Single and multi-point crossovers define cross points as places between loci where a chromosome can be split. Uniform crossover generalizes this scheme to make every locus a potential crossover point. A crossover mask, the same length as the chromosomes structure, is created at random and the parity of the bits in the mask indicates which parent will supply the offspring with which bits. When there is a 1 in the mask, the gene is copied from the first parent and when there is 0, the gene is copied from the second parent, as shown in figure 2.14. The process is repeated with the parents exchanged to produce the second offspring. Offspring therefore contains a mixture of genes from each parent.

Parent-1	1	0	1	0	0	0	1	1	1	0	1
Parent-2	0	1	0	1	0	1	0	0	1	1	0
Mask	1	0	0	1	0	1	1	1	0	0	1
Offspring1	1	1	0	0	0	0	1	1	1	1	1
Offspring2	0	0	1	1	0	1	0	0	1	0	0

Figure 2.14: Uniform Crossover

Crossover Rate: In GA literature, the term crossover rate is usually denoted as P_c, the probability of crossover. The probability varies from 0 to1. This is calculated in GA by finding out the ratio of the number of pairs to be crossed to some fixed population.

(6) Mutation

After crossover, strings are subjected to mutation. Mutation of a bit involves flipping it: changing a 0 to 1 or vice versa. Just as p_c controls the probability of a crossover, another parameter p_m (the mutation rate), gives the probability that a bit will be flipped. The bits of a string are independently mutated - that is, the mutation of a bit does not affect the probability of mutation of other bits. The SGA treats mutation only as a secondary operator with the role of restoring lost genetic material. For example, suppose all the strings in a population have converged to a 0 at a given position and the optimal solution has a 1 at that position. Then crossover cannot regenerate a 1 at that position, while a mutation could.

Mutation Rate: it is the probability of mutation which is used to calculate number of bits to be muted. The simple genetic algorithm uses the population sizes of 30 to 200 with the mutation rates varying from 0.001 to 0.5.

(7) Reinsertion

To maintain the size of original population, the new individuals have to be reinserted into the old population. Similarly, if not all the new individuals are to be used at each generation or if more offspring are generated than the size of the old population then a reinsertion scheme must be used to determine which individuals are to exist in the new population. An important feature of not creating more offspring than the current population size at each generation is that the generational computational size is reduced and the memory requirements are smaller as fewer new individuals need to be stored while offspring are produced. While selecting which members of the old population should be replaced the most apparent strategy is to replace the least fit members deterministically.

(8) Termination of the GA

Because the GA is a stochastic search method, it is difficult to formally specify convergence criteria. As the fitness of a population may remain static for a number of generations before a superior individual is found, the application of conventional termination criteria becomes problematic. A common practice is to terminate the GA after a pre-specified number of generations and then test the quality of the best members of the population against the problem definition. If no acceptable solutions are found, the GA may be restarted or a fresh search initiated.

2.4.5 Working of Genetic Algorithms

Despite the successful use of GAs in a large number of optimization problems, progress on the theoretical front has been rather slow. A very clear picture of the workings of GAs has not yet emerged, but the *schema theory* and the *building-block hypothesis* of Holland and Goldberg capture the essence of GA mechanics [21, 23].

(1) Similarity Template

A schema is a similarity template describing a subset of strings with similarities at certain position. In other words, a schema represents a subset of all possible strings that have the same bits at certain string positions. As an example, consider strings with five bits. A schema **000 represents strings with 0s in the last three positions: the set of strings 00000, 01000, 10000, and 11000. Similarly, a schema 1*00* represents the strings 10000, 10001, 11000, and 11001. Each string represented by a schema is called an *instance* of the schema. Because the symbol * signifies that a 0 or a 1 could occur at the corresponding string position, the schema ***** represents all possible strings of five bits. The *fixed positions* of a schema are the string positions that have a 0 or a 1: in **000, the third, fourth and fifth positions. The number of fixed positions of a schema is its *order:* **000 is of order 3. A schema's *defining length* is the distance between the outer- most fixed positions. Hence, the defining length of **000 is 2, while the defining length of 1 *00* is 3. Any specific string is simultaneously an instance of 2^l schemata (l is the string length).

(2) Competition

Consider a schema with *k* fixed positions. There are 2^k - 1 other schemata with the same fixed positions that can be obtained by considering all permutations of 0s and 1 s at these k positions. Altogether, for k fixed positions, there are 2^k distinct schemata that generate a partitioning of all possible strings. Each such set of k fixed positions generates a *schema competition,* a survival competition among the 2^k schemata. Since there are 2^l possible combinations of fixed positions, 2^l distinct schema competitions are possible. The execution of the GA thus generates 2^l simultaneous schema competitions. The GA simultaneously, though not independently, attempts to solve all the 2^l schema competitions and locate the best schema for each set of fixed positions. We can visualize the GA's search for the optimal string as a simultaneous competition among schemata to increase the number of their instances in the population. If we describe the optimal string as the juxtaposition of schemata with short defining lengths and high average fitness values, then the winners of the individual schema competitions could potentially form the optimal string. Such schemata with high fitness values and small defining lengths are appropriately called *building blocks.*

(3) Schema Theorem

When we consider the effects of selection, crossover and mutation on the rate at which instances of a schema increase from generation to generation, we see that proportionate selection increases or decreases the number in relation to the average fitness value of the schema. Neglecting crossover, a schema with a high average fitness value grows exponentially to win its relevant schema competition. However, a high average fitness value alone is not sufficient for a high growth rate. A schema must have a short defining length too. Because crossover is disruptive, the higher the defining length of a schema, the higher the probability that the crossover point will fall between its fixed positions and an instance will be destroyed. Thus, schemata with high fitness values and small defining lengths grow exponentially with time. This is the essence of the schema theorem, first proposed by Holland as the "fundamental theorem of genetic algorithms" [24] described by equation 2.26.

$$N(h,t+1) \geq N(h,t)\frac{f(h,t)}{\bar{f}(t)}\left[1 - p_c \frac{\delta(h)}{l-1} - p_m o(h)\right] \quad (2.26)$$

where

f (h,t) = average fitness value of schema h in generation t

f (t) = average fitness value of the population in generation t

p_c = crossover probability

p_m = mutation probability

δ (h) = defining length of the schema

o (h) = order of the schema h

N (h,t) = expected number of instances of schema h in generation t

l = number of bit positions in a string

The factor p_c (δ(h)/l − 1) gives the probability that an instance of the schema h is disrupted by crossover, and $p_m o(h)$ gives the probability that an instance is disrupted by mutation. The GA samples the building blocks at a very high rate. In a single generational cycle the GA processes only P strings (P is the population size), but it implicitly evaluates approximately P^3 schemata. This capacity of GAs to simultaneously process a large number of schemata, called implicit parallelism, arises from the fact that a string simultaneously represents 2^l different schemata.

2.4.6 Pros and Cons of GA

In contrast to more traditional numerical techniques, which iteratively refine a single solution vector as they search for optima in a multi-dimensional landscape, genetic algorithms operate on entire populations of candidate solutions in parallel. In fact, the parallel nature of a GA's stochastic search is one of the main strengths of the genetic approach. This parallel nature implies that GAs are much more likely to locate a global peak than traditional techniques, because they are much less likely to get stuck at local optima. Also, due to the parallel nature of the stochastic search, the performance is much less sensitive to initial conditions, and hence and a GA's convergence time is rather predictable. In fact, the problem of finding a local optimum is greatly minimized because GAs, in effect, makes hundreds, or even thousands, of initial guesses. This implies that a GA's performance is at least as good as a purely random search [25]. In fact, by simply seeding an initial population and stopping there, a GA without any evolutionary progression is essentially a Monte Carlo simulation.

As appealing as a GA may seem, the parallel nature of the stochastic search is not without consequences. Although the prospects of finding global optima make it robust, the convergence of a GA is usually slower than traditional techniques. In fact, with a good initial guess close to the global optimum, a numerical technique will likely be much faster, and more accurate, than a genetic search because, in essence, the GA will be wasting time testing the fitness of sub-optimal solutions. Furthermore, due to the stochastic nature of a GA, the solution, although more likely to estimate the global optimum, will only be an estimate. Users must realize that GAs will only by chance find an exact optimum, whereas traditional methods will find it exactly [25].

2.4.8 Potential Applications of GA

Amounts of applications have benefited from the utilization of genetic algorithms. Some of the current applications [26] involve:

(1) Pattern Recognition Applications

In any pattern recognition application, a minimum feature set which yields maximum classification accuracy is desirable. Genetic algorithm is used to search the space of all possible subsets of the complete set which includes all features. The feature set is coded as a binary string which each bit represents presence or absence of a particular feature. The classification accuracy will be used as fitness. Genetic algorithm gives better classification accuracy than important score method with the expenses of a higher number of iterations required and a higher number of features required in the reduced feature set. Genetic algorithm can also be used to select a new feature set which combines original features and new features constructed by applying arithmetic operations (such as +, -, *, /) to the original features.

(2) Robotics and Artificial Life Applications

Steady-state genetic algorithms have been used to evolve sensory characteristics of artificial organism in an environment with controlled complexity. The environment model used is called a latent energy environment (LEE). Feed-forward neural networks are used to simulate organisms. Two types of sensors are interested in this study: contact and ambient. Contact sensors are presented in the organism which is required to learn avoidance tasks. Reinforcement learning is used to train motor actions which are the outputs from neural network. Ambient sensors are presented in organism which approaching task is required. Back-propagation learning is used to train sensory prediction outputs of the network. Any changes in motor characteristic of this type of organism can only be achieved via evolution. Steady-state genetic algorithm used in this study as follows: Each individual, represented by neural network, must acquire energy from atoms in the environment beyond a fixed threshold before it can asexually reproduce. If the energy level within an individual is lower than a threshold, that individual will die. Chromosome of each individual contains two parts, one in floating-point format, and the other in binary format. The connection weights in neural network are coded into the floating-point section of chromosome. Mutation is done by randomly added uniformly distributed noise to the chromosome. The types of atoms the sensors sensed are coded into the binary part. Bit-flip mutation is used here.

(3) Expert System Applications

The objective of testing an expert system is to find input combinations which will cause the expert system to give inappropriate responses. Changes can then be made to the expert system. It is exhaustive to test an expert system with all possible input combinations. Genetic algorithm can be used to generate test inputs to the expert system. This results in an optimal number of test cases which yield a good coverage of all possible input combinations to the expert system. In this case, an expert system is used to control electricity output of a power station. Plant parameters are used to code chromosomes. The fitness function used is based on the plant heat rate. Plant heat rate is the number of British Thermal Units (BTUs) required to produce one Kilowatt-hour of electricity. A combination of plant inputs and environment inputs which makes the expert system to respond inappropriately and increases the plant heat rate will result in high fitness.

(4) Electronic and Electrical Applications

Genetic programming can be incorporated in electronic circuit design. Both topology and components' value in a circuit will be automatically determined by a pool of evolving programs. These evolving programs will undergo a genetic programming evolution which includes reproduction, crossover and mutation. In order to apply genetic programming to a circuit design, electrical circuits must be mapped to program trees. Each program tree contains two main types of function: connection- modifying functions which modify the topology of the circuit and component-creating functions which insert electronic components into locations within the topology of the circuit. The examples of circuit designed by using genetic programming are crossover (woofer and tweeter) filter, low pass filter, double band-pass filter, amplifier circuit and food-forging controller for simulating behavior of a lizard.

(5) Applications in Biology and Medicine

Genetic algorithm is used to model an evolution in an immune system. Chromosome of each individual represents libraries of genetic material in the immune system. These genetic materials are used to construct antibodies which are responsible for recognizing antigens. Unlike other applications using genetic algorithm, phenotype of an individual which is represented by antibodies produced is not a complete mapping from every gene in the chromosome. Chromosome is divided into four libraries of genetic material. Each library contains eight elements. An antibody is produced by combing one element from every library. This study has demonstrated that genetic algorithm is capable of improving fitness of the population even only partial information about each individual is given to the algorithm during each generation.

2.4.9 Advances in GA

Recently, new and efficient crossover operators have been designed so that search along variables is also possible. Let us consider $X_i^{(j)}$, $X_i^{(k)}$ values of design variables X_i in two-parent string j and k. the crossover between these two values may produce the new value as [20]

$$X_i^{new} = (1 - \lambda) X_i^{(j)} + \lambda X_i^{(k)} \quad (2.27)$$

Here, the parameter is a random number between 0 and 1. The above equation calculates new value bracketing the above two-parent values. This calculation is performed for each variable in the string. This crossover has uniform probability of creating a point inside the region bounded by two parents.

(1) GA with Memory

The binary tree structure provides a way to store previous designs which permit efficient search of duplicate designs [20]. Figure 2.15 shows the pseudo code for the calculation of the objective function using binary tree. After a new generation of design strings is created, the binary tree is searched for the new design. If the design is found, the objective function value is obtained from the tree without analysis; otherwise the tree is searched for the design with identical parameters. New designs and their relevant data are then inserted into the binary tree.

```
begin
    search for the given design in binary tree
    if found
        Get objective function value from tree
    else
        Search for the design having identical parameters
    if found
        Get normalized values from the tree
    else
        Perform analysis
    end if
        Add design to binary tree
    end if
end
```

Figure 2.15: Evaluation of Objective Function Using Binary Tree

(2) Multi-Modal Optimization

Many real world problems contain multiple solutions that are optimal or near optimal. The knowledge of multiple optimal solutions in a problem provides flexibility in choosing alternate yet good solutions as and when required. In nature, for example, we recognize that multiple niches (humans and animals) exist by sharing available resources. A similar sharing concept is introduce artificially in GA population in sharing functions as given by Deb [22] which calculate the extent of sharing that needs to be done between two strings. If d_{ij} is the distance between the i^{th} and j^{th} string, then

$$Sh(dij) = \begin{cases} 1 - dij / \sigma & \text{if } dij < \sigma \\ 0 & \text{otherwise} \end{cases} \qquad (2.28)$$

Here, the parameter is the maximum distance between two strings for them to be shared and is fixed before hand. Shared fitness (Sh) is calculated and this is used for reproduction. Using this sharing strategy, GAs have solved a number of multi-modal optimization problems.

(3) Searching for Optimal Schedule

Job shop scheduling, time tabling and traveling salesman problems are solved using GA. A solution in these problems is a permutation of N objects (names of machines or cities). Although reproduction operator is based on fitness function which is nothing but the distance traveled by salesman, the crossover and mutation operators are different. The operators are designed to produce offspring which are valid and yet have certain properties of both parents as suggested by Goldberg.

(4) Multi-Objective Optimization

There are many objective functions in multi-objective optimization. The usual practice is to convert multiple objectives into one objective function as

$$\Phi = W_1 f_1 + W_2 f_2 + ... + W_n f_n \qquad (2.29)$$

where W_1, W_2 are the weights and f_1, f_2, ……, fn are multi-objective functions. Equation is one way of converting multi-objective function into single-objective optimization problem. The solution of multi-objective optimization problem can be considered as a collection of optimal solutions obtained by solving different single-objective functions formed using different weight vectors. These solutions are known as Pareto-optimal solutions. This can be solved using the concept of non-dominated sorting of population members.

CHAPTER –3
SIMULATION RESULTS ON FOUR WAVE MIXING

Four-wave mixing is analogous to inter-modulation distortion when optical signals with two or three frequencies (or wavelengths) interact through the third-order electric susceptibility of optical fiber. FWM generates new optical signals at other frequencies (or wavelengths) and creates signal depletion to pre-assigned channels in WDM systems. The efficiency of FWM depends on both channel spacing and fiber dispersion. Greater group velocity mismatch between the original and FWM signals caused by fiber dispersion or larger channel As a result, with the use of DSF in high-capacity, long-haul, repeater-less WDM systems, the efficiency of FWM increases and FWM crosstalk becomes most likely the dominant nonlinear effect.

3.1 FWM versus Dispersion

Equation 2.13 gives the frequency of radiation that would be generated by FWM, should the process occur. It does not provide any information as to whether the process will occur or not, this is dictated by the *phase matching condition*. Phase matching requires that the propagation constants of the waves satisfy a relation similar to (2.13) that ensures momentum is conserved before and after the interaction. In dispersive media, such as silica, light at different frequencies propagates at different speeds as opposed to a non-dispersive vacuum in which all frequencies propagate at the same speed. In order to achieve phase matching in an optical fiber, the dispersion of the fiber must be such that the frequencies involved in FWM also have the correct propagation constants to satisfy the phase matching condition.

The small core diameters of single mode optical fibers (typically ~ 5 to 10 μm) permit high optical intensities to be maintained over long distances and therefore fibers offer excellent conditions for non-linear effects to occur. Phase matching for four-wave mixing is dependent on the relative group velocities of the interacting signals, which is determined by the dispersion of the fiber and will be a function of the frequencies of the signals and possibly their relative polarization. Dispersion causes the frequency or polarization components of an optical signal to propagate at different velocities through the fiber - leading to pulse spreading and a consequent reduction in the maximum signal bandwidth. In the interests of maximizing the information-carrying capacity of an optical fiber link, it is desirable to reduce dispersion to the smallest possible value at the transmission wavelength. However, four-wave mixing becomes most efficient under these circumstances due to increased phase matching and can become a serious limitation under some circumstances [5].

3.1.1 Chromatic Dispersion

Optical radiation propagating in bulk materials experiences a frequency-dependent refractive index and therefore a frequency-dependent velocity through the medium. For modulated optical signals and pulses with non-zero spectral width, it is the group velocity, v_g that describes propagation of the signal through the medium. The propagation of the signal may also be described in terms of the group delay per unit length, $\tau_g = 1 / v_g$ which is also a function of optical frequency, ω, or alternatively optical wavelength, λ. The derivative of the group delay with respect to wavelength in homogenous media is referred to as the material dispersion.

For waves propagating along the core of an optical fiber, the material dispersion is accompanied by dispersion due to the interaction of the optical field with the cladding material beyond the core-cladding interface. This waveguide dispersion is always negative and adds to the material dispersion to give the overall chromatic dispersion of the fiber, D_c. For the purposes of studying phase matching, the modal propagation constant of a single mode fiber is often expanded as a Taylor series about one of the interacting frequencies, such as ω_3. The propagation constant difference due to chromatic dispersion $\Delta\beta = \beta(\omega_1) + \beta(\omega_2) - (\beta\,\omega_3) - (\beta\,\omega_4)$ can then be expressed as [1]

$$\Delta\beta = \frac{\lambda_3}{\omega_3}\Delta\omega_{13}\Delta\omega_{23}\left[D_c(\omega_3) + \left(\frac{1}{2\pi\omega_3}\right)(\Delta\omega_{13} + \Delta\omega_{23})\frac{dD_c(\omega_3)}{d\lambda}\right] \tag{3.1}$$

where $\Delta\omega_{ij} = |\,\omega_i - \omega_j\,|$ and is the vacuum wavelength of radiation with angular frequency ω_3.

The overall chromatic dispersion of a fiber can be engineered to change sign from negative to positive near a particular wavelength by balancing chromatic and waveguide dispersion to produce *dispersion shifted fiber*. At the zero dispersion wavelength, λ_0, the change in group delay with respect to wavelength is zero and signals at wavelengths close to λ_0 will propagate with almost identical group velocities eliminating pulse spreading and maximizing fiber bandwidth.

3.1.2 Polarization Mode Dispersion

For a given length of fibre, the random local birefringence leads to an output polarization that varies rapidly when the frequency of polarized launched radiation is varied except for two states of launch polarisation known as the *principle states of polarization* (PSP). Radiation launched into the fibre with either of these polarization states remains in this single polarization state throughout the entire fibre and consequently emerges with minimum pulse distortion. The difference in group velocity between the two PSPs is generally the difference in velocity between the fastest and slowest pulses in the fibre and the RMS delay between them is known as the *polarization mode dispersion* (PMD) of the fibre. Optical power may be coupled between the two principle states of polarization by fibre imperfections with spatial frequency proportional to the group velocity mismatch between them. This polarization mode coupling causes an optical pulse launched with arbitrary polarization to propagate with a group velocity that lies somewhere between those of the two principle states of polarization. The random nature of the coupling causes the optical pulse to emerge temporally broadened – reducing the maximum transmission bit rate for a given BER.

By deliberately removing the cylindrical symmetry of a single mode fibre, it is possible to produce two polarization states that are sufficiently non-degenerate that typical deviations in the fibre geometry are insufficient to cause coupling between them. Under these circumstances, light launched into one of the modes will remain in that mode along the entire length of the fibre. This is the basis of the highly birefringent *polarization maintaining fibre* (PMF) in which the two principle polarization modes have quite different group velocities. The group velocity mismatch that can be achieved by launching radiation into the orthogonal modes of highly birefringent polarization maintaining fibre may be used to achieve phase matching for four-wave mixing when chromatic and non-linear dispersion are insufficient. Since the birefringence of the fibre is dependent on the geometry, it may be tuned using a number of methods including lateral compression, bending, or varying the temperature of the fibre.

