Poznan University of Technology

Doctoral Thesis

The Use of Evolutionary Algorithms in the Next-Generation Wireless Systems

Mohammed Jasim Obaid Khafaji

Supervisor:
dr hab. inż. Maciej Krasicki

Thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy
in the
Faculty of Computing and Telecommunications

Poznań, 2022

Abstract

The rapid evolution of wireless communication techniques presents considerable challenges arising from the demanding nature of the physical medium and the complexities of these techniques. Interference and multipath-induced fading exemplify the dominant issues in wireless communications.

Over the years, metaheuristics, such as Evolutionary Algorithms (EAs), have attracted increasing interest of scientists as a tool for solving many complex optimization problems. The EAs consist in a guided random search, independent of the details of the optimization problem to be solved. They model a rudimentary form of memory to archive the best solutions encountered. Therefore, they provide reasonable solutions under all conditions. One of the most widely used metaheuristics is Genetic Algorithm (GA), which has gained the attention as an optimization framework in the field of wireless communication systems.

This dissertation discusses the use of GA for Multi-User Detection (MUD) purposes in a Multi-User Multiple-Input Multiple-Output (MU-MIMO) scenario, in which the receiver is equipped with a few antennas and individual users transmit their signals concurrently in the same bandwidth.

To converge, GA applies several operators, such as selection, crossover, and mutation, to a set of temporal solutions. They are controlled by some parameters, the optimal values of which are obtained empirically as part of the work. With the aim of enhancing the GA convergence the search begins with some known good solutions rather than a random set, *i.e.*, the use of a Zero-Forcing (ZF) detector is used in the initial processing phase. The joined ZF-GA detector manages to outperform the original ZF reference in terms of Bit Error Rate (BER) vs. Signal-to-Noise Ratio (SNR).

The author contributes a new detection approach to generate the initial GA population, which conceptually evokes the Successive Interference Cancellation (SIC) procedure: an initialization scheme uses the ZF outcome with respect to the strongest signal as a reference set of solutions to create new individuals, thereby giving an initial direction to a good search region. This novel directed initialization improves the performance of the GA-driven MU-MIMO detector at no extra computational cost in comparison to the first ZF-GA solution.

The second scope of this dissertation is the use of GA as a learning mechanism to improve the performance of an adaptive channel equalizer. A novel Uni-Cycle Genetic Algorithm (UCGA) is designed with the aim of working in real time. The proposed solution has the advantage of requiring only one generation per signalling interval, which reduces computational costs, significantly, when compared to other GA-driven channel equalizers. This approach can outperform conventional solutions in terms of convergence speed and channel tracking reliability. This makes the proposed solution attractive for future wireless systems.

Streszczenie

Błyskawiczny rozwój łączności bezprzewodowej stwarza istotne problemy wynikające z kłopotliwej natury medium transmisyjnego, jak i stopnia skomplikowania samych technik transmisji. Interferencja sygnałów i zaniki, których przyczyną jest transmisja wielodrogowa, stanowią dominujące przeszkody w zapewnieniu niezawodnej łączności bezprzewodowej.

Na przestrzeni lat metaheurystyki, takie jak algorytmy ewolucyjne, cieszyły się rosnącym zainteresowaniem naukowców, którzy widzieli w nich metodę rozwiązywania wielu skomplikowanych problemów optymalizacyjnych. Algorytmy ewolucyjne mają charakter wspomaganego przeszukiwania losowego, niezależnie od szczegółów problemu. Wykorzystują przy tym prostą formę pamięci do przechowywania znalezionych dotąd najlepszych rozwiązań. Są w stanie dostarczyć rozwiązań o akceptowalnej jakości (choć niekoniecznie faktycznie optymalnych), niezależnie od warunków danego zagadnienia. Jednym z najbardziej popularnych przykładów metaheurystyk jest algorytm genetyczny. Zyskał on uznanie jako algorytm optymalizacyjny w dziedzinie systemów łączności bezprzewodowej.

Niniejsza praca dotyczy zastosowania algorytmu genetycznego do detekcji sygnałów w systemie łączności bezprzewodowej typu Multi-User MIMO (MU-MIMO), w którym odbiornik wyposażony jest w wiele anten, a poszczególni użytkownicy nadają swoje sygnały jednocześnie w tym samym pasmie.

Dla utrzymania postępu w swoim działaniu, algorytm genetyczny realizuje względem zbioru aktualnie rozpatrywanych rozwiązań swoiste operacje, jak selekcja, krzyżowanie i mutacja. Parametry tych operacji zostały w pracy dobrane empirycznie. Dla poprawy szybkości, z jaką algorytm genetyczny znajduje pożądane rozwiązanie, zamiast całkowicie losowej inicjalizacji do populacji początkowej wprowadzany jest wynik działania prostego detektora typu ZF (*Zero Forcing*). Powstały w ten sposób detektor ZF-GA przewyższa zwykły detektor ZF pod względem wartości bitowej stopy błędu (BER) w danych warunkach stosunku sygnału do szumu.

W niniejszej pracy autor przedstawia alternatywną metodą inicjalizacji populacji algorytmu genetycznego, która pod względem koncepcji wywodzi się z kręgu metod sukcesywnej redukcji interferencji: wszystkie osobniki populacji początkowej reprezentują

wynik działania detektora ZF, ale tylko dla najsilniejszego sygnału. Taka inicjalizacja nadaje wyraźny kierunek działania algorytmu genetycznego – w stronę obszaru, w którym leżą pożądane rozwiązania problemu optymalizacyjnego. Jest to rozwiązanie, które poprawia działanie detektora w systemie MU-MIMO bez wzrostu nakładów obliczeniowych względem poprzedniej propozycji autora, tj. detektora GA-ZF.

Dalsza część pracy została poświęcona wykorzystaniu algorytmu genetycznego jako narzędzia do adaptacji korektora kanału radiowego. Zaproponowane z myślą o działaniu w czasie rzeczywistym rozwiązanie, w którym na dany odstęp modulacji przypada tylko jedno pokolenie algorytmu genetycznego, pozwala znacznie ograniczyć nakłady obliczeniowe względem innych korektorów kanału wspomaganych algorytmem genetycznym. Przedstawiony sposób sterowania korektorem zapewnia szybką zbieżność i dobre śledzenie stanu kanału w porównaniu z rozwiązaniami znanymi z literatury – jest zatem obiecującym rozwiązaniem pod kątem możliwych zastosowań w przyszłych systemach łączności bezprzewodowej.

Acknowledgements

Foremost, I would like to thank my wife, *Hind*. Her support, encouragement, quiet patience and unwavering love were undeniably the bedrock upon which the past seventeen years of my life have been built. Her tolerance of my occasional vulgar moods is a testament in itself to her unyielding devotion and love. Sweet thank is also to my lovely son, *Aiham*, for his sweet smile and understanding.

I would like to show my sincere gratitude to my supervisor, dr hab. inż. *Maciej Krasicki*. It is impossible to get this thesis without his brilliant advice, guidance, support, and encouragement during my PhD study. For everything you've done for me, Dr *Krasicki*, I thank you.

Last but not least, a special thanks to my second family, *Piotr*, *Hanna*, *Lilianna* and *Jan*. I deeply thank my friend *Piotr*, who offered invaluable support and humour over the years.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

4G	Fourth-generation
5G	Fifth-generation
ACO	Ant Colony Optimization
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BPSK	Binary Phase-Shift Keying
CDMA	Code Division Multiple Access
DDFSE	Delayed Decision Feedback Sequence Estimation
DFE	Decision Feedback Equalizer
DS-CDMA	Direct Sequence -Code Division Multiple Access
EAs	Evolutionary Algorithms
EC	Evolutionary Computing
EP	Evolutionary Programming
ES	Evolution Strategies
FIR	Finite Impulse Response
FPGAs	Field-Programmable Gate Arrays
GA	Genetic Algorithm
GBML	Genetic-Based Machine Learning
GP	Genetic Programming
GSM	Global System for Mobile Communications
GWO	Grey-Wolf optimizer
IC	Interference cancellation
ICD	Inverse Channel Detection
IGA	Island Genetic Algorithm
IIR	Infinite Impulse Response
IoT	Internet of Things
ISI	Inter-Symbol Interference
LMS	Least Mean Square
LMS-AE	Least Mean Square -based Adaptive Equalizer

LS	Least Squares
MAI	Multiple Access Interference
MANET	Mobile ad hoc networks
MC-CDMA	Multicarrier- Code Division Multiple Access
MIMO	Multiple-Input Multiple-Output
ML	Maximum-Likelihood
MLSE	Maximum Likelihood Sequence Estimation
MMSE	Minimum Mean Square Error
mmWave	millimeter Wave
M-QAM	M element Quadrature Amplitude Modulation
MSE	Mean Square Error
MS-MMSE	Multi Stage- Minimum Mean Square Error
MUD	Multi-User Detection
MU-MIMO	Multi User-Multiple Input Multiple Output
OFDM	Orthogonal Frequency Division Multiplexing
PIC	Parallel Interference Cancellation
PSO	Particle Swarm Optimization
QP-SIC	Quadratic Programming- Successive Interference Cancellation
QPSK	Quadrature Phase Shift Keying
RGA	Real-coded GA
RLS	Recursive Least Square
RLS-AE	Recursive Least Square -based Adaptive Equalizer
RSGA-MUD	Redundancy-Saving strategy for the GA-MUD
RSSE	Reduced State Sequence Estimation
SDMA	Spatial Division Multiple Access
SDM-MIMO	Spatial Division Multiplexing MIMO
SIC	Successive Interference Cancellation
SINR	Signal-to-Interference-plus-Noise Ratio
SISO	single-input single-output
SNR	Signal-to-Noise Ratio
SP	Shortest Path

STBC	Space -Time Block Coding
SU-MIMO	Single-User Multiple-Input Multiple-Output
TTCM-GA-MUD	Turbo Trellis Coded Modulation-assisted GA-aided reduced complexity MUD
UCGA	Uni-Cycle Genetic Algorithm
UCGA-AE	Uni-Cycle Genetic Algorithm- based Adaptive Equalizer
UMTS	Universal Mobile Telecommunications Service
USA	United States of America
UWB	Ultra-Wideband
V-BLAST	Vertical-Bell Laboratories Layered Space-Time
VDFGA	Volterra Decision Feedback Genetic Algorithm
WiMAX	Worldwide Interoperability for Microwave Access
ZF	Zero-Forcing

Chapter 1

1.1 Overview of Genetic Algorithm Applications

In the 1960s, Holland was the first to refer to the GA concept [Hol75]. Since then, GAs have developed rapidly due to an increased interest in this area [Gol89, GD91, GB89]. In the framework of optimization and machine learning, Goldberg further developed GAs. He examined the operating principles, structure and implementation of a classifier system, as well as one type of Genetic-Based Machine Learning (GBML) [Gol89]. Davahli *et al.* suggested in [DSA20] the hybridization of GA and Grey-Wolf optimizer (GWO) for the Internet of things (IoT) intrusion detection system to decrease the dimensionality of the massive wireless network traffic. Search-based GAs have demonstrated notable performance in a wide range of global search and optimization problems, including neural networks [Gre93, AH94], adaptive processes [Vos99], besides many different optimization problems that traditional search engines can't solve [Gol89, GC00].

In the context of wireless communications, the first known study of GA application in MUD was presented by Juntti *et al.* in 1997 [JSL97]. Wang *et al*. proposed a detector for multi-user communications based on Maximum-Likelihood (ML) decision and utilizing GA for detection [WLA98]. In [RCA16], a modified real-coded GA (RGA) has been employed to find the optimal configuration of the base stations by optimizing the power consumption. The application of GA in finding the optimal location assignment for mesh routers was presented in [OEB16]. A novel spectrum sharing method in Wi-Fi/WiMAX integrated networks based on GA was proposed in [KNK14]. Cheng and Yang in [CY10] proposed to utilize GA to solve the dynamic shortest path (SP) problem in mobile ad hoc networks (MANET). GAs were also applied to Space-Time Block Coding (STBC) aided MUD scenarios [DC03], beamforming MIMO detection problems [WAC04], and Spatial Division Multiple Access (SDMA) based MIMO OFDM systems [MAG07, HK06], revealing their full potential in wireless communication systems.

1.2 Motivation

Wireless communication is a crucial aspect of the telecommunications industry. Together with its applications and underlying technologies, it is among the most active areas of technology development due to the increment in demand for wireless connectivity and the success of the fourth-generation (4G) digital wireless standard. Developing MUD techniques is one of the most considerable current approaches as the end users' demands have been growing. Complexity is the main challenge associated with multi-user detection and can take two forms: computational and informational complexity. The number of resources required to perform a particular receiver algorithm is referred to as the computational complexity. Informational complexity refers to the amount of knowledge a receiver requires to process the received signals effectively [Che13]. The optimal ML MUD, typically, consists in an exhaustive search, which imposes computational costs increasing exponentially with the number of simultaneous users, thereby making its implementation unfeasible in high-user-load scenarios [Ver98, VPE01].

