2008

Optimization of bit interleaved coded modulation using genetic algorithms

Raghu Chaitanya Doppalapudi
West Virginia University

Recommended Citation

Doppalapudi, Raghu Chaitanya, "Optimization of bit interleaved coded modulation using genetic algorithms" (2008). *Graduate Theses, Dissertations, and Problem Reports*. 2668.
https://researchrepository.wvu.edu/etd/2668

Optimization of Bit Interleaved Coded Modulation using Genetic Algorithms

by

Raghu Chaitanya Doppalapudi

Thesis submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Electrical Engineering

Matthew C. Valenti, Ph.D., Chair
Arun A. Ross, Ph.D.
Brian Woerner, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2008

Keywords: BICM-ID, Genetic Algorithms, Error-floor, Quadratic Assignment Problem

Abstract

Optimization of Bit Interleaved Coded Modulation using Genetic Algorithms

by

Raghu Chaitanya Doppalapudi
Master of Science in Electrical Engineering

West Virginia University

Matthew C. Valenti, Ph.D., Chair

Modern wireless communication systems must be optimized with respect to both bandwidth efficiency and energy efficiency. A common approach to achieve these goals is to use multi-level modulation such as quadrature-amplitude modulation (QAM) for bandwidth efficiency and an error-control code for energy efficiency. In benign additive white Gaussian noise (AWGN) channels, Ungerboeck proposed trellis-coded modulation (TCM), which combines modulation and coding into a joint operation. However, in fading channels, it is important to maximize diversity. As shown by Zehavi, diversity is maximized by performing coding and modulation separately and interleaving bits that are passed from the encoder to the modulator. Such systems are termed "BICM" for bit-interleaved coded modulation. Later, Li and Ritcey proposed a method for improving the performance of BICM systems by iteratively passing information between the demodulator and decoder. Such systems are termed "BICM-ID", for BICM with Iterative Decoding. The bit error rate (BER) curve of a typical BICM-ID system is characterized by a steeply sloping "waterfall" region followed by an error floor with a gradual slope.

This thesis is focused on optimizing BICM-ID systems in the error floor region. The problem of minimizing the error bound is formulated as an instance of the Quadratic Assignment Problem (QAP) and solved using a genetic algorithm. First, an optimization is performed by fixing the modulation and varying the bit-to-symbol mapping. This approach provides the lowest possible error floor for a BICM-ID system using standard QAM and phase-shift keying (PSK) modulations. Next, the optimization is performed by varying not only the bit-to-symbol mapping, but also the location of the signal points within the two-dimensional constellation. This provides an error floor that is lower than that achieved with the best QAM and PSK systems, although at the cost of a delayed waterfall region.

Acknowledgments

First of all I would like to thank my advisor and committee chair, Dr. Matthew C. Valenti for giving me an opportunity to work with him. There are no words in any language to express my profound gratitude towards Dr. Valenti. Without his feedback and support this thesis would not be possible. Dr. Valenti constantly supervised my work and gave me his valuable suggestions regarding my programming skills which was very helpful in the successful completion of this project. I am proud to be a student of Dr. Valenti.

I would also like to thank Dr. Woerner for being on my committee. I would like to specially thank Dr. Ross for being on my committee and courses taught by him helped in designing the genetic algorithm for this thesis. I would also like to thank Dr. Don Torrieri for his valuable feedback regarding the results obtained in this thesis.

This project is funded by Parabon Computation, Inc., under a contract from NASA. I would also like to thank Steve Armentrout, Mario Bulhoes, and Dabe Murphy at Parabon Computation, Inc.,

I would like to thank my parents, brother and my uncle Amar Yalavarthy for giving me the motivation to pursue higher studies and gave me lot of valuable suggestions which are helpful during my stay here in US. I want to say thanks to all of my friends at WVU, WCRL who stayed with me all the time and made my stay at WVU so memorable. I dedicate this thesis to my grand father Dasari Poornayya.

Contents

List of Figures

List of Tables

Notation

We use the following notation and symbols throughout this thesis.

$(\cdot)^H$	:	Complex conjugate transpose
$(\cdot)^*$	:	Complex conjugate
$E[\cdot]$	:	Expectation operator
$F_X(x)$	:	Cumulative Distribution Function (CDF) of a random variable X
$p_X(x)$	:	Probability density function (pdf) of a random variable X
$f(\cdot)$	:	Likelihood function (proportional to probability)

Bold upper case letters denote matrices and bold lower case letters denote vectors.

Chapter 1

Introduction

1.1 Introduction

In this fast-growing world of new technologies, wireless communication plays a key role, because of growing demands for Internet usage through mobiles and new software that use the Internet as a key medium to communicate. With increasing demands for wireless communication, techniques are required which can increase the data rates and reliability of the data by optimizing the existing technologies. When the system is bandlimited, it is desirable to maximize the supported data rates while minimizing the error performance. Such objectives are especially challenging in fading environments. Fading is a main concern in wireless communication which deteriorates the signal strength while traveling in the channel. A Lot of techniques were proposed to solve this fading problem.

The solution for the above problem started in the year 1982 when Ungerboeck published a paper on Trellis Coded Modulation [1]. This paper gave an intelligent overview of how to increase the data rates without increasing the bandwidth even the coding is applied. In the Ungerboeck's model, coding and modulation are considered jointly. While TCM performs well in additive white Gaussian noise (AWGN) channels, it does not perform well in fading channels because of less diversity. Diversity is an important factor in reducing the fading. Later, Zehavi [2] proposed a way to achieve high diversity order by using a bit interleaver instead of a symbol interleaver. By doing so the correlation between the symbols is reduced and the system is able to perform well in fading channels. Later Li and Ritcey [3] proposed a more effective, albeit more complex, way to decode the "BICM" signal by using iterative decoding. Such systems are termed "BICM-ID", for BICM with Iterative Decoding. Because of its good performance in fading, BICM has become a standard feature in all next generation cellular satellite and wireless networks.

BICM-ID is similar in spirit to turbo coding, but with BICM-ID there is only a single constituent encoder in contrast with turbo coding which requires two constituent codes. BICM-ID can be considered to be a serially concatenated code, with the inner encoder being replaced with a multi-level modulator. Just like a turbo code, BICM-ID systems exhibit waterfall region where the bit error rate (BER) suddenly drops as the signal-to-noise ratio (SNR) increases and an error-floor region where the BER decreases very slowly with increasing SNR. The error performance in the error floor region can be approximated with an error bound which is dependent on the symbol labeling and constellation choosed. By finding an optimal mapping and constellation, we can optimize the BICM-ID in the error floor region.

The problem of minimizing the error bound can be formulated as an instance of a particular combinatorial optimization problem known as the *quadratic assignment problem* (QAP) [4]. The QAP is an NP hard problem because no known algorithm can solve the problem in linear time. This problem dates back to year 1957 [5] when Koopmans and Beckmann first introduced QAP.

Many algorithms exist to solve QAP, but most become intractable as the problem size increases, To overcome this problem, genetic algorithms, which are widely used to solve combinatorial optimization problem can be used to solve this problem even the dimension increases it won't get stuck in the local optimal. Based on this, an optimal mapping can be obtained and new constellation can be evolved by using some heuristic techniques to evolve a design which has good performance compared to the existing ones in the error floor region.

1.2 Thesis Outline

In this thesis, the main focus is the optimization of the Bit Interleaved Coded Modulation with Iterative Decoding (BICM-ID) especially in the error floor region. The optimization is achieved by formulating it as an instance of QAP and solved using genetic algorithms to optimize the error bound. In chapter 1, a basic introduction and history about BICM systems is given. In chapter 2 BICM-ID system model was explained and the QAP formulation of the error bound is also explained. In chapter 3, a basic introduction about QAP and its application is explained along with some of the algorithms to solve this problem was explained. In chapter 4 a basic introduction to genetic algorithms and how they are useful in solving the mapping optimization problem, also tuned parameters for genetic algorithm are presented in this chapter. Later using the tuned parameters mapping was optimized and chapter 5 a detailed explanation of the constellation design was given and the optimized BICM-ID was obtained compared to the previous existing methods. Finally in the chapter 6 how this genetic algorithm can be applied to other systems and how else this algorithm could be parallelized and run on a grid computer.

Chapter 2

Bit Interleaved Coded Modulation with Iterative Decoding (BICM-ID)

2.1 System Model

Every communication system has three key ingredients: A transmitter, a channel and a receiver. At the transmitter of a digital communication system, the bits are grouped together, encoded and then modulated to form symbols. These symbols while traveling in the channel will become corrupted by AWGN noise and fading. The receiver reverses the actions performed by the transmitter by demodulating and then decoding the signal, thereby obtaining an estimate of the transmitted bits. While many systems have been developed over the years, the main distinction among these systems lies in the way that they perform modulation and coding.

A basic communication system is shown in Fig. 2.1. On the transmitter side, bits are encoded using various coding techniques to over come the noise and fading. Next, the encoded symbols are modulated and transmitted over the channel. In the channel, noise affects the signal which becomes deteriorated by the time it reaches the receiver. At the receiver, the signal is demodulated and decoded to obtain the originally sent message bits. Different kinds of modulation coding schemes have been proposed to overcome the noise and fading in the channel. In this thesis we are going to study a system with similar features as proposed. This system is termed as Bit-Interleaved Coded Modulation with

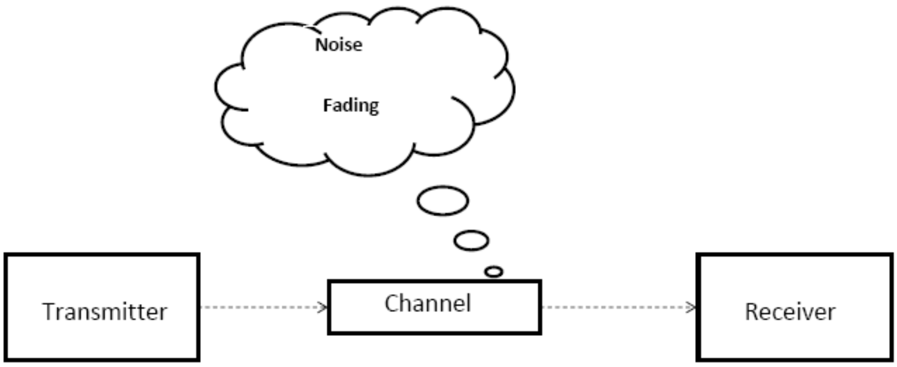


Figure 2.1: Basic Wireless Communication System

Iterative Decoding which is mostly used with band limited channels and in the channels where fading is a main concern. The name of the system itself indicates that the encoded bits are interleaved and modulated before sending over the channel. At the receiver the signal is demodulated and iteratively decoded.

A BICM-ID system is shown in Fig. 2.2 and described in the following subsections.

2.1.1 Transmitter

In the transmitter, a vector **u** of message bits is passed through a rate $r_c = \frac{B}{A}$ binary convolutional encoder. The code is denoted by (A, B, C), where A denotes the number of output bits, B denotes the number of input bits and C is the memory order of the convolutional code. A simple binary convolutional encoder is shown in Fig. 2.3. The convolutional encoder described in the Fig. 2.3 is rate $\frac{1}{3}$ encoder with constraint length of 3. The encoder output bits are computed as follows. This convolutional encoder is a finite state machine with 2^C states. Usually a trellis diagram is used to represent all the possible states. The output bits n_0, n_1, and n_2 are computed from the current input bit m_0 and last two inputs m_1, m_2 as follows:

$$n_0 = m_0 + m_2 \tag{2.1}$$

$$n_1 = m_1 + m_2 \tag{2.2}$$

$$n_2 = m_0 + m_1 + m_2 \tag{2.3}$$

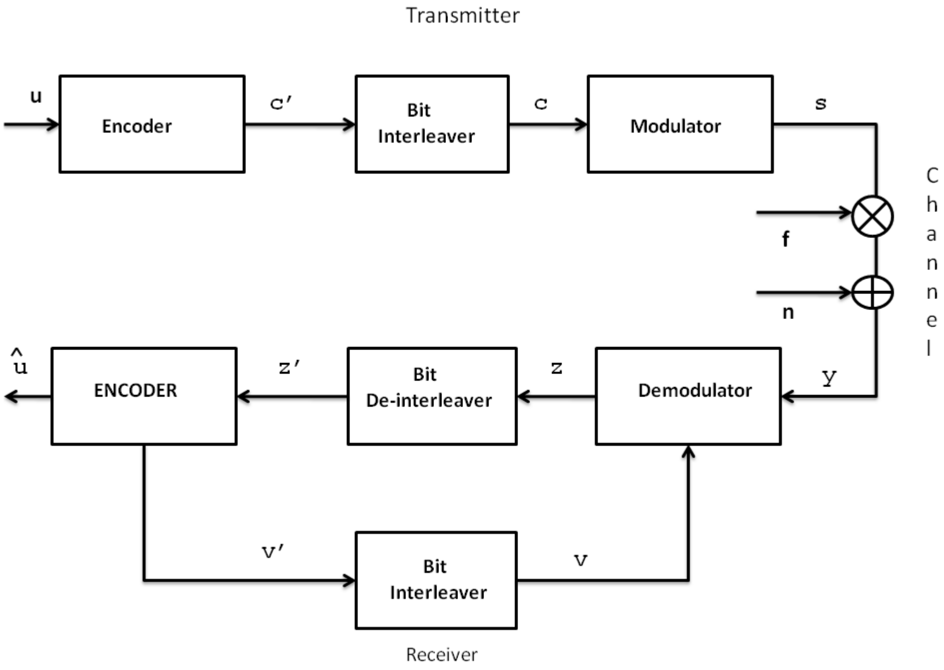


Figure 2.2: BICM-ID System Model

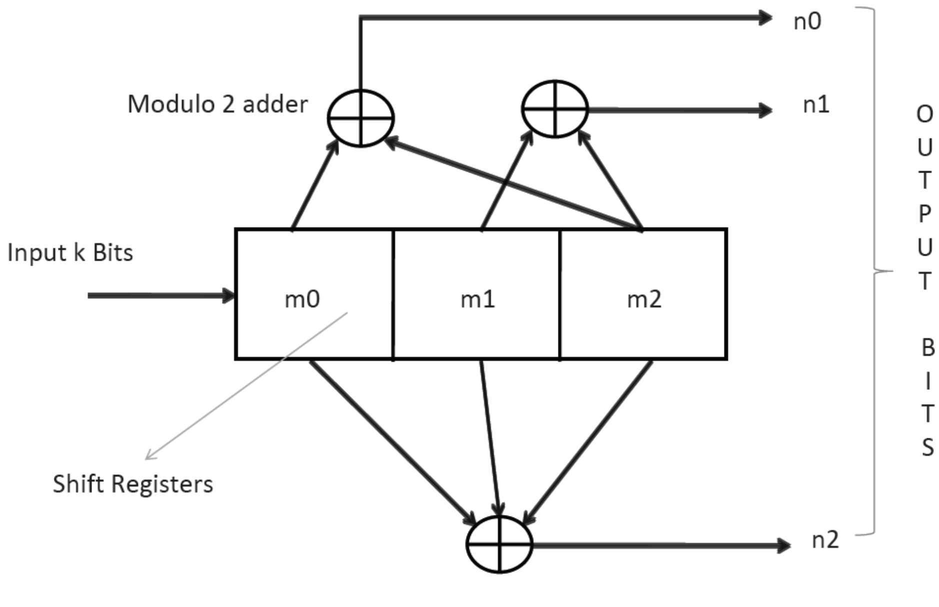


Figure 2.3: A Rate $\frac{1}{3}$ Convolutional Encoder

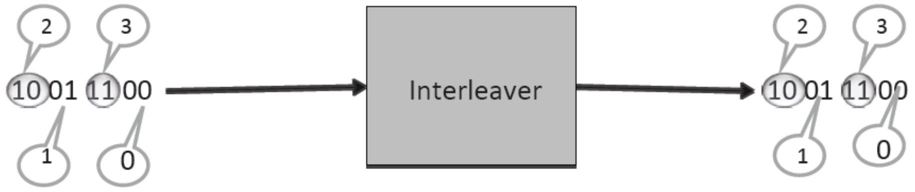


Figure 2.4: Bit Interleaver

Error correcting capability of convolutional code is given by

$$t = [\frac{d_{min} - 1}{2}] \tag{2.4}$$

where d_{min} is the minimum Hamming distance between the two code words. Decoding of convolutional codes is usually performed using the Viterbi [6] which provides maximum likelihood decoding.