3.1.3 Phase Matching

The phase matching conditions for maximum parametric gain to occur in an optical fibre require consideration of the non-linear effects of self-phase modulation and cross-phase modulation on the intense pump signals as these effects will tend to modulate the phase of the pump. When these effects are included in the analysis [5], the phase matching conditions for non-degenerate and partially degenerate four-wave mixing are given by equations 3.2 and 3.3 respectively:

$$k = \Delta\beta + \gamma\,(p_1(0) + p_2(0)) = 0 \tag{3.2}$$

$$k = \Delta\beta + \gamma\, 2p_1(0) = 0 \tag{3.3}$$

The parameter γ is the non-linearity coefficient of the fibre and $p_1(0)$ and $p_2(0)$ are the incident pump optical powers at the fibre input. It is assumed here that the optical power at the pump wavelengths is unaffected by the transfer of power to the signal wavelengths. This is known as the *undepleted pump* or *small-signal* regime. The optimum phase mismatch for the case of a depleted pump wavelength has been found to be four times smaller than in equation 3.10 for the partially degenerate case. The non-linearity coefficient is related to the third-order susceptibility through the second-order refractive index, n_2, such that

$$\gamma = \frac{n_2\,\omega}{A_{eff}\,c} = \frac{3}{8n}\mathrm{Re}(\chi_3) \tag{3.4}$$

where ω is the average optical frequency of the four signals, A_{eff} is the effective area of the fibre core, n is the linear refractive index of the medium and $\mathrm{Re}(\chi_3)$is the real part of the third-order susceptibility. The phase mismatch in equation 3.2 and 3.3 can be separated into contributions due to material dispersion, $\Delta\beta_M$, waveguide dispersion, $\Delta\beta_W$, and non-linearity, $\Delta\beta_{NL}$, such that the condition for phase matching becomes [5]

$$k = \Delta\beta_M + \Delta\beta_W + \Delta\beta_{NL} = 0 \tag{3.5}$$

In standard single mode fibres, the waveguide dispersion is very small as all modes experience almost identical waveguide parameters and phase matching must therefore be achieved by using combinations of the material dispersion and non-linear effects only. However, polarization-maintaining fibres do have non-zero waveguide dispersion due to the mismatch between the propagation constants of the two polarization modes and therefore phase matching may be achieved by balancing all three mismatch terms. Phase matching in single mode optical fibres can therefore be achieved using any of the following conditions:

1) At wavelengths below the zero dispersion wavelength of the fibre, i.e. in the *normal dispersion regime*, the phase mismatch due to material dispersion is positive, so a negative value of $\Delta\beta_W$ is required. This can be achieved using the two orthogonal modes of polarization maintaining fibre at optical powers low enough not to cause non-linear effects.

2) At wavelengths close to λ_0, the material dispersion is reduced to very low levels comparable with the waveguide dispersion. In this regime, phase matching may be achieved for low optical power levels, when $\Delta\beta_{NL} = 0$.

3) Above the zero dispersion wavelength of the fibre, i.e. in the *anomalous dispersion regime*, the phase mismatch due to material dispersion is negative and may be compensated by a correspondingly positive non-linear phase mismatch component, $\Delta\beta_{NL}$, by increasing the optical powers of the pump signals.

3.2 Simulation

The effects of dispersion on four wave mixing are studied using Optsim, the Optical Simulation tool. OptSim is a high-end optical system simulator for professional engineering and cutting-edge research of WDM, CATV and other emerging optical systems. OptSim is designed to combine the greatest possible accuracy and modeling power along with extreme ease-of-use on UNIX and Windows NT operating systems. OptSim includes the latest simulation algorithms to guarantee the highest possible accuracy and real-world results. OptSim includes an extensive component model library of the most commonly used components for the engineering of electro-optical systems, with particular emphasis on WDM and digital CATV systems. It also supports innovative optical approaches such as quasi-RZ and dispersion-managed soliton systems. OptSim architecture is composed of the following main parts:

- a graphical editor window that is the main interface for the user, it allows the creation and the simulation of network designs
- a data post-processing and display system that allows the display, the manipulation and comparison of simulation results

With the editor we set up an OptSim project, i.e. the design network to be simulated, drawing the schematic and setting up the simulation parameters and the component parameters. After that we can simulate the system created and view the simulation results launching the data display tool.

3.2.1 Simulation Techniques

In order to optimize both the accuracy and the computational effort, OptSim supports an incremental simulation approach with two different simulation techniques:

- The spectral propagation technique (SPT) where only the power spectrum is propagated and the linear effects are considered. Only the optical components are simulated.
- The variable bandwidth simulation technique (VBS) where the full vector signal is propagated, all linear and non-linear effects are considered and all system components are simulated

3.2.2. Simulation Setup

The setup for studying the effect of dispersion on four-wave mixing is shown in figure 3.1. Two WDM channels are launched over two dispersion shifted fiber spans of 100 km each. Dispersion is completely compensated at each span by fiber grating compensators. Fiber dispersion is varied from 0 to 4 ps/nm/km through parametric runs. The power of the FWM products is measured by the optical power meter. The continuous wave laser is used as source. Sin^2 modulators are used for the modulation of the input signal. The drivers used are Non-Return to Zero cosine drivers.

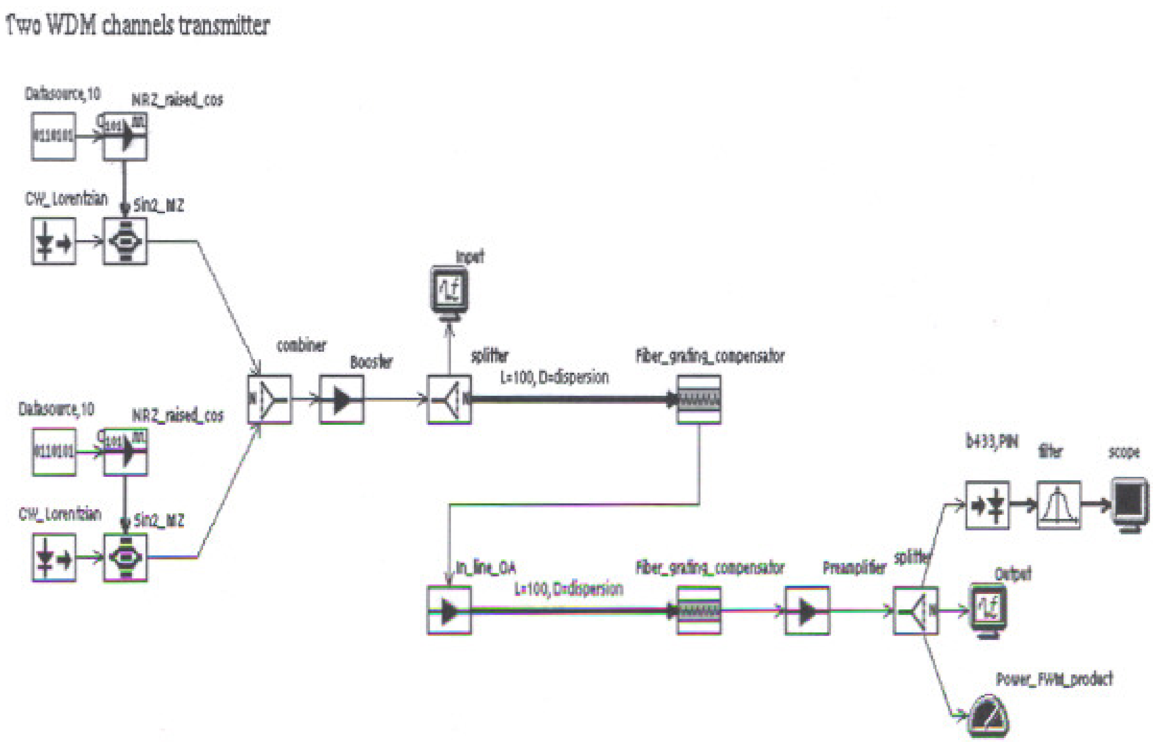


Figure 3.1: Simulation Setup for studying the effect of dispersion on four wave mixing

3.2.3 Parameters used

Dispersion = Run 1: 0 ps/nm/km

Run 2: 1 ps/nm/km

Run 3: 2 ps/nm/km

Run 4: 3 ps/nm/km

Run 5: 4 ps/nm/km

f_1 = 193.025 THz

Δf = 0.05 THz

3.2.3 Components used in the setup

(1) Data Source

This component simulates a pseudo-random or a deterministic logical signal generator. Baud rate, sequence length and logical signal level (number of bits per symbol) can be customized. When the logical signal level is greater than 1, the generated serial sequence is loaded into the output logical signal after a serial-to-parallel conversion.

Parameters used:
Bit Rate: 10 Gbit/s
Baud Rate: 10 Gbaud/s
Samples per bit: 31
Number of bits per symbol: 1
Sequence: Pseudorandom
Generating polynomial: Random

(2) CW Lorentzian Laser

This model implements a simplified continuous wave (CW) laser. Laser phase noise is taken into account by generating a Lorentzian emission line shape whose FWHM (Full Width Half Maximum) is specified by the parameters. Two options are available for laser phase noise bandwidth: ideal (infinite bandwidth) or realistic (bandwidth-limited).

Parameters used:
Laser phase: Deterministic
Initial laser phase: 0.3 rad
Laser noise bandwidth: Ideal
Line width: FWHM = 10 MHz
Full width at -20 dB = 99.498 MHz
Center emission frequency: f_1 THz for laser 1
$f_1 + \Delta f$ THz for laser 2.

(3) NRZ Cosine Driver

A driver converts logical input signal, a binary sequence of zeros and ones into an electrical signal. This component simulates the NRZ raised cosine driver. The output signal has two levels: one for ones and the other for zeros. Switching between the two levels is not instantaneous: it follows a raised cosine shape with a given roll-off. The difference between this driver and the RZ raised cosine pulse driver lies in the fact that in the RZ modulation the signal is forced to return to the "0" level at the end of the each bit, also if there are two consecutive bits at the "1" level. The raised cosine waveform, when the driver is connected to a linear optical modulator, can shape either the optical amplitude or the optical power.

Parameters used:

Signal Dynamics: Low level = 0

High level = 5

Resulting raised cosine shape through linear modulator is in: Power

(4) Sin^2 Amplitude Modulator

This model implements a single arm Mach-Zehnder Amplitude Modulator with sin^2 electrical shaped Input-Output P-V characteristic. This transfer function is typical for a Mach-Zehnder external modulator based on the electro-optic effects in the $LiNbO_3$ devices. This model also incorporates the spectral characteristics of the device adding a filter-section at the electrical input. The applied filter has the following transfer function:

$$H(f) = \frac{\sin\left(\frac{\pi f}{B_0}\right)}{\frac{\pi f}{B_0}}$$

where $B_0 \approx 2.25 \cdot B_W$

Parameters used:

Excess loss (dB): 0

Maximum Transmissivity offset voltage Von: 5V

Extinction ratio: Ideal

Chirp factor: 0

Voltage swing V_π: 5 volts

(5) **Optical combiner**

This component simulates an optical combiner.

Parameters used:

Attenuation on each input (dB): 0

Sum correction factor: Peak power = 0 dB

Average power = 0 dB

(6) Booster (Fixed output power amplifier)

The fixed output power model set the amplifier gain so to have a total output power equal to a given fixed value, unless the resulting gain is greater than the specified small signal gain, which turns out to have also the meaning of maximum achievable gain for this model. In the wavelength dependent model, the gain is reduced by a scalar value to satisfy the fixed output power constraint.

Parameters used:
Output power: 6 dBm
Gain shape: Flat
Maximum small signal gain: -35 dBm

(7) Fixed Gain Optical Amplifier

The fixed gain model is the simplest available but can be useful in several different situations. For example, while working on a simulation where the attention is not focused on amplification problems, but to fiber transmission properties. In this case, a very simple model for the EDFA is sometimes preferred in order to easily fix power levels and gains along the link. Saturation effects are not taken into account in this model, so that the gain and the noise figure are selected as constant scalar numbers (flat case) or as a function of the wavelength (through an external file).

Parameters used:
Gain (dB): 20
No noise

(8) Optical Splitter

This component simulates an optical splitter.
Parameters used:
Attenuation on each output: 0 dB

(9) Ideal Fiber Grating Compensator

This component simulates a fiber grating device used to compensate, over a fiber span L km long, the accumulated dispersion DL or β_2L and its slope ($\partial D/\partial \lambda$)L or ($\partial \beta_2/\partial \omega$)L. An ideal fiber grating introduces a certain amount of dispersion, without influencing the optical power spectrum if the bandwidth of the channels is neglected, like in SPT simulations. Therefore, in SPT simulations, the ideal fiber gratings do not influence the optical power spectrum.

Parameters used:
Reference frequency: 193.05 THz
Total dispersion slope to compensate at reference frequency: -5.937 ps/nm^2

(10) Optical Spectrum Analyzer

This component simulates an optical spectrum analyzer. The spectrum is estimated using the method of the modified periodograms, i.e. sectioning the entire data sequence into a number of parts each containing N_P samples, where N_P is computed in order to guarantee the number of points requested by the Number of Spectrum Points over the Simulation Bandwidth parameter. The amplitudes squared of the Fourier transforms of each part are then summed up and averaged.

Parameters used:
Number of spectrum points over simulation bandwidth: 2000 for input analyzer
1000 for output analyzer

(11) Optical Power Meter

This component simulates an optical power meter: it evaluates the power, defined as the mean square value, of an optical signal. The evaluation can be performed over the whole time domain simulation bandwidth or in a selected subrange of frequencies (Limited Bandwidth Optical Power Meter).

Parameters used:
Measure bandwidth: Limited
Center frequency: $f_1 + 2*\Delta f$ THz
Bandwidth: 10 GHz

3.3 Results and Discussion

The input to the system is shown in figure 3.2 and the respective outputs for different runs are shown in figure 3.3. The spectrum of the received signals show that the FWM products decrease with increasing dispersion. The input spectra starts at -30 mW/THz and the maxima lie at 20 mW/THz. It is clear from figure 3.3 that as the number of runs increases, the voltage level reduces. For the lower frequency range the voltage level reduces while the maxima in the middle of the frequency range remains at the same value. In the higher frequency range the voltage level again begins to fall with increasing runs. The number of runs here conveys the dispersion value in ps/nm/km as given in section 3.2.3. Thus, increasing the number of runs enhances the dispersion value, which reduces the chances of phase matching conditions to be satisfied and consequently the power of the four wave mixing products gradually decreases.

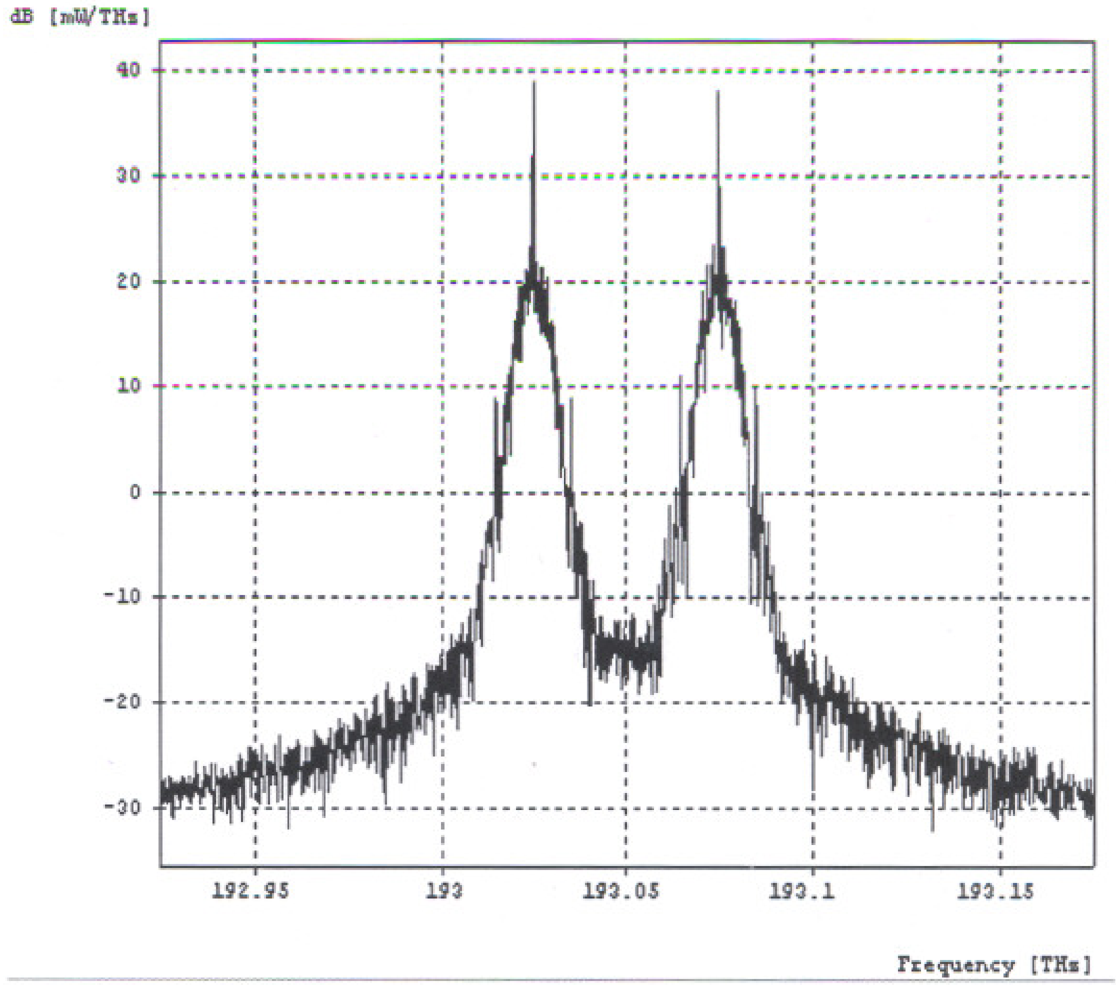
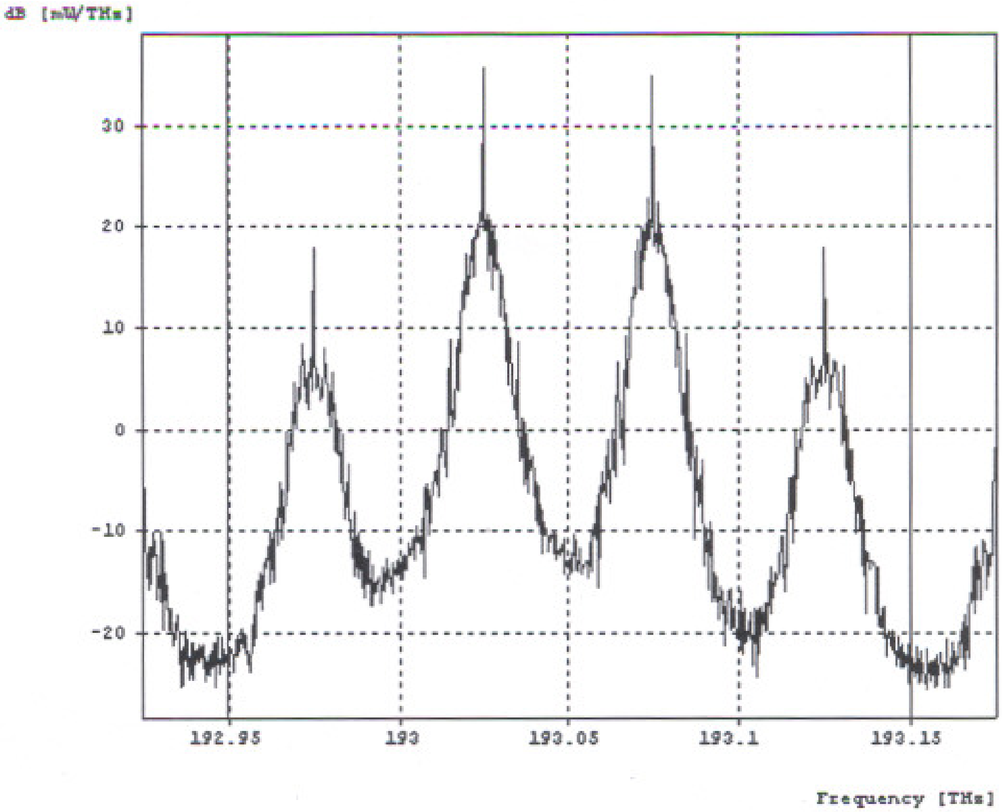


Figure 3.2: Input to the Setup

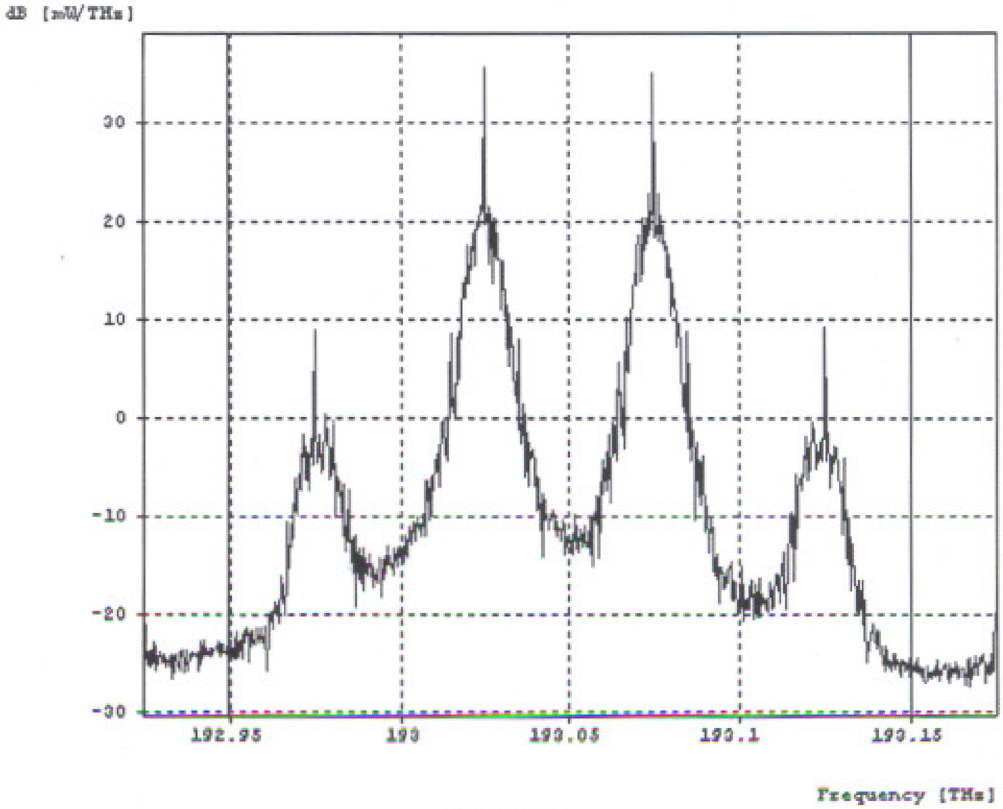
In the first run, the dispersion value is 0 ps/nm/km. The starting voltage level of the signal is at -10 mW/THz at the initial frequency. As the frequency increases the signal rises at about 5 mW/THz and finally at about 20 mW/THz. The signal repeats itself further.