Several suboptimal nonlinear MUDs have been proposed in the literature, such as the MUDs that utilize the SIC technique [VPE01] or parallel interference cancellation (PIC) [Che13, JH07] that require iterative algorithms to reduce the impact of the interfering signals during each detection stage.

As the wireless market continues to grow, which brings a higher number of users and higher capacity demands, dealing with the wireless channel fading and multipath effects is becoming a more and more challenging task. Channel equalization is essential for reliable communication. A desirable equalizer should be capable of achieving high performance while maintaining low computational costs. The high computational complexity of ML MUD was the reason for which the researchers focused their attention on suboptimal receivers with lower complexity, such as linear and non-linear equalizers; the research objective is to reach the balance between performance and complexity metrics in hardware and theoretical development [MAG07, CW98].

GAs have been applied for optimization of a wide variety of parameters in wireless networks for many reasons. GA is a powerful search engine, capable of solving optimization problems with large search spaces. It can adapt to unknown environments, which is crucial in wireless networks, where the uptime decisions must be made automatically [MQA16].

Different approaches to the MUD and adaptive equalizer based on a computationally efficient GA are the topics of this thesis.

1.3 Thesis and Main Goals

The thesis of the dissertation is as follows:

The use of genetic algorithms can diminish the rate of erroneous MU-MIMO detector's decisions on transmitted symbols and boost the convergence of wireless channel equalizer.

Based on the above-stated thesis, the research focuses on two issues, namely:

- *Improving the ZF-GA MUD performance by mitigating the error propagation effect*. In [KK18, Kha17] the author of the thesis has proposed an original GA-aided solution to the MUD problem, in which one individual of the initial population represents the result outputted by a simple ZF detector. In the dissertation, he revisits that concept. The new strategy bases on the Successive Interference Cancellation (SIC) idea. An employed ZF detector assesses the reliability of signals transmitted by individual stations to give an initial direction to a good search region when the optimization algorithm starts.

- *Boosting the convergence and tracking capability of the adaptive equalizer performance.* Slow convergence and complexity are the major disadvantages of Least Mean Square (LMS) and Recursive Least Square (RLS) algorithms, respectively. A novel UCGA is proposed as a learning tool to optimize the coefficients of the adaptive linear equalizer. In the proposed GA scheme, the individual represents the estimated equalizer coefficients. The algorithm is optimized for working on a real-time basis. To meet such needs, it considers only one population generation per one signalling interval, which has not been practiced, yet.

1.4 Thesis Organization

This thesis is organized in six chapters:

- Chapter 2 briefly presents basic optimization concepts and introduces EAs and their features. Moreover, the basic operators and terminology used in evolutionary algorithms are also discussed. The chapter focuses on the topic of genetic search, and introduces various operators involved with GA. In the last part of the chapter, it is explained how the GA can find the global minimum.

- Chapter 3 provides an overview of MUD techniques. The author demonstrates how GA can be applied in the case of MIMO MUD. Furthermore, some simulation results are presented to display the performance of the reference GA-based solutions, including his own ZF-GA MUD proposal.

- In Chapter 4, SIC fundamentals are presented. In its main part, the chapter brings a re-designed method to generate the initial GA population. BER measurements are investigated to examine the performance of improved SIC-Inspired GA-ZF MU-MIMO detector compared with the ZF benchmark.

- Chapter 5 gives details of a novel approach to an adaptive equalizer, *i.e.*, the so-called UCGA. The performance of the investigated system is evaluated with some experimental results.

- Chapter 6 concludes the work and suggests some interesting future research directions.

Chapter 2

Metaheuristics have become the basis of many studies on optimization over the past few years. Almost all of these metaheuristics are independent of the details of the optimization problem being solved and, therefore, can be applied to address a wide variety of optimization problems. Randomness is used as the basis for a metaheuristic search, but past knowledge is also included to guide the search. One of the most well-known metaheuristics is GA. Their optimization procedure was inspired by the biological phenomenon of evolution [Lev06].

GAs are the earliest set of methods exemplifying the implementation of evolutionary techniques. They operate on a population of possible solutions and simulate the Darwinian theory of survival of the fittest among individuals of successive generations for solving a problem [SD08, Mir19].

This chapter gives a brief introduction to optimization concepts. Also, it shortly discusses the EAs. Afterwards, the chapter moves on to GA. The basic operators involved in GA and the various terminologies are discussed in this chapter. It ends with an example of how the GA can find the global minimum.

2.1 Optimization

Optimization involves defining the alternative solutions of achieving a designated objective and then selecting the one that accomplishes the objective most efficiently, subject to definite constraints; this means the value of an objective function is minimized or maximized by determining the adequate solution [ZTB09].

An optimization problem is a search problem that can be solved by identifying a specific object in a space of alternative solutions, which is generally large. It can be concluded that the problem-solving process can be considered as a search among a combination of options to locate the best choice [ES03, SD08]. Optimization problems in most cases have three main components. The first one is the objective function. The second component is a set of variables whose values can be manipulated to optimize the formalized optimization goal. Constraints are the third component of the optimization problem that limit the values of these variables [CLV07].

2.1.1 Single-Objective Optimization

The author assumes that the considered optimization problems are single-objective problems. Generally, the single-objective problem comprises an objective function $f(\mathbf{x})$ that needs to be optimized (minimized or maximized) subject to a number of constraints, where $\mathbf{x} = (x_1, \dots, x_k)$, $\mathbf{x} \in \Omega$ is denoted as a k-dimensional decision variable vector or solution, Ω contains all feasible solutions. The solution that provides the optimum value of the objective function in comparison to all potential solutions in the search space (the group of all feasible solutions in which the required solution exists) is called global optimum [CLV07]. All the optimization problems covered in this thesis involve finding the global minimum. In general, the global minimum of a single-objective problem can be expressed as:

$$f(\mathbf{x}_{min}) \leq f(\mathbf{x}) \ : \forall \mathbf{x} \in \Omega, \qquad (2.1)$$

where $\mathbf{x}_{min}$ is the global minimum solution and f refers to the objective function. The objective function is called the cost function when the purpose is to minimize a function, and it is called the fitness function in the case of maximizing the function [SD08].

2.1.2 Optimization Algorithms and Evolution

Optimization algorithms are computational processes that compare different possible solutions to find the best one [KW19]. Optimization algorithms can be grouped according to many different ways. In general, they are classified into three categories: (a) enumerative based (b) deterministic (c) random (stochastic) [Vik16, KD21]. Figure 2.1 demonstrates the classification of the optimization algorithms [Yan10].

Enumerative search procedures can be applied to find solutions for various problems. The main advantage of this strategy lies in its computational simplicity because each possible solution is evaluated sequentially within a finite search space. This technique is inefficient with a large search space due to time consuming and, as such, not practical [CLV07]. By contrast, deterministic algorithms attempt to find acceptable solutions in reasonable time by using the knowledge of problem domain. Deterministic algorithms follow a specified action chain. Thus, the objective function, given a particular input, can have the same results and follows the same computation steps [Özk18]. Hill-Climbing, as an example, is a deterministic algorithm, and whenever the program is run, the algorithm follows the same path when the starting point is the same [Yan10]. Gradient-Based algorithms are also deterministic algorithms; they use the functions values and their derivatives. Their effectiveness is reduced when there is some discontinuity in the objective function [CLV07, Yan10].

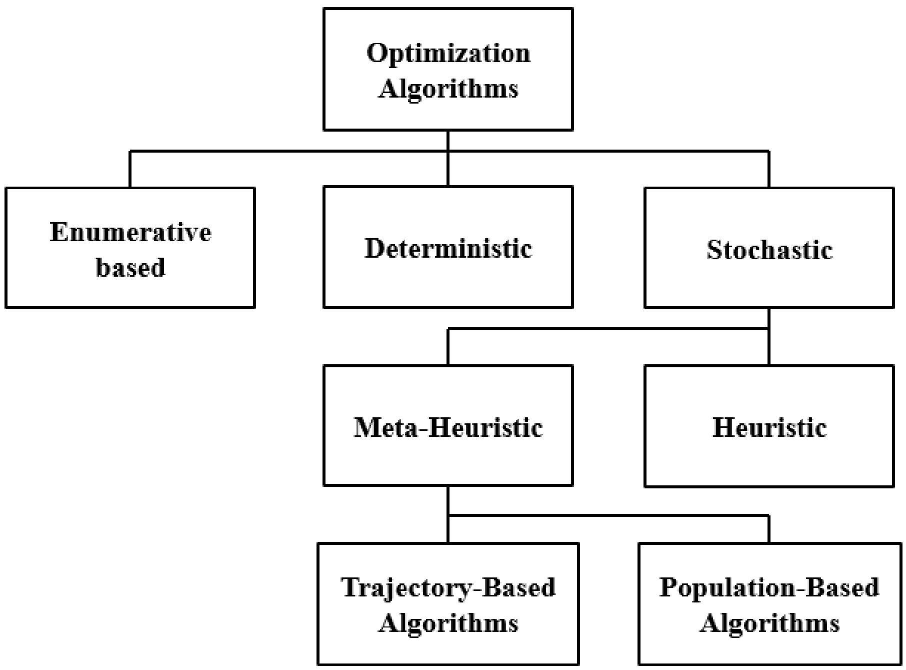


Figure 2. 1 Optimization algorithms classification [Yan10]

Stochastic methods were developed as alternative approaches for solving high-dimensional, discontinuous, multimodal problems [CLV07]. Stochastic algorithms are classified as heuristic or metaheuristic even though the difference is inconsiderable. The heuristic approach seeks a rapid solution with an acceptable level of accuracy; thus, the heuristic algorithms are usually utilized when exact solutions are computationally complicated and the approximate solutions are adequate [Yan10]. In general, metaheuristic algorithms perform better than simple heuristic algorithms. They are usually defined as a top-level strategy of the heuristic algorithms. Figure 2.2 describes the basic principle of metaheuristic algorithms. They create a set of new feasible solutions from the current ones using definite creation operators [Lev06]. The literature contains many applications of metaheuristics for telecommunications problems; e.g., Kim et al. [KCK00] proposed employing a simulated annealing algorithm to allocate nominal channels to the cells of a cellular system. In [ACM03], Amaldi et al. proposed two randomized greedy methods and a tabu search algorithm for locating the Universal Mobile Telecommunications Service (UMTS) base stations to minimize installation costs. Ganame et al. introduced in [GYG19] an implementation of a metaheuristic algorithm based on swarm intelligence to minimize the number of 5G network base stations and optimize their positions at millimeter wave (mmWave) frequencies while meeting user data rates requirement.

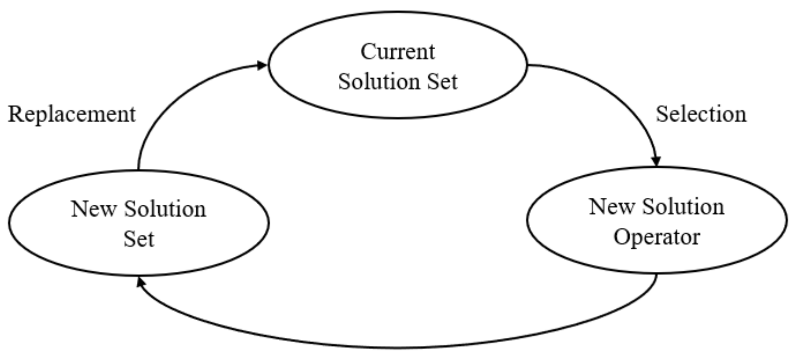


Figure 2. 2 Metaheuristic algorithm basic principle [Lev06]

Metaheuristic algorithms can be grouped into two general groups, *i.e.*, population-based algorithms and trajectory-based algorithms. Population-based algorithm is a generic term for EAs that simulate the natural evolutionary procedures. Selection and randomization are the main features of any metaheuristic algorithm. Selection of the best solutions assures the convergence to the optimality, while randomization increases the diversity of the solutions and, at the same time, prevents the algorithm from being trapped in local optima [Yan10]. Figure 2.3 shows a basic scheme for solving optimization problems using EA.