Above encoder is used to produce a codeword $\mathbf{c}'$. The codeword is bitwise interleaved by a permutation matrix $\mathbf{\Pi}$ which is randomly generated to produce the bit-interleaved codeword $\mathbf{c} = \mathbf{c}'\mathbf{\Pi}$. The bit interleaver will spread burst errors, such as the errors produced in a deeply faded channel, to widely separated locations in the code word so that number of errors in each code word stays with in the correctible number of errors. The result is that a burst of errors in the channel after interleaving becomes a few scarcely spaced single symbol errors, which are more easily correctable.

In BICM-ID system the interleaver shown in Fig. 2.4 plays an important. An ideal interleaver should have the following characteristics [7]

1. Increase minimum Euclidian distance between any two code words

2. Error propagation needs to be reduced while performing iterative decoding.

Based on the above said characteristics some design rules has been proposed in [7]. But there is a drawback in this system by using interleaver which causes random modulation, this problem is resolved by using iterative decoding.

The bit-interleaved codeword $\mathbf{c}$ is then passed through a modulator to produce the vector of complex symbols $\mathbf{s} \in \mathcal{X}^M$ where $\mathcal{X} = [x_0, x_1, ..., x_{M-1}]$ is a constellation of symbols with cardinality M and average energy $\mathcal{E}_s$.

$$\mathcal{E}_s = \frac{1}{M}\sum_{i=0}^{M-1}|x_i|^2 \tag{2.5}$$

The overall rate of the system is $R = m.r_c$, where $m = \log_2 M$, and the average energy per information bit is $\mathcal{E}_b = \mathcal{E}_s/R$.

The modulator selects symbols from $\mathcal{X}$ based on the constellation labeling map and the corresponding group of m consecutive bits at the modulator input. Each signal in $\mathcal{X}$ is labeled with a unique m-bit sequence $\mathbf{b}$. Define the integer representation of a length-m binary vector $\mathbf{b}$ as

$$q(\mathbf{b}) \;\; = \;\; \sum_{k=0}^{m-1} b_k 2^{m-k-1}. \tag{2.6}$$

Let $\mathbf{x} = [x_0, x_1, ..., x_{M-1}]$ be a vector containing each signal in $\mathcal{X}$. The signals in vector $\mathbf{x}$ are indexed according to a natural mapping so that the label $\mathbf{b}$ of x_i satisfies $q(\mathbf{b}) = i$. Let $\mathbf{x}'$ be an interleaved version of the vector $\mathbf{x}$ such that $\mathbf{x}'_k = \mathbf{x}_{\mu(k)}$ where $\mu(k)$ is a permutation function. Let $\boldsymbol{\mu} = [\mu(0), \mu(1), ..., \mu(M-1)]$ be a vector containing the integers 0 through $M-1$ permuted according to the function $\mu(k)$. When the input to the modulator is $\mathbf{b}$, then the modulator selects symbol $s = x'_{q(\mathbf{b})} = x_{\mu(q(\mathbf{b}))}$ for transmission. Thus, the vector $\boldsymbol{\mu}$ specifies the labeling map for a particular ordered signal set $\mathbf{x}$.

2.1.2 Channel

The channel in a wireless communication system affects the signal in various ways. The signal at the receiver is composed of number of scattered waves caused by reflection and diffraction of transmitted signal by the objects in surrounding environment. All these waves are combined at the receiver to give a signal varying in amplitude and phase. Factors influencing this fading are multi-path propagation, mobility of reflecting objects and scatterers and relative motion between receiver and transmitter. There are many fading channel models in the literature, including:

- Rayleigh fading
- Rician Fading

- Nakagami Fading
- Dispersive Fading

In order to overcome this fading, many diversity techniques has been developed in time, frequency and space domain. Common techniques used to overcome signal fading are

- Diversity Reception and Transmission: In this thesis we used this techniques in the form of Bit interleaving which achieves time diversity.
- Orthogonal Frequency Division Multiplexing (OFDM)
- Rake Receivers
- Space Time Codes
- Multiple Input Multiple Output (MIMO)

Each coded symbol passes through a frequency-nonselective channel with complex-fading coefficient f. In this thesis, uncorrelated Rayleigh fading was assumed such that the f's are i.i.d. zero-mean complex Gaussian with unit power. The output of the channel is $y = f \times s + n$ where n is a sample of a white complex-Gaussian process with variance $N_0/2$ per dimension.

2.1.3 Receiver

The received signal at the receiver is decoded using iterative decoding. This iterative decoding philosophy was taken from turbo codes which showed very good performance for concatenated schemes. Iterative decoding works on "turbo principle" which is a kind of feedback system. At the receiver, a demapper within the demodulator processes each received symbol to produce a vector $\mathbf{z}$ of bit likelihoods. This vector provides extrinsic information that is deinterleaved and passed to the decoder. The soft-output decoder produces extrinsic information that is interleaved and provided to the demapper as a conventional decoding i.e bit metric generation and Viterbi decoding of BICM fails in the gaussian channels due vector $\mathbf{v}$ of *a priori* information.

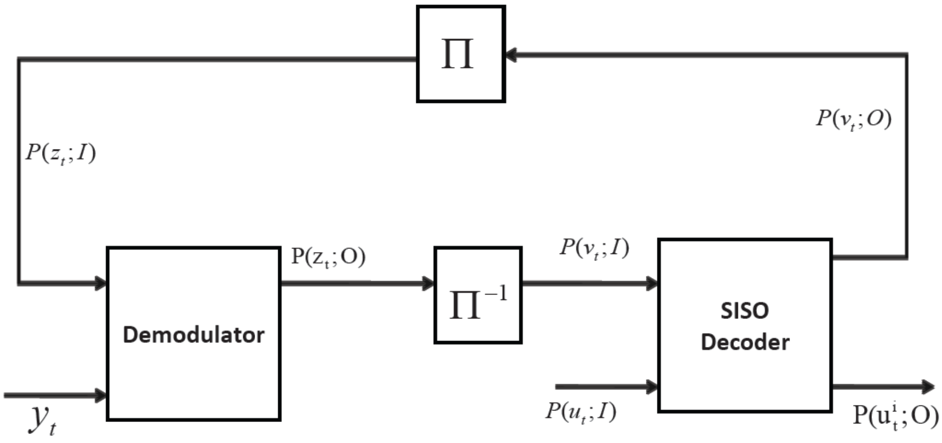


Figure 2.5: Iterative Decoder

In the above shown decoder we are using soft feedback rather than hard feedback which has improved performance. In the figure 2.5 a SISO[1] decoder and soft output demodulator are used. In the first iteration all the received signals are assumed equi-probable and the posterior probabilities generated by the SISO decoder are interleaved and this probability forms as the prior information to the demodulator and it de-modulates as soft output which is obtained from the MAP rule. In this way the information is iteratively exchanged between decoder and demodulator which is called extrinsic information used extensively in the turbo code literature.

The output of the demodulator for bit k is [8]

$$z_k = \log \frac{\displaystyle\sum_{x' \in \mathcal{X}_k^{(1)}} p(y|x') \prod_{\substack{j=0 \\ j \neq k}}^{m-1} \exp\left[\beta_j(x') v_j\right]}{\displaystyle\sum_{x' \in \mathcal{X}_k^{(0)}} p(y|x') \prod_{\substack{j=0 \\ j \neq k}}^{m-1} \exp\left[\beta_j(x') v_j\right]} \tag{2.7}$$

where the function $\beta_j(x')$ returns the j^{th} bit of the label of x' and $\mathcal{X}_k^{(b)}$ is the set of all symbols in $\mathcal{X}$ labeled with $b_k = b$.

[1] soft input soft output

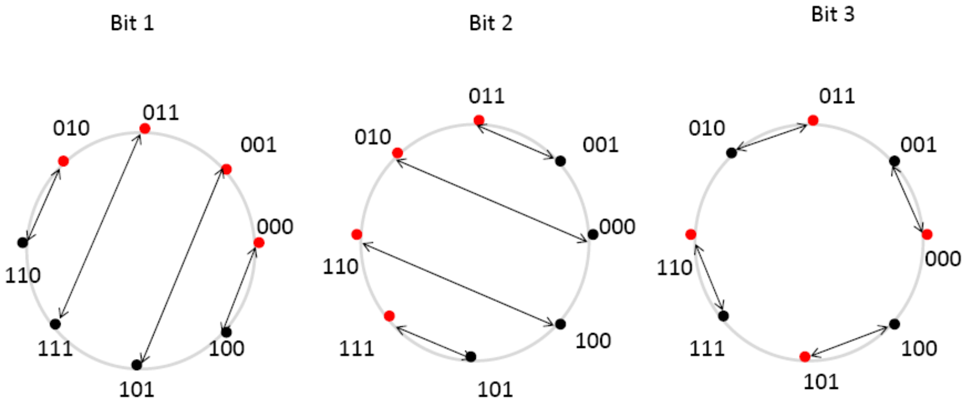


Figure 2.6: Error Free Feedback Pairs

2.2 Error Free Feedback (EFF) Bound

The union bound on the bit error probability is [9]

$$P_b \leq \frac{1}{k_c}\sum_{d=d_f}^{\infty} W_I(d) f_p\left(d, \frac{\mathcal{E}_s}{N_0}\right) \tag{2.8}$$

where d_f is the minimum distance of the convolution code, $W_I(d)$ is the total input weight of error events in the convolutional code at Hamming distance d, and $f_p(d, \mathcal{E}_s/N_0)$ is the pairwise error probability (PEP). The PEP depends on the choice of constellation $\mathcal{X}$ and its labeling map $\boldsymbol{\mu}$.

Let the function $g_k(x)$ return the symbol whose label is identical to the label of symbol x except for the k^{th} position, which is complemented. The signal x and $g_k(x)$ form an *error-free feedback (EFF) signal set*. In the following, $\{x, g_k(x)\}$ is called an *EFF pair* and $g_k(x)$ is said to be the *EFF companion* of x. The *EFF bound* on BER is found from (2.8) under the assumption that the demodulator is provided with perfect *a priori* information by the decoder, in which case the LLR of the unknown bit k has the form

$$z_k = \log p(y|x') - \log p(y|g_k(x')) \tag{2.9}$$

where x' is one of the 2^{m-1} symbols whose k^{th} position is labeled with a one, i.e. $\beta_k(x') = 1$. The decision rule for bit k involves just one EFF pair for that bit, and the identity of the pair depends on the *a priori* information fed back from the decoder. Because the decoder does not always provide perfect *a priori* information to the demodulator, and because (2.8) is an upper bound, the EFF bound is not actually a bound but rather is an approximation. However, the EFF bound is usually able to accurately predict the performance in the error floor region.