An increase of dispersion value in run 2 reduces the initial value to -25 mW/THz, increasing gradually to 2 mW/THz in the middle, the maxima being the same at 20 mW/THz. A further increase in dispersion lowers the signal level more. Thus in run 3, the signal level goes upto even -8 mW/THZ in the middle range, the initial and the higher values remaining the same as run 2.

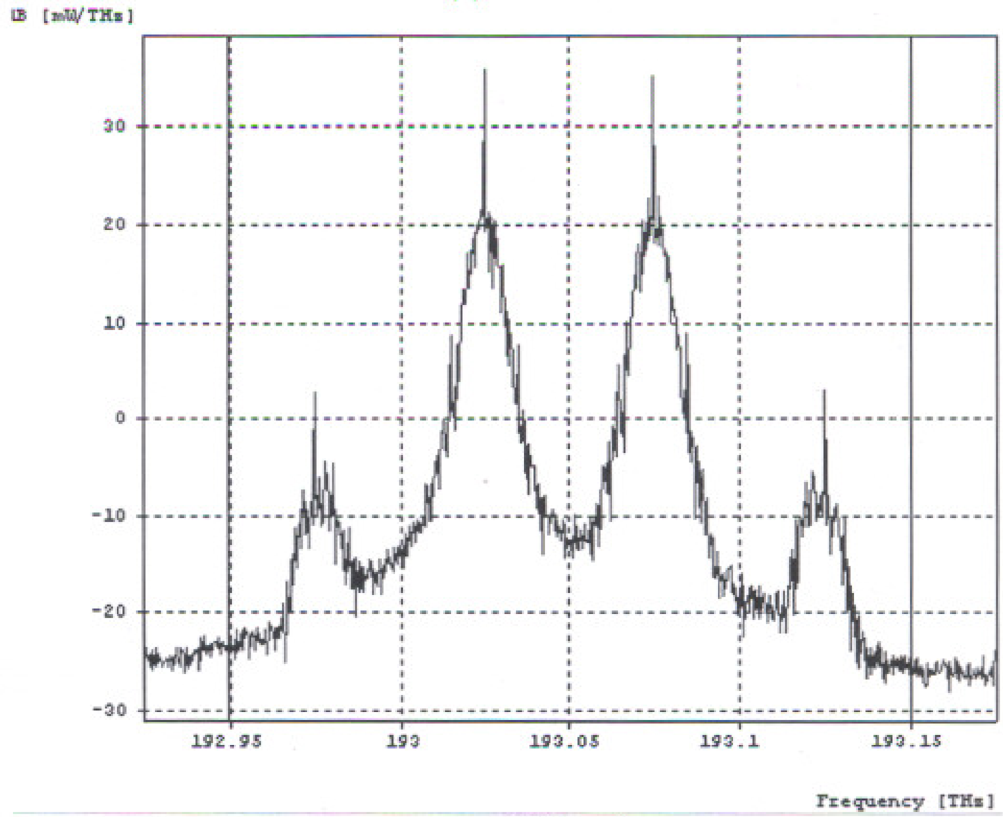
The run 4 provides the initial value of the signal the same as that of run 2 and run 3, but the middle range value is again decreased, resulting in a value of -11 mW/THz. In the last run this value goes upto -13 mW/THz.



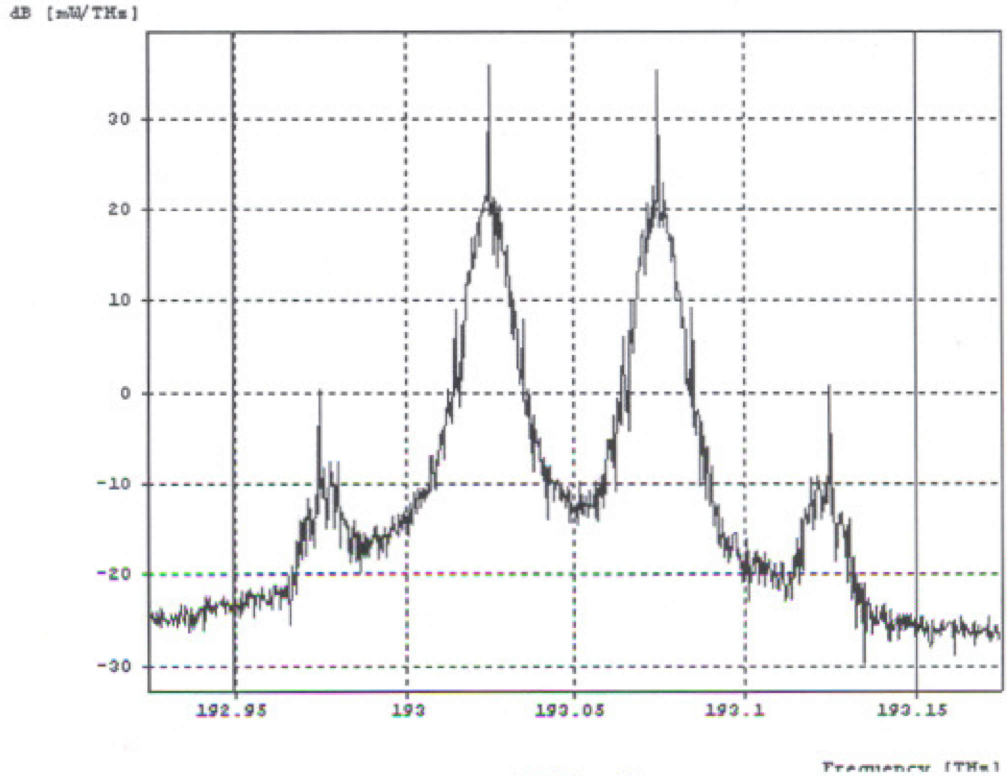
(a) Run 1


dB [mW/THz]
30
20
10
0
-10
-20
-30
192.95
193
193.05
193.1
193.15
Frequency [THz]

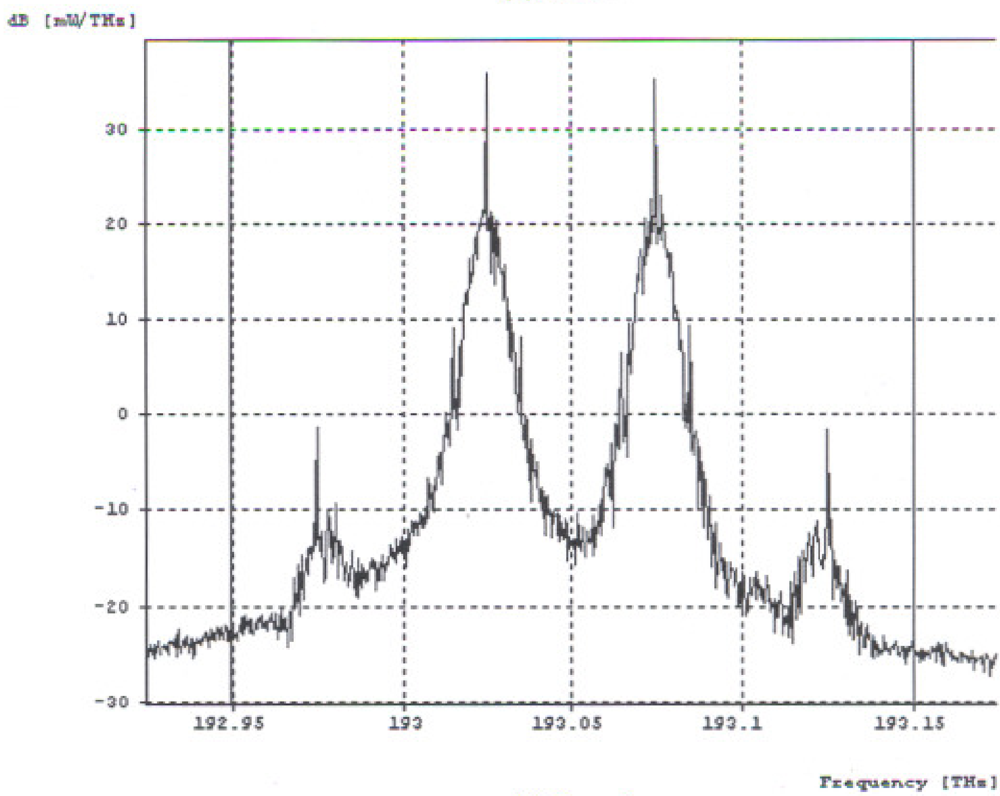
(b) Run 2


dB [mW/THz]
30
20
10
0
-10
-20
-30
192.95
193
193.05
193.1
193.15
Frequency [THz]

(c) Run 3



(d) Run 4



(e) Run 5

Figure 3.3: Output Spectra for different runs

Table 3.1: Effect of Dispersion on the Power of FWM Products

RUN	DISPERSION	POWER (dBm)	POWER(mW)
1	0	-13.157	0.483E-01
2	1	-21.755	0.668E-02
3	2	-27.491	0.178E-02
4	3	-30.400	0.912E-03
5	4	-32.474	0.566E-03

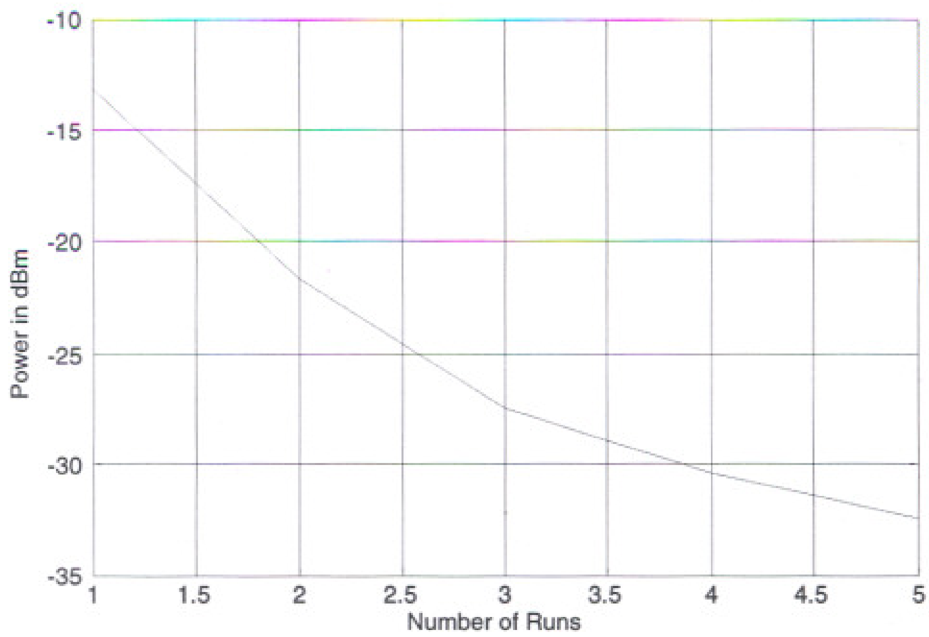


Figure 3.4: Effect of Dispersion on the Power of FWM Products

CHAPTER –4

GOLOMB RULER ALGORITHMS

An n- mark Golomb ruler is a set of n distinct nonnegative integers (a_1, a_2, ..., a_n) called marks, such that the positive differences | a_i - a_j |, computed over all possible pairs of different i,j = 1,.....n with $i \neq j$ are distinct. Let a_n be the largest integer in an n-mark golomb ruler. Then an optimal Golomb ruler with n marks (0,......, a_n) is an n-mark Golomb ruler if

1. There exists no other n-mark Golomb ruler having smaller largest mark a_n, and
2. The ruler is written in canonical form as the 'smaller' of the equivalent rulers (0, a_2,......,a_n) and (0,.....$a_n - a_2$,a_n), where smaller means the first differing entry is less than the corresponding entry in the other ruler.

4.1 Problem Formulation

Assume that N channels (i.e., frequencies or wavelengths) are used in a given WDM system. For any three co-propagating optical signals with frequencies f_i, f_j and f_k and the new frequencies f_{ijk}'s generated by FWM are represented by

$$f_{ijk} = f_i + f_j - f_k \tag{4.1}$$

for i, j, k ϵ [1, N] and $k \neq \{i, j\}$ [4,27]. Considering all the possible permutations, nine new optical signals are generated by these three co-propagating signals. Those new signals, which overlap with the original ones, are considered as crosstalk and will interfere with the normal operation of the WDM channels.

The unequal-spaced channel-allocation design begins with the division of the available optical bandwidth into equal frequency slots of width Δf [27] as shown in Fig. 4.1. Let f_0 be the center frequency of the first channel and $f_i = f_0 + n_i \Delta f$ be the center frequency of the i^{th} channel (or slot), where the integer n_i represents the slot number of the i^{th} channel and N is the total number of channels. In addition, $m_i = n_{i+1} - n_i$ is defined as the channel spacing (in integer) between the i^{th} and $(i+1)^{th}$ channels for i = {1, 2, 3, ..., N-1}. Therefore, the new frequencies f_{ijk}'s created by FWM in (4.1) can equivalently be written in terms of slot number n_{ijk} so that

$$n_{ijk} = n_i + n_j - n_k \qquad (4.2)$$

for i, j, k ϵ [1, N] and $k \neq \{i,j\}$. In other words, to ensure that no FWM signals can fall on the pre-assigned WDM channels, the channel-allocation problem can be treated as finding a set of distinct slot numbers so that $n_{ijk} \neq \{n_1, n_2, n_3, \ldots\ldots., n_N\}$. As suggested by Forghieri *et al.*, the problem can be modified, according to the following lemma [16]:

Lemma: To prevent FWM signals from falling onto the channels of an unequal-spaced WDM system, the frequency separation between any two channels must be distinct.

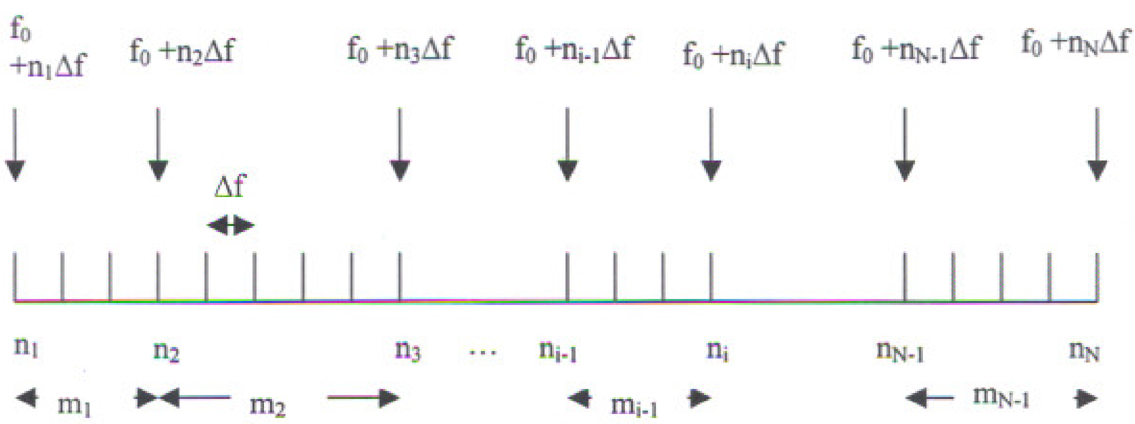


Figure 4.1: Unequal Spaced Channel Allocation Design of a WDM System

By the lemma, this channel allocation problem resembles the construction of optical orthogonal codes (OOC's) in optical code division multiple access (CDMA). As in constructing OOC's, the autocorrelation property of a codeword is inspected by considering the relative time delay (or separation) between any two consecutive pulses. All of the linear combinations of jointly connected relative delays are constructed and checked with no same values in order to ensure that there is no autocorrelation side lobe greater than one. This condition is similar to have no frequency overlaps in the channel allocation problem, according to lemma. Therefore, to obtain solutions to the problem, it is equivalent to find optical CDMA codeword with a predetermined pulse separation and aperiodic autocorrelation side lobes no greater than one.

4.2 Constraints

To further formulate the allocation problem, we consider the physical system parameters. The slot width Δf should be large enough to accommodate the optical signal in a channel with minimum distortion, even with some instability in channel frequencies. On the other hand, to reduce unwanted spectral sidebands entering into a desired channel, the channel frequency-separation Δf_c should also be large enough. For example, to have reasonable system performance, Forghieri [27] suggested the required minimum values of slot width (i.e., $\Delta f \geq 2R$) and channel-to-channel separation (i.e., $\Delta f_c \geq 10R$) as an integer multiple of bit rate in order to avoid significant crosstalk created by FWM spectra and adjacent WDM channels, respectively. These two requirements impose a constraint that relates the minimum channel separation to the slot width (i.e., $\Delta f_c = n\Delta f$) in terms of an integer multiple n.

Constraint 1: Since $m_i = n_{i+1} - n_i$ denotes the integer channel spacing between the *i*th and *(i+1)*th channels, the inequality

$$m_i \geq n \tag{4.3}$$

must be satisfied for all i= {1, 2, 3, ….. , N-1}. Furthermore, to minimize the total optical bandwidth occupied by the WDM channels, an additional constraint on the total number of slots is needed while solving the channel-allocation problem.

Constraint 2: The total number of slots

$$S = \sum m_i = n_N \tag{4.4}$$

must be as small as possible.

4.3 Lower-Bound on Total Optical Bandwidth

A lower bound to the total optical bandwidth required B_{un} can be found just from the condition that the m_i's must be different from each other (and larger than n*)*. It follows that [28]

$$B_{un} \geq [1 + ((N/2)-1)/n)]\, B_{eq} \tag{4.5}$$

where $B_{eq} = (N - 1)\Delta f_c$ is the total optical bandwidth of a conventional WDM system with the channels equally spaced. Figure 4.2 shows the bandwidth expansion factor, defined as

Bun/Beq, versus the number N of channels in the WDM system for various values of the minimum separation parameter n. It can be observed that for n ≥ 5 and up to 10 channels the lower bound is achievable. In general, for any value of N and n there are several optimum solutions.

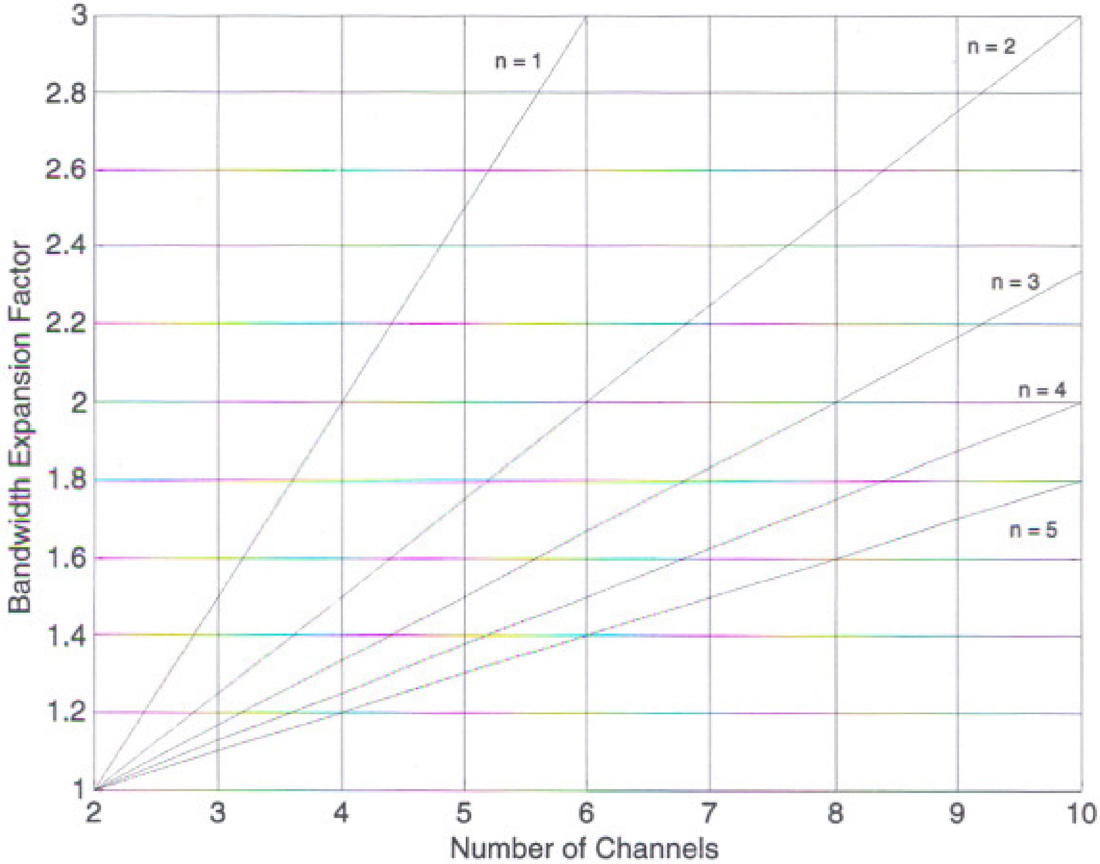


Figure 4.2: Bandwidth Expansion Factor (B_{un}/B_{eq}) Vs Number of Channels for Various Values of $N = \Delta F_c / \Delta F$

4.4 Golomb Ruler Algorithms

There are many algorithms to construct the Golomb ruler sequences. Here two methods are described which have been totally implemented in Matlab. The Matlab codes for both constructions are given in appendix B and their results and comparison are shown in the next section.

4.4.1 EQC Algorithm

In the following, optimal sequences/solutions with n = (3P – 1)/2, N = P + 1, and S = P (2P – 1) are constructed algebraically for a given prime number P [16]. The construction starts with the number theoretic concept of quadratic congruence by generating extended quadratic congruence (EQC) sequences with weight P and periodic autocorrelation constraint one. We then construct "modified" sequences with weight P + 1 and aperiodic autocorrelation constraint one. Among these modified sequences, we show that the lower bound can be achieved by at least one sequence and thus the algorithm is optimum.