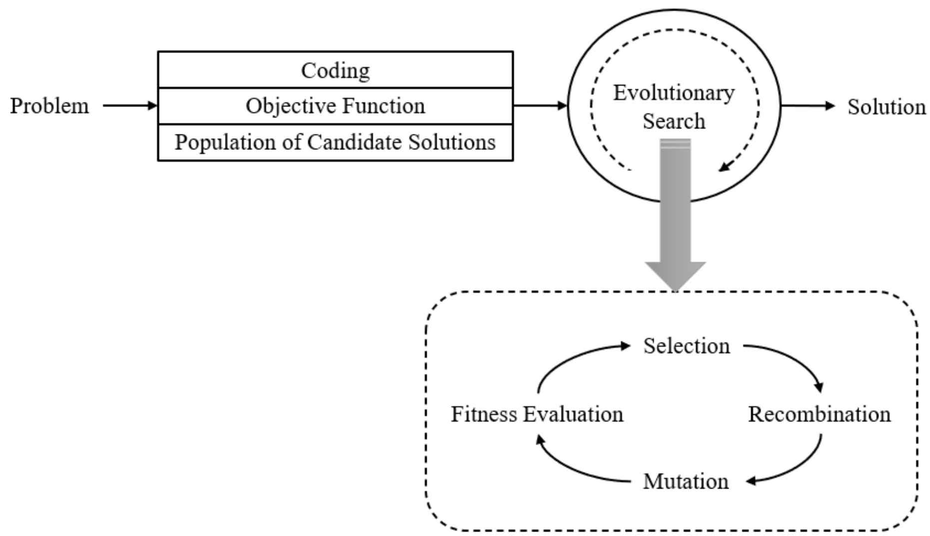


Figure 2. 3 Basic scheme of solving optimization problems using EA [OGS05]

A fitness function assigns a measure of quality to each candidate solution in the population associated with the problem under consideration, this value is the quantitative knowledge the algorithm depends on to guide the search [Hol92]. A new set of approximations is iteratively produced by the process of selecting solutions according to their level of fitness in the problem domain and developing them together using operators acquired from natural genetics (Subsequent sections bring details). This process leads to the evolution of the population in a sense that the current solutions are better suited to their environment than that they were created from, just as in natural adaptation [SD08].

2.2 Evolutionary Algorithms

Darwinian principles were first applied to automated problem-solving in the 1940s, long before computers became widely available [Fog98]. The evolutionary search was introduced by Turing in 1948, and by 1962, Bremermann had actually conducted computer experiments on optimization through evolution [ES03]. Three different applications of the fundamental concept of evolution were created in different locations during the 1960s. Evolutionary programming was first developed in the USA by Fogel, Owens, and Walsh [FOW65, FOW66], but Holland referred to his approach as a genetic algorithm [Jon75, Hol73, Hol92]. In Germany, Rechenberg and Schwefel developed evolutionary strategies [Rec73, Sch95]. These areas had been developed separately for about 15 years till the

1990s when they were found to be different representations of one technology known as Evolutionary Computing (EC) (with the exception for the genetic programming that developed in the early 1990s [BNK98, Koz94]). The algorithms involved in EC are referred to as EAs [Vik16].

There are different EC paradigms based on how the EA components are implemented [Eng07, ES03]:

a) GA that simulates genetic evolution.
b) Genetic Programming (GP). It is similar to GA except for the individuals represented as trees of objects, including programming language commands.
c) Evolutionary Programming (EP). It refers to phenotypic evolution not the genetic model, *i.e.*, the simulation of adaptive behaviour in evolution.
d) Evolution Strategies (ES). It considers both genotypic and phenotypic evolution and is based on the notion of the evolution of evolution. Individuals are represented by their genetic materials and a set of strategy parameters that determine the behaviour of individuals in their environments.

EAs handle a population of individuals (possible solutions). The initial step of an EA is the representation or mapping all possible solutions onto chromosomes. It is similar to creating a link between the problem-solving space, where evolution takes place, and the original problem context. A chromosome, which is essentially an abstract representation, is used to describe each solution [ES03]. An objective function is applied as a fitness measure [ZYC19]. In order to explain the concept of evolutionary cycle, the author refers to Figure 2.4 as a reference. Some of the better candidates are chosen to seed the next generation based on their fitness values, and then recombination (so-called crossover) and mutation are applied to them [CLV07, ES03]. Offspring is created by applying these operators to the selected candidates. After that, these offspring take part in the evolution process and compete with the old candidates for a place in the subsequent generation based on their fitness. This procedure is repeated until a suitable solution, *i.e.*, candidate with desired quality, is found [ES03].

The strategy of EAs shows that to solve optimization problems, EAs simply need the solutions to be represented and evaluated in order to complete the search, despite the fact that problem structural information may be missing. EAs, in particular, can be used without objective function gradient information; all that is required is the fitness of a solution to be evaluated. Hence, EAs can be considered general-purpose optimization algorithms [ZYC19].

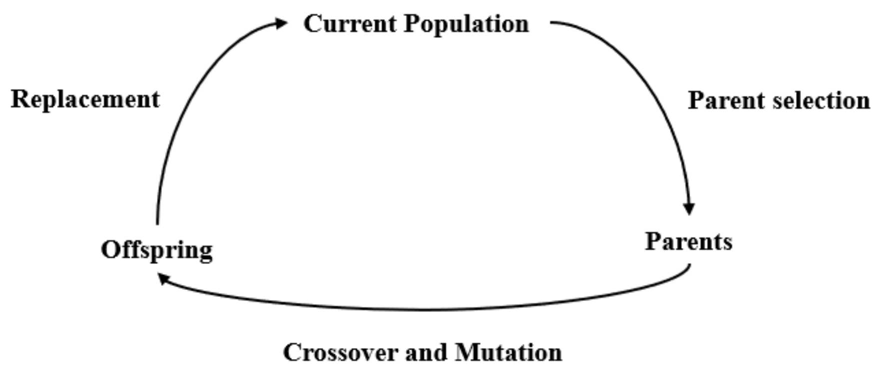


Figure 2. 4 General scheme of evolutionary cycle

2.3 Genetic Algorithms

This section presents the genetic algorithm and its operation as an evolutionary tool. GA is the earliest well-known method exemplifying the implementation of evolutionary techniques. It is an iterative procedure that simulate the natural evolution process by using operators (selection, crossover, and mutation) according to some probabilistic rules [Gen96]. It was first introduced by Fraser, followed by Bremermann and Reed. The substantial work was accomplished by Holland, who generalized GAs [Eng07, Hol75].

GA was found a valuable technique for solving optimization problems. To solve a problem, GA operates on a population of possible solutions and simulate the Darwinian theory of survival of the fittest among individuals of successive generations [Mir19]. It enables a population to evolve to a condition that minimizes the cost function under defined selection methods and can find the best solution from a large candidate set [SD08].

GA, based on the individuals' representation, can be classified into Binary GA and Continuous (so-called real-coded) GA. Both strategies follow the same procedures to minimize the cost and find the optimal solution. Binary GA represents the candidate solutions as an encoded binary string, while the other represents them by floating-point numbers [HH04].

2.3.1 Genetic Algorithm Terminology

In this section, the author explains the fundamental terminology required to comprehend GA.

2.3.1.1 Population and Generation

GA is a population-based procedure. **Population** is a set of candidate solutions, so-called **individuals**, at a particular generation. The individual in the actual real-world solution space refers to the feasible solution and is referred to as a **phenotype**. Meanwhile, in the computing space, where the evolution takes place, the solutions are represented in a form that can be easily processed by a computing system, and the individual is known as **genotype** [SD08, MQA16].

Individuals are encoded as **chromosomes**, *i.e.*, strings of a specific length, such that their values are uniquely re-represented in the phenotypic domain. One or more chromosomes form a genotype [MQA16].

Each chromosome symbolizes a string of a specific length that contains a part of an individual's genetic information. This information refers to the variables of the optimization problem. The chromosome is composed of **genes** which represent the variables to be optimized and take a particular value (**allele**) [ES03]. These elements are pictured in Figure 2.5; in the presented case, the genotype consists of only one chromosome.

The population size remains invariant during the search. It can affect the efficiency of GA significantly. GA can perform poorly when the population size is relatively small. A large population is, in fact, beneficial. (With a large population size, the search space is easier to explore.) However, it needs more processing cost, time, and memory [Gre86, Gol89].

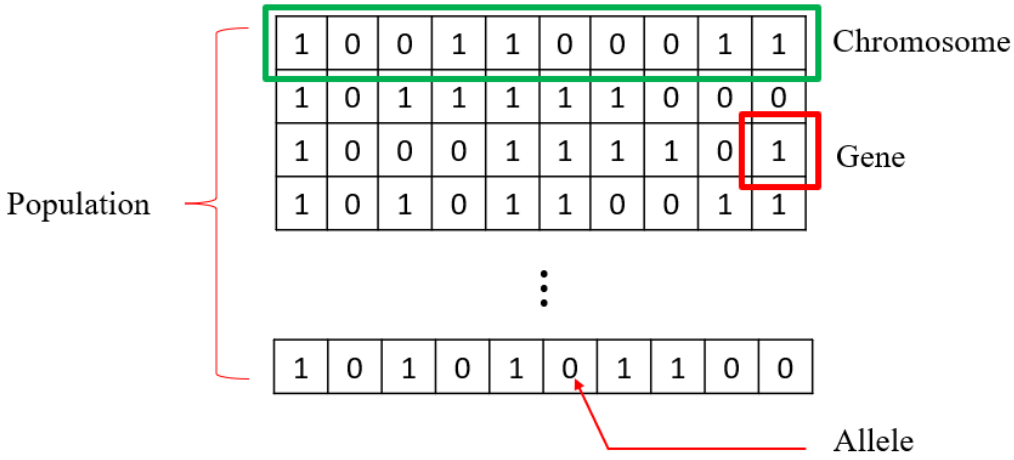


Figure 2. 5 GA basic terminology

At each cycle (iteration), GA applies a set of rules to the existing population to cause its evolution. A generation is a term referring to each iteration of a GA, while the number of generations specifies the number of iterations that GA will perform. The average distance between individuals in a population is referred to as **diversity**. If the average distance is considerable, a population has high diversity as shown in Figure 2.6. Besides increasing the computational capacity of the GA, diversity also allows it to search a broader region of the space [Abr07].

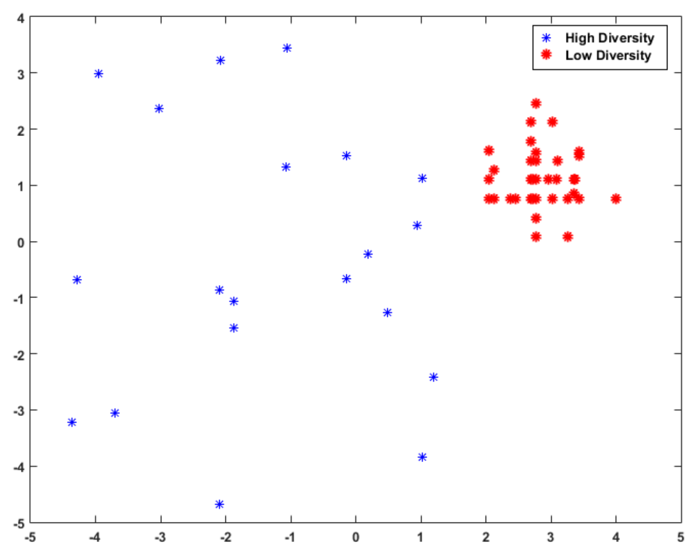


Figure 2. 6 The population on the left has a high level of diversity, whereas the population on the right has a low level of diversity

2.3.1.2 Objective Function and Fitness

It is necessary to have some criteria to evaluate a certain solution in order to know how "good" it is. These criteria are expressed as an objective function, a computable function of the decision variables. Objective function provides a fitness measure that is used in determining the relative performance of the individuals in the problem domain and considered as the basis of the selection operator [Eng07, Sim13]. It is important to understand that the fitness of a possible solution (individual) is defined as the value of an objective function given for its phenotype, *i.e.*, the chromosome has to be decoded first, and then the objective function has to be assessed. The fitness indicates how close to the optimal solution the given chromosome is. When it comes to a minimization problem, the chromosome with the lowest numerical value of the related objective function will be the fitter in the population [KD21, Abr07].

2.3.1.3 Parents and Children

The GA selects fitter individuals from the current population, known as **parents**, and uses them to produce **offspring** (**children**) for the subsequent generation [SD08].

2.3.2 Genetic Algorithm Procedures

The GA process is summarized in the following steps [SD08, ES03]:

1. The method starts by generating a random initial population.
2. Next, the algorithm generates a series of new populations by performing the following steps:
   (a) Each individual of the current population is scored by calculating its fitness value.
   (b) The algorithm selects parents according to their fitness.
   (c) Some individuals in the current population are selected as elite. These individuals are allowed to pass their traits to the new generation population.
   (d) The GA applies crossover and mutation operators to produce children.
   (e) The existing population is replaced by the newly generated offspring to form the next generation population.

3. Once the stopping criterion is met, the algorithm stops.

Figure 2.7 depicts a flowchart of the GA subsequent steps.