Consider how to determine $f_p(1, \mathcal{E}_s/N_0)$, i.e. the probability of a solitary bit error. Assume without loss of generality that a bit $b = 1$ is transmitted. An error will occur if the corresponding LLR z is negative

$$f_p\left(1, \frac{\mathcal{E}_s}{N_0}\right) = Pr[z < 0|b = 1] \tag{2.10}$$

Since there are m bits labeling each symbol, the PEP is found by averaging the m bit error probabilities

$$f_p\left(1, \frac{\mathcal{E}_s}{N_0}\right) = \frac{1}{m}\sum_{k=0}^{m-1} Pr[z_k < 0|b_k = 1] \tag{2.11}$$

The error probability for each bit may be found by averaging over the 2^{m-1} symbols that are labeled with $b_k = 1$

$$Pr[z_k > 0|b_k = 1] = \frac{1}{2^{m-1}}\sum_{x' \in \mathcal{X}_k^{(1)}} Pr[z_k < 0|x']. \tag{2.12}$$

Let $P_{z_k|x'}(z)$ denote the conditional CDF of z_k given that symbol x' was sent. Substituting (2.12) into (2.11) allows the PEP to be expressed as

$$f_p\left(1, \frac{\mathcal{E}_s}{N_0}\right) = \frac{1}{m2^{m-1}}\sum_{k=0}^{m-1}\sum_{x' \in \mathcal{X}_k^{(1)}} P_{z_k|x'}(0). \tag{2.13}$$

The PEP can be found using moment generating function techniques [9, 10]. Define $\phi_{z_k|x'}(s) = \mathcal{L}\left\{p_{z_k|x'}(z)\right\}$ to be the Laplace transform of the pdf of z_k given that symbol x' was sent. In Rayleigh fading [11],

$$\phi_{z_k|x'}(s) = \frac{1}{1 + s(1 - sN_0)||x' - g_k(x')||^2}. \tag{2.14}$$

From the integration property of the Laplace transform and the fact that the CDF is the integral of the pdf

$$P_{z_k|x'}(z) = \mathcal{L}^{-1}\left\{\frac{\phi_{z_k|x'}(s)}{s}\right\} \tag{2.15}$$

The PEP can thus be found by substituting (2.15) into (2.13), resulting in

$$f_p\left(1,\frac{\mathcal{E}_s}{N_0}\right) \quad = \quad \mathcal{L}^{-1}\left\{\frac{\psi(s)}{s}\right\}_{z=0} \tag{2.16}$$

where

$$\psi(s) \quad = \quad \frac{1}{m2^{m-1}}\sum_{k=0}^{m-1}\sum_{x'\in\mathcal{X}_k^{(1)}}\phi_{z_k|x'}(s). \tag{2.17}$$

When $d > 1$, the PEP is the probability that the sum of d independent bit LLRs is less than zero given that the corresponding transmitted bits are all ones. Replacing z_k in the first line of (2.11) with this sum and conditioning on the event that all bits are ones results in

$$f_p\left(d,\frac{\mathcal{E}_s}{N_0}\right) \quad = \quad Pr\left[\sum_{i=0}^{d-1} z_i < 0 \middle| \prod_{i=0}^{d-1} b_i = 1\right] \tag{2.18}$$

where i is used to index bit error events, which will generally be in different symbols. By using the convolution property of the Laplace transform, the PEP may be found using

$$f_p\left(d,\frac{\mathcal{E}_s}{N_0}\right) \quad = \quad \mathcal{L}^{-1}\left\{\frac{[\psi(s)]^d}{s}\right\}_{z=0} \tag{2.19}$$

Once the PEP is found for each d, the the bit error probability is found by substituting into (2.8).

Asymptotically, as $N_0 \to \infty$, the BER can be approximated by [9], [10]

$$P_b \quad \approx \quad \kappa\left[d_h^2\frac{\mathcal{E}_s}{N_0}\right]^{-d_f} \tag{2.20}$$

where κ is a constant and

$$d_h^2 \quad = \quad \left[\frac{1}{m2^{m-1}}\sum_{k=0}^{m-1}\sum_{x'\in\mathcal{X}_k^{(1)}}||x'-g_k(x')||^{-2}\right]^{-1} \tag{2.21}$$

is the *harmonic mean* of the squared-Euclidian distances between signals in each EFF signal set.

Taking the base-10 logarithm of (2.20), defining $X_{dB} = 10\log_{10} X$, and recalling that $\mathcal{E}_s = R\mathcal{E}_b$ yields

$$\log_{10} P_b \approx \frac{-d_f}{10}\left[\left(Rd_h^2\right)_{dB} + \left(\frac{\mathcal{E}_b}{N_0}\right)_{dB}\right] + \kappa_{dB}. \tag{2.22}$$

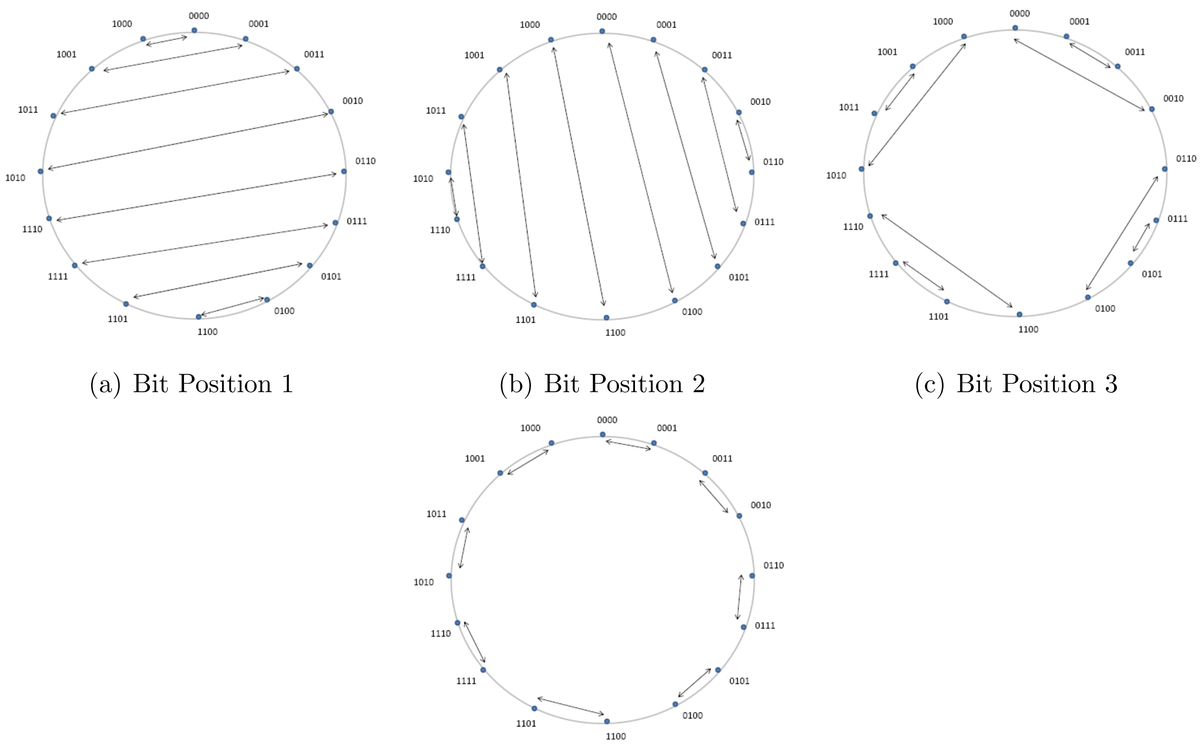
A plot of $\log_{10} P_b$ versus $(\mathcal{E}_b/N_0)_{dB}$ will be a straight line with negative slope given by d_f, which is the diversity gain. The horizontal offset of the curve is controlled by d_h^2, with larger values of d_h^2 translating into lower error rates.

From (2.21), it is clear that the choice of signal set $\mathcal{X}$ and symbol labeling map $\boldsymbol{\mu}$ influences the harmonic mean, and from (2.22) the harmonic mean determines the offset of the BER curve. Thus, to minimize the EFF bound for a given outer code, it is necessary to maximize d_h^2. For a given $\mathcal{X}$, the maximization is performed over the set of all possible $\boldsymbol{\mu}$. Such maximization is discussed in chapter 4. If one is free to chose not only $\boldsymbol{\mu}$ but also the signal set $\mathcal{X}$, then the maximization is over both functions, as described in chapter 5.

2.3 Role of Mapping

In this section we will discuss the role of mapping on maximizing the Harmonic Mean (2.21). Lets consider the case of $16 - psk$ in BICM-ID system, if we apply gray mapping to the system the resulting harmonic mean can be computed from (2.21) as 2.3142 which is not an optimal value so lets consider another mapping X which is obtained by running genetic algorithm. Which yields a harmonic mean of 3.1142 so gray label mapping is not the optimal one. From the above Figure (2.3) we can see how the EFF pairs are distributed and we can observe that they are not optimally placed i.e each EFF pair in each bit position is not placed as far as possible. So some sort of optimization of the mapping needs to be done. In this thesis techniques to optimize this mapping will be discussed, for example we can look at the following mapping which is obtained after optimizing the mapping.

In the figure 2.3 we can observe that all the EFF pairs are located at farther distances which is an optimal placement. From the above figures we can observe that d_h^2 harmonic mean is optimized by selecting a proper label mapping, in this thesis we are going to generate



(a) Bit Position 1

(b) Bit Position 2

(c) Bit Position 3

(d) Bit Position 4

Figure 2.7: EFF Pairs for Gray Labeling (a), (b) ,(c) and (d)

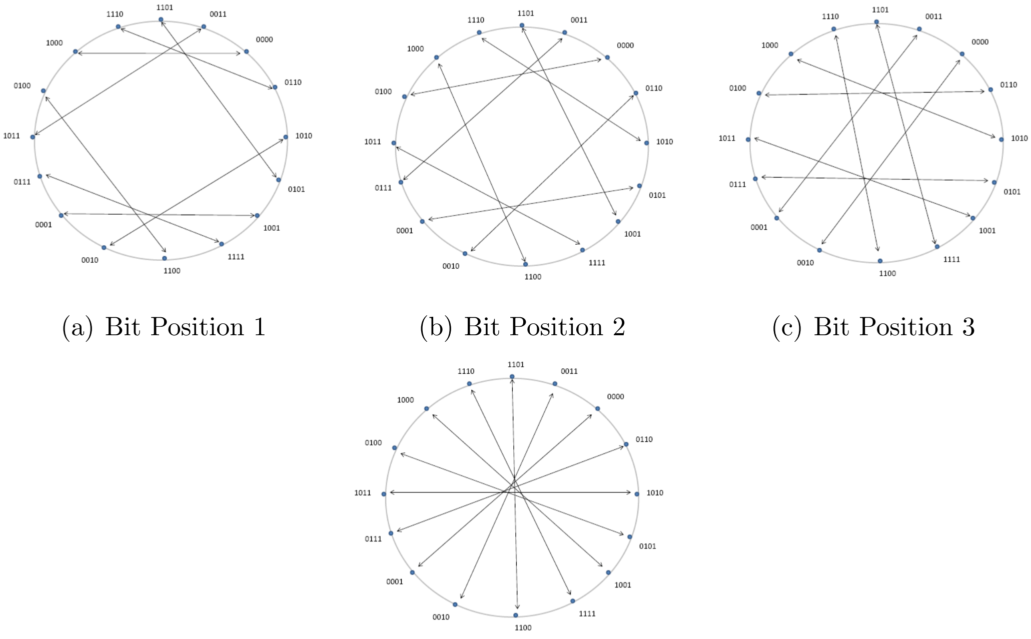
optimized label mappings using genetic algorithms which will be discussed in chapter 4

2.4 QAP Formulation

Maximizing the harmonic mean given by (2.21) over all $\boldsymbol{\mu}$ is equivalent to finding the minimum of a cost function:

$$\min_{\boldsymbol{\mu}} \sum_{k=0}^{m-1} \sum_{x \in \mathcal{X}_k^{(1)}} ||x - g_k(x)||^{-2}. \tag{2.23}$$

As discussed in [12], the minimization given by (2.23) can be formulated as an instance of the *Quadratic Assignment Problem* (QAP) [4]. Define the *flow matrix* $\mathbf{F}$ such that element $f_{i,j} = 1$ if the labels of x_i and x_j are different in just one bit position. Since the vector $\mathbf{x}$ is indexed according to a natural labeling, $f_{i,j} = 1$ whenever the binary expansion of the integers i and j have a Hamming distance of one. Otherwise, $f_{i,j} = 0$. Define the *distance*



(a) Bit Position 1

(b) Bit Position 2

(c) Bit Position 3

(d) Bit Position 4

Figure 2.8: EFF Pairs of mapping obtained from GA (a), (b) ,(c) and (d)

matrix **D** with elements

$$d_{i,j} \;=\; \begin{cases} ||x_i - x_j||^{-2}, & i \neq j \\ 0, & i = j \end{cases} \tag{2.24}$$

For a given **F** and **D**, the cost function in (2.23) may be found as

$$\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} f_{i,j} d_{i,j}. \tag{2.25}$$

The previous expression is the cost function when a natural mapping is applied. If another mapping is applied by permuting $\mathbf{x}$ to obtain $\mathbf{x}'$, then a new distance matrix $\mathbf{D}'$ must be found. However, since $x'_k = x_{\mu(k)}$, $\mathbf{D}'$ is merely a column-and-row permuted version of $\mathbf{D}$. Thus, the cost function under mapping $\boldsymbol{\mu}$ can be expressed as [12]

$$\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} f_{i,j} d_{\mu(i),\mu(j)}. \tag{2.26}$$

The optimization is to minimize the above expression with respect to all possible mapping functions (permutations) $\boldsymbol{\mu}$

Chapter 3

Quadratic Assignment Problem

3.1 Introdution and History

The Quadratic Assignment Problem was first introduced by Koopmans and Beckmann [5] in the year 1957 for modeling a plant location problem. It is described as one of the NP hard combinatorial optimization problem. NP hard means Non Deterministic polynomial time hard problem. For these type of problems no known algorithm exists that solves the problem in polynomial time. Many combinatorial optimization problems fall into this category some of the NP hard problems are Traveling salesman problem, minimum spanning tree problem etc.

Let us suppose if we want to solve the QAP by trying out each possible combination, i.e Consider $n = 30$, for which the possible combination would be $30! = 2.65 * 10^{32}$ combinations. Using the present days compute power, if the computer performs trillion calculations per second it takes around 300 years to solve this problem. Motivated by the complexity of this problem, many algorithms have been developed which can give a optimized solution in a given amount of time.

The Quadratic Assignment Problem can be described simply as follows. Suppose we have n locations and n facilities and we want to assign these n facilities to the locations that minimize the total cost of moving the goods between these facilities. We are also given the distance between each pair of locations and the flow between each pair of facilities. The cost can be given as the sum of the products of the distance and flows of the corresponding

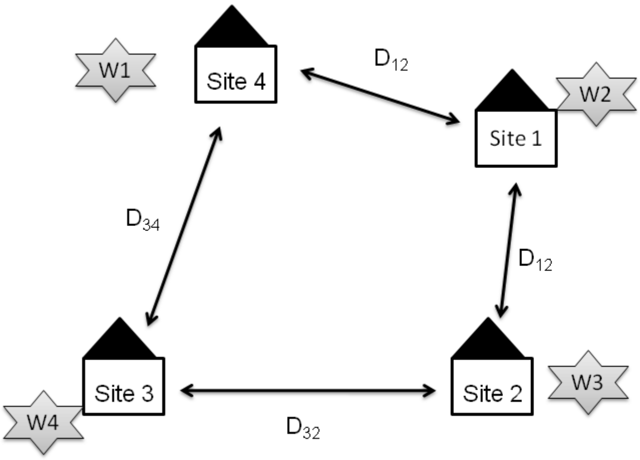


Figure 3.1: Facility Location Problem

facilities.

In the Fig. 3.1 distance between two sites is represented as $d(i,j)$ and the flow between two facilities i and j is given by $w(i,j)$ and the objective function or the cost function is given by the following equation cost function

$$Cost = \min_{\pi} \sum_{j=1}^{n} \sum_{i=1}^{n} d(i,j) \times w(i,j) \tag{3.1}$$

Above mentioned (3.1) is a simple formulation of QAP, it can also be formulated in many other ways.