Let GF (P) = {0, 1, 2, ..., P-1}denote the Galois field of a prime number P. An EQC sequence $R_i = [r_{i,0}, r_{i,1}, \dots r_{i,j}, \dots r_{i,P-1}]$ is a P-tuple over GF (P) with the element

$$r_{i,j} = i \otimes [j(j+1)/2] \qquad (4.6)$$

where "⊗ " denotes modulo- multiplication and i = {1, 2, 3, ... P – 1}. The corresponding EQC binary codeword $c_i = [c_{i,0}, c_{i,1}, \dots, c_{i,j}, \dots, c_{i,P(2P-1)-1}]$ is obtained, according to the rule

$$c_{i,j} = \begin{cases} 1, & \text{if } j = r_{i,k} + k\,(2P-1), \\ 0, & \text{otherwise.} \end{cases} \qquad (4.7)$$

Based on the mapping, c_i consists of P sub-blocks of 2P – 1 bits each, whereas $r_{i,j}$ indicates the location of a bit "1" (or pulse) at the j^{th} sub-block. This gives code words of length P (2P – 1) with periodic autocorrelation side lobes no greater than one. Each of the original P – 1 EQC sequences R_i's is now taken as a seed from which a group of new sequences can be generated. For group i, a modified EQC $R_{i,k,l}$ sequence is constructed by adding all elements $r_{i,j}$'s of R_i with an integer k and left-rotating them l times, i.e.,

$$R_{i,k,l} = [r_{i,k,l,0}, r_{i,k,l,1}, \dots r_{i,k,l,j-1}, r_{i,j,k,l,j}, \dots, r_{i,k,l,P-1}] \qquad (4.8)$$
$$= [r_{i,l} + k, r_{i,l+1} + k, \dots r_{i,P-1} + k, r_{i,0} + k, \dots, r_{i,l-1} + k]$$

where k and l ϵ GF (P) and the additions are modulo-P. Each modified EQC sequences $R_{i,k,l}$ is mapped into a binary codeword $\{c_{i,k,l} = [c_{i,k,l,0}, c_{i,k,l,1}, \ldots, c_{i,k,l,P(2P-1)-1}]; i = \{0, 1, 2, \ldots, P-1\}; k = \{0, 1, 2, \ldots, P-1\}; l = \{0, 1, 2, \ldots, P-1\}$ according to the rule

$$c_{i,k,l,j} = \begin{cases} 1, & \text{if } j = r_{i,k,l,m} + m(2P-1), \\ & \text{for some } m \in \{0, 1, \ldots, P-1\} \\ 0, & \text{otherwise} \end{cases} \tag{4.9}$$

resulting in P^2 (P-1) modified EQC binary code words of length P (2P – 1) each.

Using these modified EQC binary sequences, new code words $d_{i,k,l}$'s of length P(2P – 1) + 1 are constructed such that the aperiodic correlation is no greater than one. The (P (2P- 1) + 1) – tuples $\{d_{i,k,l} = [d_{i,k,l,0}, d_{i,k,l,1}, \ldots, d_{i,k,l,j}, \ldots, d_{i,k,l,P(2P-1)}]$; $i = \{1, 2, 3, \ldots, P-1\}$; $k = \{0, 1, 2, \ldots, P-1\}$; $l = \{0, 1, 2, \ldots, P-1\}\}$ are given by equation (4.10), where $q = r_{i,k,l,0}$ represents the number of zeros before the first bit '1' in the sequence $c_{i,k,l}$..

$$\begin{aligned} d_{i,k,l,j} &= c_{i,k,l,j+q}, \quad \text{for } j = \{0, 1, 2, \ldots, P(2P-1)-1-q\} \\ d_{i,k,l,j} &= 0, \quad \text{for } j = \{P(2P-1)-1-q+1, P(2P-1)-1-q, \ldots, P(2P-1)-1\} \\ d_{i,P(2P-1)} &= 1, \quad \text{otherwise} \end{aligned} \tag{4.10}$$

This procedure reduces the number of binary sequences from P^2 (P – 1) to P (P^2 – 1) / 2 since $d_{i,k1,l} = d_{i,k2,l}$ for some distinct k_1 and k_2.

4.4.2 Search Algorithm

As observed in the last section, the EQC technique restricts the smallest delay (i.e., n) in the code words to (3P – 1) / 2 for a given number P and results in a limited number of solutions to the channel-allocation problem.

In this section, a construction without the restriction on the minimum pulse separation constraint n is introduced. Based on the last section, the following search algorithm generates optimal sequences/solutions, with N = P + 1 and S = (n + (P – 1) / 2) P for some given prime number P and minimum pulse separation n [18].

1. For a given prime number P and pulse separation n, the first delay vector (or channel spacing vector) m_1= [m0, m_1, …. , m_{P-1}] = [n, n+1, … , n + P – 1] is constructed.
2. The j^{th} delay vector m_j = [l_0, l_1, … , l_k , …. , l_{P-1}] for j = {1, 2, 3, …., P – 1} are constructed with $l_k = m_{j \otimes k}$, where $\otimes$ denotes modulo-P multiplication.
3. The j^{th} codeword s_j = [s_0, s_1, s_2, …… , s_q, …… , s_P] with weight P + 1 are constructed from m_j, according to the rule $s_q = l_{q-1} + s_{q-1}$ where q = {1,2, 3, …. , P} and $s_0 = 0$.
4. Finally, find the code words s_j 's with aperiodic correlation constraint one.

4.5 Results and Discussion

4.5.1 Results from EQC Sequences

Taking as an example, let us suppose given P = 5, four EQC sequences R_1 = 01310, R_2 = 02120, R_3 = 03430 and R_4 = 04240 are constructed, according to equation (4.6). Based on equation (4.8), these four R_i are the seeds to generate P^2 (P – 1) = 100 modified EQC sequences, where $R_{1,0,0}$ = 01310, $R_{1,1,0}$ = 12421, $R_{2,0,0}$ = 02120, $R_{2,3,0}$ = 30403, $R_{3,0,0}$ = 03430, $R_{3,0,1}$ = 34300, $R_{4,0,0}$ = 04240 and $R_{4,2,3}$ = 12214 are some examples. Using equations (4.9) and (4.10), all of the 100 modified EQC sequences are converted to codeword $d_{i,k,l}$'s. Since some code words are repeated (e.g., the code words from $R_{1,0,0}$ = 01310 and $R_{1,1,0}$ = 12421), there are only $P(P^2 – 1) / 2$ = 60 code words. For example, the corresponding code words for k = l = 0 are $d_{1,0,0}$ = 100 000 000 010 000 000 000 100 000 010 000 000 100 000 000 1, $d_{2,0,0}$ = 100 000 000 001 000 000 010 000 000 001 000 000 100 000 000 1, $d_{3,0,0}$ = 100 000 000 000 100 000 000 010 000 000 100 000 100 000 000 1 and $d_{4,0,0}$ = 100 000 000 000 010 000 001 000 0000 000 010 000 100 000 000 1. Among them $d_{1,0,0}$ and $d_{2,0,0}$ are two codewords with $m_i \geq 7$, resulting in two possible solutions (with S = 45, N = 6, and n = 7) to the channel allocation problem. Both give the slot vector n = [0, 10, 21, 28, 36, 45] and [0, 11, 19, 29, 36, 45] respectively. Table 4.1 shows some results obtained from the algorithm where P is a prime number, N is the number of unequal-spaced WDM channels, S is the total number of slots occupied by these channels, n is the minimum channel separation, n is the slot vector, N_{un} is total number

Table 4.1: Some Simulation Results Obtained From the Modified EQC Constructions

N	S	n	Example of slot vector	N_{un}	N_{eq}	B_{un}/B_{eq}
4	15	4	[0,4,9,15]	28	12	2.33
6	45	7	[0,10,21,28,36,45]	140	35	4
8	91	10	[0,14,29,45,55,66,78,91]	378	70	5.4
12	231	16	[0,26,46,71,90,114,132,155,172,194,210,231]	1441	176	8.2
14	325	19	[0,27,56,87,107,129,153,179,207,237,256,277, 300,325]	2340	247	9.5
18	561	25	[[0,37,78,106,138,174,214,241,272,307,346, 372,420,436,474,499,528,561]	5203	425	12.2
20	703	28	[0,41,86,116,150,188,230,276,307,342,381, 424,452,484,520,560,604,633,666,703]	7163	532	13.5
24	1035	34	[0,55,97,149,188,237,273,319,375,418,471, 511,561,598,645,679,723,777,818,869,907, 955,990,1035]	12650	782	16.2

N	Number of channels
S	Total number of slots occupied by these channels
n	Minimum channel separation
N_{un}	Total number of slots for unequal spacing
N_{eq}	Total number of slots for equal spacing
B_{un}/B_{eq}	Bandwidth expansion factor

of slots for unequal spacing, N_{eq} is the total number of slots for equal spacing, N_{un} is the total number of slots for unequal spacing, and the ratio of these two is the parameter (B_{un} / B_{eq}) called the bandwidth expansion ratio. The table gives an example of sequence and calculates bandwidth expansion factor for different number of channels. This BEF variation with number of channels is shown in figure 4.3. It can be seen that the BEF curve, as expected, increases linearly with the number of channels.

The total time taken was noted down for each simulation varying the value of p (number of channels). The table giving values of bandwidth and time taken for different values of channels for the EQC algorithm is given in table 4.2. Figures 4.4 and 4.5 show respectively the effect of increasing the number of channels on the bandwidth and time taken in seconds graphically. It can be easily seen from figure 4.4 that the minimum numbers of slots required grow exponentially with the increasing number of channels. While from figure 4.5 it is seen that the time taken remains almost constant up to eight channels, starts increasing linearly afterwards and after about 14 channels, it increases exponentially. The various sequences for a particular number of channels as obtained by the simulation of the above EQC algorithm are shown in figure 4.6.

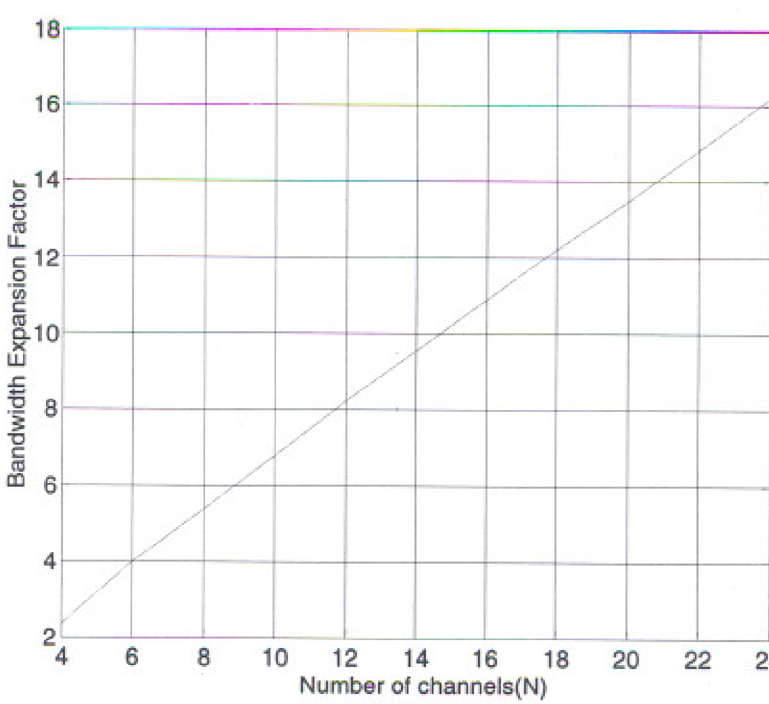


Figure 4.3: Simulation Results of Bandwidth Expansion Factor (B_{un}/B_{eq}) Vs Number of Channels for Results by EQC Algorithm

Table 4.2: Results of Bandwidth Requirements and Time Taken for Modified EQC Sequences for Different Number of Channels

Number of channels	Minimum number of slots	Time taken in seconds
4	31	0.0310
7	140	0.1560
10	378	1.11
16	1419	24.8750
19	2327	85.0780
25	5151	618.1410

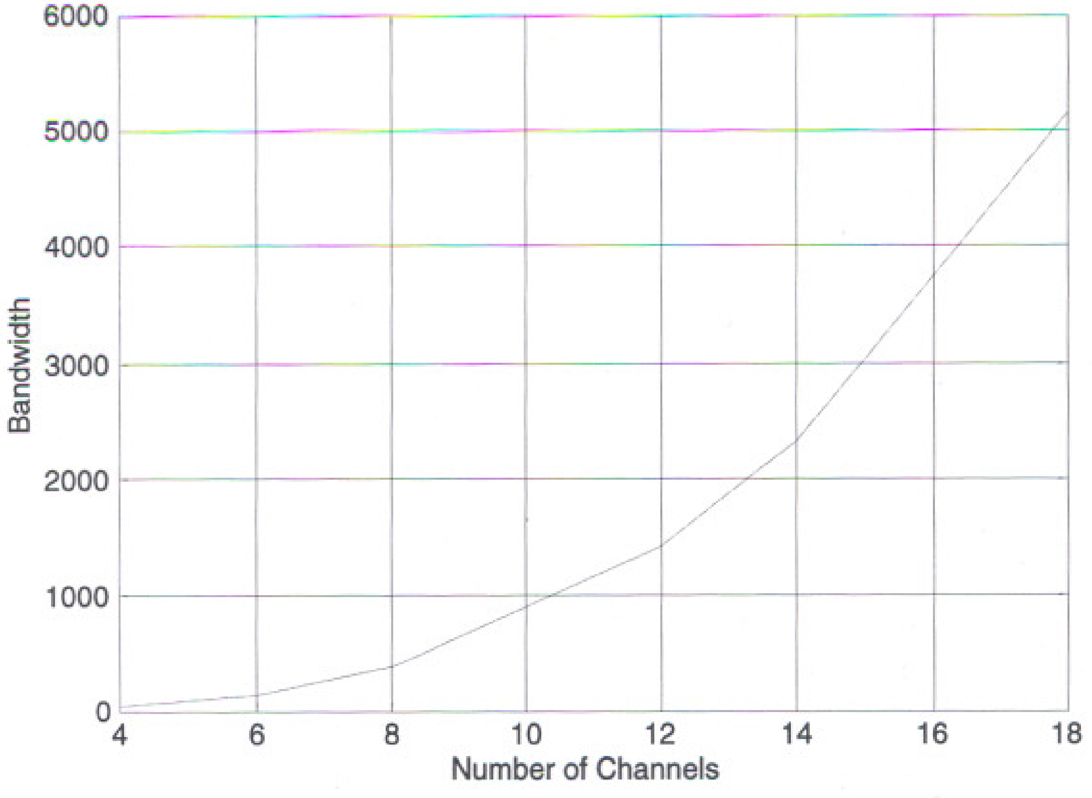


Figure 4.4: Simulation Results of Bandwidth versus Number of Channels by EQC Algorithm

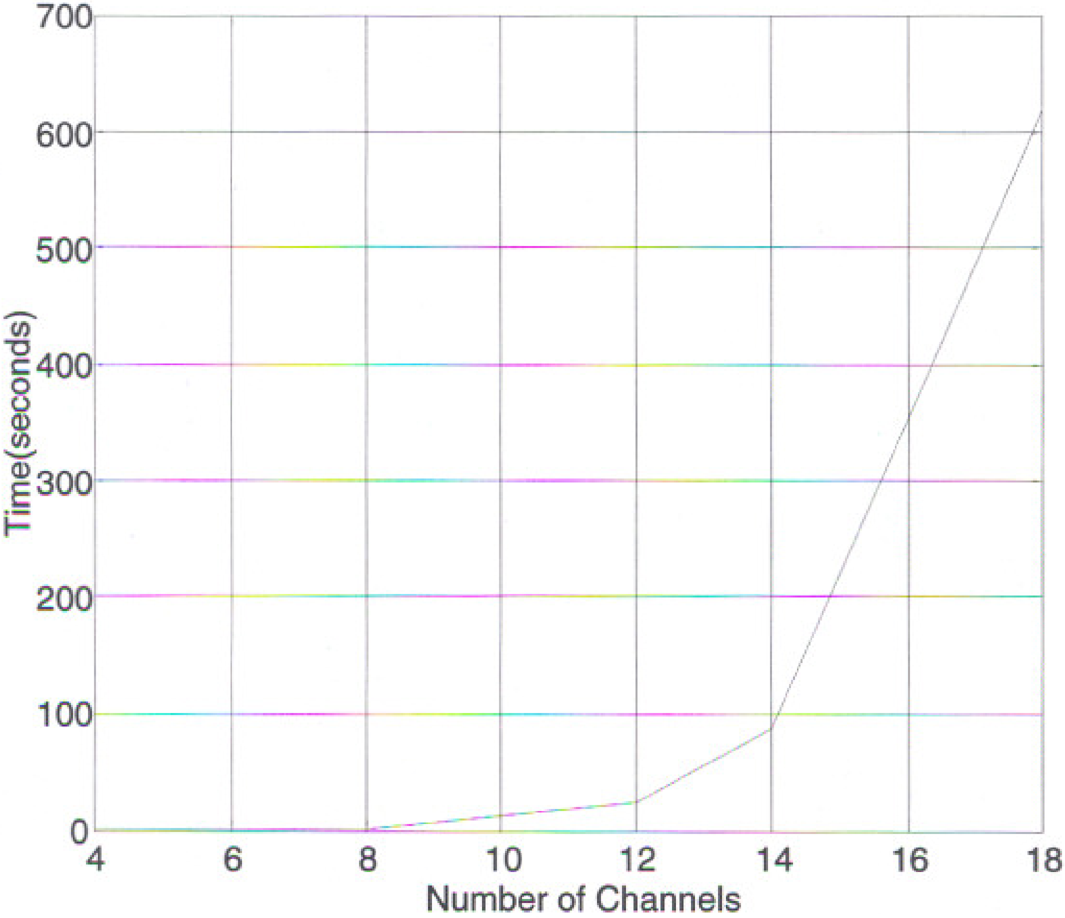


Figure 4.5: Simulation Results of Time Taken in Seconds versus Number of Channels by EQC Algorithm

0	5	11	15
0	6	10	15
0	5	12	15
0	7	10	15
0	6	11	15
0	7	12	15

N = 4

0	9	19	30	37	45
0	10	21	28	36	45
0	9	20	28	38	45
0	11	19	29	36	45
0	9	21	31	39	45
0	12	22	30	36	45
0	9	22	29	40	45
0	13	20	31	36	45
0	11	18	31	40	45
0	13	22	27	38	45
0	10	22	31	37	45
0	11	21	30	38	45

N = 6

0	13	27	42	58	68	79	91
0	14	29	45	55	66	78	91
0	13	28	45	57	71	80	91
0	15	32	44	58	67	78	91
0	13	29	41	56	67	81	91
0	16	28	43	54	68	78	91
0	13	30	44	55	70	82	91
0	17	31	42	57	69	78	91
0	13	31	40	54	66	83	91
0	18	27	41	53	70	78	91
0	13	32	43	53	69	84	91
0	19	30	40	56	71	78	91
0	17	28	40	53	67	82	91

N = 8

0	21	43	66	90	115	130	157	174	192	211	231
0	22	45	69	94	109	136	153	171	190	210	231
0	21	44	69	85	114	134	156	169	195	212	231
0	23	48	64	93	113	135	148	174	191	210	231
0	21	45	72	91	113	127	155	175	198	213	231
0	24	51	70	92	106	134	154	177	192	210	231
0	21	46	64	86	112	131	154	170	190	214	231
0	25	43	65	91	110	133	149	169	193	210	231
0	21	47	67	92	111	135	153	176	193	215	231
0	26	46	71	90	114	132	155	172	194	210	231
0	21	48	70	87	110	128	152	171	196	216	231
0	27	49	66	89	107	131	150	175	195	210	231
0	21	49	73	93	109	132	151	177	199	217	231
0	28	52	72	88	111	130	156	178	196	210	231
0	21	50	65	88	108	136	150	172	191	218	231
0	29	44	67	87	115	129	151	170	197	210	231
0	21	51	68	94	107	129	149	178	194	219	231
0	30	47	73	86	108	128	157	173	198	210	231
0	21	52	71	89	106	133	148	173	197	220	231

N = 12

0	25	51	78	106	135	152	183	202	235	256	278	301	325
0	26	53	81	110	127	158	177	210	231	253	276	300	325
0	25	52	81	112	132	154	178	204	232	262	281	302	325
0	27	56	87	107	129	153	179	207	237	256	277	300	325
0	25	53	84	105	129	156	186	206	229	255	284	303	325
0	28	59	80	104	131	161	181	204	230	259	278	300	325
0	25	54	87	111	126	158	181	208	226	261	287	304	325
0	29	62	86	101	133	156	183	201	236	262	279	300	325
0	25	55	77	104	136	160	176	210	236	254	277	305	325
0	30	52	79	111	135	151	185	211	229	252	280	300	325
0	25	56	80	110	133	162	184	212	233	260	280	306	325
0	31	55	85	108	137	159	187	208	235	255	281	300	325
0	25	57	83	103	130	151	179	201	230	253	283	307	325
0	32	58	78	105	126	154	176	205	228	258	282	300	325
0	25	58	86	109	127	153	187	203	227	259	286	308	325
0	33	61	84	102	128	162	178	202	234	261	283	300	325
0	25	59	76	102	137	155	182	205	237	252	276	309	325

N = 14

Figure 4.6: Result Sequences by EQC Algorithm

4.5.2 Results from the Search Algorithm

Now taking an example of the constructions based on searching, le us suppose P = 4 and n = 3, we have m_1 = [3, 4, 5, 6, 7]. From m_1, m_2 = [3, 5, 7, 4, 6], m_3 = [3, 6, 4, 7, 5] and m_4 = [3, 7, 6, 5, 4] are found to have aperiodic correlation no greater than one. Therefore the three code words are s_2 = 10 010 00 010 00 000 10001 00000 1, s_3 = 10 010 00 001 00 010 00 000 10 000 1, and s_4 = 10 010 00 000 10 000 01 000 01 000 1, respectively, resulting three possible solutions (with S =25, N = 6, and n = 3) to the channel allocation problem.