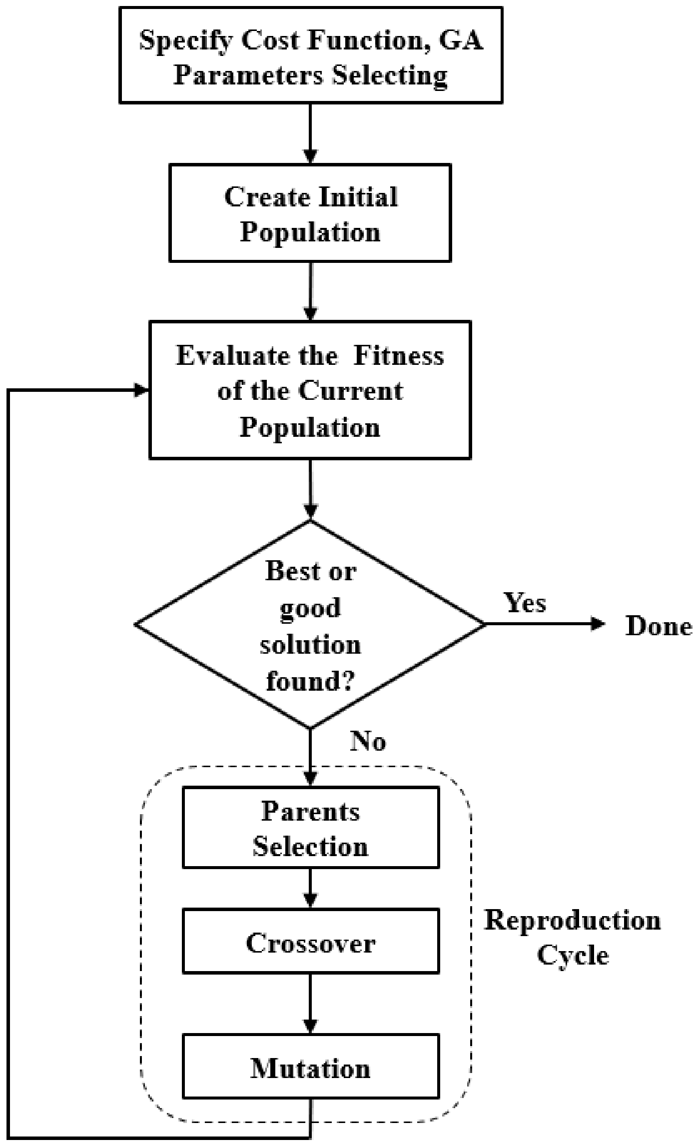


Figure 2. 7 GA flowchart

2.3.3 Encoding (Representation) and Decoding

Encoding (often referred to as Representation) is a procedure of representing the candidate solution [SD08]. It describes the transformation from the solution space (phenotype space) to the computation space (genotype space), while Decoding involves the mapping from genotype to phenotype [ES03]. The phenotype space and the genotype space can be extremely different. It is necessary to comprehend that the entire evolutionary search (selection, recombination, and mutation) takes place within the genotype level, whereas individuals are evaluated at the phenotype level. So, encoding or mapping between the phenotype and the genotype spaces is needed [GD91].

Encoding approach can be completed using bits, real or integer numbers. For example, given an optimization problem with integer solutions, which represent the set of phenotypes, one may choose to define the candidate solutions by their binary code. For illustration, the number 75 is perceived as a phenotype, and 0100 1011 is seen as a genotype expressing it.

In this section, some of the most generally utilized representations are presented.

- *Binary String Representation.* This is one of the most often used representations in GA. Each chromosome is expressed by a binary string (0s and 1s) as in Figure 2.8. It is possible for each bit in the string to represent some characteristics of the solution. Therefore, every bit string is a solution, but not always the optimal one. More bits can be used to increase the precision of a binary representation, but this will slow down the algorithm [JM91].

Chromosome	1	0	1	0	1	1	0	0	1	1	**Binary Representation**

Figure 2. 8 Binary representation

- *Real and Integer Representation.* Real values are usually the best way to represent a solution to a problem when the variables, to be represented as genes, come from a continuous distribution. With this type of representation, every chromosome is a string of a real values as illustrated in Figure 2.9. The precision of these real values on a computer is limited by the implementation, therefore they will be referred to as floating-point numbers, which is a form used in computers to approximate real numbers [SD08, ES03].

Chromosome	1.42	0.8	3.54	7.25	1.63	0.11	**Real values Representation**

Figure 2. 9 Chromosome with real representation

Compared to binary representation, floating-point representation requires less storage since a single floating-point number describes the variable instead of a few integers. GA with floating-point basis is inherently faster than that with binary basis because the chromosomes do not need to be decoded prior to the evaluation of the cost function [HH04].

With integer representation, the chromosome is represented by a string of integer value [SD08] as in Figure 2.10.

Chromosome	1	3	2	9	4	4	7	5	8	6	**Integer values Representation**

Figure 2. 10 Chromosome with integer representation

2.3.4 Initial Population

After deciding the representation, generating an initial population is the first step when using GA to solve an optimization problem. GA initialization has a considerable impact on its performance. It is usually initialized with a random population. This is the simplest and often used technique of initialization [Eng07]. However, there may be cases when the population is initialized with some known solutions using a kind of heuristic to seed the initial population. In such cases, the population’s mean fitness is high, initially, which may boost the algorithm in finding effective solutions faster. Initializing GA with existing solutions can also lead the search to the parts in the search space that contain good solutions [SD08].

2.3.5 Creating the Next Generation

At each iteration, GA creates a new set of approximations by selecting individuals, as parents, according to their level of fitness to create the children for the next generation. Three types of children are created by the GA [SD08, Abr07]:

a) Crossover children are generated by merging genes from two parents' chromosomes in the current iteration and producing a new child.
b) Mutation children are generated by applying some random changes to the genes of a single individual.
c) Elite children. The children described as elite are individuals with the highest fitness values in the current generation. Their genes are passed down generation after generation.

An example for clarification; assuming the population size is 100, the number of elite children is 10, and the crossover fraction is 0.9. The offspring structure in the next generation will be:

a) 10 Elite Children.
b) 81 crossover children (crossover fraction (0.9) * the number of individuals other than elite children (90)).
c) The remaining 9 individuals are mutation children.

2.3.5.1 Selection

Selection is the process of selecting individuals from the current generation to be parents for reproducing. It seeks to identify and maintain fitter individuals within the population. GA is driven by the selection operator, which guides its search to promising regions within the search space. Selection operates at chromosome level and recurs in two places in the evolutionary cycle: parent selection stage and the replacement (survivor selection) stage [LL12, SD08].

A. Parent Selection

Parent selection (so-called mate selection) is the process of identifying individuals based on their quality and selecting the better ones to become parents that mate and recombine to produce offspring for the subsequent generation [SD08]. In GA, this mechanism is probabilistic, which means that the fitter individuals have a better chance to become parents than the individuals with low "quality". Selection operator is fitness-dependent. Generally, there are three categories of selection procedures: fitness-proportionate selection, ordinal-based selection, and threshold-based. Other schemes are grouped under these headings [ES03]. The classification of parent selection methods is shown in Figure 2.11 and described as follows:

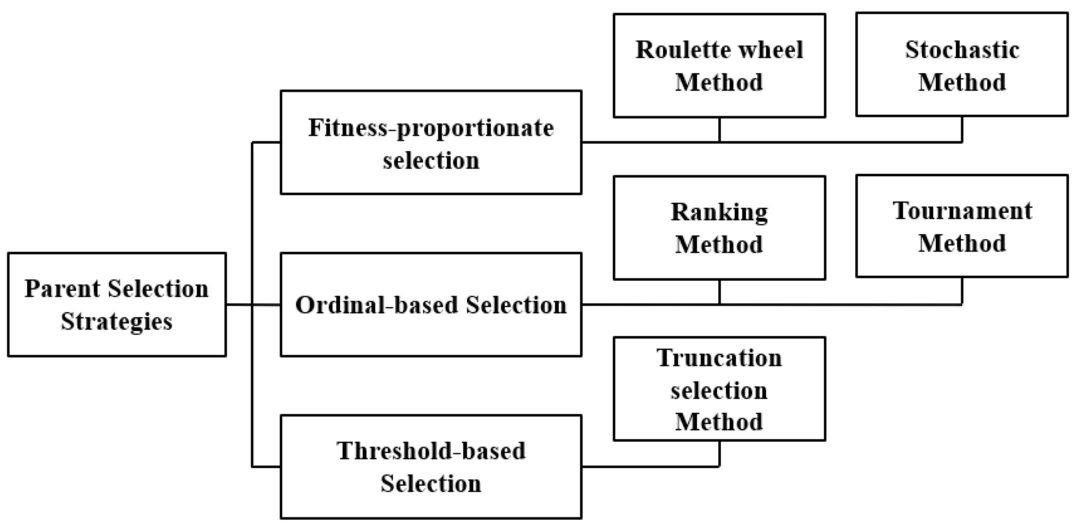


Figure 2. 11 Parent selection methods

- *Fitness-proportionate selection.* It is one of the most often used methods for parent selection. With this method, the individual's probability of becoming a parent is proportionate to its fitness value relative to the fitness of the other individuals in the population. Thus, fitter individuals have a higher chance to be selected for reproduction [GD91]. There are several approaches to implementing fitness-proportionate selection, such as Roulette Wheel Selection and Stochastic Selection.

- *Ordinal-based selection.* It first sorts the individuals in accordance with the objective values and then ranks them. According to its own rank, each individual is then given a selection probability [Whi89, Bak85].
- *Threshold or Truncation-based Selection.* This method sorts the individuals based on their fitness, selects a portion φ of the fittest individuals, and then replicates it $(1/\varphi)$ times. Only those that are above a certain threshold are decided as parents [GD91].

The author aims to explain the principles of Roulette-Wheel method in the subsequent paragraph.

Roulette-Wheel selection is a conventional GA selection approach inspired by real-world roulette wheels. It is a stochastic selection approach in which an individual's selection probability Pr_l is proportional to its fitness. The roulette selection concept is based on a linear search through a roulette wheel with a number of slices measured in accordance with the fitness values of the individuals, The number of slices corresponds to the number of individuals in the population [SD08]. Each individual is given a slice of the roulette wheel, the size of which is proportional to the individual's fitness. This wheel is spun p times, where p is the total number of individuals in the population. Each time the wheel spins, the individual under a fixed point on the wheel circumference is chosen to become a parent for the next generation [GD91]. The selection probability Pr_l of individual l, in a population with p individuals, is expressed as the individual's raw fitness value f_{x_l} relative to other individuals in the population:

$$Pr_l = \frac{f_{x_l}}{\sum_{i=1}^{p} f_{x_i}}\,. \tag{2.2}$$

The probability distribution $P_d = \{Pr_1, \dots, Pr_p\}$ is then simply used to select the parents for recombination [LL12].

Chromosome	Fitness Value
A	8.5
B	4.6
C	3.1
D	2.2
E	4.6
F	0.9

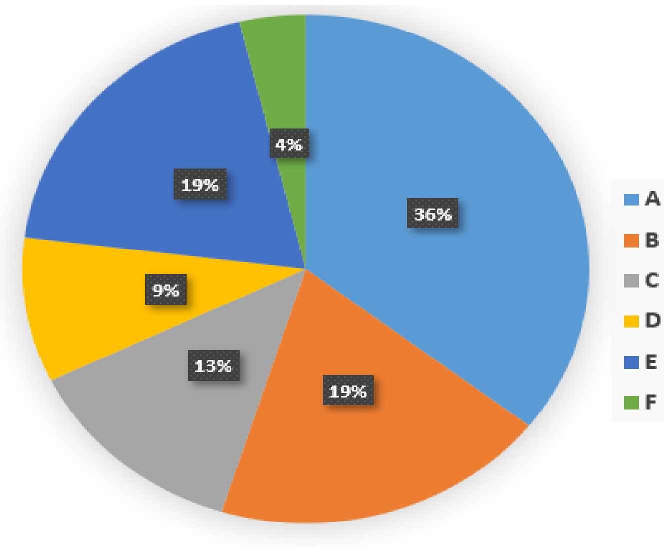


Figure 2. 12 Roulette-Wheel selection

For illustration, refer to Figure 2.12, where a fitter individual (A) occupies more space on the wheel and has a higher probability of being chosen.

B. Replacement (Survivor Selection)

Replacement, comparable to parent selection, is used to characterize the individuals based on their quality but is utilized in a different phase of the evolutionary cycle. This mechanism comes after the creation of the offspring; the author preferred to introduce Replacement in this section to preserve context [SD08, ES03].

The size of the population is almost always constant, so Replacement defines which of the current individuals should be replaced by newly generated offspring to form the next generation population using a specific replacement scheme. Replacement procedures are typically based on the fitness values; higher-quality individuals are preferred, while the concept of age is also frequently applied. Survivor selection is often deterministic, in contrast to the parent selection, which is often stochastic [Bak85].

In general, Replacement can be implemented using two methods, Fitness-Based method and Age-Biased strategy [SD08]. Figure 2.13 demonstrates the classification of the replacement method.

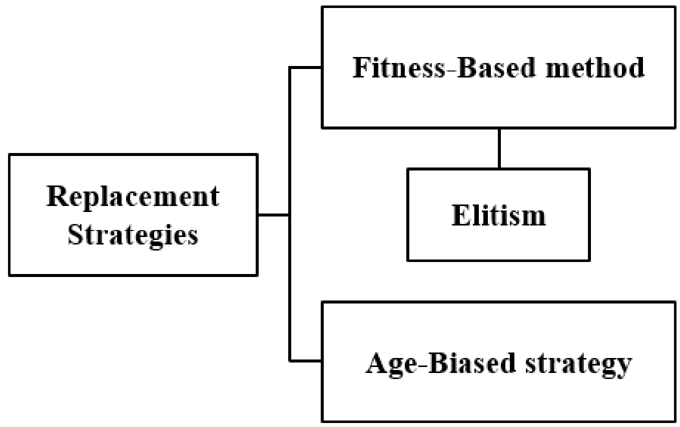


Figure 2. 13 Replacement method classification

The children obtained by means of Fitness-Based selection method replace the population's least fit members. Figure 2.14 shows an example. Individuals with less fitness (individuals 2 and 5) were replaced by the created offspring. It is worth mentioning that when the individuals have the same fitness value (individuals 4 and 5), the choice of which individual should be removed from the population is arbitrary.