- Koopmans-Beckman QAP:

  Let C and D be two n x n matrices such that $C = [c_{ij}]$ and $D = [d_{ij}]$. As above, consider the set of positive integers $1, 2, ..., n$ and let S_n be the set of permutations of $1, 2, ..., n$. Then the Quadratic Assignment Problem can be defined as follows:

$$\min \sum_{j=1}^{j=n} \sum_{i=1}^{n} c_{ij} d_{\pi_{(i)}\pi_{(j)}} \tag{3.2}$$

Over all permutations $\pi \in S_n$. The above formulation in (3.1) is known as the Koopmans-Beckman QAP. For convenience above formulation can be represented as QAP(C;D). The objective of the Quadratic Assignment Problem with cost matrix C and distance matrix D is to find the permutation $\pi_0 \in S_n$ that minimizes the double summation over all i; j. $d_{\pi_{(i)}\pi_{(j)}}$ as used above, refers to permuting the rows and columns of the matrix D by some permutation π.

- Quadratic 0-1 formulation:

  This formulation is based on the one-to-one relationship between the permutations $\pi \in S_n$ and a set of so-called permutation matrices defined as follows. Let $X = [x_{ij}]$be an n x n matrix. Then X is called a permutation matrix is it satisfies the following three conditions:

$$\sum_{i=1}^{i=n} x_{ij} = 1; j = 1, ..., n; \tag{3.3}$$

$$\sum_{j=1}^{j=n} x_{ij} = 1; i = 1, ..., n; \tag{3.4}$$

$$x_{ij} \in 0, 1; i, j = 1, ...n; \tag{3.5}$$

If all the above conditions are satisfied we can formulate QAP(C,D) as follows

$$\min_{\mathbf{X}} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} c_{ij} d_{kl} x_{ik} x_{jl} \tag{3.6}$$

3.2 Different Techniques to Solve QAP

Many techniques in the litarature have been developed to solve this problem. QAP is an NP hard problem no known algorithm exist to solve this in a polynomial time. A lot of techniques compete among themselves in terms of time complexity, memory complexity and performance in obtaining good results. In the early days when QAP [13] formulation was made many tried to solve it using an exact algorithm which would give promising results. But by sacrificing time and these algorithms became intractable as the dimension of the problem increased. Many people arrived at the algorithms which were inspired by the physics, nature and human thinking. A short description of all these techniques is given below and in this thesis the optimization was done using genetic algorithm and the results are validated using reactive tabu search [14]

3.2.1 Exact Algorithms

These algorithms try to give exact solution for the optimization problem by traversing a subset of solution space by applying constraints to the direction of traversal in the solution space, these techniques gave exact results for all QAP problem instances with $n < 30$ but very few were able to solve nugent instances of QAP. Some of the exact algorithms are discussed below.

- Branch and bound:

  This algorithm is used in solving optimization problems generally combinatorial. As QAP is a combinatorial optimization it can be solved using this techniques, among all the exact branch and bound gave good results but as the dimension of the problem increased this also became intractable. This technique evaluates all the candidate solutions and removes the useless candidates by calculating lower and upper bounds on the candidate solutions. the name of the algorithm itself indicated that the algorithm performs branching and bounding, this algorithm's performance is mainly dependent on the branching initially the solution candidates are split to smaller subsets whose union gives actual set, upon these subsets the bounding operation is performed which gives the lower and upper bounds, based on these values the algorithm whether to divide that branch has any promising results or not. If that branch has promising results, it is again branched to subsets, the above process is repeated until the subset reaches a single element. In the literature some of the instances exist for solving the Quadratic assignment problem using branch and bound [15]

- Cutting planes:

  In order to apply this technique we need to formulate quadratic assignment problem as an convex-optimization problem [13]. These cutting planes are hyper planes which separate current point from the optimal points. In this technique the solution space is reduced by these cutting planes. This technique is highly compute intensive and is not preferred for solving QAP, which also would not give the optimal solutions.

- Dynamic programming:
  This is a technique used to solve the problems which exhibit properties of overlapping and optimal substructure. There is some what close relationship between dynamic programming and branch-bound tenchnique. If the problem can exhibit the above said properties then only we can solve it using dynamic programming, programming here is nothing to do with coding it is just term used in place of mathematical solving. Our mapping optimization problem will not exhibit the above said properties so dynamic programming does not fit our frameowork.

3.2.2 Simulated Annealing

Simulated annealing derived its origin from physics where the substance like is heated initially and them gradually cooled so that the atoms in the initial stage wander randomly and they get settles as the temperature is reduced and obtain stability, this kind of phenomena is used in combinatorial optimization problems, like quadratic assignment problem. Solving QAP using simulated annealing is not a good idea because simulated annealing solves the search space whose cost function is smooth and has some hills and valleys very well than the space whose cost function is so irregular. there are lot other algorithms similar to simulated annealing like gradient descent search, neural networks but are not preferable for solving QAP.

3.2.3 Reactive Tabu Search

. Tabu Search was first introduced in the year 1986 by Fred Glover [16]. It is an iterative procedure and was able to solve many difficult optimization problem's accurately. Tabu search is used in many optimization problems like job shop scheduling, Traelling Salesman Problem. Reactive Tabu Search(RTS) is an instance of tabu search. It is an optimization problem that falls into the category of local search techniques. It works on the principle of “taboo” means prohibiting termed as “tabu”, where some of the candidate solutions are eliminated if they occur certain amount of times and it takes care that these candidates are not again repeated. Tabu search takes help of the memory in improvising the cost function. This is a kind of learning algorithm which keeps track of past solutions that were already visited, based on this memory the algorithm tries to taboo particular move for certain amount of generations. Reactive tabu search is a type of tabu search, which reacts immediately to the actions that are being taking place while the algorithm is running. Tabu search does two mechanisms for obtaining better solutions i.r intensification and diversification. In the intensification process it tries to run the algorithm intensively in the search space where a global optimal can be found. where as in the diversification process it tries to produce diversified solutions from the search space. In this thesis RTS was independently used to optimize the symbol, and these mappings were compared against those we obtained using out genetic algorithm which will be discussed in chapter 4.

3.2.4 Genetic Algorithms

Genetic algorithms perform very good at solving NP-hard combinatoiral optimization problems. Genetic algorithms works on the principle of natural evolution and selection Application of genetic algorithms increased over period of time in solving very hard optimization problem. Genetic algorithm’s application to optimization problem started by Holland in the year 1975. From then onwards GA’s attracted lot of people due to its superior solving capabilities and ease of understanding. A number of people have proposed genetic algorithms to solve QAP, initial algorithms were not able to obtain accurate results [17] In the year 2000 Ahuja proposed [18] a greedy genetic algorithm to solve Quadratic Assignment Problem which was able to solve most of the QAPLIB problems accurately. A detailed description of genetic algorithms is given in chapter 4.

3.2.5 Hybrid Algorithms

Now new techniques are being developed to solve QAP by taking advantages from each technique proposed above and combining them to one. In this type of algorithms exisitng heuristics are combined with genetic operators to solve QAP's [19]. So these hybrid alogorithms also seems to be potential solution for Quadratic Assignment Problem.

3.3 Applications of QAP

QAP has entered into real world which has many applications. Many engineering optimization problems fall into this category. QAP has entered other areas like operation research, statistics, netoworking problems. In this section some of the most commonly used applications of QAP are discussed to prove that how this QAP is very useful in solving optimization problems.

- Distributed Computing Applications:

  In distributed computing applications job scheduling is a very important task. In order to complete jobs in given amount of time we need to optimize the scheduling process, we can formulate this as an instance of QAP and solve using above described methods. In this problem computer nodes can be compared to the warehouses in the problem described in 3.1 and distance travelles by the packets between these nodes is the actual distance described above. we need to assign the jobs to the data ceneters such that distance travelled by the packets is very less and the jobs gets done in the given amount of time. This problem needs to be solved in real time, so we need to come up with algorithms which can solve the QAP in real time although it is an NP hard problem. The genetic algorithm proposed in this thesis could be used to solve such kind of real world problems.

- Wiring Problem in VLSI:

  This problem arises when we want to reduce wiring between the components of printed circuit boards. This problem could also be formulated as an instansce of QAP by considering the components on the PCB equivalent to the warehouses described in the 3.1. and the length of wiring between each component is considered as the distance between warehouses. We can optimize this problem by formulating it as Quadratic 0-1 Integer Formulation described in 3.1

- Hospital Layout Problem:

  Designing a hospital is a very difficult task. In such an environment where many lives are at stake, it is important that the design team take the necessary precautions to ensure that the facility layout is the most beneficial to both the patients and the care providers. An optimal assignment as defined by Elshafei [20] is one which minimizes the total distance traveled by patients between clinics, measured in patient- meters per year.The poor placement of the clinics combined with the increasingly overwhelming volume of traffic between them was causing delays and heavy congestion. So hospital layout needs to be optimized for serving the patients quickly and easily. This problem could also be solved by using the Quadratic 0-1 Integer Formulation.

3.4 Motivation for Genetic Algorithm(GA)

The major problem with algorithms that solve optimization problems is that they might get stuck in local minima, techniques which employ heuristics need to be careful how to get out of this local minima. Usually some sort of diversification to the solutions is added for such type of heuristics. Among all the techniques mentioned above, exact algorithms were able to perform well for the problem with less number of dimensions for example n=30 and they all failed to obtain good results for the large dimensions. So many people started working on other techniques which could be applied for large dimension even though they don't give a global optimum solution these algorithms try to give an optimal solution with in a given amount of time. In this thesis we have chosen genetic algorithm compared to other heuristics because this is not memory intensive and easy to implement compared to other ones and there is a large scope for parallelizing the GA as our dimension of QAP problem in this thesis is 64,32,16,8 so exact algorithms are intractable and other techniques like RTS are memory intensive and tuning of parameters is difficult. So we chose genetic algorithm. Before moving into next chapter about genetic algorithms, let me explain in which scenarios we can use genetic algorithms.

- If we don't have any mathematical analysis of the problem

- Search space is very large and complex
- Traditional search methods fail
- Knowledge about the problem is very less

Chapter 4

Genetic Algorithms

4.1 Introduction and History

The name Genetic Algorithm itself suggests that this algorithm uses some kind of biological convention. These algorithms fall into a broad field named evolutionary computing. Initial steps to genetic algorithm's lead from Darwin's evolutionary theory, that explains about the human's survival and evolution over the years. Before moving into the detailed explanation about the genetic algorithm let's discuss about the biological terms. Genes are fundamental parts of chromosomes which in turn form basis for cells and in turn for living organisms. Each gene carries important information which is called trait for examples information about color of hair, color of skin etc., Now let's discuss how species evolve, in the initial stages of evolution genes mutate by just interchanging the information and new genes are formed. The mutated genes and the parent genes overlap to form new genes which inherit the traits from the two parent genes, survival of these will be moving to new generation i.e. the best ones in the previous generations. In this way the traits in the genes get better and better by evolution.

Natural Selection: This is a process which makes species adapt to the environment present around it. With this process very good traits become common in upcoming generations and bad traits become uncommon. Genetic Drift: This is the process that induces changes in the traits at certain frequency. With this process new traits come into existence and survive if they were able to adopt to the surrounding environment otherwise they disappear in upcoming generations

The above two processes form basis for any form evolution that occurred naturally. The interesting behavior of human evolution caught the mind of many computer scientists. They thought if human beings are evolving over centuries based on Darwin's theory why can't we use the same theory to solve problems in the engineering world. In the year 1975 Holland with his students developed genetic algorithm which [**?**] Genetic algorithm is a search technique which is used to optimize the cost function by using natural evolution and selection process. Now-a-days lot of people are optimizing the systems using genetic algorithms, which gave very good results. NASA used genetic algorithms to evolve new antenna shapes which are used on the space shuttle.

In order to optimize the system using genetic algorithm, we need to have two requirements fulfilled. The genetic algorithm must be able to encode the problem into genotype or phenotype. The genetic algorithm must be driven by a fitness function.

If we were able to find ways to obtain above said requirements for the problem we can solve it using genetic algorithm without any other information.

Following is the pseudo code of simple genetic algorithm.

```
Initially start with random population
Evaluate fitness of each individual
Repeat the below procedure until stopping criteria is met
    Apply selection procedure and obtain good population.
    Apply breeding technique to generate new population.
    perform mutation operation to tweak some population.
        Evaluate fitness of the population and replace worst-
                 with best in the newly generated population.
```

Initially every genetic algorithm starts with random population, fitness of every individual is evaluated Genetic algorithms don't involve any kind of intelligence or memory to solve the optimization. They just evolve over time. GA [1] are simple, robust, flexible and parallel. Speed of these algorithms can be improved by parallelizing them and adopting some greedy algorithms to make them faster. Genetic algorithms are extensively used in solving combinatorial optimization problems and they are less prone to stuck in local optimum but they are compute intensive. This compute intensive problem could be solves using a grid

[1] Genetic Algorithms

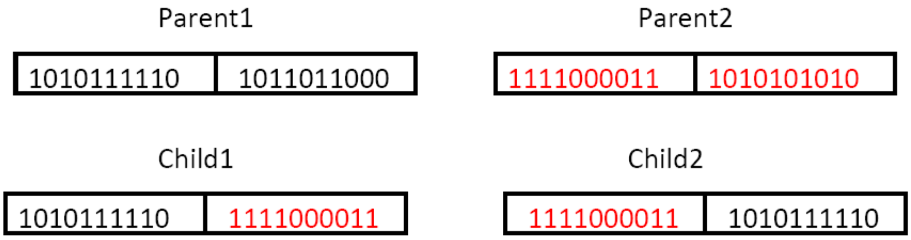


Figure 4.1: Cross-Over Operation

computer an architecture how to run genetic algorithm in parallel is proposed in chapter 6 Some of the parameters that effect the behavior of genetic algorithms are as follows.

- Crossover Rate: It is one of the important parameters of genetic algorithms to inherit some of the properties from parents to the offsprings. It is also termed as cross over probability. It indicates how often the crossover operation is performed. If the cross over probability is 100% the offspring is entirely built entirely from crossover of two parents. This percentage gives the measure of offspring's inherited properties from parent's. Some times this crossover operation is performed at single points and multiple points. In the figure shown 4.1 two parent chromosomes are represented and crossover operation is performed at single point and the first child that is formed is a half copied from first parent and another half from second parent.