Table 4.3 shows some results obtained form the search algorithm, where P is the prime number, N is the number of unequal-spaced WDM channels, S is the total number of slots occupied by these channels, n is the minimum channel separation is the slot vector, N_{un} is the total number of slots for unequal spacing, N_{eq} is the total number of slots for equal spacing and the ratio of these two is the parameter (B_{un} / B_{eq}) called the bandwidth expansion ratio (BER). The table gives an example of sequence and calculates bandwidth expansion factor for different number of channels. This BEF variation with number of channels is shown in figure 4.7. It can be seen that the BEF curve, as expected, increases linearly with the number of channels.

The total time taken was noted down for each simulation varying the value of p (number of channels). The table giving values of bandwidth and time taken for different values of channels for the search algorithm is given in table 4.4. Figures 4.8 and 4.9 show respectively the effect of increasing the number of channels on the bandwidth and time taken in seconds graphically. It can be easily seen from figure 4.8 that the minimum numbers of slots required grow exponentially with the increasing number of channels. While from figure 4.9 it is seen that the time taken remains almost constant up to eight channels, starts increasing linearly afterwards and after about 14 channels, it increases exponentially. The various sequences for a particular number of channels as obtained by the simulation of the above EQC algorithm are shown in figure 4.10.

Table 4.3: Some Simulation Results Obtained From the Search Algorithm

N	S	n	Example of slot vector	N_{un}	N_{eq}	B_{un}/B_{eq}
4	15	4	[0,4,9,15]	28	12	2.33
6	45	7	[0,7,15,24,34,45]	125	35	3.6
8	91	10	[0,10,21,33,46,60,75,91]	336	70	4.8
12	231	16	[0,16,33,51,70,90,111,133,156,180,205,231]	1276	176	7.3
14	325	19	[0,19,39,60,82,105,129,154,180,207,235,264,294,325]	2093	247	8.5
18	561	25	[0,25,51,78,106,135,165,196,228,261,295,330,366,403 , 441,480,520,561]	4641	425	10.9
20	703	28	[0,28,57,87,118,150,183,217,252,288,325,363,402,442 , 483,525,568,612,657,703]	6457	532	12.1
24	1035	34	[0,34,69,105,142,180,219,259,300,342,385,429,474, 520,567,615,664,714,765,817,870,924,979,1035]	11408	782	14.6

N	Number of channels
S	Total number of slots occupied by these channels
n	Minimum channel separation
N_{un}	Total number of slots for unequal spacing
N_{eq}	Total number of slots for equal spacing
B_{un}/B_{eq}	Bandwidth expansion factor

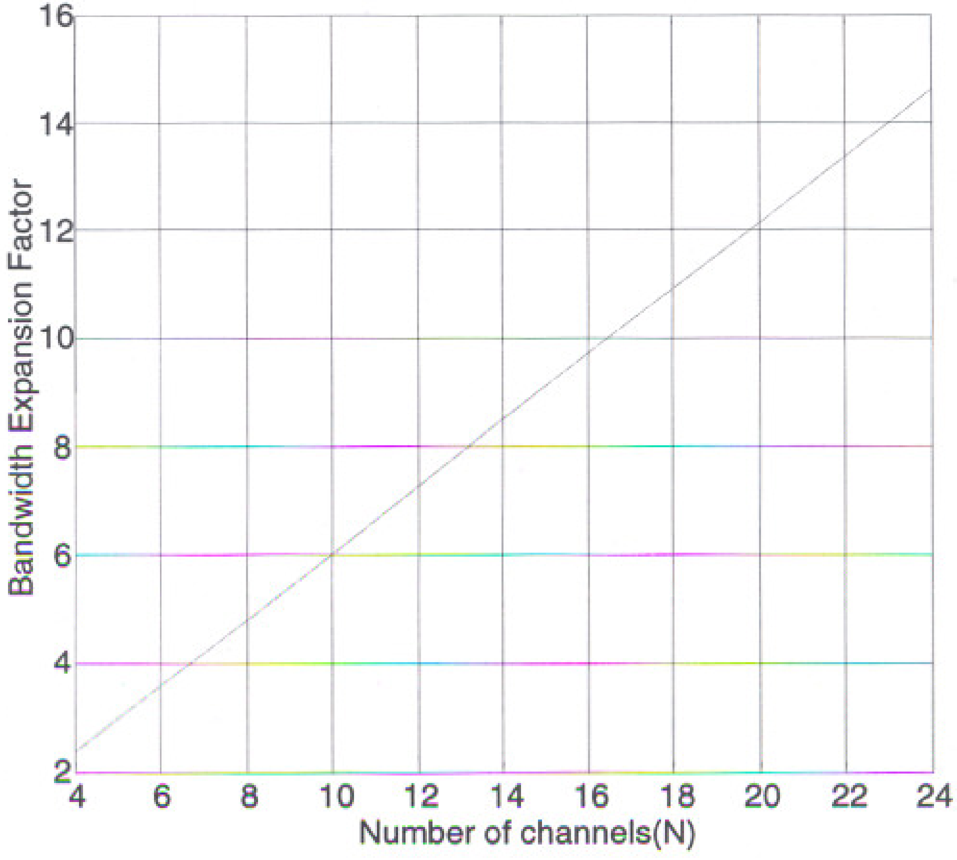


Figure 4.7: Simulation Results of Bandwidth Expansion Factor (B_{un}/B_{eq}) Vs Number of Channels for Results by Search Algorithm

Table 4.4: Results of Bandwidth Requirements and Time Taken by for the Search Algorithm for Different Number of Channels

Number of channels	Minimum number of slots	Time taken in seconds
4	28	0.0160
7	125	0.0160
10	336	0.0160
16	1276	0.0160
19	2093	0.0160
25	4641	0.0160

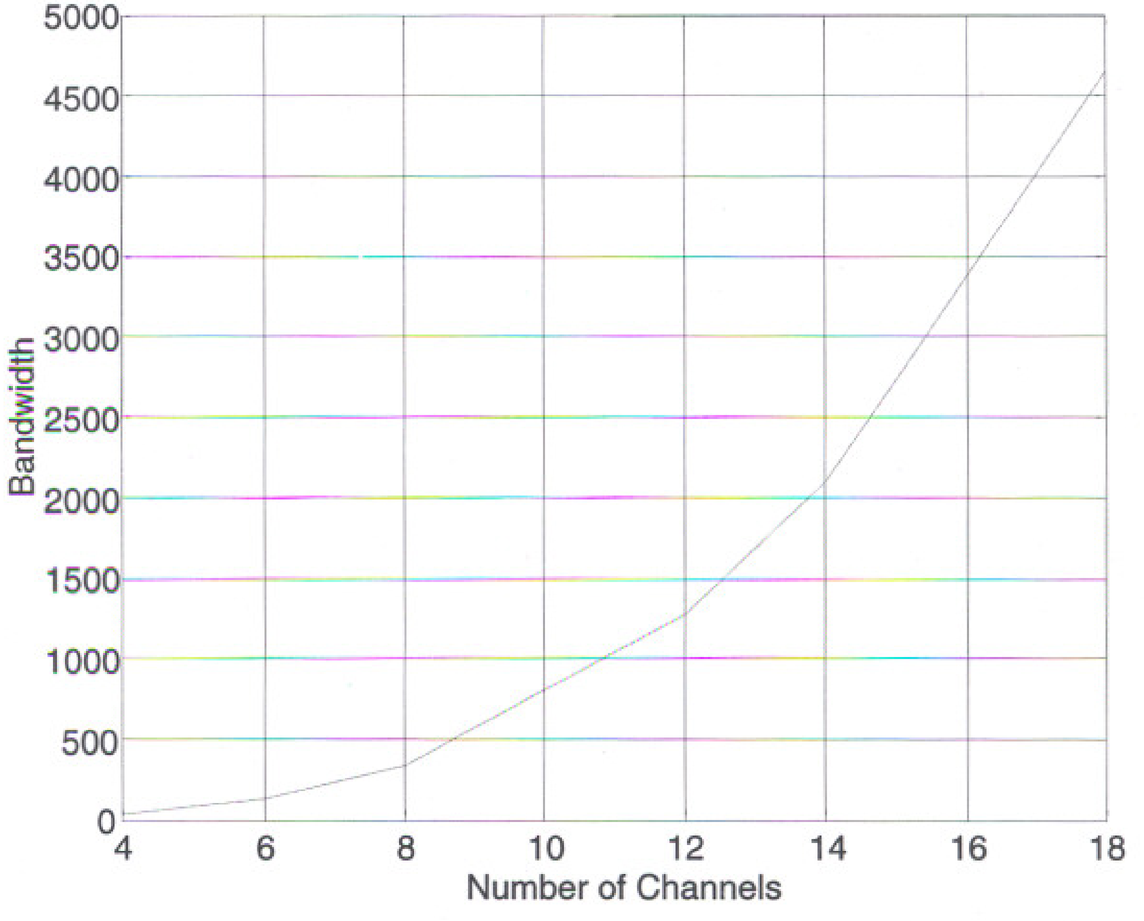


Figure 4.8: Simulation Results of Bandwidth versus Number of Channels for the Search Algorithm

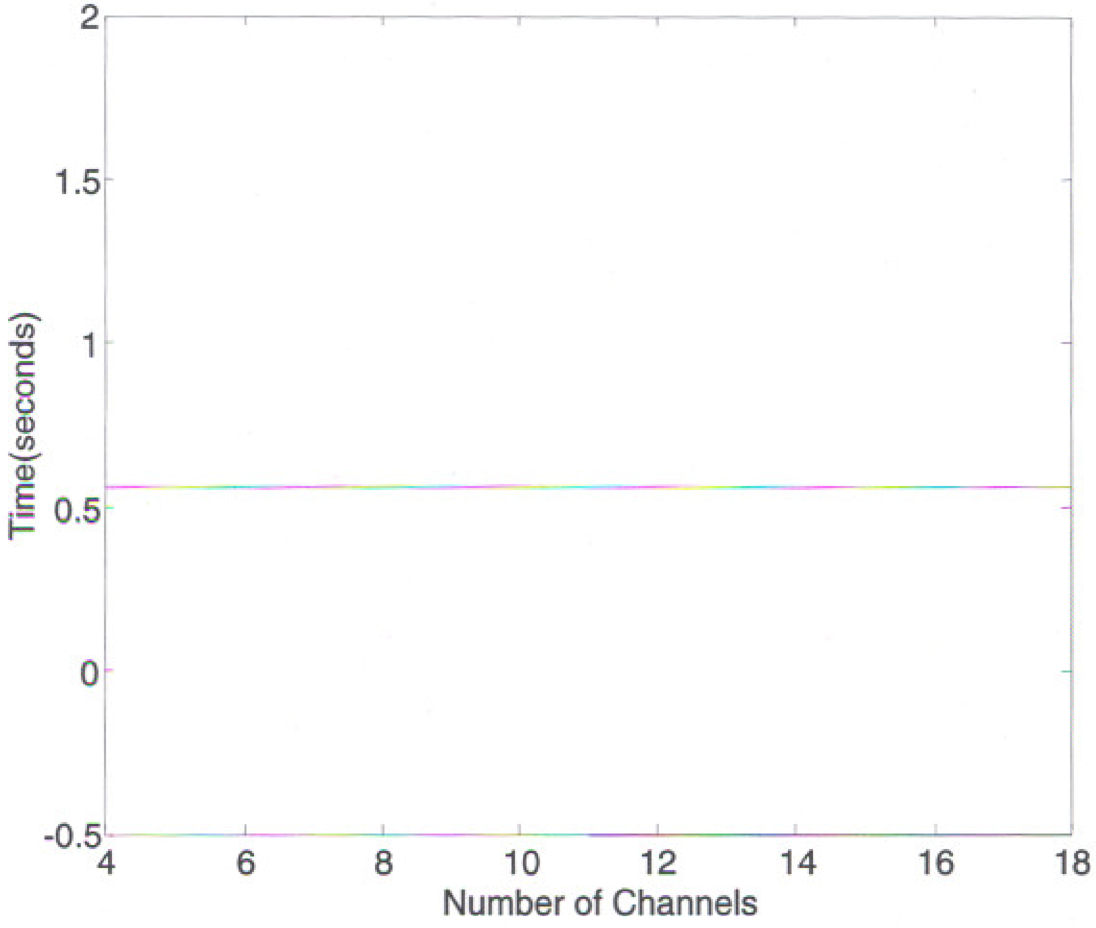


Figure 4.9: Simulation Results of Time Taken in Seconds versus Number of Channels for the Search Algorithm

0	4	9	15
0	4	10	15

N = 4, n = 4

0	5	11	18
0	5	12	18

N = 4, n = 5

0	2	5	9	14	20
0	2	6	12	15	20
0	2	7	10	16	20
0	2	8	13	17	20

N = 6, n = 2

0	3	7	12	18	25
0	3	8	15	19	25
0	3	9	13	20	25
0	3	10	16	21	25

N = 6, n = 3

0	5	11	18	26	35
0	5	12	21	27	35
0	5	13	19	28	35
0	5	14	22	29	35

N = 6, n = 5

0	4	9	15	22	30	39	49
0	4	10	18	28	33	40	49
0	4	11	21	27	36	41	49
0	4	12	17	26	32	42	49
0	4	13	20	25	35	43	49

0	4	14	23	31	38	44	49

N = 8, n = 4

0	5	11	18	26	35	45	56
0	5	12	21	32	38	46	56
0	5	13	24	31	41	47	56
0	5	14	20	30	37	48	56
0	5	15	23	29	40	49	56
0	5	16	26	35	43	50	56

N = 8, n = 5

0	5	11	18	26	35	45	56	68	81	95	110
0	5	12	21	32	45	60	66	74	84	96	110
0	5	13	24	38	44	53	65	80	87	97	110
0	5	14	27	33	43	57	64	75	90	98	110
0	5	15	30	39	53	61	74	81	93	99	110
0	5	16	22	34	41	54	62	76	85	100	110
0	5	17	25	40	51	58	72	82	88	101	110
0	5	18	28	35	50	62	71	77	91	102	110
0	5	19	31	41	49	55	70	83	94	103	110
0	5	20	34	47	59	70	80	89	97	104	110

N = 12, n = 5

0	7	15	24	34	45	57	70	84	99	115	132
0	7	16	27	40	55	72	80	90	102	116	132
0	7	17	30	46	54	65	79	96	105	117	132
0	7	18	33	41	53	69	78	91	108	118	132
0	7	19	36	47	63	73	88	97	111	119	132
0	7	20	28	42	51	66	76	92	103	120	132
0	7	21	31	48	61	70	86	98	106	121	132
0	7	22	34	43	60	74	85	93	109	122	132
0	7	23	37	49	59	67	84	99	112	123	132
0	7	24	40	55	69	82	94	105	115	124	132

N = 12, n = 7

Figure 4.10: Result Sequences by Search Algorithm

4.5.3 Comparison of the Two Algorithms

Figures 4.11 and 4.12 show the comparison of the two methods to construct the OGR sequences with respect to bandwidth and time respectively. In both the criteria, the search algorithm is better as it can be seen that it requires lesser number of total slots and the much lesser time for computation. In addition, the time required does not increase linearly with the number of channels, rather it remains constant.

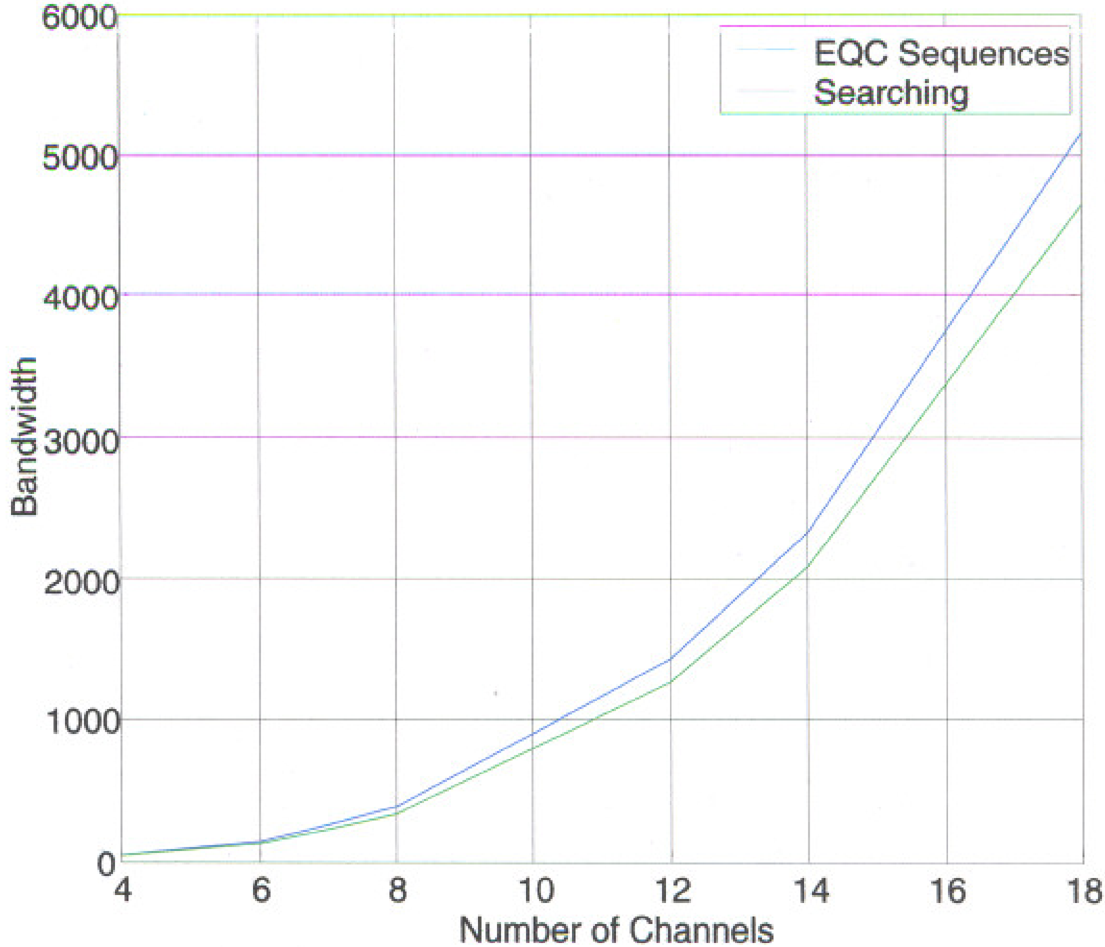


Figure 4.11: Simulation Results of Comparison of the Two Methods with respect to Bandwidth versus Number of Channels

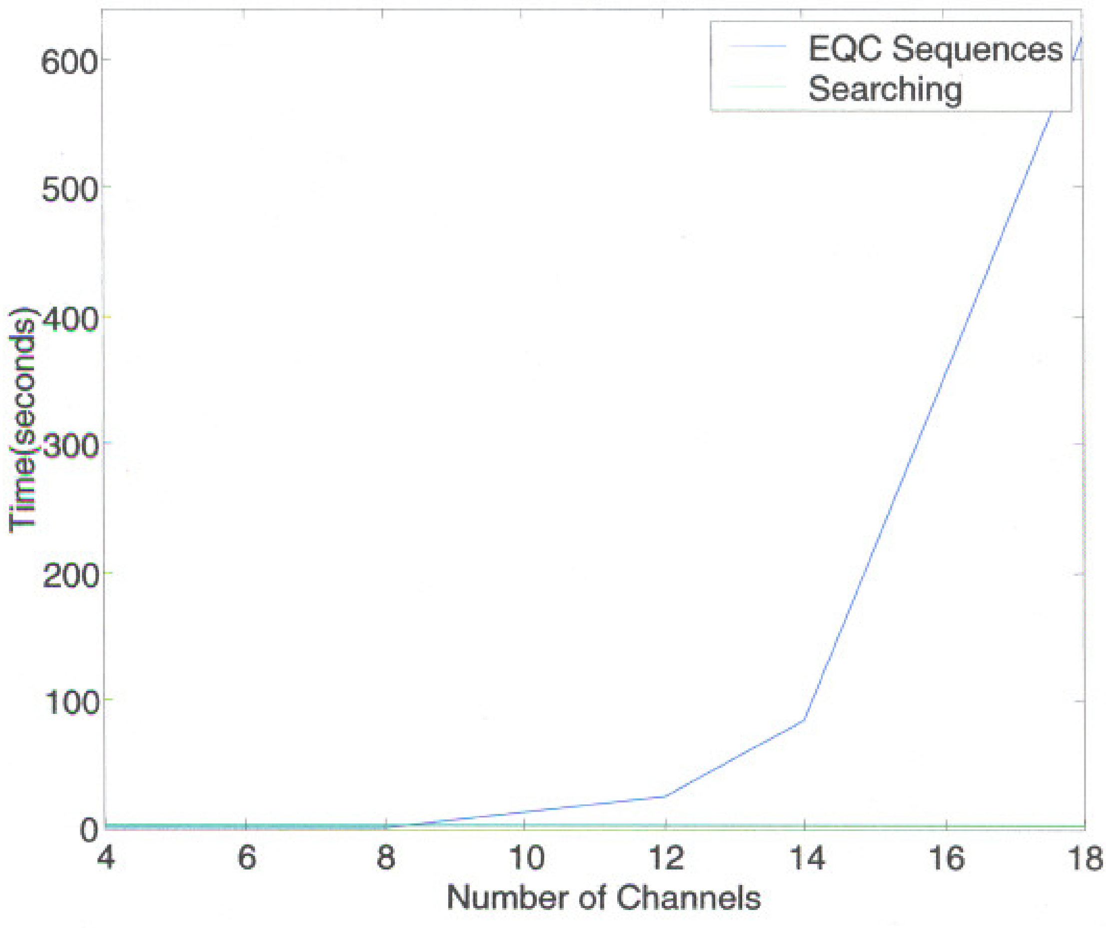


Figure 4.12: Simulation Results of Comparison of the Two Methods with respect to Time Taken Versus Number of Channels.