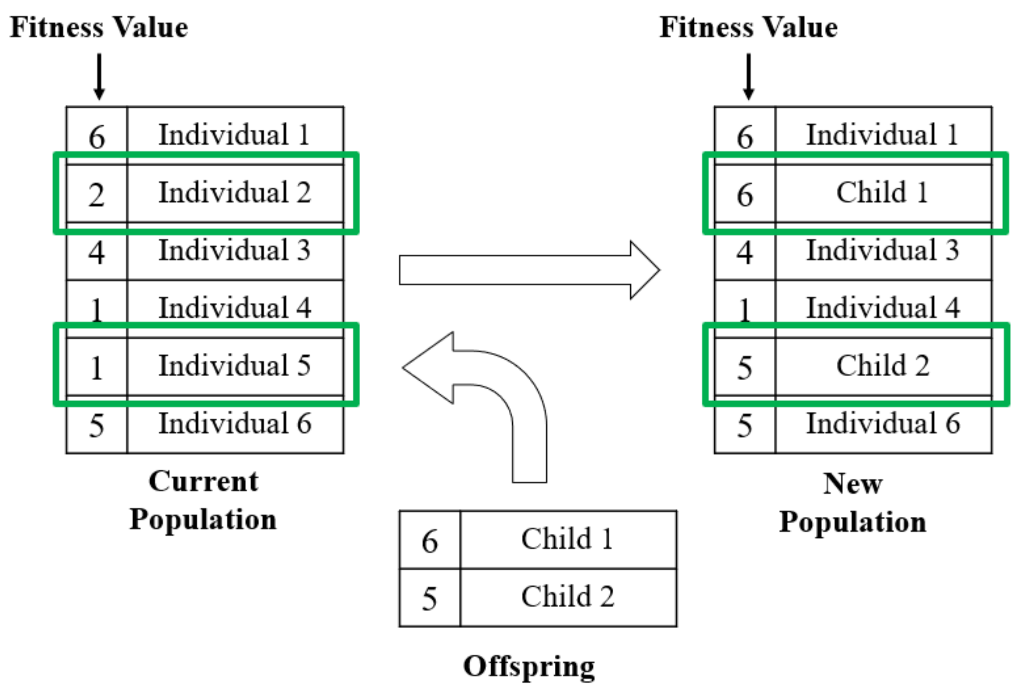


Figure 2. 14 Fitness-Based replacement method

In Age-Biased selection, individuals' fitness is not taken into account. Each individual is allowed to reproduce for finite generations before being kicked out, regardless of how fit they are. For illustration, in Figure 2.15, the number of generations an individual has been in the population is referred to as individual age. Individuals (3 and 5) are the oldest in the population; this is the reason for their replacement [SD08].

In order for evolution to occur, some EAs apply the **Elitism** strategy, which means the current fittest chromosomes are allowed to reproduce their traits to the population in the next generation. This approach improves convergence rate, *i.e.*, the number of generations required by the GA to obtain a population including the fittest individuals [HH04].

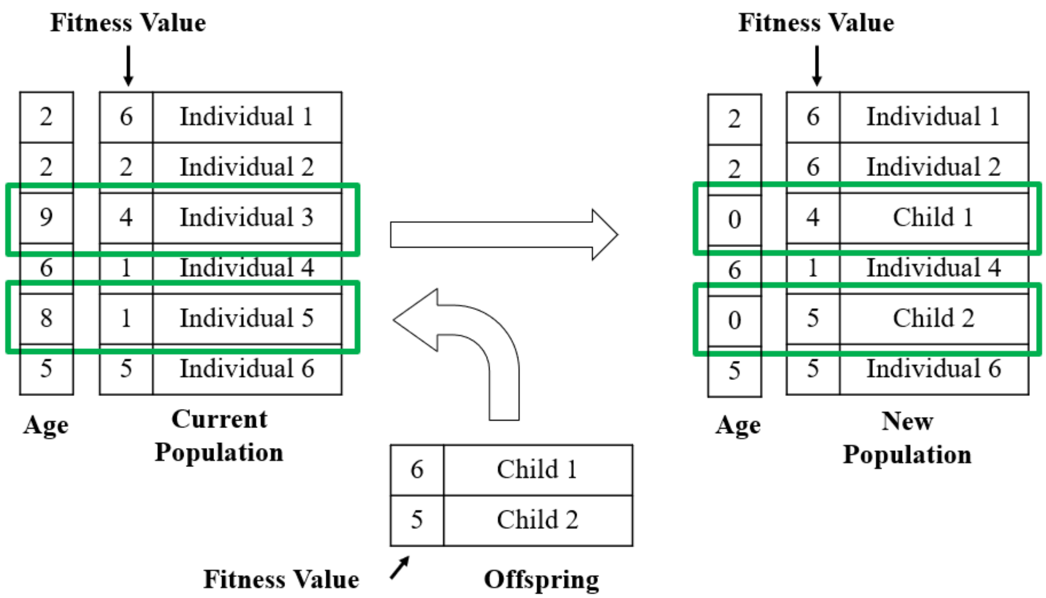


Figure 2. 15 Age-Biased replacement method

2.3.5.2 Crossover

The process of creating a child from two parent solutions is known as crossover or recombination. This operator is equivalent to the biological crossover. It combines two parent genotypes to produce one or two child genotypes depending on a stochastic process. Selection process, described above, does not, actually, create new individuals. Instead, it leads the population to get enhanced with better individuals; therefore, crossover is applied with the hope that it produces more suitable offspring [SD08].

There are different forms of crossover operators depending on the type of representation used, *e.g.*, single-point crossover, two-point and multi-point crossover are generally used for binary representations. These methods start with two parents and produce two children, although they are expanded to include many parents [EKK95]. The simplest method to implement a crossover can be summarized in three steps:

1. Two individuals with desirable characteristics are selected.
2. One or more positions are chosen at random as crossover points over the string length.
3. Alleles after the crossover point are swapped in each individual. The produced offspring merges both of those characteristics.

Figure 2.16 exemplifies a single-point crossover in a binary GA. Assume there are two $k-$bit vector solutions: $\mathbf{x}^{(1)} = \left(x_1^{(1)}, \dots, x_k^{(1)}\right)$ and $\mathbf{x}^{(2)} = \left(x_1^{(2)}, \dots, x_k^{(2)}\right)$. The gene $g_{cr} \in \{1, \dots, k-1\}$ was selected randomly as the crossover point along the length of the mated individuals. The two individuals exchange all of their bits after the g_{cr} position:

$$\begin{aligned} \check{\mathbf{x}}^{(1)} &= \left(x_1^{(1)}, \dots, x_{g_{cr}}^{(1)}, x_{g_{cr}+1}^{(2)}, \dots, x_k^{(2)}\right), \\ \check{\mathbf{x}}^{(2)} &= (x_1^{(2)}, \dots, x_{g_{cr}}^{(2)}, x_{g_{cr}+1}^{(1)}, \dots, x_k^{(1)}). \end{aligned} \tag{2.3}$$

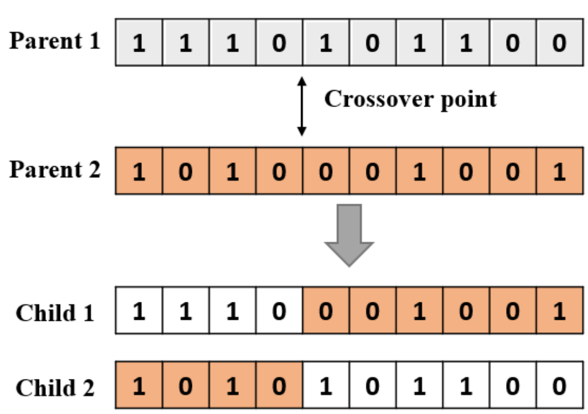


Figure 2. 16 Single-point crossover

Works [Eng07, ES03] provide a comprehensive review of standard forms of crossover used for binary representations.

With the floating-point representation of the chromosome structure, there are three options for crossover [ES03]:

a) Employing the same methods as for bit-strings and split floats, accordingly. That time, the allele is one floating-point value rather than one bit. This type of crossover operator, in the case of floating-point representations, is referred to as discrete crossover. Assume that child $\breve{\mathbf{x}}$ is created from parents $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$. Then the allele value of gene i is given by $\breve{x}_i = x_i^{(1)}$ or $x_i^{(2)}$ with equal probability.
b) Setting each allele in the offspring for a value lying between the parents' alleles on respective gene positions. As a result, crossover can create a new gene material. Operators of this type are known as arithmetic or intermediate crossover.
c) Using operators known as blend crossover. The operator of this type creates, in each gene position of the offspring, a new allele with a value close to that of one of the parents. It may be lower or higher than the reference. The created offspring has a new gene material.

In the subsequent part, the author aims to discuss the fundamentals of Intermediate Crossover method as it has been used during the simulation experiments, reported in the further part of the thesis. The said method generates new offspring by a random weighted average of the parents. The following rule governs the production of offspring

$$\breve{x}_i^{(1)} = x_i^{(1)} + \omega_i\left(x_i^{(2)} - x_i^{(1)}\right) \qquad i \in (1, \dots, k), \tag{2.4}$$

where ω_i is a scaling factor selected at random in the interval, $[B_L, B_U] \subseteq \mathbb{R}$, typically [-0.25, 1.25]. $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ are the parent individuals with k genes. According to (2.6), each gene in the offspring is the consequence of merging the genes of the parents with a scaling factor ω_i, set individually per each pair of parent genes [Abr07].

Crossover Probability

By definition, crossover probability p_x is a parameter that specifies the percentage of the next generation individuals, other than elite individuals, that are produced by crossover. In a 100% crossover probability case, all offspring, other than elite individuals, are created by crossover. In turn, when $p_x = 0\%$, the new population will be formed only by generating copies of chromosomes from the old population, but – thanks to mutation - the new population is not the same as the previous one [SD08, Eng07, ES03]. These extremes of p_x are not an effective improvement strategy. A different setting for p_x can yield the best result [Abr07].

2.3.5.3 Mutation

Mutation is a small random change in the chromosome that results in a different solution, *i.e.*, it uses simply one parent and produces one child by involving some type of randomized change to the genotype. Mutation is used to keep the GA from being trapped in the local minimum and is applied to preserve diversity in the population. Mutation is rare in biology, and it is also rare in most GA implementations. It is applied with a low probability p_m [Gol89]. When the mutation probability is set too high, the GA behaves like a random search, which isn't the best way to find the best solution. However, if the mutation probability is too low, inbreeding and evolutionary dead-ends occur, which prevent the GA from finding a solid solution [Sim13].

It is worth mentioning that mutation operator is representation-dependent. For instance, flipping a bit can be used as a mutation operator when genotypes are bit-strings, as shown in Figure 2.17. One offspring chromosome is created by flipping a bit at a randomly selected position in the parent chromosome.

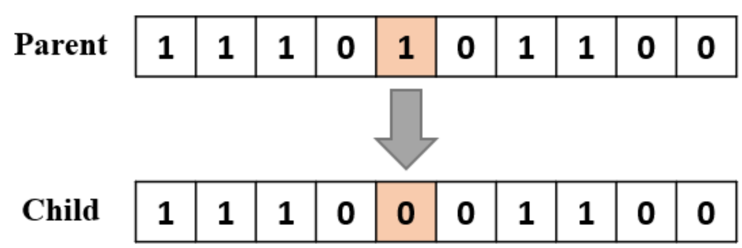


Figure 2. 17 Bitwise mutation for binary representation

When a floating-point representation is used, mutation means random allele value changes of all genes (within a definite range). In terms of the probability distribution from which the new gene values are drawn, there are two types of mutation: uniform and nonuniform (Gaussian) mutation [ES03].

a) Uniform Mutation. With this operator, the allele value is drawn uniformly randomly from a single interval $[B_L, B_U]$.
b) Non-uniform Mutation. This is performed by adding a random number from a Gaussian distribution, with zero mean and user-specified standard deviation, to the current gene value.

Below, the author presents a general explanation of Gaussian mutation as it was used during the experiments.

With Gaussian approach [ES03, Abr07], each element of the parent vector is given a random zero-mean number derived from a Gaussian distribution. At each successive generation, the mutation amount, which is proportional to the distribution's standard deviation, often decreases. Scale and Shrink parameters allow for controlling the average amount of mutation applied to a parent in each generation. The standard deviation of the mutation in the first generation is defined by a Scale multiplied by the initial population range. The Shrink parameter describes the reduction in standard deviation over successive generations. Standard deviation shrinks linearly throughout generations, with its last value equal to (1 - Shrink) times its value at the initial generation. The standard deviation of the random number added to the chromosomes in generation $l > 1$ is

$$\sigma_l = \sigma_{l-1} \cdot \left(1 - shrink \cdot \frac{l}{generations}\right). \tag{2.5}$$

The standard deviation is constant when the Shrink parameter is set to 0. When the Shrink parameter is set to 1, the standard deviation reduces linearly to zero as the last generation approaches.