- Mutation Rate : Mutation Rate is another important parameter of genetic algorithm's which helps to bring out the algorithm that is stuck in local optima. It induces different kinds of individuals into population by performing random mutations. A simple mutation operation is performed in the following figure 4.2. The parent chromosome is given before and after mutation is given, mutation is performed by inverting some of the bits in the individual. Number of inverted bits depend on the mutation probability.

- Population Size: Population size plays an important role in the evolving of genetic algorithms [21] [?] if we choose less population we might stuck in local minima or it might take more number of generations to converge to optimal value so in the literature

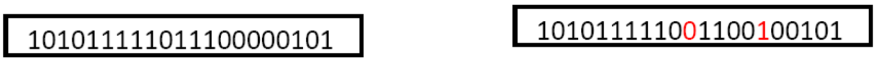


Figure 4.2: Mutation Operation

techniques were proposed to obtain population size for faster convergence.

- Stopping Criteria: This criteria is used when to stop genetic algorithm, usually genetic algorithms are stopped after certain amount of generations. They can be stopped manually after a certain optimal value is reached, in this thesis the genetic algorithm we proposed stops after certain amount of generations. Many different stopping criteria exist and depends on the situation the genetic algorithm is being used. If a genetic algorithm is used to solve a real time problem the stopping criteria would be "stop after certain amount of generations".

A detailed explanation how the parameters tuning effect the performance of genetic algorithm is discussed in the section 4.4. Based on the above parameters lot of different genetic algorithms exist and in thesis we tweaked some of these parameters to see which parameters are influencing more on the optimization criteria. Some of the genetic algorithms were discussed in the next section and the actual genetic algorithm is described in the section 4.3

4.2 Different Types of Genetic Algorithms

Different types of genetic algorithms exist, classification of GA depends on various properties. For example if population variation is considered as parameter, genetic algorithms are classified to two kinds.

1. Generational
2. Steady State

If there is a complete replacement of population between generations, it is called Generational Genetic Algorithm. If partial amount of population is replaced with new population that

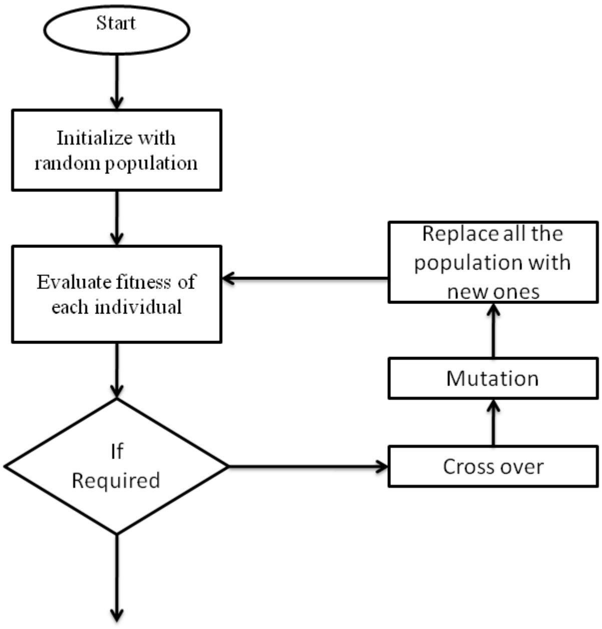

Start
Initialize with random population
Evaluate fitness of each individual
If Required
Cross over
Mutation
Replace all the population with new ones

Figure 4.3: Flowchart of Generational GA

is obtained by mutation and breeding this type of genetic algorithm is called steady state genetic algorithm. Flow charts of these 2 types of algorithm's is given below in the Figures 4.3, 4.4.

From the Fig's. 4.3 4.4 both genetic algorithms starts with random population. The main difference between both the algorithms is replacement of population from one generation to another.

In this thesis the genetic algorithm proposed is steady state, i.e after each generation weak parent population is replaced with newly obtained population from breeding and mutation operations. A detailed explanation of the history of genetic algorithm is given, its time to dwell into the actual algorithm that is used for mapping optimization.

4.3 Genetic Algorithm for Mapping Optimization

The genetic algorithm used to optimize the mapping is based on the algorithm proposed in [22]. Pseudo code of the algorithm is as follows.

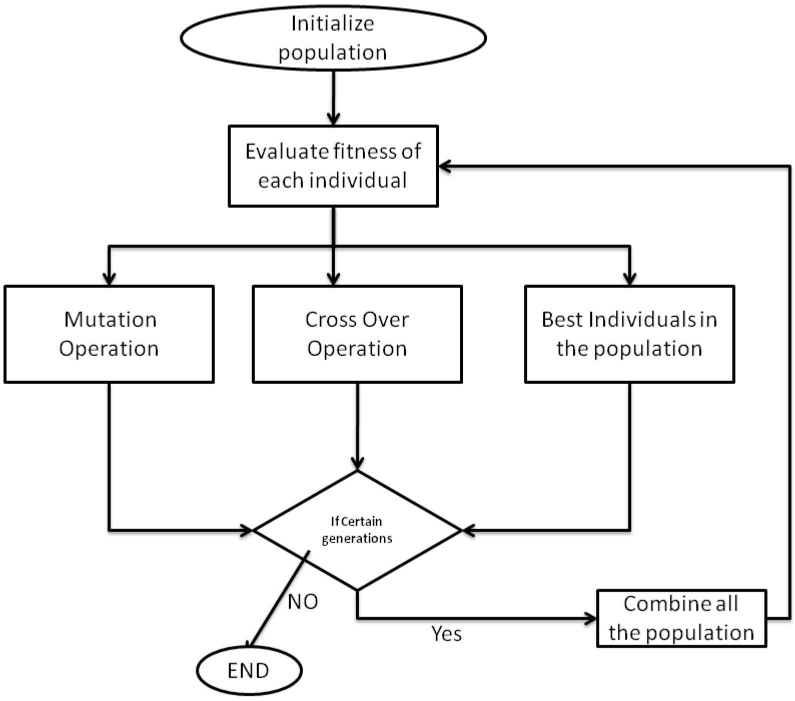

Initialize population
Evaluate fitness of each individual
Mutation Operation
Cross Over Operation
Best Individuals in the population
If Certain generations
NO
Yes
Combine all the population
END

Figure 4.4: Flowchart of Steady State GA

```
Begin
    Choose N random mappings
    Sort the Mappings from Best to Worst
    Pick first parent from anything other than Best
    Pick second parent that is better than first
    Apply Breeding to selected parents
    Start Mutation Process if children worse than parent
    If not Tc generations
        Replace parent with child if better
    else
        Replace worst among the population with best child
  Repeat the above process for certain amount of generations
End
```

The algorithm is initialized with a population of N randomly drawn mapping functions $\boldsymbol{\mu}_i, 0 \leq i \leq N-1$. The mapping functions are indexed in increasing order of cost so that mapping $\boldsymbol{\mu}_0$ is the best. Whenever the population changes, it is re-indexed.

Breeding requires the selection of two parents $\boldsymbol{\mu}_j$ and $\boldsymbol{\mu}_k$. The first parent is selected at random from the entire population except for the best $(1 \leq j \leq N-1)$, while the second is selected at random from those mappings that are better than the first $(0 \leq k \leq j)$. This *preferential parent selection rule* was not contemplated in [22–24] and improves the rate of convergence.

The two parents produce two children $\tilde{\boldsymbol{\mu}}_j$ and $\tilde{\boldsymbol{\mu}}_k$. The children are initially empty vectors of length M. Pictorial representation of the parents and children is as follows.

- Two Parents are chosen

7	**1**	6	3	2	**5**	4	0		6	**7**	1	2	5	**3**	0	4

- Copy cross over points from parents

X	**7**	X	X	X	**3**	X	X		X	**1**	X	X	X	**5**	X	X

- copy down the elements from direct parents

X	**7**	6	X	2	**3**	4	0		6	**1**	X	2	X	**5**	0	4

- Place the remaining elements in the unfilled positions

1	7	6	5	2	3	4	0		6	1	3	2	7	5	0	4

A child's *direct* parent is the parent with the same index, while its *indirect* parent is the parent with a different index. The children are generated by randomly selecting λ *crossover points*, where λ is a fixed number for all breedings. The crossover points indicate the positions of each child that are inherited from its indirect parent. At the crossover points, values from $\boldsymbol{\mu}_k$ are copied into $\tilde{\boldsymbol{\mu}}_j$ and values from $\boldsymbol{\mu}_j$ are copied into $\tilde{\boldsymbol{\mu}}_k$. The remaining positions of a child $\tilde{\boldsymbol{\mu}}_\ell$ are inherited from its direct parent $\boldsymbol{\mu}_\ell$, where $\ell = \{j, k\}$. Whenever possible, values of parent $\boldsymbol{\mu}_\ell$ that are not already in its child $\tilde{\boldsymbol{\mu}}_\ell$ are copied into $\tilde{\boldsymbol{\mu}}_\ell$ at the same position. The remaining values that could not be copied from parent to child into the same position are copied into the closest open position. A pictorial representation of the crossover operation is represented in Fig. 4.3

The cost of each of the two children are evaluated. If a child $\tilde{\boldsymbol{\mu}}_\ell$ has a lower cost than its parent $\boldsymbol{\mu}_\ell$, then the parent is replaced with the child. If the child is not better than the parent, then an optional *mutation* process is started. A sample mutation process is shown as follows

- Parent

7	1	6	3	2	5	4	0		6	7	1	2	5	3	0	4

- children

1	7	6	5	2	3	4	0		6	1	3	2	7	5	0	4

- Do mutation by swapping two positions

1	7	6	5	2	3	4	0	$\Longrightarrow$	1	7	4	5	2	3	6	0

A mutant is generated from $\tilde{\boldsymbol{\mu}}_\ell$ by exchanging the values stored in randomly chosen pairs of positions in $\tilde{\boldsymbol{\mu}}_\ell$. The *mutation rate* ρ is the probability that any given position in $\tilde{\boldsymbol{\mu}}_\ell$ will become exchanged with some other randomly chosen position. During the mutation process, a total of L mutants are created and the cost of each is evaluated. If any of the mutants have a lower cost than parent $\boldsymbol{\mu}_\ell$, then the parent is replaced with the lowest cost mutant.

During each generation, a certain number of breedings occur. While this number is arbitrary, we have set the number of breedings per generation to equal N, the overall population size. As proposed in [22], our algorithm periodically performs a *culling* operation once every T_c generations. During a culling generation, each child $\tilde{\boldsymbol{\mu}}_\ell$ is no longer compared against its direct parent. Instead, the child is compared against the current worst mapping $\boldsymbol{\mu}_{N-1}$ and if the child is better than the worst mapping, it will replace the worst mapping. It is possible that the population produced by this genetic algorithm will contain duplicates of the same mapping. During culling generations, these duplicates are removed and replaced with new random mappings. This is a brief overview of the genetic algorithm that is developed to evolve new mappings to generate low error floors.

4.4 Tuning of the Genetic Algorithm

Genetic algorithm proposed in the section 4.3 can be tuned by changing different parameters like varying the population size, varying descent search depth, adaptive depth. Usually a genetic algorithm need to be tuned before starting actual optimization process.

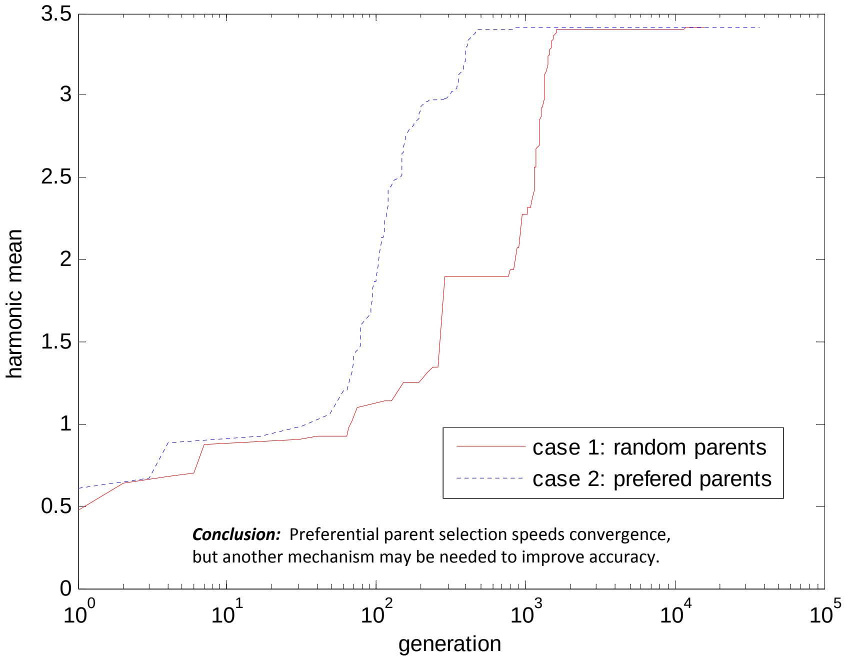


Figure 4.5: Random Vs Preferred Parent

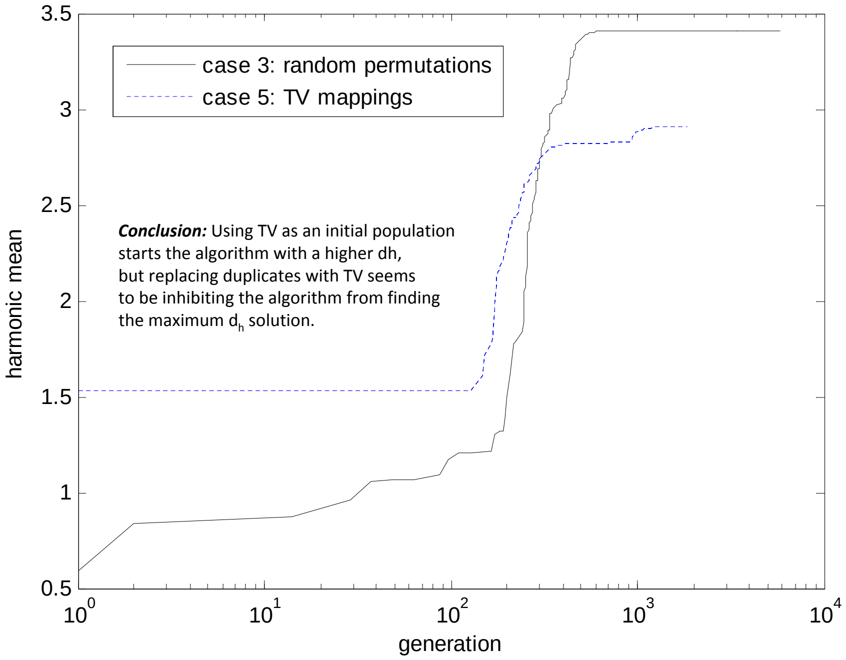
4.4.1 Random Parents vs. Preferential Parents Selection for GA

During the selection process of genetic algorithm i.e selecting the individuals that needs to be forwarded to next generation, here we can apply two different methods to select the individuals.