4.6 Conclusion

To reduce four-wave-mixing crosstalk in high-capacity, long-haul, repeater less, WDM light wave systems, an algebraic framework for finding the solutions to the unequal-spaced channel-allocation problem has been described. Two algebraic algorithms on finding the frequency locations of unequal spaced channels have been introduced based on finding optical CDMA code words with a predetermined pulse separation and aperiodic autocorrelation side lobes no greater than one. Numerical examples have been given to illustrate the constructions. The constructions provide a fast and simple alternative to solve the problem, besides the recently proposed computer-search method. Using the eight-channel case as an example, the optimum slot vector used in the experiment in was [0, 8, 19, 29, 36, 42, 51, 56], where n = 5 and S = 56. Using the assumptions in that $\Delta f = 2R$ and $\Delta f_c = n \cdot \Delta f = 10$ R, the total optical bandwidth occupied by these eight channels was $B_{opt} = S \cdot \Delta f = 112$ R. With the parameters P = 7, N= 8, S = 56 and n=5, the Construction given in search algorithm can be used to obtain similar optimum solutions, such as [0, 5, 12, 21, 32, 38, 46, 56] for j = 2 and [0, 5, 13, 24, 31, 41, 47, 56] for j = 3 in Table 4.2, whereas the latter can be modified by left-rotating its delay vector once to get [0, 8, 19, 26, 36, 42, 51, 56]. Moreover, an optimum slot vector for these eight channels can possibly be obtained by using P = 7 in the construction given in EQC algorithm where the slot vector is given as [0, 14, 29, 45, 55, 66, 78, 91] in the row with p = 7, N = 8, S = 91 and n = 10 of table 4.1. Let $\Delta f = R$. the minimum channel spacing Δf_c is still equal to 10 R. however; the total optical bandwidth is now only equal to 91 R, an 18.75% improvement. The only trade-off is the reduction of the slot width Δf by half, which may potentially increase the crosstalk of channel-FWM spectra overlap.

The comparison of both the algorithms is also provided which proves that the latter one, the search algorithm is better and faster one. The bandwidth requirements are comparatively smaller than the EQC algorithm, and further, the time remains constant as the number of marks increase.

CHAPTER – 5

OPTIMIZATION OF GOLOMB RULERS USING GENETIC ALGORITHM

In an ideal world we want the best solution to a problem. However, in the real world of budgets and deadlines we find that the choice is usually to do things fast and cheap. Here is where the goal of optimization comes in. Given the constraints of time and of money we cannot always find the perfect solution to a problem, and so we must find good solutions. The focus here is to find high order good Golomb Rulers.

5.1 Introduction and Literature Survey

The success of genetic algorithm in finding relatively good solutions to NP-complete problems such as the traveling salesman problem [23] and job-shop scheduling problem provided a good starting point for a machine intelligent method of finding Golomb Rulers [29]. These rulers have been applied to radio astronomy, X-ray crystallography, circuit layout and geographical mapping. The nature of NP-complete makes the search for higher order rulers difficult and very time consuming. While the shortest lengths for each other are important as a mathematical exercise, finding relatively short high order valid rulers has a more important impact on real world applications. Genetic algorithms have shown good results in finding good usable Golomb Rulers in minutes or hours instead of weeks or months.

5.1.1 Need of optimization

The goal of optimizing Golomb rulers is to get them as short as possible, while not duplicating any measured distances. As an example, let us consider the ruler of four marks. The various possible Golomb rulers are (0,1,4,6); (0,2,5,9); (0, 4, 10, 23); (0,1, 5,11) and so on. Among the all given above, the ruler having shortest length is the first one, having length 6. The rulers generated by the algorithms given in chapter 4 can be optimized using genetic algorithm. The optimum rulers help in conserving the total bandwidth occupied by the channels when used in WDM channel allocation.

5.1.2 Use of Genetic Algorithms

Golomb rulers represent a class of problems known as NP-complete. Unlike the traveling salesman problem, which may be classified as a 'complete ordered set', the Golomb Ruler may be classified as an 'incomplete ordered set'. The exhaustive search of such problems is impossible for higher order models. As another mark is added to the ruler, the time required to search the permutations and to test the ruler becomes exponentially greater. The success of genetic algorithm in finding relatively good solutions to NP-complete problems such as the traveling salesman problem and job-shop scheduling problem provided a good starting point for a machine intelligent method of finding Golomb Rulers. The theorem that provides a means of removing unnecessary permutations in the heuristic-based approach also lends a hand in the GA approach. When the GA begins, the rulers will contain low order Golomb Rulers within themselves. As the population approaches the optimum point, the rulers will contain higher order Golomb rulers.

5.1.3 Literature Survey

The genetic algorithm approach to the search for Golomb Rulers by Stephen W. Soliday et. al. [29] produces *short* rulers for each of the orders. Typically Golomb Rulers are represented by the position of marks on the ruler. It was described that this method created problems with developing a good crossover operator because the Golomb Ruler represented permutations of an incomplete ordered set. The representation used here is one in which the length of each segment is represented by the element of an integer array. The position of a segment on the ruler corresponds to an index for the array, for example, if ruler (5) = 7, then the fifth segment on the ruler has a length of seven. In order to assure unique segment lengths, the first representation tried contained not one but two lists of ruler segments. The first list was the *ruler* and the second list was the *surplus*. Mutation was performed by either swapping ruler segments with each other, thereby creating a new permutation, or by swapping a ruler segment with a surplus segment. By introducing multiple fitness criteria, a simpler representation replaced the two list method and only the ruler segments were maintained in the list. This effectively reduced the search space but required modifications to the simple GA. Each of the modifications was discussed in the paper. The GA in this paper was written in C++ using object oriented programming.

Genetic algorithm has been used to construct sub-optimal Golomb arrays. In the approach used by J.P. Robinson [30], the cells of the array are arranged in a spiral order. The chromosomes encode the presence or a 1 in the next available position. After each 1 is inserted, all the positions that violate the Golomb ruler condition are removed from the available positions. It is also shown that the chromosome representation is capable of encoding the entire search space. An approach using simultaneous evolution of multiple chromosome population is also proposed. However, the pruning approach is computationally expensive

Sub optimal Golomb Rulers using genetic algorithms have also been generated by Sharat S. Chikkerur [31]. The genetic algorithm toolbox for Matlab™ was used to run the simulations. The problem statement is given as: (1) Given the size of the sequence (N) what is the maximum density of the Golomb ruler that can be achieved. (2) Obtain a sub optimal Golomb ruler of m marks with length as close as possible to the optimal length. There are two approaches described to solve this problem

1. Assume the length of the ruler (N) and obtain rulers with variable number of marks
2. Assume the number of marks and obtain rulers with variable lengths

Each chromosome or a bit string represents an candidate solution to the optimization problem. The bit string represents its *genotype.* In case where the bit strings are decoded into integer or real variables to represent the parameters, they are known as the *'phenotype'* (physical manifestation of the genotype). To avoid complexity each chromosome is assume to be of fixed length.

The encoding of the chromosomes is done to have results in the form of complementary or orthogonal codes. The direct encoding of the chromosome results in non feasible descendants. Therefore although the fitness function of the successful populations converges very quickly, the overall fitness of the population oscillates across the generations. The effectiveness of the cost function is calculated for each bit string and plotted for increasing number of generations.

5.2 Theory

As stated earlier that the Golomb Ruler is an incomplete ordered set. Golomb Rulers are classified by their order (*n*), where *n* indicates the number of marks. For a ruler this means that there are $n - 1$ segments. For a given order n and a maximum segment length *m* the number of permutations, or possible rulers [29] that may be formed is:

$$\text{Number of rulers} = \frac{1}{2}\binom{m}{n-1} = \frac{m!}{2(m-n+1)!} \qquad (5.1)$$

The 2 in the denominator of equation 5.1 reflects the fact that mirror images do not produce unique rulers. The number of tests made on an order n ruler in order to determine redundancy of measurement and thereby check the validity of the ruler, is:

$$\text{Number of spans} = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} \qquad (5.2)$$

Let us imagine for a moment that we possessed a hypothetical computer able to measure and compare the discrete intervals of our Golomb Rulers at a fictitious speed of one measurement and test every nanosecond (10^{-9} seconds). How long would it take to exhaustively search for the shortest n mark ruler with a maximum individual segment length of m?

$$\text{Search time} = \frac{n\,(n-1)m!}{4\,(m-n+1)!}\; 10^{-9}\ \text{sec} \qquad (5.3)$$

Clearly it can be seen that exhaustively searching for the shortest Golomb Rulers is impractical and sometimes impossible. The use of genetic algorithm removes the unnecessary permutations thus *optimizing* the ruler.

5.2.1 Lower-bound on Optimal Rulers

It is very difficult to obtain the closed form expression for the lower bound for the length of a ruler with m marks. To predict a lower bound for the length of a sub optimal ruler we adopt a coarse approximation based on the lengths of the already known optimal rulers. Assuming the lower bound and mark are related by a second order polynomial we can fit a polynomial curve over the existing data as shown in Figure 5.1. The curve represented by the equation $0.847\,x^2 - 3.169\,x + 4.72$ seems to fit very closely with the existing data [32] However no upper bound exists for the sub optimal rulers. It is interesting to observe that $0.847 \approx \left(\frac{1}{2}\right)^{\frac{1}{4}}$, $3.169 \approx \pi$ and $4.72 \approx \pi / \log(2)$. Perhaps this indicates that there is a mathematical structure in optimal Golomb Rulers.

5.3 Construction of Rulers Using Genetic Algorithms

The genetic algorithm toolbox for Matlab GEATBX was used to run the simulations. The toolbox provides various functions to simulate the program efficiently.

5.3.1 Objective Function

The objective function indicates the relative fitness of each chromosome in the population. Here, we want to optimize (minimize) the length of the ruler. Our objective is to minimize the total bandwidth occupied by the channels. Thus, if channel spacing between any pair of channels is denoted as CS and the total number of channels is N, we can say that we have to minimize the ruler length, denoted as RL, where RL is given as:

$$\mathrm{RL} = \sum_{i=1}^{N} (CS)_i \qquad (5.4)$$

subject to $(CS)_i \neq (CS)_j$

If each individual in the population is a Golomb Ruler, the sum of all elements of an individual (genes) forms the objective function, and the aim is to minimize it.

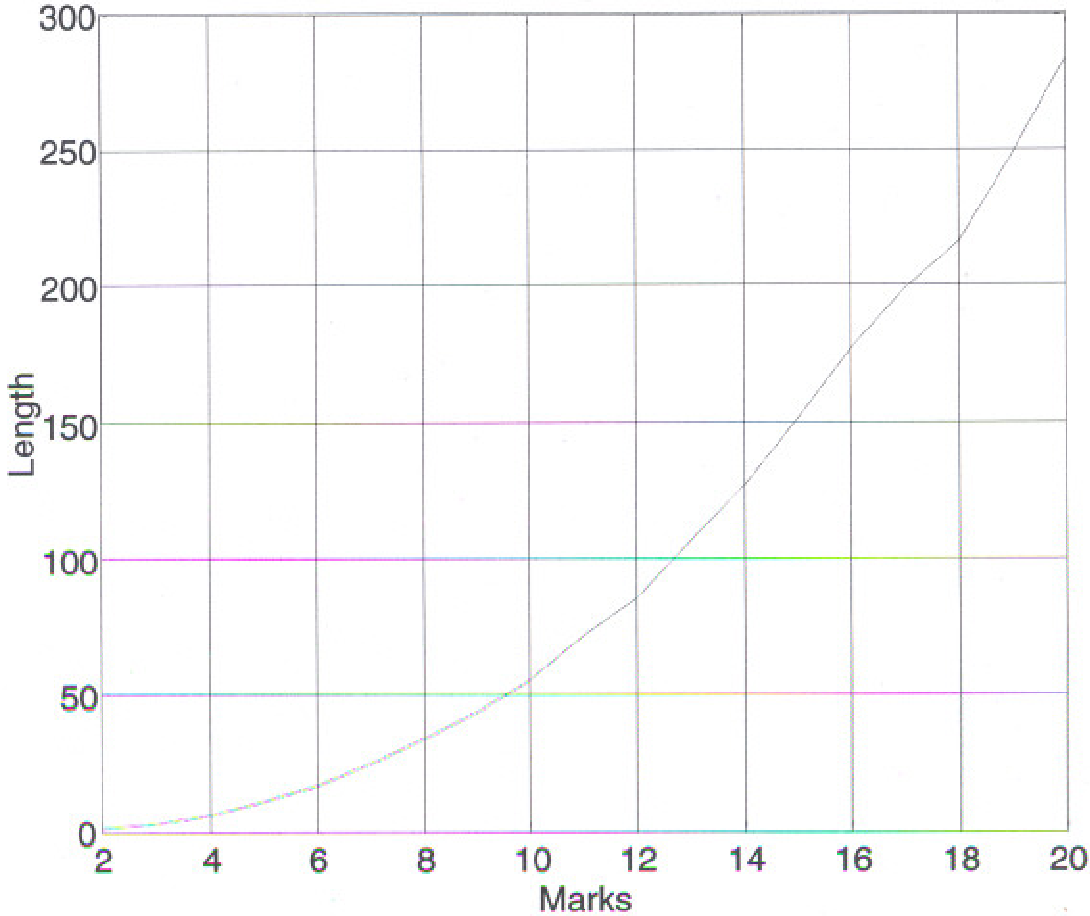


Figure 5.1: Lower Bound versus Number of Marks

5.3.2 Steps for Simulation

The number of channels was prompted to be input by the user and then the following steps were simulated:

1. Generate the initial population where each individual consists of a randomly generated sequence of integers, the number of integers in each individual (genes) being equal to the number of spacings input by the user.
2. Check the golombness of each individual. If it satisfies the conditions for golombness, retain that individual; if it does not, delete that particular individual from the population generated from step 1.

3. If a population of 20 individuals is not generated then go to step 1.
4. Apply the objective function to each of the individual of the population and evaluate the fitness value of the same.
5. Rank the individuals based on the fitness values.
6. Select the best fit individuals, that is, the individuals that have the highest ranks or in other words the greatest fitness values.
7. Apply the crossover and mutation operators to the selected individuals.
8. Check the new offsprings, generated by crossover and mutation, for golombness. Those who are Golomb Rulers are retained.
9. Reinsert the offsprings from step 8 in the original population of step 3 to get a new population.
10. Repeat these steps until maximum number of generations is reached.

5.3.3 Parameters Used

The parameters chosen for the genetic algorithm simulation are as follows:

Table 5.1: Parameters Used For Genetic Algorithm Simulation

Parameter	Value
Number of individuals in each population	20
Crossover method	Single point
Crossover probability	0.8
Mutation probability	0.01
Selection method	Stochastic universal sampling
Number of generations	20

5.4 MATLAB GA Toolbox

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

•Mathematical computation

•Algorithm development

•Modeling, simulation, and prototyping

•Data analysis, exploration, and visualization

•Scientific and engineering graphics

•Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or FORTRAN.

The name MATLAB stands for *matrix laboratory*. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB uses software developed by the LAPACK and ARPACK projects, which together represent the state-of-the-art in software for matrix computation. MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of application-specific solutions called *toolboxes*. Very important to most users of MATLAB, toolboxes allow you to *learn* and *apply* specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems; neural networks, fuzzy logic, wavelets, simulation, and many others. These are numerous functions for 2-D and 3-D graphics as well as for animation. Also for these who cannot do without their FORTRAN or C codes, MATLAB even provides an external interface to run those programs from within MATLAB. MATLAB's language is very easy to learn and to use.

The basic building block of MATLAB is the matrix. The fundamental data-type is the array. Vectors, scalars real matrices and complex matrices are all automatically handled as special cases of the base data type. There is no need to declare the dimensions of a matrix. MATLAB simply loves matrices and matrix operations.

Whilst there exist many good public-domain genetic algorithm packages, none of these provide an environment that is immediately compatible with existing tools in the control domain. The MATLAB Genetic Algorithm Toolbox [32] aims to make GAs accessible to the control engineer within the framework of an existing package. This allows the retention of existing modeling and simulation tools for building objective functions and allows the user to make direct comparisons between genetic methods and traditional procedures.

5.4.1 Data Structures

MATLAB essentially supports only one data type, a rectangular matrix of real or complex numeric elements. The main data structures in the GA Toolbox are chromosomes, phenotypes, objective function values and fitness values. The chromosome structure stores an entire population in a single matrix of size *Nind* × *Lind*, where *Nind* is the number of individuals and *Lind* is the length of the chromosome structure. Phenotypes are stored in a matrix of dimensions *Nind*× *Nvar* where *Nvar* is the number of decision variables. A *Nind* × *Nobj* matrix stores the objective function values, where *Nobj* is the number of objectives. Finally, the fitness values are stored in a vector of length *Nind*. In all of these data structures, each row corresponds to a particular individual.

5.4.2 Toolbox Structure

The GA Toolbox uses MATLAB matrix functions to build a set of versatile routines for implementing a wide range of genetic algorithm methods. Here we outline the major procedures of the GA Toolbox.

Population representation and initialization: crtbase, crtbp, crtrp

The GA Toolbox supports binary, integer and floating-point chromosome representations. Binary and integer populations may be initialised using the Toolbox function to create binary populations, crtbp. An additional function, crtbase, is provided that builds a vector describing the integer representation used. Real-valued populations may be initialised using crtrp. Conversion between binary and real-values is provided by the routine bs2rv that also supports the use of Gray codes and logarithmic scaling.

Fitness assignment: ranking, scaling

The fitness function transforms the raw objective function values into non-negative figures of merit for each individual. The Toolbox supports the offsetting and scaling method of Goldberg [23] and the linear-ranking algorithm. In addition, non-linear ranking is also supported in the routine ranking.

Selection functions: reins, rws, select, sus

These functions select a given number of individuals from the current population, according to their fitness, and return a column vector to their indices. Currently available routines are roulette wheel selection, rws, and stochastic universal sampling, sus. A high-level entry function, select, is also provided as a convenient interface to the selection routines, particularly where multiple populations are used. In cases where a generation gap is required, i.e. where the entire population is not reproduced in each generation, reins can be used to effect uniform random or fitness-based re-insertion.

Crossover operators: recdis, recint, reclin, recmut, recombin, xovdp, xovdprs, xovmp, xovsh, xovshrs, xovsp, xovsprs

The crossover routines recombine pairs of individuals with given probability to produce offspring. Single-point, double-point and shuffle crossover are implemented in the routines xovsp, xovdp and xovsh respectively. Reduced surrogate crossover is supported with single-, xovsprs, and double-point, xovdprs, crossover and with shuffle, xovshrs. A general multi-point crossover routine, xovmp that supports uniform crossover is also provided. To support real-valued chromosome representations, discrete, intermediate and line recombination are supplied in the routines, recdis, recint and reclin respectively. The routine recmut performs line recombination with mutation features. A high-level entry function to all the crossover operators supporting multiple subpopulations is provided by the function recombin.

Mutation operators: mut, mutate, mutbga

Binary and integer mutation are performed by the routine mut. Real-value mutation is available using the breeder GA mutation function, mutbga. Again, a high-level entry function, mutate, to the mutation operators is provided.

Multiple subpopulation support: migrate

The GA Toolbox provides support for multiple subpopulations through the use of high-level genetic operator functions and a function for exchanging individuals amongst subpopulations, migrate. A single population is divided into a number of subpopulations by modifying the data structures used by the Toolbox routines such that subpopulations are stored in contiguous blocks within each data element. The high-level routines, such as select and reins, operate independently on each subpopulation contained in a data structure allowing each subpopulation to evolve in isolation from the others. Based on the *Island* or *Migration* model, migrate allows individuals to be transferred between subpopulations. Uni- and bi-directional ring topologies as well as a fully interconnected network are selectable via option settings as well as fitness-based and uniform selection and re-insertion strategies.

5.4.3 A Simple GA in MATLAB

Figure 5.2 shows the MATLAB code for a Simple GA. The first few lines of the code set the parameters that the GA uses, such as the number and length of the chromosomes, the crossover and mutation rates, the number of generations and, in this case, the binary representation scheme. Next, an initial uniformly distributed random binary population, Chrom, is created using the GA Toolbox function crtbp. The objective function, objfun, is then evaluated to produce the vector of objective values, ObjV. Note that as we do not need the phenotypic representation inside the GA, the binary strings are converted to real values within the objective function call. The initialization complete, the GA now enters the generational loop. First, a fitness vector, FitnV, is determined using the ranking scheme. Visualization and preference articulation can be incorporated into the generational loop by the addition of extra functions. In this example, the routine plotgraphics displays the performance of the current best controller allowing the user to asses the state of the search. Individuals are then selected from the population using the stochastic universal sampling algorithm, sus, with a generation gap, GGAP = 0.9. The 36 (GGAP × NIND) selected individuals are then recombined using single-point crossover, xovsp, applied with probability XOV = 0.7. Binary mutation, mut, is then applied to the offspring with probability MUTR = 0.0175, and the objective function values for the new individuals, ObjVSel, calculated. Finally, the new individuals are re-inserted in the population, using the Toolbox function reins, and the generation counter, gen, incremented. The GA terminates after MAXGEN iterations around the generational loop. The current population, its phenotypic representation and associated cost function values remain in the user's workspace and may be analyzed directly using MATLAB commands.

```
LIND = 15;                                  % Length of individual vars.
NVAR = 2;                                   % No. of decision variables
NIND = 40;                                  % No. of individuals
GGAP = 0.9;                                 % Generation gap
XOV = 0.7;                                  % Crossover rate
MUTR = 0.0175;                              % Mutation rate
MAXGEN = 30;                                % No. of generations

 % Binary representation scheme
FieldD = [LIND LIND; 1 1; 1000 1000; 1 1; 0 0; 0 0; 0 0];

% Initialise population
Chrom = crtbp(Nind, Lind*NVAR);             % Create binary population
ObjV = objfun(bs2rv(Chrom, FieldD));        % Evaluate objective fn.
```

```
Gen = 0;                                    % Counter

% Begin generational loop
while Gen < MAXGEN

% Assign fitness values to entire population
FitnV = ranking(ObjV);

% Visualisation
plotgraphics

% Select individuals for breeding
SelCh = select('sus', Chrom, FitnV, GGAP);

% Recombine individuals (crossover)
SelCh = recombin('xovsp', SelCh, XOV);
% Apply mutation
SelCh = mut(SelCh, MUTR);

% Evaluate offspring, call objective function
ObjVSel = objfun(bs2rv(SelCh, FieldD));
% Reinsert offspring into population
[Chrom ObjV] = reins(Chrom, SelCh, 1, 1, ObjV, ObjVSel);
% Increment counter
Gen = Gen+1;
End

% Convert Chrom to real-values
Phen = bs2rv(Chrom, FieldD);
```

Figure 5. 2: MATLAB Code for a Simple GA

5.5 Results and Discussion

5.5.1 Sequences

The optimum sequences generated by the algorithm are shown in figure 5.3 for different values of N. It can be seen that are sequences are Golomb Rulers. It can also be observed that the rulers reach their optimum values after a certain number of generations, it means that as the number of generations are increased, the results become more and more effective, but after a certain limit, the results are same. This is the point where the results are optimum and no further improvement is seen, that is, we now have got the best results.