Mutation Probability

This probability demonstrates how frequently mutations occur on various parts of chromosomes. In the absence of mutations, offspring are produced directly after crossover (or immediately copied) without any alteration. 100% mutation probability changes the entire chromosome, 0% mutation probability does not [Abr07, ES03].

2.3.6 Termination

When it comes to deciding when a GA run terminates, the termination condition is crucial. For this purpose, the following options are frequently used [SD08]:

1. The algorithm stops when the determined number of generations has been reached.
2. The procedures end when the maximally allowed time elapses.
3. The algorithm terminates if the objective function does not improve for several successive generations (**Stall generation**) or during an interval of time equal to Stall time limit.

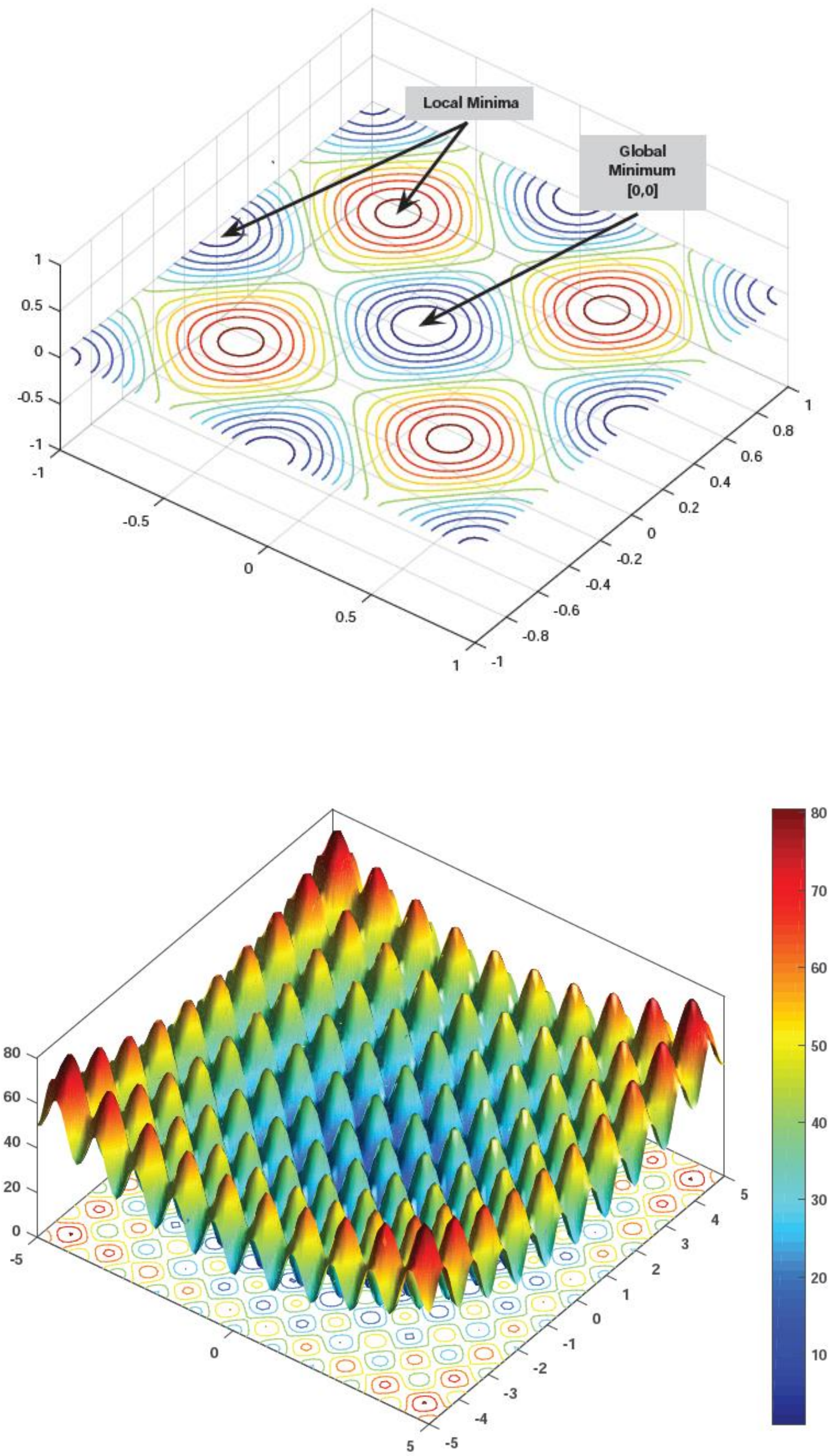
2.4 Example of How Genetic Algorithm Works

Rastrigin function is frequently used to assess the GA performance since its multiple local minima make finding the global minimum difficult for typical gradient-based methods. This section includes an example that describes how to find the minimum of this function using MATLAB GA solver.

On k-dimensional domain, Rastrigin function is expressed as

$$f_{Ras}(\mathbf{z}) = A \cdot k + \sum_{l=1}^{k}[z_l^2 - A\cos 2\pi z_l], \tag{2.6}$$

where k refers to the number of independent variables, $z_l \in [-5.12,\ 5.12], \forall l$ and $A = 10$. This function has many local minima. However, it has one global minimum at $z = 0$ where $f_{Ras}(\mathbf{z}) = 0$. At any local minimum, the value of $f_{Ras}(\mathbf{z}) > 0$ and it increases as the distance from the origin grows. Rastrigin function is illustrated in Figure 2.18 in its two-dimensional form.



(a)

(b)

Figure 2. 18 Rastrigin function: (a) Contour plot shows the alternating maxima and minima, (b) Function value increases as the distance from the origin increases

The author assumes two independent variables, thus Rastrigin function can be written as

$$f_{Ras}(z) = 20 + z_1^2 + z_2^2 - 10(\cos 2\pi z_1 + \cos 2\pi z_2). \tag{2.7}$$

Table 2.1 presents the settings of the GA parameters used for the simulation experiment.

Table 2. 1 GA parameters

Parameter	Value
Population Size	200
Number of generations	50
Selection function	Roulette wheel
Creation function	Uniform
Crossover function	Intermediate
Crossover probability	0.8
Mutation function	Gaussian
Elite count	10
Initial Range	[-5,5]

After running the GA solver, the obtained objective function value is 0, which is the actual minimum; the final point is [1.13e-09, -1.73e-09], very close to the actual optimal solution. The small discrepancies result from data representation rounding. In the following paragraphs, let us follow the GA routine step by step.

Initial Population

As illustrated in Figure 2.19, the algorithm starts by generating a random initial population. There are 200 individuals in the initial population. They are located between -5 and 5, because the initial range was set to [-5, 5].

Next Generations

Figure 2.20 shows the individuals of generation 10, 30, and 50. With an increase in the generation number, the individuals in the population get closer to each other, approaching the global minimum point, simultaneously.

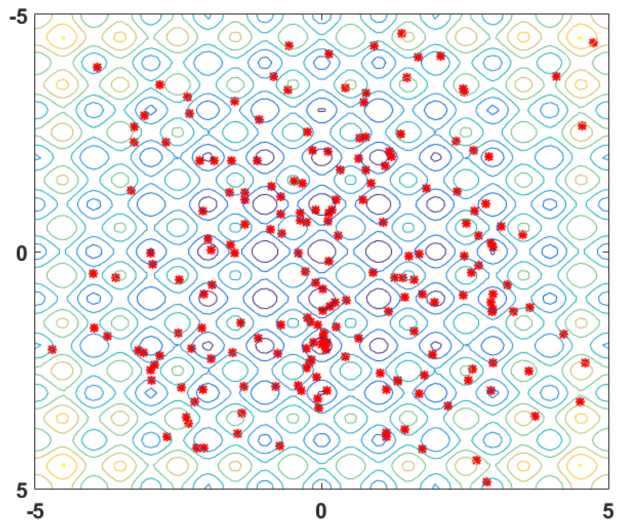


Figure 2. 19 Initial population

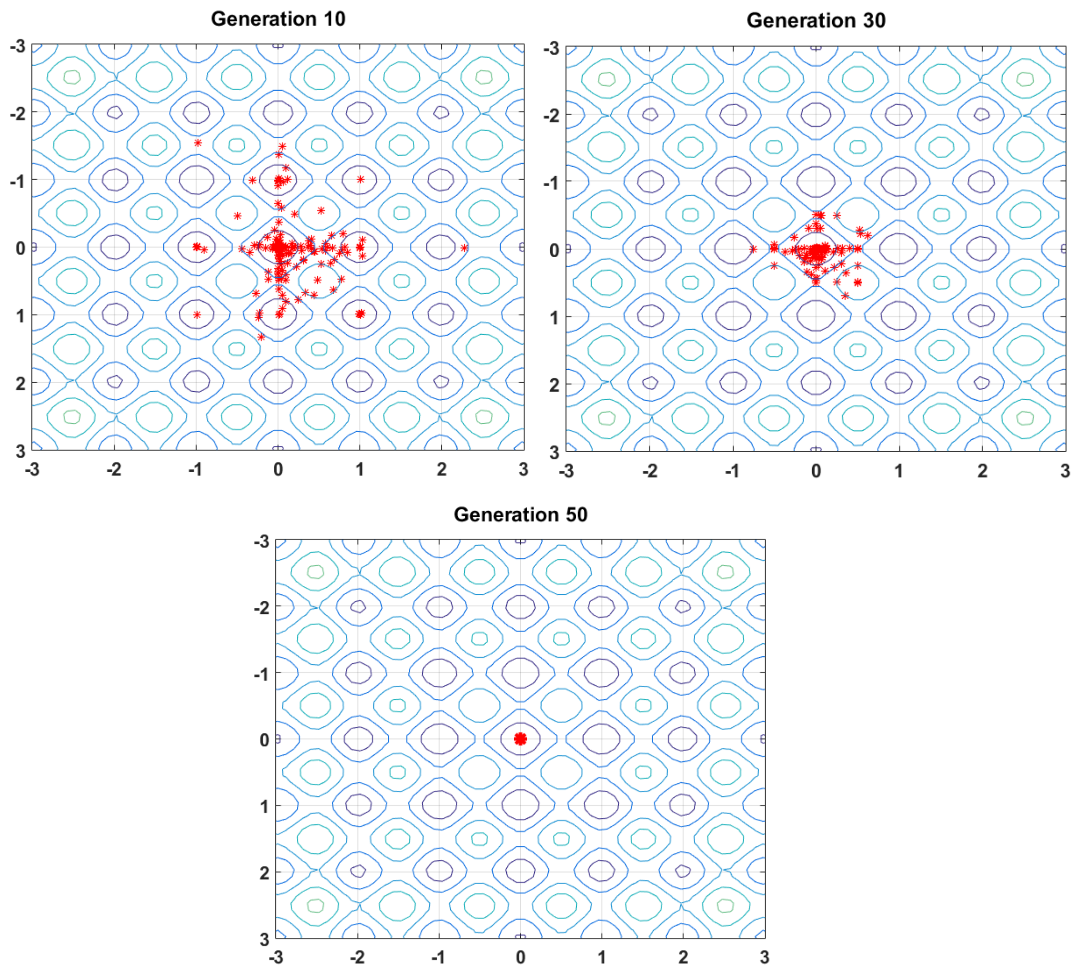


Figure 2. 20 Individuals' convergence at further generations

2.5 Summary

The details of a simple GA have been presented in this chapter. GAs work on populations of possible solutions. To create a new population, selection (reproduction), crossover, and mutation are applied. These operators make the algorithm converge over successive generations towards the global (or near-global) optimum. Using Rastrigin function as an example to assess the GA performance has helped illustrate the algorithm's details and robustness.

In general, there are some key differences between GAs and conventional optimization techniques [SD08]. They can be summarized as follows:

a) At each generation, GA creates a number of solutions rather than a single one, as for almost all the classical algorithms.
b) Evaluation of GA is based on fitness functions instead of derivatives. Thus, they can be used to solve any continuous or discrete optimization problem.
c) In GAs, transition operations are probabilistic, whereas, in traditional continuous optimization methods, transition operations are deterministic.