1. Random Parents

2. Preferential Parents

In the Random Parents selection process, the individuals are selected randomly without any rule and in preferential process individuals are selected based on the cost i.e best individuals are selected. Plot of the harmonic mean vs generations is given for both techniques in Fig 4.5. From the plot we can see that preferential parent selection converged optimal value early than random parent selection. Though random parent selection converged to optimal value. So genetic algorithm that does preferential selection work's better.



Conclusion: Using TV as an initial population starts the algorithm with a higher dh, but replacing duplicates with TV seems to be inhibiting the algorithm from finding the maximum d_h solution.

Figure 4.6: TV mappings for initial mappings and replacement

4.4.2 TV Mappings for Replacement and Initial Population

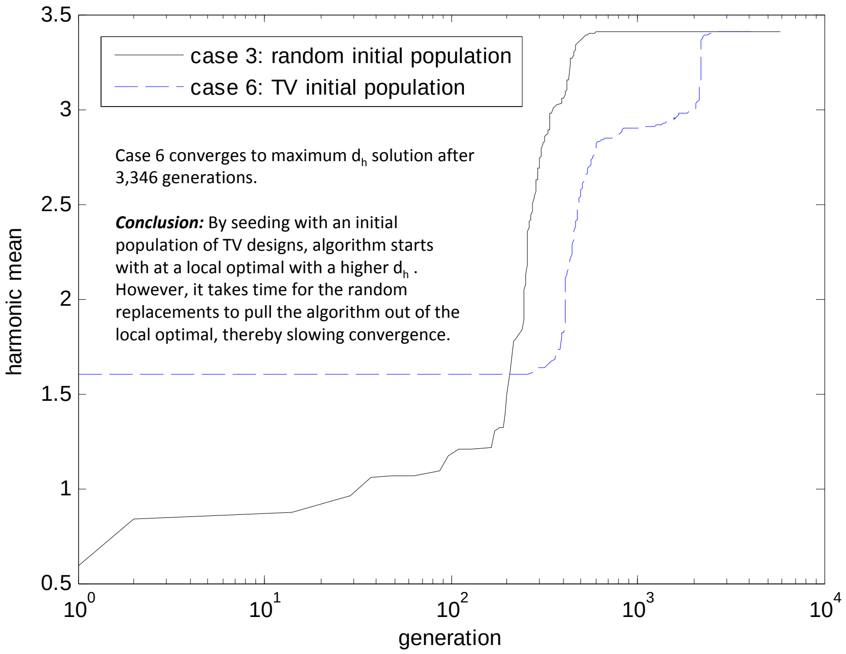
Usually every genetic algorithm starts with some initial population i.e random population is considered. But now in this GA we are going to select TV[2] mappings as initial population. Before moving onto the running of this GA let's discuss about what TV mappings are.

TV label mappings: These label maps are similar to other mappings but they have features like hamming distance between the adjacent symbols is atleast $M-1$ where M is the number of bits in the symbol proposed in [25]. These mappings showed better performance for BICM-ID system compared to the gray mapping.

During the replacement of the duplicates in genetic algorithm we use TV mappings for this purpose. With the above settings genetic algorithm was ran and the plot was given in Fig. 4.6. Running the genetic algorithms with TV mappings did not improve the performance a lot.

In the Fig. 4.7 TV mappings are used as initial mappings and the genetic algorithm was

[2] Torrieri Valenti



Case 6 converges to maximum d_h solution after 3,346 generations.

Conclusion: By seeding with an initial population of TV designs, algorithm starts with at a local optimal with a higher d_h . However, it takes time for the random replacements to pull the algorithm out of the local optimal, thereby slowing convergence.

Figure 4.7: TV mappings for initial mappings and random mapping for duplicate replacement

started, later for replacing the duplicates random mappings are used. Even this combination did not yield very good results.

4.4.3 Effect of Population Size

Population size plays an important role with genetic algorithms so this genetic algorithm was tested with different amount of populations. A Plot of the Harmonic vs. the generation number with different population sizes is given in the Fig 4.8. From the above figure it is clear that large population seems to slow down the convergence process so a small population like 100 is enough for convergence with optimal value.

4.4.4 Effect of Adaptive Depth Search

In the proposed genetic algorithm we had a parameter depth search, Here the mutation operation is performed on the off spring for certain amount of times. After certain mutations this new off spring is compared with parent to see if there is any improvement in the cost, if off spring is better than the parent it is replaced other wise it remains the same. So there

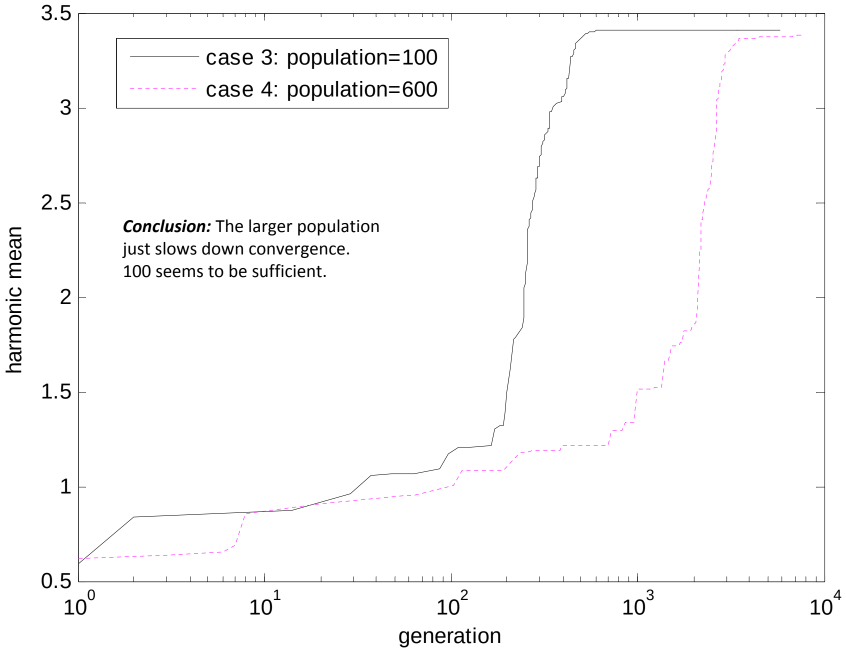


Figure 4.8: population size

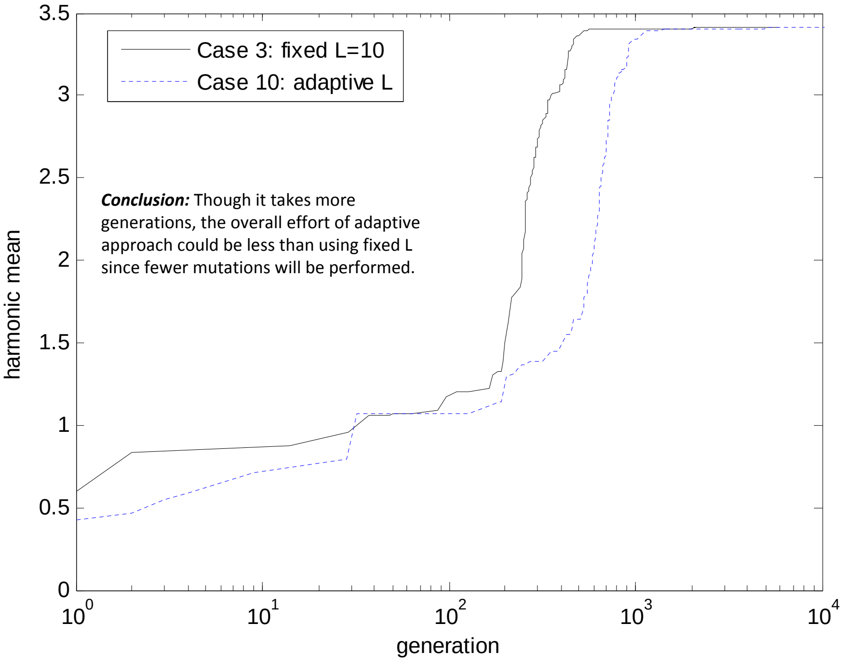
is a performance issue with the variation of this adaptive depth. In the Figure 4.9 a plot for adaptive depth and fixed depth is shown which indicates that both converge to the optimal value but adaptive depth search does less mutations in the initial stages and if there is no improvement in the child for certain amount of generations the depth value is increased.

Pseudo code for Adaptive depth search is as follows

```
Start with L=0
    When culling (i.e. once every Tc generations),
         determine if a new best mapping has been identified since the last Culling
      If yes, then L = L+1
      If not, then L = L-1
Make sure that 0 < L < Lmax.
```

From Fig. 4.9 it is clear that adaptive depth search does less effort compared to the fixed depth. So now we need to fine tune the value of fixed depth search, which value of depth search is good for obtaining the convergence fast. Plot for various depth search is given in the Fig. 4.10

From the above figure it is clear that with very small depth search convergence takes lot



Conclusion: Though it takes more generations, the overall effort of adaptive approach could be less than using fixed L since fewer mutations will be performed.

Figure 4.9: Adaptive Depth

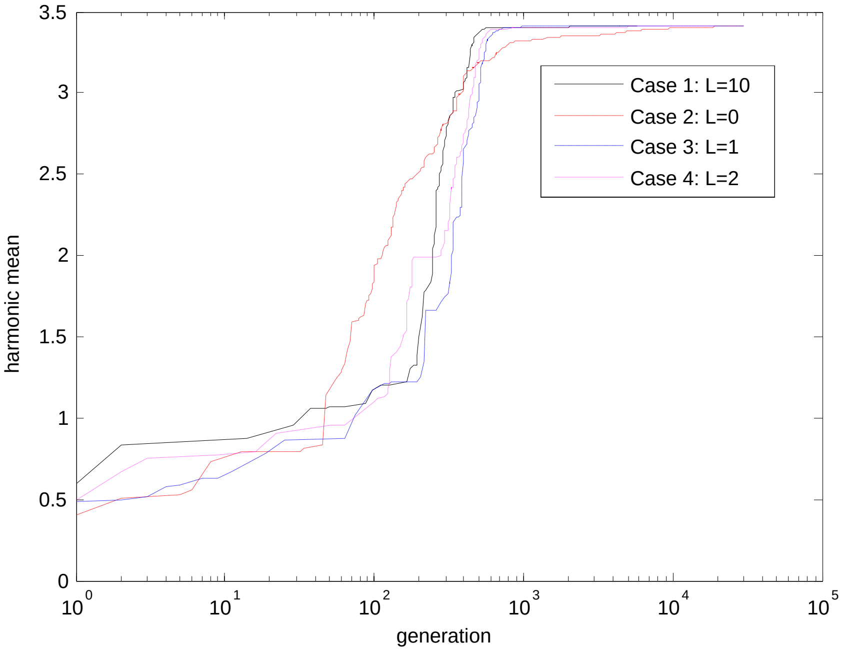
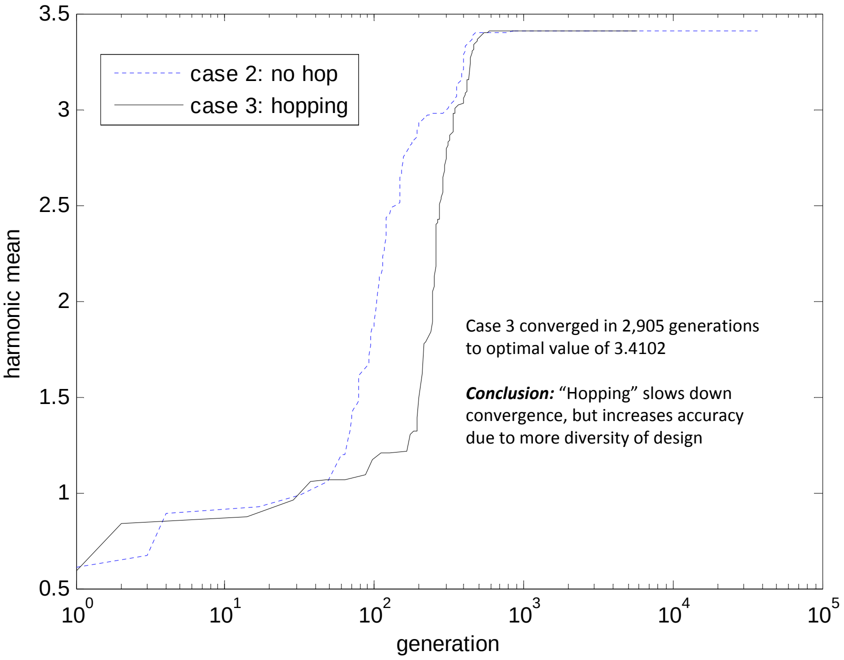


Figure 4.10: Descent Search Depth



Case 3 converged in 2,905 generations to optimal value of 3.4102

Conclusion: "Hopping" slows down convergence, but increases accuracy due to more diversity of design

Figure 4.11: Culling with random hop

of time and for L=0 and L=1 cases it did not even reach the optimal value.

4.4.5 Culling with Random Hop

For every T_c generations culling operation is performed so before this culling operation we can increase diversity of designs by performing a random hop as shown below.

- Start with child C and pick a random value 6

7	1	6	3	2	5	4	0

- Delete randomly chosen value from C producing vector of length $M - 1$

7	1	3	2	5	4	0

- Insert the removed element back in random position

7	1	3	2	6	5	4	0

In the Figure4.11 GA with hopping and no hopping. In the plot we can observe that hopping

Table 4.1: Value of d_h^2 obtained using RTS and the proposed genetic algorithm. Also listed is the number of generations required for the GA to converge.