5.5.2 Effect of Increasing Generations on Total and Average Bandwidth

As the number of generations increase, the total and average bandwidth of the sequences tends to decrease, that is, we are approaching towards the optimal solution This can be seen in tabular form in tables 5.2 (a), 5.2 (b) and 5.3(c) for different number of channels (N) and graphically in figures 5.3, 5.4 and 5.5.

Table 5.2: Effect of Increase in Generations on Bandwidth and Average Bandwidth for (a) N=3 (b) N = 4 and (c) N = 6

Generations	Bandwidth	Average Bandwidth
1	16	27.9
5	7	17.2
10	5	7.7
15	4	4.1
20	4	4.1
25	4	4.1
30	4	4.1

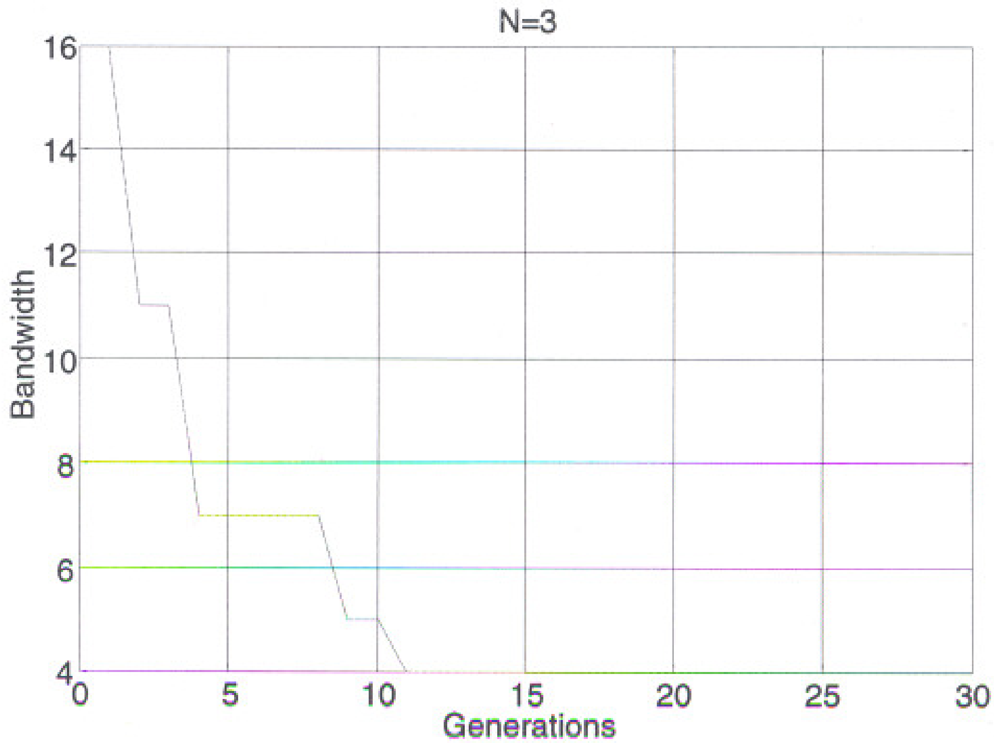
(a)

Generations	Bandwidth	Average Bandwidth
1	24	35.7
5	23	24.8
10	23	23.1
15	21	23.2
20	20	22.1
25	18	20.1
30	18	19.7

(b)

Generations	Bandwidth	Average Bandwidth
1	34	48
5	32	34
10	31	32.3
15	29	31.6
20	29	30.8
25	28	29

(c)



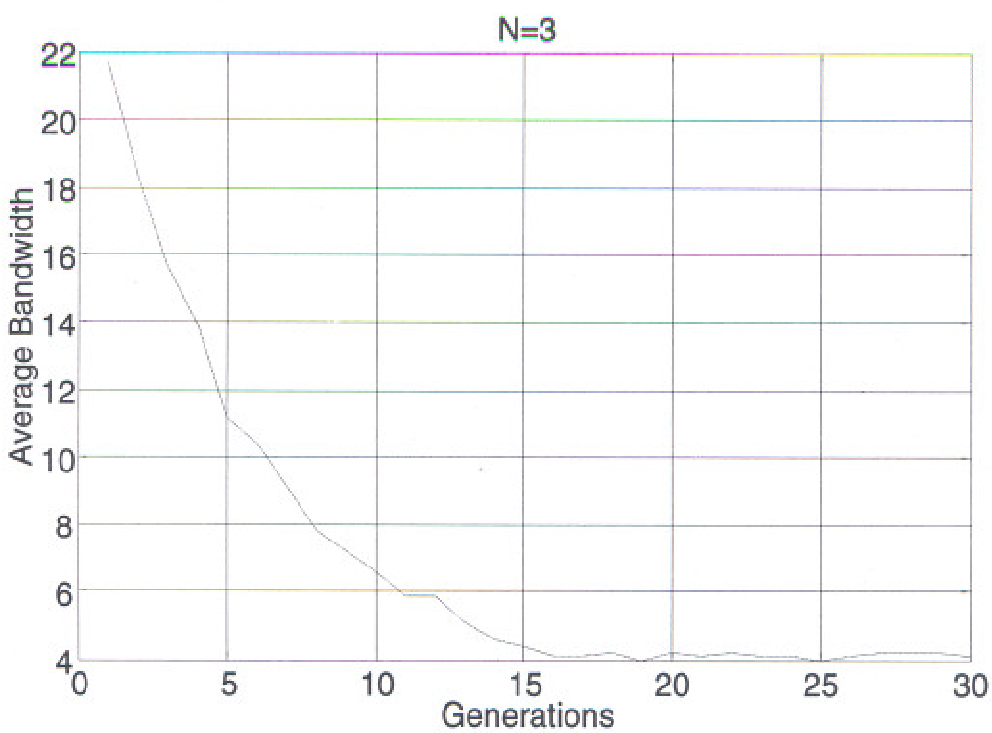


Figure 5.3: Bandwidth and Average Bandwidth vs. No. of Generations for N = 3

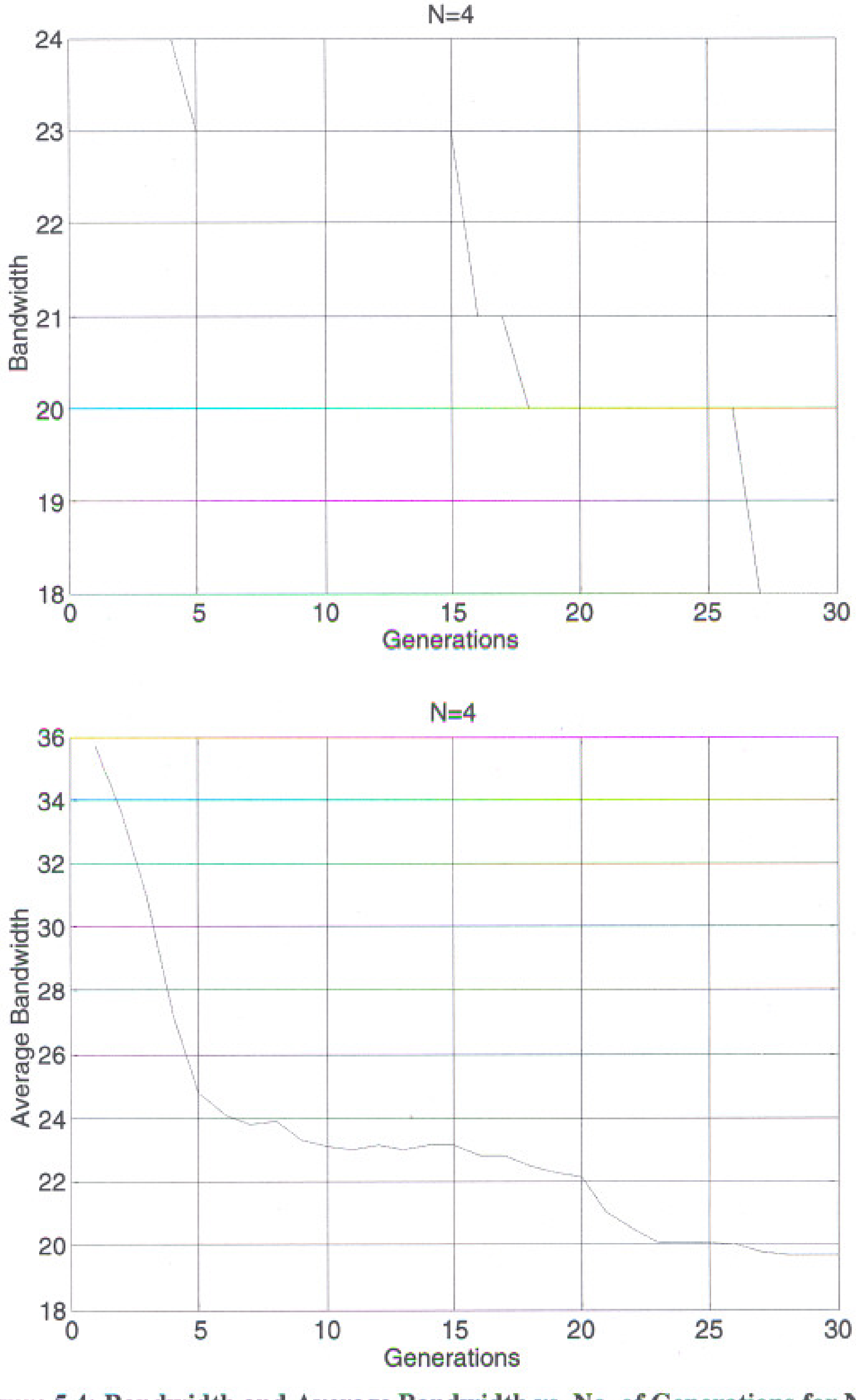


Figure 5.4: Bandwidth and Average Bandwidth vs. No. of Generations for N = 4

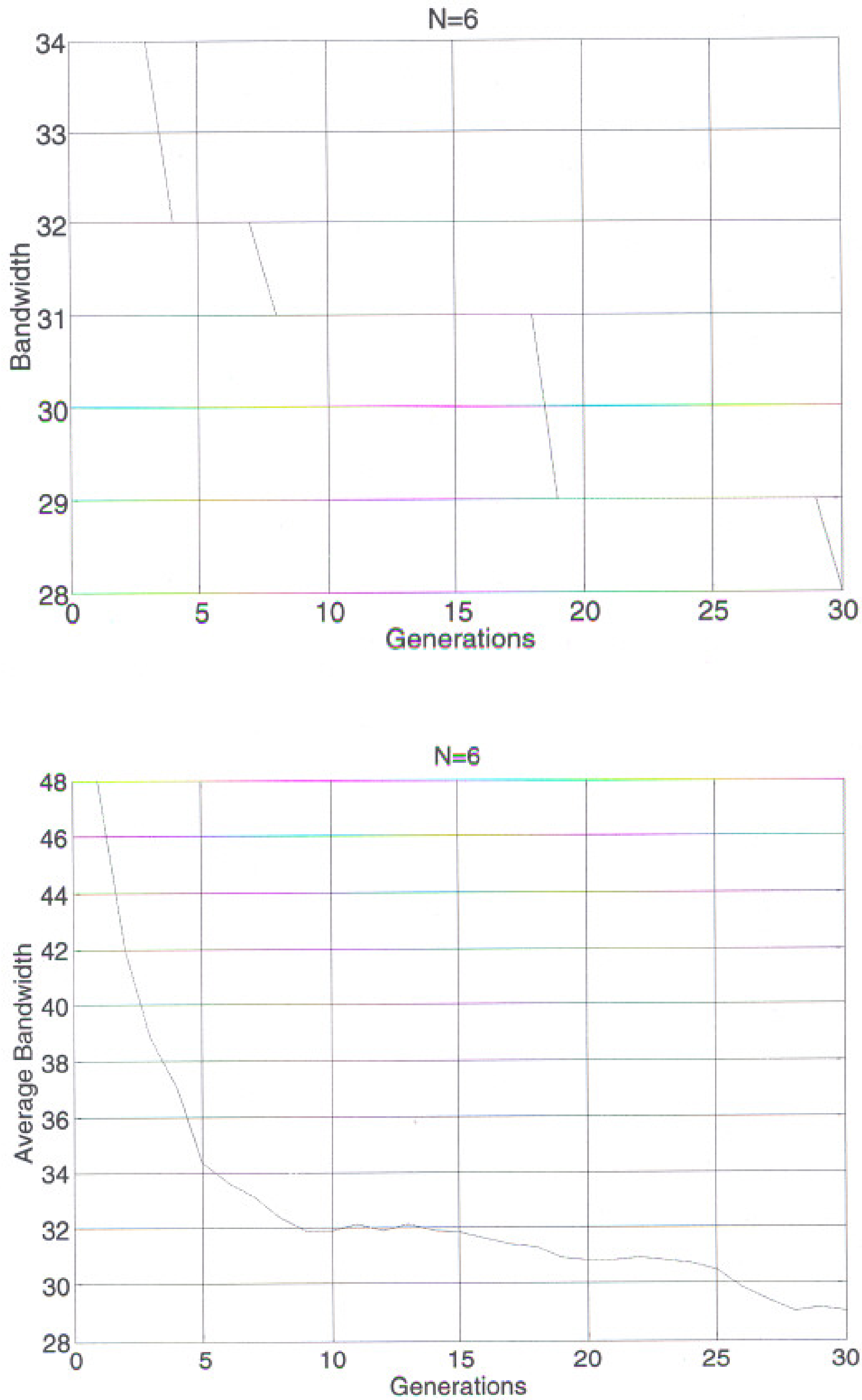


Figure 5.5: Bandwidth and Average Bandwidth vs. No. of Generations for N = 6

5.5.3 Comparison with Previous Results

The aim to use genetic algorithms here was to optimize the length of the ruler so as to conserve the bandwidth. Comparing the simulation results of genetic algorithm with the simulation results of chapter 4, i.e., the conventional methods of generating Golomb Ruler sequences, the EQC construction and the search algorithm; we observe that there is a very significant improvement with respect to the length of the ruler and thus the total bandwidth occupied by the use of genetic algorithm. Table 5.3 and figures 5.6 and 5.7 illustrate the comparison of the three algorithms.

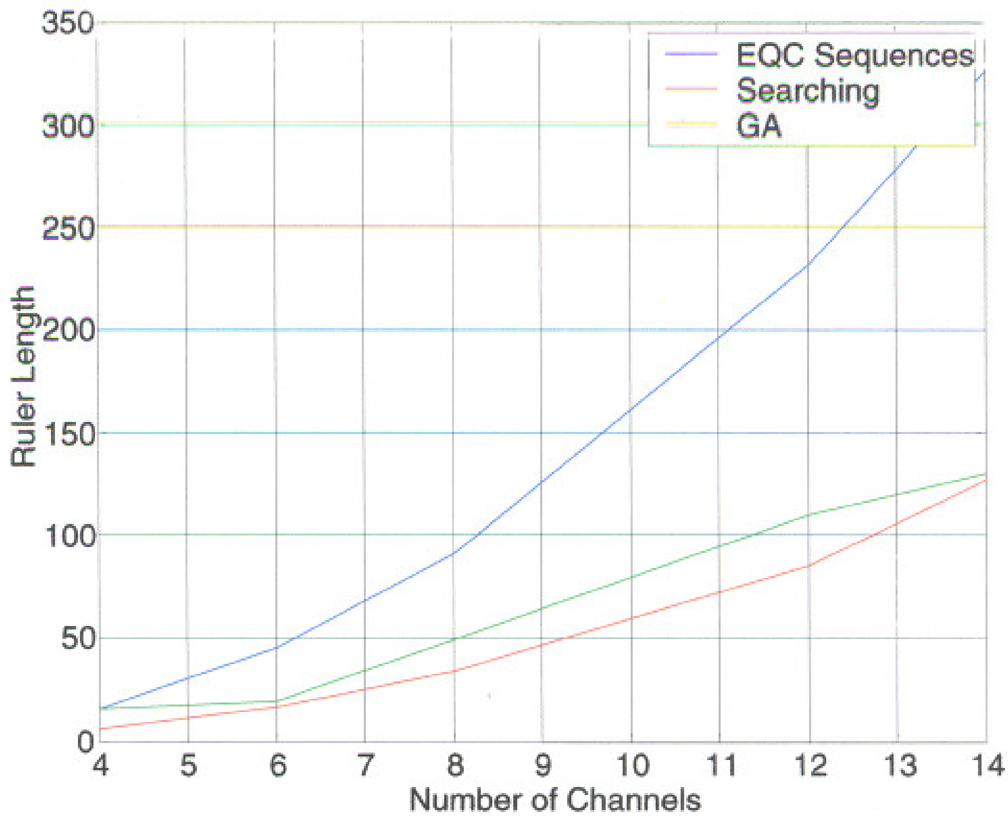


Figure 5.6: Simulation Results of Comparison of the Three Algorithms with respect to the Length of the Ruler

Table 5.3: Comparison of the Results Obtained by the Three Algorithms

N	EQC	SEARCH	GA
4	15	15	6
6	45	20	17
8	91	49	34
12	231	110	85
14	325	130	127

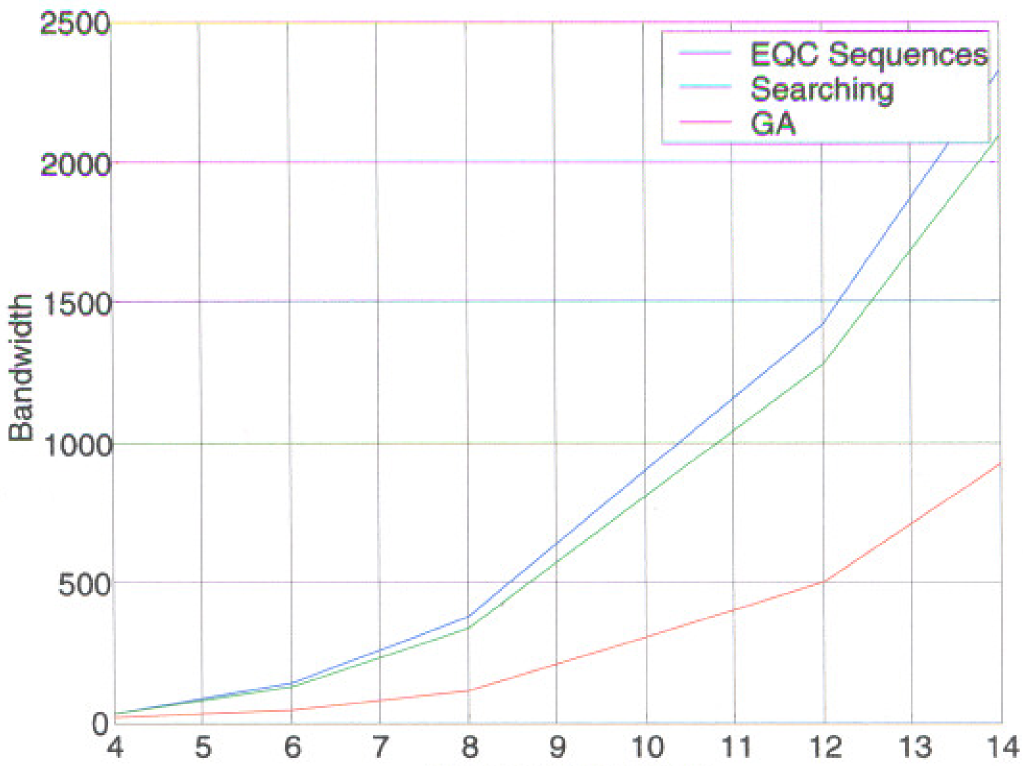


Figure 5.7: Simulation Results of Comparison of the Three Algorithms with respect to Bandwidth of the Ruler

5.6 Conclusion

The purpose of machine intelligent optimization is not necessarily to produce perfect results, but to produce the best results under the constraints of time and cost. The GA described here is successful in producing *short* rulers for each of the orders.

Genetic algorithm is a new computational paradigm suitable for solving non parametric and 'NP complete' optimization problems. It is a biologically motivated programming technique that arrives at an optimal solution through natural selection of sub optimal approaches mimicking the nature of biological evolution. Golomb Rulers represent a class of NP-complete problems. As another mark is added to the ruler, the time required to search the permutations and to test the ruler becomes exponentially great. The use of GA removes the unnecessary permutations and provides rulers with short lengths.

It is seen that with the use of genetic algorithms, the bandwidth is reduced to about three times of that achieved by the previous algorithms. This results in a large saving of the bandwidth occupied by the unequally-spaced channels. The length of the ruler is also achieved smallest as compared to the conventional algorithms. Thus, the genetic algorithms provide *good* as well as *short* rulers such that both the time and cost parameters are reduced, thus providing the optimum solutions for channel allocation.

CONCLUSIONS AND FUTURE SCOPE OF WORK

Dense WDM transmission in which individual wavelength channels are modulated at rates of 10 Gb/s offers capacities of N × 10 Gb/s, where N is the number of wavelengths. To transmit such high capacities over long distances requires operation in the 1550-nm window of dispersion-shifted fiber. In addition, to preserve an adequate signal-to-noise ratio, a 10-Gb/s system operating over long distances and having nominal optical repeater spacings of 100 km needs optical launch powers of around 1 mW per channel. For such WDM systems, the simultaneous requirements of high launch power and low dispersion give rise to the generation of new frequencies due to four-wave mixing.