Chapter 3

Interference mitigation is one of the most crucial aspects of developing multi-user communications systems. It is the case of contemporary wireless networks, such as mobile cellular networks, since obtaining high spectral efficiency necessitates aggressive frequency reuse [TV05]. In this chapter, interference is understood as signals originating from other users (or, more broadly, transmitted data) connected to the same system, *i.e.*, Multiple Access Interference (MAI). One of the receiver design strategies for recovering the desired signal(s) from interference and noise is MUD (also known as joint detection). MUD technique can be used to demodulate digital information sent concurrently by several desired users, sharing a multiple access channel [BLM12]. The essential issue connected with MUD techniques is their complexity despite their superior performance [Che13]. Maximum-Likelihood (ML) MUDs can reach the best performance as a result of exhaustive search, which entails computational complexity increasing exponentially with number of concurrent users increases [JH07]. For illustration, assuming $N_T = 8$ transmit stations, 16-QAM modulation (N_B = 4 bits per modulation period), and one symbol per modulation period, the total number of considered transmitted signals samples' variations would reach $(2^{N_B})^{N_T} \approx 4.3 \cdot 10^9$.

In [KK18, Kha17] the author considered the use of GA for MUD purposes. With the aid of an evolutionary optimization strategy, represented by the GA, the computation complexity has been reduced with a comparable BER performance. This solution will be discussed in this chapter and considered a benchmark for the recently developed approach (described in Chapter 4).

There are different GA-related contributions in the literature, demonstrating the significant interest of many research communities in both the theory and applications of GAs, with regard to MUD methods, a hybrid approach is suggested in [EH00] by Ergun and Hacioglu that combines GAs with multistage MUD in the context of a Code Division Multiple Access (CDMA) system. Ng *et al.* presented in [NYH02] a Turbo Trellis Coded Modulation-assisted GA-aided reduced complexity MUD (TTCM-GA-MUD) that can provide a considerable coding gain without any bandwidth expansion while maintaining low complexity in comparison to the optimal MUD. Du and Chan have invoked GA for sub-optimal detection in STBC aided MUD systems [DC05]. In [UKC15], Island Genetic Algorithm (IGA) has shown attractive BER when applied to Multicarrier Code Division Multiple Access (MC-CDMA) receiver to find out the weight vectors. A Redundancy-Saving strategy for the GA-MUD (RSGA-MUD) was proposed in [THL10]; that method is based on the cost statistics of GA solutions in a synchronous Direct-Sequence Code Division Multiple Access (DS-CDMA) system.

The current chapter provides an overview of MUD techniques, including the use of multiple antennas at both ends (transmitter and receiver), generally known as MIMO. This chapter also discusses how GA is applied in the case of MIMO MUD. Several simulation results are presented to give a better understanding of the system's performance.

3.1 Introduction to MIMO system

There are several challenges facing wireless system designers, such as the limited radio frequency spectrum availability and complex space-time-variable wireless environments. Furthermore, higher data rates, better quality of service, higher network capacity, and higher number of users per spectrum unit per area unit, are in demand. Recent years have seen the emergence of MIMO systems as the most promising technology for these measures [BO13, IEE12, IEE12]. It has received a lot of attention since the pioneering works [FG98] and [Tel99].

MIMO system, as the term implies, uses multiple antennas at both the transmitting and receiving ends. A core concept of MIMO is that it combines signals sampled in a spatial domain at both ends in such a way that it creates multiple parallel spatial data channels (thereby increasing the data rate) and/or adds diversity, *i.e.*, the method for minimizing the impact of fading resulted from multipath propagation [Wes01], to enhance communication quality (BER) [JH07, BCC07].

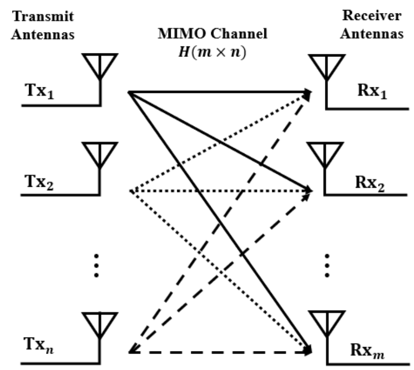


Figure 3. 1 General schematic representation of a MIMO system utilizing n transmit antennas and m receiver antennas [JH07]

The major benefits of MIMO systems over single-input single-output (SISO) systems can be summarized as follows [RC98, PGN04]:

- Both the system capacity and spectral efficiency increase significantly.
- The enhanced diversity dramatically reduces the impacts of fading.

Figure 3.1 shows a schematic of the generic MIMO system. The MIMO channel can be described by the transfer function, which is a matrix

$$H = \begin{bmatrix} h_{11} & \cdots & h_{1j} \\ \vdots & \ddots & \vdots \\ h_{i1} & \cdots & h_{ij} \end{bmatrix},$$

where the number of rows equals to the number of receiving antennas *i*, and the number of columns equals to the number of transmitting antennas j. A single matrix element h_{ij} represents the fading coefficient (random channel gain) affecting the symbol transmitted from the jth transmit antenna to the ith receive antenna.

Work [PGN04] provides a comprehensive review of MIMO techniques, including performance limits, channel models, *etc*. Single-User MIMO (or multi-antenna MIMO) and Multi-User MIMO (MU-MIMO) are two forms of the MIMO system. In the case of Single-User MIMO (SU-MIMO) system, only one device can transmit its data stream and another device can receive it in a given time [LLF11]. In recent years, many SU-MIMO techniques have been developed and ratified in standards, *e.g.*, 802.11n. In general, if not taking into account cross-domain techniques (such as space-time or space-frequency coding), the data stream can be spatially encoded (Vertical Bell Labs layered Space-Time (V-BLAST) [WFG98], index modulation [MWW19], *etc.*) or spatially multiplexed (the data stream is divided into substreams, transmitted concurrently by individual antennas).

In contrast to SU-MIMO, MU-MIMO approach allows simultaneous transmissions between multiple users and a base station, where the spatial dimension is exploited to serve many users in parallel. It is important to note that each form has its advantages and disadvantages, summarized as follows [BCC07]:

- SU-MIMO increases the data rate for a single user, while the use of MU-MIMO increases capacity.
- SU-MIMO ensures a higher throughput in the case of a low SNR. Under a high signal-to-noise ratio, MU-MIMO offers higher throughput.
- The key advantage of SU-MIMO is interference reduction. Multiplexing gain is the key feature of MU-MIMO.

3.2 Multi-User Detection

MUD, in brief, refers to the mode in which a single receiver jointly detects multiple simultaneous transmissions, as depicted in Figure 3.2 [JH07].

The MUD method is often a set of algorithms combined to easily detect the incoming multi-user symbols, *i.e.*, a series of symbols transmitted by individual users, concurrently. One of the most challenging aspects of designing multi-user systems is reducing interference resulting from multi-user data stream simultaneously arriving at the receiver [BCC07].

Note the difference between multi-user detection and interference suppression. The primary distinction is that a multi-user detector attempts to recover multiple transmitted symbols, whereas interference suppression implies that the receiver is only interested in one signal among the received superposition of the transmitted signals [BCC07, Lou91].

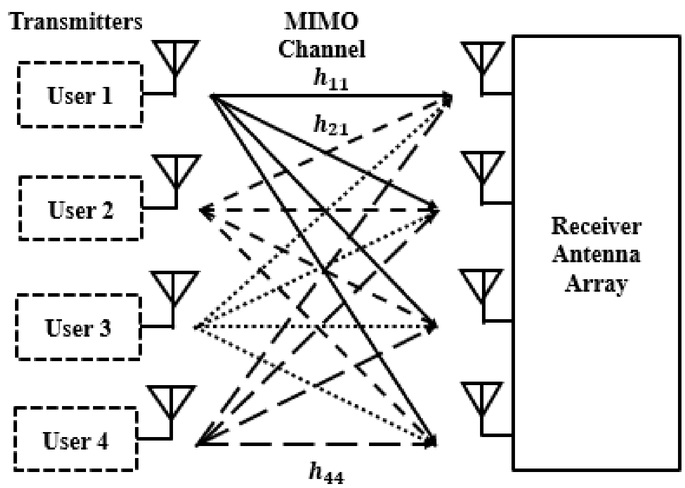


Figure 3. 2 Schematic of general MU-MIMO communication system

This chapter provides an overview of the most popular techniques for estimating the transmitted multi-symbol based on the received multi-symbol. As demonstrated in Figure 3.3, the detection schemes can be divided into Maximum Likelihood (ML), linear, and nonlinear methods [JH07]. The linear MUDs, such as ZF or Minimum Mean Square Error (MMSE), is characterized by a low level of complexity at the cost of limited performance. Different suboptimal nonlinear MUDs have also been proposed in the literature, some of which are based on PIC or SIC methods. They require iterative algorithms in order to eliminate the influence of the interfering users during each detection stage [Che13, JH07].

The ML MUD scheme can achieve the best performance, but it imposes a computational complexity that increases exponentially with the number of users served simultaneously [APA15].

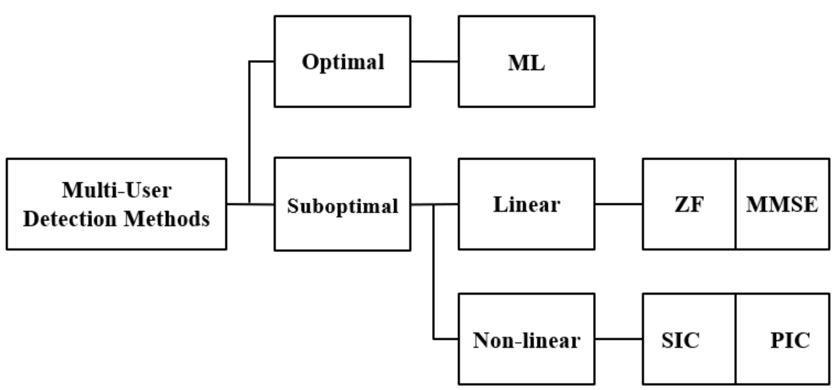


Figure 3. 3 Different type of MUD receivers [JH07]

3.3 Optimal MUD Method

The optimal detector makes the decision on the transmitted multi-symbol in each signalling interval according to the result of the correlation between the observed signal $\mathbf{r}$ and $(\mathbf{H} \cdot \mathbf{s}_i)$, where $\mathbf{s}_i$ is a hypothetical multi-symbol for $i = 1,2 \ldots b$, and b is the number of possible solutions. $b = 2^{N_T N_B}$, where N_T is number of transmitting users and N_B is number of bits per modulated symbol [APA15].

3.3.1 Maximum Likelihood (ML) Detection

ML detection is a high-performance method used as a benchmark for BER performance. From the perspective of minimizing the likelihood of error, this is the optimal detector. It is a search algorithm that compares all potentials solutions and finds the one exhibiting the best cost function value [Tad11, BCC07]. The maximum likelihood detector solves the following problem:

$$\dot{\mathbf{s}}_{ML} = \arg\min_{\check{\mathbf{s}} \in \eta^{N_T}} \|\mathbf{r} - \mathbf{H}\check{\mathbf{s}}\|^2, \tag{3.1}$$

where $\dot{\mathbf{s}}_{ML}$ is the receiver estimate, $\mathbf{H} \in \mathbb{C}^{N_R \times N_T}$ is the channel matrix, $\|\mathbf{r} - \mathbf{H}\check{\mathbf{s}}\|^{\mathbf{2}}$ refers to the ML metric, and η indicates a set of potentially transmitted multi-symbols. It is assumed that $\mathbf{H}$ is known, *i.e.*, has already been estimated by the receiver, regardless of the specific channel matrix structure given by the scenario under consideration.

ML manifests very high accuracy compared to other detection methods presented in this chapter. However, its computational complexity rises exponentially as modulation order and/or the number of transmit antennas increases because of the necessity to test all possibilities. To minimize computational complexity, suboptimal linear receivers comprising channel inverse matrix computations, such as ZF and MMSE detectors, could be employed. [APA15].

3.4 Linear MUD Methods

In linear detection methods, the transmitted signals are treated as interference, except for the required signal sent from the target antenna. Thus, to detect the required signal from the target transmit antenna, signals emitted by other antennas should be cancelled or attenuated, at least. [CKY10].

To achieve the estimates of all the required signals, transmitted simultaneously through several antennas, the impact of the channel is reversed by a weighting matrix **W** such that [SK15]:

$$\dot{\mathbf{s}} = \boldsymbol{W}\boldsymbol{r}, \tag{3.2}$$

where $\dot{\mathbf{s}}$ is the estimate of the transmitted multi-symbol, while **r** is the received signal. The interpretation of (3.2) is that a linear combination of the received multi-symbols determines the detection of each symbol. The complexity of linear detectors is equivalent to that of inverting a matrix of dimensions $(N_R \times N_T)$.

The ZF and MMSE algorithms, described below, are the standard linear detection methods [CKY10].

3.4.1 Zero-Forcing Linear Detection

ZF is considered one of the simplest algorithms available. It is an Inverse Channel Detection (ICD) method, which comprises a multiplication of the received multi-symbol by the channel matrix inverse to obtain the estimate of the transmitted multi-symbol. ZF technique directly uses the pseudo-inverse of the channel matrix in the role of the filtering matrix [SOZ10]

$$\boldsymbol{W}_{ZF} = (\boldsymbol{H}^H\boldsymbol{H})^{-1}\boldsymbol{H}^H, \tag{3.3}$$

where $(\cdot)^H$ represents the Hermitian transposition. ZF solution requires only the channel state information (the channel gain matrix) and has a low computational payload. The disadvantage of the ZF scheme is that it suffers from noise amplification, resulting in suboptimal performance. As a result, the ZF algorithm is excellent for noiseless channels because it successfully reduces all Inter-Symbol Interference (ISI), but it might be useless for noisy channels as it increases the noise experienced by the receiver without attempting to compensate for it [APA15].