M	Modulation	d_h^2 from RTS	d_h^2 from GA	generations
16	QAM	2.7190	2.7190	103
	PSK	3.1142	3.1142	1,249
32	QAM	2.8154	2.8154	542
	PSK	3.2916	3.2916	530
64	QAM	2.8742	2.8460	15,421
	PSK	3.4102	3.4102	8,987

slows down the convergence process but increases the accuracy by inducing new population. Running the genetic algorithm with above hopping is beneficial in obtaining accurate results.

4.5 Numerical Results

With the proposed tunning techniques in the previous section 4.4. The genetic algorithm was run to optimize the mappings of PSK and QAM with $M = \{16, 32, 64\}$ [26]. For each case, the population was $N = 100$, the number of crossover points $\lambda = \lceil M/5 \rceil$, the mutation rate $\rho = 0.02$, and the culling period $T_c = M$. The number of mutants L per child was variable. Initially, $L = 0$. If after T_c generations a new best mapping is not identified, then L is incremented up to a maximum of 20. On the other hand, if a new mapping is identified within T_c generations, then L is decremented. The value of d_h^2 after each generation, up to generation 4×10^4, is shown in Fig. 4.14. The final value of d_h^2 along with the generation that the best mapping was identified is listed in Table 4.1. Also shown are the values of d_h^2 obtained using RTS. As can be seen, the genetic algorithm has found the same optimal value as was found using RTS with the exception of 64-QAM, in which case the genetic algorithm's result is slightly inferior to that of RTS. The constellation labeling maps found by the genetic algorithm for $M = 16$ are shown in Fig. 4.12.

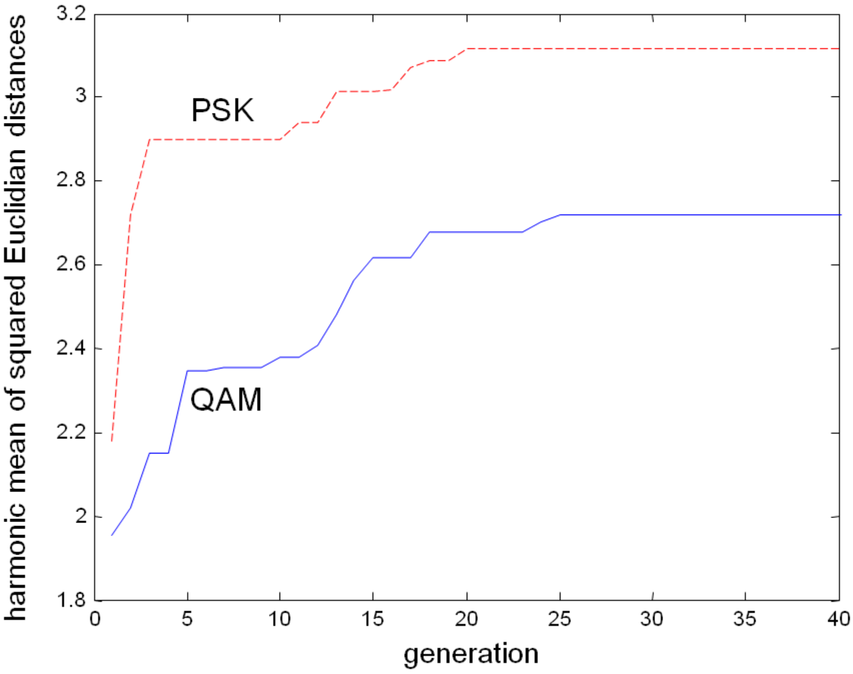


Figure 4.12: QAM and PSK constellation with optimized mapping generated using Genetic Algorithm

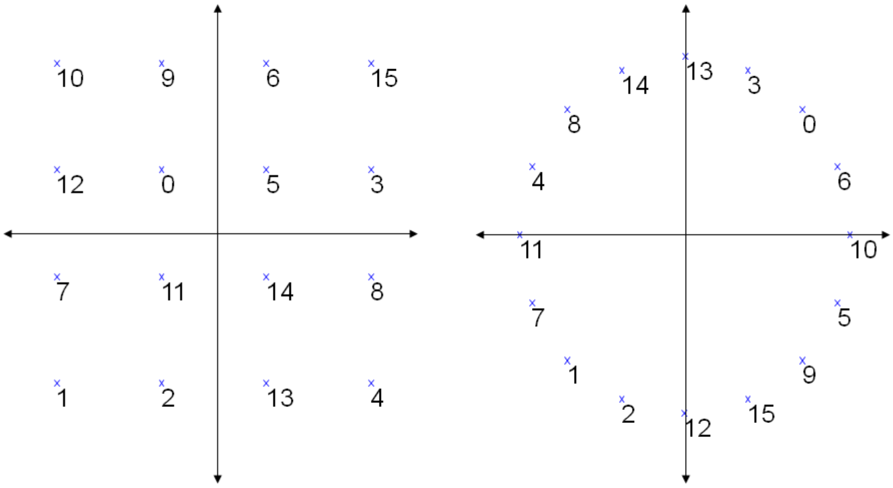


Figure 4.13: Harmonic mean of the squared-Euclidian distance between EFF signal pairs after each generation of the genetic algorithm.

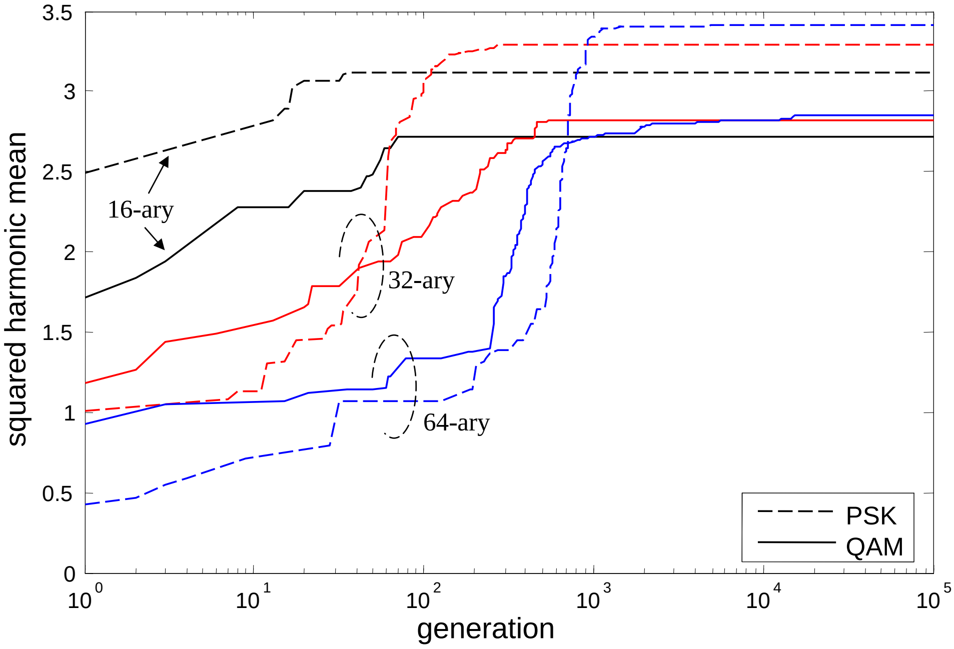


Figure 4.14: Harmonic mean of the squared-Euclidian distance between EFF signal pairs after each generation of the genetic algorithm.

Chapter 5

Optimization of Two Dimension Modulation

In chapter 2 it was mentioned that BICM-ID system in the error floor region depends on the mapping chosen. The location of the signal points within the constellation also affects the Bit Error Rate (BER) in error floor region. Solution of finding good mapping is formulated as an instance of Quadratic Assignment Problem (QAP). In chapter 4 mapping was optimized for good Results.Now, in this chapter a meta-heuristic to optimize constellation is proposed. This meta-heuristic does the optimization by moving the constellation points in the signal space. It is obvious that well placed constellation points and mappings yield good Bit Error Rate in error floor region.

5.1 Meta-heuristic to optimize the 2D Modulation

Before dwelling into the actual algorithm following is pseudo code of the algorithm.

1. Choose PSK Constellation as it has best d_h^2 Next the
2. Constellation is optimized in an attempt to increase d_h^2
3. Pick an EFF pair that has minimum d_e, defined by

$$d_e = \min_{\substack{x' \in \mathcal{X}_k^{(1)} \\ 0 \le k \le m-1}} ||x' - g_k(x')||. \tag{5.1}$$

4. These two points are then forced to be at distance αd_e, where $\alpha \geq 1$, arbitrarily chosen as 1.01

5. The new constellation is renormalized to $\mathcal{E}_s$ and reoptimized for labelling map using GA

6. Above process is repeated iteratively from step 3

In the previous chapter, the constellation $\mathcal{X}$ was fixed and only the labeling map $\boldsymbol{\mu}$ was optimized. However, d_h^2 depends not only on the labeling map but also on the constellation. We seek to increase d_h^2 beyond the values possible for QAM and PSK by moving signal points in the constellation. Ideally, the goal is to perform the following optimization

$$\min_{\mathcal{X},\boldsymbol{\mu}} \sum_{k=0}^{m-1} \sum_{x' \in \mathcal{X}_k^{(1)}} ||x' - g_k(x')||^{-2}. \tag{5.2}$$

Because the number of possible $\mathcal{X}$ is infinite, this optimization is more challenging than the one given by (2.23).

A heuristic method for performing the optimization in (5.2) is proposed [26]. The optimization begins with a standard constellation. Because PSK has a better potential d_h^2 than QAM, it is a more appropriate starting constellation. Several generations of the genetic algorithm are run on the PSK constellation to obtain an initial mapping. Next, the constellation is modified in an attempt to increase d_h^2. This can be done by first picking an EFF pair whose distance is is at the minimum d_e, where

$$d_e = \min_{\substack{x' \in \mathcal{X}_k^{(1)} \\ 0 \leq k \leq m-1}} ||x' - g_k(x')||. \tag{5.3}$$

If there are multiple EFF pairs that are distance d_e apart, then a pair is selected at random. The two points are then forced to be further apart. Let $\alpha > 1$ represent a scale factor, such that the selected EFF pair is forced to be distance αd_e apart with the same centroid. When this adjustment is made, the average energy of the constellation will typically increase, and so it must be renormalized to $\mathcal{E}_s$. Also, after the pair is pushed apart, the previously chosen labeling map might no longer be optimal, and thus it is redesigned by running one generation of the genetic algorithm with an initial population of mappings set to the final

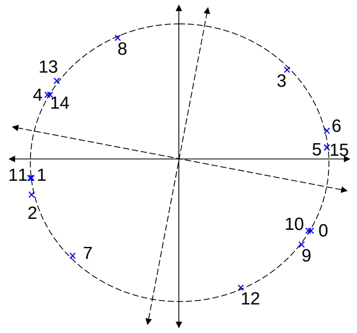


Figure 5.1: 16-ary constellation optimized using a genetic algorithm. Tilted axes have been superimposed to emphasize the symmetry that has evolved.

population of the previously optimized constellation. The process continues iteratively, with each iteration consisting of moving one pair of points apart, renormalizing the constellation, and genetically reoptimizing the constellation labeling map.

Starting with the 16-PSK constellation shown in Fig. **??**, the optimization was run with 1000 iterations and $\alpha = 1.01$. The resulting constellation is shown in Fig. 5.1. The optimization resulted in $d_h^2 = 3.6841$, which is considerably larger than the best d_h^2 found with QAM and PSK. The signal values for the constellation are listed in Table 5.1. Inspecting the constellation reveals an interesting geometry. Tilted axes have been superimposed on the figure to emphasize the symmetry in the evolved constellation. Each quadrant of the tilted constellation contains four symbols whose labels differ in exactly two bit positions; thus, there are no EFF pairs in the same quadrant. Of the four symbols in each quadrant, the first two symbols are nearly colocated, a third symbol is distance ~ 0.12 from the colocated pair, and a fourth symbol is distance ~ 0.62 from the colocated pair. The fourth symbol's EFF companions all lie on the opposite quadrant (for instance the EFF companions of '8' are $\{0, 9, 10, 12\}$). For all other symbols, two of their EFF companions are in one quadrant and the other two EFF companions are in another quadrant.

Constellations for $M = 32$ and $M = 64$ were optimized in a similar fashion. The resulting constellations had $d_h^2 = 3.5477$ for $M = 16$ and $d_h^2 = 3.5178$ for $M = 64$ which are again better than the values found for PSK and QAM. Note that for the genetically-designed constellation, d_h^2 actually decreases with increasing M, which is opposite the trend observed for QAM and PSK modulation. Thus the relative gains achieved by evolving the constellation

Table 5.1: Signal values for the constellation shown in Fig. 5.1.

Labeling	Real	Imaginary	Labeling	Real	Imaginary
0	0.8803	-0.4859	8	-0.4123	0.8977
1	-0.9894	-0.1141	9	0.8254	-0.5905
2	-0.9885	-0.2297	10	0.8704	-0.4840
3	0.7245	0.6716	11	-0.9993	-0.1120
4	-0.8704	0.4840	12	0.4123	-0.8977
5	0.9907	0.1081	13	-0.8254	0.5905
6	0.9885	0.2297	14	-0.8803	0.4859
7	-0.7160	-0.6677	15	0.9894	0.1141

decrease with increasing M.

5.2 Numerical Results

The EFF bound was calculated for the PSK and QAM constellations described in Section ?? as well as the genetically-designed constellations described in Section ??. The bounds are shown in Fig. 5.2 for the rate-1/2 convolutional code with octal generators $(7, 5)$ and $d_f = 5$. While all curves have the same slope due to the common value of d_f, there are gains in moving from one modulation choice to another. In general, there is a gain in moving to a larger alphabet and there are gains in moving from QAM to PSK and from PSK to the genetically-designed constellation. For instance, the genetically-designed constellations provide gains of 1.32 and 0.88 dB, respectively, over the 16-QAM and 64-QAM constellations. These trends are the opposite of the trends observed with uncoded modulation or BICM without feedback, indicating that BICM-ID achieves its gains in a much different manner. Indeed, the constellation shown in Fig. 5.1 would perform quite poorly without coding and iterative decoding.