Four-wave mixing (FWM) is a type of optical Kerr effect and occurs when light of two or more different wavelengths is launched into the fiber. Generally speaking, FWM occurs when light of three different wavelengths is launched into the fiber, giving rise to a new wave, the wavelength of which does not coincide with any of the others. Thus, FWM is the interaction of two or more wavelengths (channels) which results in sidebands (or ghost channels) and is caused by non-linear effects. The sidebands can coincide with other channels resulting in distortion. The effect on DWDM channels can be similar to crosstalk if only few channels are used and similar to noise if a number of channels are used. The use of dispersion compensating techniques in WDM systems further enhances the FWM efficiency. The effect of dispersion on four wave mixing is studied in chapter 3. Two WDM channels are launched over two dispersion shifted fiber spans. Dispersion is completely compensated at each span. Fiber dispersion is varied from 0 to 4 ps/nm/km through parametric runs. The power spectrum of received signals show that the FWM products decrease with increasing dispersion. Thus, FWM effect should be minimized in optical systems employing WDM and dispersion-shifted fibers.

All WDM systems for which FWM crosstalk is the main source of performance degradation can be substantially improved if FWM generation at the channel frequencies is avoided. It is therefore important to develop algorithms to allocate the channel frequencies in order to minimize the effect of FWM. If the frequency separation of any two channels of a WDM system is different from that of any other pair of channels, no FWM waves will be generated at any of the channel frequencies, thereby suppressing FWM crosstalk. A design methodology of channel spacings is presented to satisfy the above requirement. The method is a generalization of what had been proposed in the 1950's to reduce the effect of 3rd-order inter-modulation interference in radio systems. The use of proper unequal channel spacing keeps FWM waves from coherently interfering with the signals.

The term 'Golomb Ruler' refers to a set of non-negative integers such that no distinct pairs of numbers from the set have the same difference. Since the difference between any two numbers is distinct, the new FWM frequencies generated would not fall into the one already assigned for the carrier channels. Thus, to reduce four-wave-mixing crosstalk in high-capacity, long-haul, and repeater-less WDM lightwave systems, an algebraic framework for finding the solutions to the unequal-spaced channel-allocation problem has been described. The problem has been formulated algebraically as constructing Golomb Ruler sequences, which can be used as spacings between the channels effectively. Two algebraic algorithms on finding the frequency locations of unequal spaced channels have been described, which have been simulated in Matlab. The first algorithm starts with the number theoretic concept of quadratic congruence by generating extended quadratic congruence (EQC) sequences with weight P and periodic autocorrelation constraint one. We then construct "modified" sequences with weight P + 1 and aperiodic autocorrelation constraint one. This algorithm generates the sequences of any length defined by the user, the difference between any two numbers being unique, or in other words, the Golomb Ruler sequences of any number of marks can be generated. The second algorithm generates the sequences without the restriction of minimum channel spacing. The parameter, Bandwidth Expansion Factor, which gives the factor of bandwidth expanded with the use of unequal channel allocation for a given number of channels to that of equal channel allocation for the same number of channels, is calculated for some sequences generated by both the algorithms. Clearly, the BEF increases with the number of channels in both the cases. The bandwidth requirements and the time taken for both the algorithms are calculated, plotted and compared. It is seen that the search algorithm needs lesser bandwidth than the EQC algorithm. Moreover, in EQC algorithm, the time taken increases as the number of channels are increased, while search algorithm generates sequences in much less time and the time taken remains constant even if numbers of channels are increased. Thus, the search algorithm is more reliable and efficient.

Perfection is rare among Golomb Rulers. In fact Golomb has supplied a proof that says that no ruler with more than four marks can attain perfection. In this range, we search for what are termed 'Optimum Golomb Rulers'. The absence of perfection implies that some distances (within the maximum limit) cannot be measured by the ruler. But the ruler is as short as possible, while not duplicating any of the measured distances, thus saving the bandwidth occupied by the unequal spaced channels. Golomb rulers represent a class of problems known as NP-complete. Unlike the traveling salesman problem, which may be classified as a 'complete ordered set', the Golomb Ruler may be classified as an 'incomplete ordered set'. The exhaustive search of such problems is impossible for higher order models. As another mark is added to the ruler, the time required to search the permutations and to test the ruler becomes exponentially greater. The success of genetic algorithm in finding relatively good solutions to NP-complete problems such as the traveling salesman problem and job-shop scheduling problem provided a good starting point for a machine intelligent method of finding Golomb Rulers. The theorem that provides a means of removing unnecessary permutations in the heuristic-based approach also lends a hand in the GA approach. When the GA begins, the rulers will contain low order Golomb Rulers within themselves. As the population approaches the optimum point, the rulers will contain higher order Golomb rulers.

Genetic algorithms are search and optimization algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search. In every generation, a new set of artificial creatures (strings) is created using bits and pieces of the fittest of the old; an occasional new part is tried for good measure. They efficiently exploit historical information to speculate on new search points with expected improved performance.

To optimize the Golomb Rulers using genetic algorithm, the genetic algorithm toolbox for Matlab GEATBX was used to run the simulations. The objective function, to be minimized, was taken as the total bandwidth occupied by the ruler. The program was simulated for 30 generations and the total and average bandwidth noted for various generations. It is seen that as the number of generations are increased the bandwidth reduces thus we are approaching the objective. At a limit, the bandwidth starts remaining constant, this is the point where we have got the optimal results. The results are taken for different number of channels. The optimum sequences generated by the algorithm are also given. By comparing these results with those generated by the previous algorithms, we find that there is a drastic improvement in terms of bandwidth requirements by using genetic algorithms. Thus, the genetic algorithms provide *good* as well as *short* rulers such that both the time and cost parameters are reduced, thus providing the optimum solutions for channel allocation.

As the computations involved are time consuming and complex, we suggest that parallel GA may be tried for larger number of channels. Also the performance of such complex problems may be evaluated on a cluster of computers. Various approaches can also be used to generate optical Golomb ruler sequences such as fixed length approach, fixed mark approach in which the chromosomes encode the index of the distance [31], the greedy approach which places a mark at the next available position and the BDD approach [30] that is based on encoding for Golomb arrays and always produces a valid ruler. These approaches can be compared for their performance with respect to the length of the ruler or bandwidth and time requirements.

REFERENCES

[1] G.P. Aggarwal, "Nonlinear Fiber Optics", Second Edition, Academic Press, San Diego, CA, 2001.

[2] Govind P. Aggarwal, "Fiber-Optic Communication Systems", Third Edition, John Wiley & Sons, New York, 2002.

[3] Gerd Keiser, "Optical Fiber Communications", Third Edition, McGraw-Hill, New York, 2000.

[4] Andrew R. Chraplyvy, "Limitations on Lightwave Communications Imposed by Optical-Fiber Nonlinearities," Journal of Lightwave Technology, Vol. 8, No. 10, October 1990.

[5] R. Billington, "A Report of Four-Wave Mixing in Optical Fibre and its MetrologicalApplications", National Physical Laboratory, Teddington, January 1999

[6] Inoue K., " Four-Wave Mixing in an Optical Fiber in the Zero-Dispersion Wavelength Region", IEEE Journal of Lightwave Technology, vol. 10, pp. 1553-1561, 1992.

[7] Qiang Lin and Govind P. Aggarwal, "Vector theory of four-wave mixing: polarization effects in fiber-optic parametric amplifiers", J. Opt. Soc. Am. B/Vol. 21, No.6/June 2004

[8] C. W. Thiel, "Four-Wave Mixing and Its Applications", www.physics.montana.edu

[9] David F. Geraghty, Robert B. Lee, Marc BVerdiell, Mehrdad Ziari, Atul Mathur and Kerry J. Vahala, "Wavelength Conversion for WDM Communication Systems Using Four-Wave Mixing in Semiconductor Optical Amplifiers", IEEE Journal of Selected Topics in Quantum Electronics, Vol.3, No.5,October 1997

[10] Bostjan Batagelj, Matjaz Vidmar, " Fiber Nonlinear coefficient measurement scheme based on Four Wave Mixing method with externally modulated laser source", ICTON, IEEE 2002.

[11] John Zweck and Curtis R. Menyuk, "Reduction of intrachannel Four-Wave Mixing using Sub channel Multiplexing", IEEE Photonics Technology Letters, 2002

[12] Chu Xiang and James F. Young, "Wavelength Shift Keying Technique to Reduce Four-Wave Mixing Crosstalk in WDM", IEEE Lasers & Electro-Optics Society Annual Meeting Proceedings, Paper WZ2, pp.609-610, San Francisco, CA, November1999

[13] Vrizlynn L. L. Thing, P. Shum and M.K. Rao, "Channel allocation algorithm for WDM systems",Optics Express, Vol. 11, No. 11, June 2003

[14] W.C. Babcock, "Intermodulation interference in radio systems", Bell Systems Technical Journal, pp. 63-73, January 1953.

[15] James B. Shearer, "Some New Optimum Golomb Rulers", IEEE Transactions on Information Theory, IT-36:183-184, January 1990.

[16] Wing C. Kwong, and Guu-Chang Yang, "An Algebraic Approach to the Unequal-Spaced Channel-Allocation Problem in WDM Lightwave Systems", IEEE Transactions on Communications, Vol. 45, No. 3, March 1997.

[17] Apostolos Dollas,William T. Rankin, and David McCracken, "A New Algorithm for Golomb Ruler Derivation and Proof of the 19 Mark Ruler", IEEE Transactions on Information Theory, Vol. 44, No. 1, January 1998

[18] "Project OGR", http://www.distributed.net /OGR

[19] Gary S. Bloom and Solomon W.Golomb, "Applications of Numbered Unidirected Graphs", Proceedings of the IEEE, 65: 562-570, April 1977.

[20] S. Rajasekaran and G.A. Vijayalakshmi Pai, "Neural Networks, Fuzzy Logic, and Genetic Algorithms – Synthesis and Applications", Prentice Hall of India Pvt. Ltd., New Delhi, 2004.

[21] Melanie Mitchell, "An Introduction to Genetic Algorithms", Prentice Hall of India Pvt. Ltd., New Delhi, 2004.

[22] K. Deb, "Optimization for Engineering Design – Algorithms and Examples", Prentice Hall of India Pvt. Ltd., New Delhi, 2003.

[23] David E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison Wesley, USA, 1989.

[24] M. Srinivas and Lalit M. Patnaik, "Genetic Algorithms: A Survey", Proceedings of the IEEE, June 1994.

[25] Richard Baker, "Genetic Algorithms in Search and Optimization", Financial Engineering News, Issue No. 5, July 1998.

[26] N. Chaiyarataiia and A. M. S. Zalzala, "Recent Developments in Evolutionary and Genetic Algorithms: Theory and Applications", Genetic Algorithms in Engineering Systems: Innovations and Applications, Conference Publication No. 446, IEE, Sep.1997

[27] F. Forghieri, R. W. Tkach, A. R. Chparlyvy and D. Marcuse, "Reduction of Four-Wave Mixing Crosstalk in WDM Systems Using Unequally Spaced Channels", IEEE Photonics Technology Letters, Vol. 6, No. 6, June 1994.

[28] F. Forghieri, R. W. Tkach and A. R. Chparlyvy, "WDM Systems with Unequally Spaced Channels", Journal of Lightwave Technology, Vol. 13, No. 5, May 1995.

[29] S. W. Soliday, A. Homaifar and G. L. Lebby, "Genetic Algorithm Approach to the Search for Golomb Rulers", Proceedings of the Sixth International Conference on Genetic Algorithms, Vol. 1, pp. 528-535, University of Pittsburgh, PA, July 1995.

[30] J. P. Robinson, "Genetic Search for Golomb Arrays", IEEE Trans. on Information Theory, Vol. 46, No.3, May 2000.

[31] Sharat S. Chikkerur, "Generation of Sub Optimal Golomb Rulers Using Genetic Algorithms", ssc5@eng.buffalo.edu.

[32] A. J. Chipperfield and P. J. Fleming, "The MATLAB Genetic Algorithm Toolbox", IEEE Colloquium on Applied Control Techniques Using MATLAB, Digest No. 1995/014, January 1995.

APPENDIX- A

TABLE OF OPTIMUM GOLOMB RULERS

Marks	Ruler
1	0
2	0 1
3	0 1 3
4	0 1 4 6
5	0 1 4 9 11
6	0 1 4 10 12 17
7	0 1 4 10 18 23 25
8	0 1 4 9 15 22 32 34
9	0 3 9 17 19 32 39 43 44
10	0 1 6 10 23 26 34 41 53 55
11	0 1 4 13 28 33 47 54 64 70 72
12	0 2 6 24 29 40 43 55 68 75 76 85
13	0 7 8 17 21 36 47 63 69 81 101 104 106
14	0 5 28 38 41 49 50 68 75 92 107 121 123 127
15	0 6 7 15 28 40 51 75 89 92 94 121 131 147 151
16	0 1 4 11 26 32 56 68 76 115 117 134 150 163 168 177
17	0 5 7 17 52 54 56 67 80 81 100 122 138 159 165 168 191 199
18	0 2 10 22 53 56 82 83 89 98 130 148 153 167 188 192 205 216
19	0 4 13 15 42 56 59 77 93 116 126 138 146 174 214 221 240 245 246
20	0 24 30 43 55 71 75 89 104 125 127 162 167 189 206 215 272 275 282 283
21	0 4 23 37 40 48 68 78 138 147 154 189 204 238 250 251 256 277 309 331 333
22	0 1 9 14 43 70 106 122 124 128 159 179 204 223 253 263 270 291 330 341 353 356
23	0 6 22 24 43 56 95 126 137 146 172 173 201 213 258 273 281 306 311 355 365 369 372

APPENDIX – B

MATLAB CODES FOR GOLOMB RULER ALGORITHMS

E.1 Matlab Code for EQC Construction

```
clc
clear all;

%% Promp user to input value of N
p = input ('value');

%% Calculate parameters
n = ((3*p)-1)/2;
N = p+1;
S = p*((2*p)-1);

%% Declare vectors
z = [];
e = [];
f = [];
k1= 0;
result = [];
res2 = [];
ind1 = [];
ind4 = [];
tic;

%% Perform Modulo-multiplication
for i =1:1:p-1
   for j =1:1:p-1
      r = i*(j*(j+1)/2);
      R(i,j) = rem(r,p);
end
end

%% Perform modulo-addition of all elements with k.

b = zeros(p-1,1);
c = horzcat(b,R);
o = ones(p-1,p);

for k = 0:1:p-1
```

```
   m = k*o;
   M = c+m;
  K = mod(M,p);
  z = [z;K];
end
z;



%% Left-rotate from 1 to p times

for n = 1:length(z)
 z1 = z(n,1:p);
for j = 1:p
b = z1(1);
for i = 2:length(z1)
   z1(i-1) = z1(i);
end
z1(p) = b;
e = [e;z1];
end
end

%% Generate binary codewords
e;
a = 1;
z = zeros(1,(2*p-2));
h1 = horzcat(a,z);
h = horzcat(a,z);

for i = 1:(length(h)-1)
h(i+1) = h(i);
h(i) = 0;
f = [f;h];
end

f = [h1;f];
se = size(e);
sf = size(f);
k4 =1;


%% Map sequences in binary codewords
for c =1:se(1)
    for d =1:se(2)
       q = e(c,d);
```

```
        for j =1:sf(2)
          k1(c,k4) = f(q+1,j);
          k4 = k4+1;
        end
      end
      k4 =1;
  end

      k1;
      w1 = size(k1);

      %% Find indices of ones
      for w =1:w1(1)
        w2 =k1(w,:);
        one = [1];
        w3 = horzcat(w2,one);
        index = find(w3);
        res = index-1;


        res1 = [];

result = [result;res];

%% Check correlation and delete rows having correlation > 1

for i2 =1:length(res)
    for j2 =1:length(res)
    y1 = res(j2);
    if i2<j2
    x1 = res(i2)-y1;
    z1 = abs(x1);
    res1 = [res1,z1];
    end
end
end

res1;

res2 = [res2;res1];
end
result;
res2;
si = size(res2)
    for i3 =1:si(1)
        row = res2(i3,:);
```

```
   for i1 = 1:(length(row)-1)
   for j1 = (i1+1):length(row)
      if i1<j1
      if row(i1) = row(j1)
         row1 = 0;
         ind = i3;
         ind1= [ind1, ind];
      end
   end
end
end
end
ind1;
result(ind1,:) = [];

siz = size(result);
for i6 =1:(siz(1)-1)
   for j6 = (i6+1):siz(1)
      if result(i6,:) = result(j6,:)
         ind3 = i6;
         ind4 = [ind4, ind3];
      end
   end
end
ind4;
result (ind4,:) = []
toc
```

E.2 Matlab Code for Search Algorithm

```
clc
clear all;

%% Prompt the user to enter values of p and n
p = input ('enter the value of p');
n = input ('enter the value of n');
mod2 = [];
tic;

%% First delay vector
m = n:1:n+p-1;

%% Modulo multiplication to form next delay vectors
for j=1:1:p-1
   mod1=[];
```

```
  for k=0:1:p-1
     mul = j*k;
     modulo =mod(mul,p);
     pick = m(modulo+1);
     mod1 = [mod1,pick];
     end
     mod1;
     mod2 =  [mod2;mod1];
  end
  mod2;

  %% Formation of codewords

  si = size(mod2);
  s1 = [];
  for q =1:si(1)
     row =mod2(q,:);
     s(1) = row(1);
     for p = 2:p
        s(p) = row(p)+s(p-1);
     end
     s1 = [s1;s];
  end
  s1;

  ze = 0;
  seq = [];
  siz = size(s1);
  for d =1:siz(1)
     row1 = s1 (d, :);
     frow = horzcat(ze,row1);
     seq = [seq;frow];
  end
  seq
  toc
sz1=size(seq);
total1=[];
for n1=1:sz1(1)
  rowers = seq(n1,:);
  total =sum(rowres);
  total1= [total1, total];
end
total1;
minimum = minval(total1)
```

APPENDIX – C

MATLAB CODES FOR GENETIC ALGORITHM

```
clc
clear all;
nn = 20;

k1 = 0;
pop = [];


indv = [];

%% Prompt the user to input the value of spacing

n = input ('enter value of spacing');

while k1 < 10
c = [];

%% Generate initial population
for in =1:n
chrom = initip(1,[1;20]);
c = [c,chrom];
end
c;
e = size(c);

%% Check golombness and delete rows that are not golomb
for s =1:e(1)
  res = [];
   indv =c(s,:);
for i =1:length(indv)
   for j  =1:length(indv)
   y = indv(j);
   if i<j
   x = indv(i)-y;
   z = abs(x);
   res = [res,z];
end
end
end
res;
```

```
a1 = [];
for i1=1:(length(res)-1)

   for j1 = (i1+1):(length(res))
       if i1<j1
      if res(i1) = res(j1)
         a = 0;
     else
        a =1;
     end
     a1= [a1, a];

   end
end
end
a1;



d = 0;
for h =1:length(a1)
   if a1(h) = 1
      d = d+1;
   end
   d;
end


if d = length(a1)
   k1=k1+1;
   pop = [pop;indv];

end
end
end

pop;

pop1=abs(pop);

gen = 0;
maxgen = 50;

%% Find objective values
```

```
objv = fitn(pop1);
bestind1 = [];

%% Generation counter

 while gen<maxgen

ind1= [];
res3 = [];
re = [];

%% Rank the individuals
fitnv = ranking(objv);

%% Select the best fit individuals
selch = selsus(fitnv,8);

newpop = pop1(selch,:);

%% Perform recombination

newpop1 = recsp(newpop,0.8);

%% Perform mutation

newpop2 = mutint(newpop1,[1:n;1:n]);

se = size(newpop2);
res = [];

%% Check golombness and delete rows that are not golomb

for i5 =1:se(1)
    row1= newpop2(i5,:);
   re = [];
for i2 =1:length(row1)
    for j2 =1:length(row1)
    y1= row1(j2);
    if i2<j2
    x1 = row1(i2)-y1;
    z1=abs(x1);
    re=[re,z1];
    end
end
end
re;
```

```
res3 = [res3; re];
end
res3;
si =size(res3)

   for i3=1:si(1)
      row2=res3(i3,:);

   for i1=1:(length(row2)-1)
   for j1=(i1+1):length(row2)

      if i1<j1
      if row2(i1)==row2(j1)
         row3=0;
         ind=i3;
         ind1=[ind1,ind];
      end
   end
end
end

end
ind1;

  ind5=[];
for i=1:(length(ind1)-1)

   if ind1(i)==ind1(i+1)
      indu=i;
      ind5=[ind5,indu];
   end

end
ind5;
ind1(ind5)=[];
ind2=ind1;


newpop2(ind2,:)=[];

%% Find fitness values of new population
objvsel=fitn(newpop2);

%% Reinsert into old population

[pop objv]=reins(pop1,newpop2,1,1,[],objv,objvsel)
```

```
gen=gen+1;


fitnv1=ranking(objv);
selchrom=maxpop(fitnv1);

%% Find best individual

bestchrom=pop(selchrom,:)
bestind=objv(selchrom);
bestind1=[bestind1;bestind];
end
bestind1
```

Function for Calculating Objective Values

```
function obj = fitn(x)
result = [];
si = size(x);
for c=1:si(1)
      row = x(c,:);
      total = sum(row);
      result = [result; total];
   end
   obj = result;
```

Function for Finding Best Individual

```
function best = maxpop(g)
j = max(g);
for i =1:length(g)
   if g(i) = j;
      best = i;
   end
end
```

LIST OF PUBLICATIONS

- Published and presented paper on "Four Wave Mixing – Introduction and Applications" in National Conference in Electronic Circuits and Communication Systems at TIET, Patiala.

- Published and presented paper on "Latest Applications of Four Wave Mixing" in 8^{th} Punjab Science Congress Conference at Punjabi University, Patiala

- Published and presented paper on "Golomb Ruler Algorithm for WDM Channel Allocation" in 2^{nd} National Conference on Intelligent Systems and Networks at HEC, Jagadhri.