To demonstrate the tightness of the EFF bound at high SNR, the three 16-ary BICM-ID systems were simulated. The simulations used a codeword length of $N = 24,000$ bits over an uncorrelated Rayleigh fading channel. The results of the simulation are shown in Fig. 5.3. Also shown on the plot are the corresponding EFF bounds. As can be seen, the EFF bound

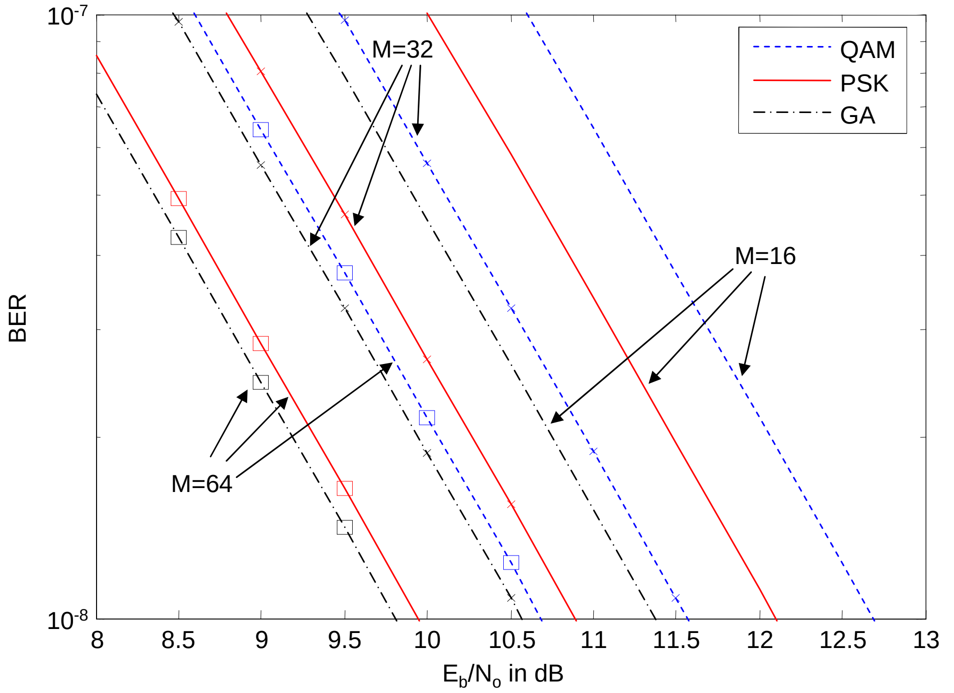


Figure 5.2: The EFF bound on BER when using a $(7,5)$ convolutional code. For each M, QAM has the highest error floor and the constellation designed with the genetic algorithm (GA) has the lowest error floor.

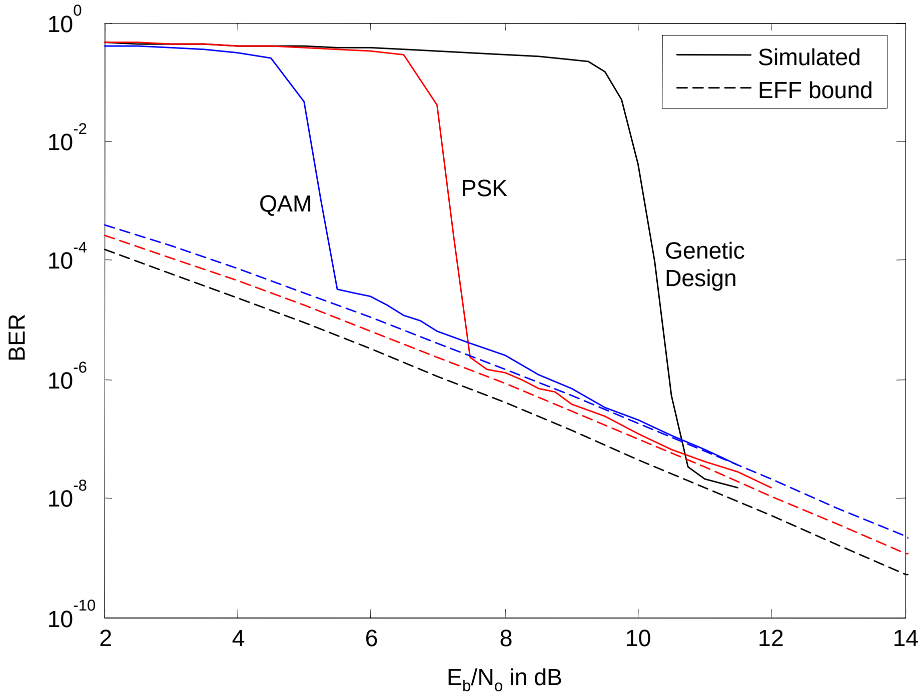


Figure 5.3: Simulated bit error performance using 16-ary modulation, a $(7, 5)$ covolutional code, and $24, 000$-bit codewords.

accurately predicts the performance at high SNR. However, it can also be seen that lowering the EFF bound comes at a cost. The waterfall region, i.e. the range of SNR characterized by a rapid drop in BER, occurs at a higher SNR for each modulation that has a lower EFF bound.

Chapter 6

Conclusion

6.1 Summary

Bit Interleaved Coded Modulation in the error floor region was optimized by choosing optimized constellation and optimized mapping obtained using genetic algorithm. This optimization of error free feed back bound was solved by formulating it as an instance of Quadratic Assignment problem. First the mapping of the BICM is optimized using genetic algorithm and the results obtained are compared with reactive tabu search. Later the constellation points are moved apart based on suggested heuristic procedure. Using the heuristic procedure and genetic algorithm, optimized constellation and mapping are obtained respectively. The EFF bound obtained with above constellation and mapping proved to more efficient than previously proposed results. It is clearly defined that the performance of BICM-ID system is effected mainly by its mapping and constellation. In this thesis role of mapping is given and various techniques to optimize this EFF bound by formulating it as an instance of QAP. Among all the suggested techniques Genetic Algorithms proved to be more efficient as the dimension of the problem increased and they can be executed in parallel. Later a heuristic procedure to optimize the constellation was suggested using this constellation points are moved in the signal space until the EFF bound is optimized for the given mapping. EFF bound's for all 16,32,64 PSK and QAM are obtained.

6.2 Future Research

In this thesis BICM-ID in error floor region was only studied i.e the high SNR region. However, for many systems, the location of the water-fall region is equally important. Such systems need to be optimized in this region, too. There is no bound in the waterfall region to be optimized, some new techniques need to be studied how to optimize the system in water fall region. The performance of genetic algorithm can also be improved by parallelizing it. By parallelizing these genetic algorithm's can be ran on a grid computer in parallel. In this section a framework to run genetic algorithm in parallel is explained.

Following are the possible ideas for continuing or branching from the research of this thesis.

6.2.1 Optimization of BICM-ID system in the waterfall region

In this thesis optimization of BICM-ID was done in error-floor region. We had a error-bound in the error floor region which is used for optimization. However, there is no such type of bound in waterfall region. Therefore, other techniques needs to be studied how to optimize in the waterfall region. One of the techniques is using EXIT charts we can know at which SNR point does the waterfall occur, so some insight into the exit charts and combination of various coding techniques might optimize in the waterfall region. This is an area where lot of potential research can be done to optimize BICM-ID.

6.2.2 Genetic Algorithms foray into other areas of wireless communication

In this thesis new designs were evolved using genetic algorithms which showed better performance compared to the previously existing designs. New space time block codes have been evolved by Valenti and Torrieri [27] which showed very good performance in fading environments than the existing designs. In the same way we can solve lot other design problems using genetic algorithms. Many people are keeping effort in application of genetic algorithms to the field of wireless communication.

6.2.3 Model of the Genetic Algorithm for running on Grid Computer

As genetic algorithm's are highly compute intensive we can run these algorithms on a grid computer to use the computer power by modeling the algorithm to run in parallel. In this section we will be talking how the proposed genetic algorithm to optimize mappings could be parallelized, so that it could be run on grid computer. Now-a-days grid computing has become a buzz word. Many people are interested in cloud computing, distributed computing and grid computing to run highly intensive computation tasks. Grid computer is a form of distributed computer where a cluster of networked systems are loosely connected to run the jobs in parallel. Many software development kits for writing grid computing applications. Using these SDK's applications could be developed which run on grid computer in parallel. Origin is a Java-based software development platform for developing distributed evolutionary computation and genetic programming applications. This Origin has two models of operation they are as follows.

- The Master/Slave Model:

  Master/Slave model suits the problem where the evaluation of cost function is highly compute intensive and needs lot of compute power. In this scenario the Origin application generates the population, does the breeding operation and mutation operation locally and evaluation of fitness of each individual is done remotely. These remote jobs evaluates the fitness of each individual in parallel and report back to the Origin application running locally. Here Origin application acts as master and remote nodes acts as slaves. Master submits the jobs onto grid and gets back results from slaves after required computation is done.

- The Opportunistic Evolution Model: In this model evolution is done on the grid nodes in parallel. After certain number of generations these remote nodes submit the population to local origin application, this merges the population from all the remote nodes and breeds a new set of population which can be submitted to remote nodes for evolution again.

Opportunistic Evolution model is preferable for the genetic algorithm proposed in this thesis. Where group of individuals are distributed to the remote nodes to evolve for certain amount of generations, then the final individuals from each node are collected and combined. Then the local origin applications submits the new subset of individuals to be evolved. This type of population transfer can also be termed as "Immigration" as people from one country migrate to another country and they become to evolve with very good traits, in the same way after certain amount of generations individuals are exchanged between the generations. This is kind of "Distributed Genetic Algorithm" development will be useful to run higly compute intensive genetic algorithms on a grid computer.

References

[1] G.Ungerboeck, “Channel coding with multilevel/phase signals,” *IEEE Trans. Inform. Theory*, vol. 28, pp. 56–67, Jan 1982.

[2] E. Zehavi, “8-psk trellis codes for a Rayleigh channel,” *IEEE Trans. Commun.*, vol. 40, pp. 873–884, May 1992.

[3] J.A. Ritcey Xiaodong Li Ritcey, “Bit-interleaved coded modulation with iterative decoding,” *Proc. IEEE Int. Conf. on Commun. (ICC)*, vol. 40, pp. 858–863, June 1999.

[4] E. Cela, *The Quadratic Assignment Problem: Theory and Algorithms*, Kluwer Academic Press, 1998.

[5] T. C. Koopmans and M. J. Beckmann, “Assignment problems and the location of economic activities,” *Econometrica*, vol. 25, pp. 53–76, 1957.

[6] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Trans. Inform. Theory*, vol. 13, pp. 260–269, Apr 1967.

[7] J.A. Ritcey Xiaodong Li Ritcey, “Bit-interleaved coded modulation with iterative decoding and 8 psk signalling,” *IEEE J. Select. Areas Commun.*, vol. 1250-1257, pp. 858–863, Aug 2002.

[8] M. C. Valenti and S. Cheng, “Iterative demodulation and decoding of turbo coded M-ary noncoherent orthogonal modulation,” *IEEE J. Select. Areas Commun.*, vol. 23, pp. 1738–1747, Sept. 2005.

[9] G. Caire, G. Taricco, and E. Biglieri, “Bit-interleaved coded modulation,” *IEEE Trans. Inform. Theory*, vol. 44, pp. 927–946, May 1998.

[10] A. Chindapol and J. A. Ritcey, “Design, analysis, and performance evaluation of BICM-ID with square QAM constellations in Rayleigh fading channels,” *IEEE J. Select. Areas Commun.*, vol. 19, pp. 944–957, May 2001.

[11] S. Benedetto and E. Biglieri, *Principles of Digital Transmission with Wireless Applications*, Kluwer, 1999.

[12] Y. Huang and J. A. Ritcey, “Optimal constellation labeling for iteratively decoded bit-interleaved space-time coded modulation,” *IEEE Trans. Inform. Theory*, vol. 51, pp. 1865–1871, May 2005.

[13] M.S.Bazaraa and Benders H.D.Sherali, "on the use of exact and heuristic cutting plane methods for the quadratic assignment problem," *Journal of operations Research Society*, pp. 991–1003, 1982.

[14] Don Torierri M.C.Valenti, "A genetic algorithm for designing constellations with low error floors," *Conference on Information Systems and Sciences*, pp. 991–1003, March 2008.

[15] F. Rendl Pardalos, P.M. and H. Wolkowicz, "The quadratic assignment problem: A survey and recent developments, in quadratic assignment and related problems," *Discrete Mathematics and The-oretical Computer Science*, pp. 1–42, 1994.

[16] Fred Glover, "Tabu Search–Part I," *INFORMS JOURNAL ON COMPUTING*, vol. 1, no. 3, pp. 190–206, 1989.

[17] Alice E. Smith David M. Tate, "A genetic apporach to the quadratic assignment problem," in *Computers and Operations Research*, 1995, pp. 73–83.

[18] Ravindra K. Ahuja, Ravindra K. Ahuja, James B. Orlin, James B. Orlin, Ashish Tiwari, and Ashish Tiwari, "A greedy genetic algorithm for the quadratic assignment problem," *Computers and Operations Research*, vol. 27, pp. 917–934, 2000.

[19] Charles Fleurent, Jacques, and A. Ferland, "Genetic hybrids for the quadratic assignment problem," in *DIMACS Series in Mathematics and Theoretical Computer Science*. 1994, pp. 173–187, American Mathematical Society.

[20] Alwalid N. Elshafei, "Hospital layout as a quadratic assignment problem," *Operational Research Quarterly (1970-1977)*, vol. 28, no. 1, pp. 167–179, 1977.

[21] Colin R Reeves, "Using genetic algorithms with small populations," in *Proceedings of the Fifth International Conference on Genetic Algorithms*. 1993, pp. 92–99, Morgan Kaufmann.

[22] Y. Wu and P. Ji, "Solving the quadratic assignment problems by a genetic algorithm with a new replacement strategy," *Int. J. Humanities and Social Sciences*, vol. 1, no. 3, pp. 151–155, 2007.

[23] C. Fleurent and J. Ferland, "Genetic hybrids for the quadradic assignment problem," *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pp. 173–187, 1994.

[24] Z. Drezner, "A new genetic algorithm for the quadratic assignment problem," *INFORMS J. Computing*, vol. 15, pp. 320–330, 2002.

[25] Don Torrieri and Matthew C. Valenti, "Constellation labeling maps for low error floors," *Military Communications Conference, 2007. MILCOM 2007. IEEE*, pp. 1–6, Oct. 2007.

[26] M.C. Valenti, R. Doppalapudi, and D. Torrieri, "A genetic algorithm for designing constellations with low error floors," *Information Sciences and Systems, 2008. CISS 2008. 42nd Annual Conference on*, pp. 1155–1160, March 2008.

[27] D. Torrieri and M.C. Valenti, "Efficient space-time block codes designed by a genetic algorithm," *Military Communications Conference, 2008. MILCOM 2008. IEEE*, Nov. 2008.