3.4.2 Minimum Mean Square Error

By utilizing MMSE criterion, the linear detection can be enhanced, as it minimizes the combined error caused by noise and interference, resulting in the following filtering matrix [SOZ10]:

$$\boldsymbol{W}_{MMSE} = (\boldsymbol{H}^H\boldsymbol{H} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{H}^H, \tag{3.4}$$

where $\boldsymbol{I}$ is the identity matrix and σ^2 is the Additive White Gaussian Noise (AWGN) variance. At a low SNR, the MMSE detector outperforms ZF; however, at high SNR, the MMSE detector approaches the ZF [CKY10].

3.5 MIMO Multi-user Detection Problem Formulation

The considered system is a four-user ($N_{User} = 4$) uplink communication system with one transmit antenna for each user ($N_T = N_{User}$). All the users are supposed to be synchronous and mutually independent. For each user, the uncoded bit-stream is mapped into QPSK, 16-QAM, or 64-QAM symbols before being transmitted. The signals from all users reach the receive antennas ($N_R = 4$) through an uncorrelated MIMO flat-fading Rayleigh channel, represented by the channel matrix $\mathbf{H} = \left[h_{ij}\right]_{N_R N_T}$, as shown in Figure 3.4. The channel matrix's elements h_{ij} are i.i.d. complex Gaussian random variables with zero mean and standard deviation of 1. The author assumed that $\mathbf{H}$ is estimated ideally at the receiver side ($\breve{\mathbf{H}} = \mathbf{H}$).

At a given modulation period, a vector $\mathbf{r}$ represents the symbols received through all the receive antennas and is represented as:

$$\mathbf{r} = \mathbf{Hs} + \mathbf{v}, \tag{3.5}$$

$\mathbf{s} = [s_i]_{N_T,1}: s_i \in \eta,\ \forall_i$, where η is the signal constellation set, is a multi-symbol transmitted by N_T users, simultaneously, while $\mathbf{v} = [v_i]_{N_R,1}$ is a vector of complex AWGN samples. The MUD task is to retrieve a multi-symbol estimate $\dot{\mathbf{s}}$, in which

$$\dot{\mathbf{s}} = \arg min_{\breve{\mathbf{s}}} \left\|\mathbf{r} - \breve{\mathbf{H}}\breve{\mathbf{s}}\right\|^2. \tag{3.6}$$

All feasible candidate multi-symbols $\check{\mathbf{s}} = [\check{s}_i]_{N_T,1}: \check{s}_i \in \eta,\ \forall_i$, are included in the search space. If $\dot{\mathbf{s}} = \mathbf{s}$, the MUD succeeds.

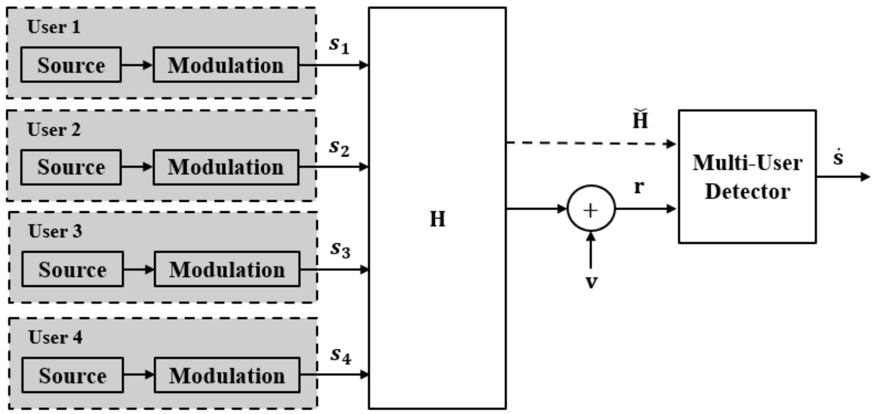


Figure 3. 4 MU MIMO system model [KK18, Kha17]

3.6 Application of GA in MIMO Multi-user Detection

3.6.1 Basic GA-based MUD

Every individual in the GA-based MUD from [KK18, Kha17] represents a potential multi-symbol $\check{\mathbf{s}} = (\check{s}_1 \dots \check{s}_{N_T})$. The employed representation is binary encoding. Let 2^{N_B} be the modulation order. Then, QPSK, 16-QAM, or 64-QAM symbols can be mapped with N_B-bit-long labels $\check{\mathbf{b}}_i \stackrel{\text{def}}{=} \beta(\tilde{s}_i)$ following a specific labelling scheme β (Gray map is used hereinafter). Therefore, in the considered optimization problem, the chromosome could simply be created as a concatenation $\check{\mathbf{b}} = (\check{\mathbf{b}}_1 \dots \check{\mathbf{b}}_{N_T})$ of the labels, assigned to the symbols transmitted simultaneously by all users. It is important to note that the chromosome length is determined both by the number of users and the modulation order. For the 16-QAM instance, as illustrated in Figure 3.5, each label contains 4 bits, and there are $N_T = 4$ users. As a result, the entire length of the chromosome is 16 bits.

It is worth noting that the GA's search space is of discrete type, *i.e.*, it has a discrete number of candidate solutions. It corresponds with the fact that the permissible constellation point positions are strictly limited. As a result, there is no discretization loss when GA is used.

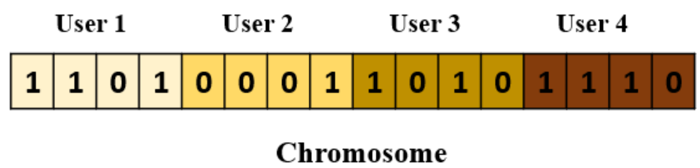


Figure 3. 5 Chromosome representing a candidate multi-symbol for which $\breve{\boldsymbol{b}}_1 = (\mathbf{1101})$, $\breve{\boldsymbol{b}}_2 = (\mathbf{0001})$, $\breve{\boldsymbol{b}}_3 = (\mathbf{1010})$, $\breve{\boldsymbol{b}}_4 = (\mathbf{1110})$ [KK18, Kha17]

The search procedure of the GA algorithm follows the flowchart shown in Figure 2.7. Table 3.1 summarizes the settings of the GA parameters used for the simulation experiment. The subsequent steps of the algorithm are briefly described below.

Table 3. 1 GA Parameters [KK18, Kha17]

Parameter	Symbol	Value
Population Size	p	50
Generation	G	500
Crossover Probability	p_x	0.95
Crossover Function		Single-point
Mutation Function		Uniform
Mutation Probability	p_m	0.005
Selection function		Roulette Wheel
Elite count	ϑ	2

Population Initialization: All initial chromosomes are created randomly. The population size p has a significant impact on the algorithm's ability to find the optimal solution, a high number yields a better solution but slows the convergence [Abr07]. In the considered scenario, the number of individuals in the population (population size) is specified to be 50 (it is a moderate value).

Fitness Evaluation: The optimum ML-based decision metric serves as the objective function for evaluating each individual's fitness and is expressed as:

$$f(\breve{\mathbf{s}}) = \left\|\mathbf{r} - \breve{\mathbf{H}}\breve{\mathbf{s}}\right\|^2. \tag{3.7}$$

The chromosome with the lowest objective function value is the best in the generation. Beginning from the second iteration of the GA algorithm, the individuals with the worst chances of surviving in the current population will die, maintaining the population size through subsequent iterations. This way, the survivors can serve as the parents for the next step.

Selection: The roulette wheel selection rule [SD08, Sim13] is used. According to that approach, each individual has a chance of being selected, but the selection probability is proportional to the individual's fitness. The selection probability of the l^{th} individual is given by eq. (2.2). It is equal to the individual's raw fitness value relative to other individuals in the population. A small number (elite count, $\vartheta = 2$) of the fittest individuals are copied into the next generations. Ultimately, elitism ensures that the quality of individuals gained by the GA does not deteriorate with generations.

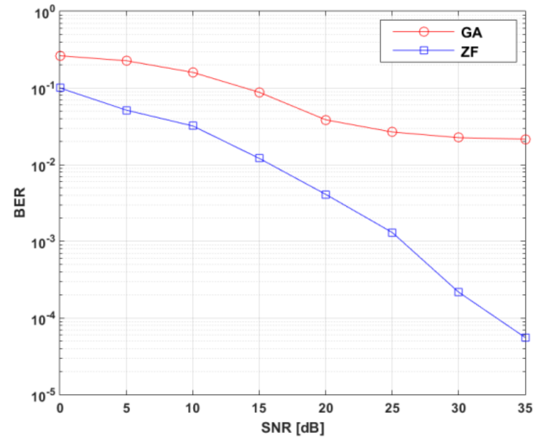
Crossover: Single-point crossover operator is used for binary string representations (refer to Section 2.4.4). It means that the offspring are created by swapping segments of genes between the parents, instead of single genes. Here, the crossover point is selected randomly. The part of the new population that crossover produces is assumed to be 95% of the old population (Crossover Probability $p_x = 95\%$).

Mutation: For binary representations, the uniform (random) mutation is used, where bit positions of the children’s chromosome are selected randomly, and the corresponding bit value is flipped (0 to 1 or vice versa). Not every child undergoes mutation. Instead, it should be a small percent of the newborns [Sim13, Abr07]. In [KK18, Kha17] it was assumed to be 0.5% (Mutation Probability $p_m = 0.005$).

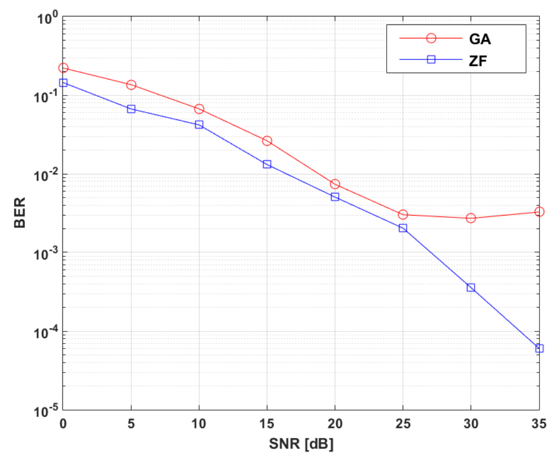
Ending Criteria: In the basic GA-MUD setup, considered in [KK18, Kha17], the stopping criterion was set in terms of the maximum number of generations (iterations) the GA performs – it was assumed to be 500.

3.6.1.1 Simulation results

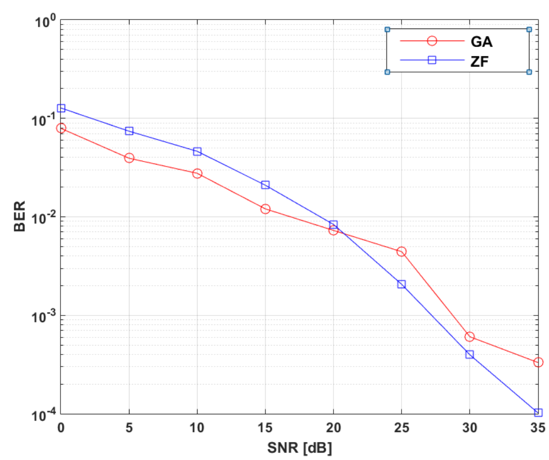
Some simulation results from [KK18, Kha17] are presented in this section to demonstrate the performance of the analyzed reference system. For that purpose, BER vs. SNR measurements are considered. The performance of the GA-based MUD is studied with respect to the modulation order (64-QAM, 16-QAM, and QPSK) and compared to the ZF benchmark; this is illustrated in Figure 3.6. The results show that the 64-QAM system with



a. 64-QAM modulation



b. 16-QAM modulation



c. QPSK modulation

Figure 3. 6 BER vs SNR for the compared MUDs: basic GA and reference ZF [KK18, Kha17]

GA MUD performs poorly, *i.e.*, error floor at *ca.* $2 \cdot 10^{-2}$ BER level appears, and GA MUD is inferior to the ZF reference at high SNR.

Although the error floor for 16-QAM is reasonable (around $2.5 \cdot 10^{-3}$), GA is still considerably outperformed by ZF. Moreover, the GA performance while using QPSK modulation is comparable to that of ZF. The basic GA MUD, in conclusion, is appropriate only for systems transmitting low-order modulations' signals.

3.6.2 ZF-GA MUD

3.6.2.1 Description

For the sake of preventing an inefficient, completely random search, it is helpful to provide the GA-based MUD with an initial estimate of the transmitted multi-symbols. For low-cost design, the output vector of a simple detector may be included in the GA's initial population. The receiver side of such a novel approach, contributed in [KK18, Kha17], utilizes a ZF detector to generate a seed individual (chromosome) for the following GA operations. Thus, some information about the received signal is included in the initial population. As shown in Figure 3.7, the ZF detector determines the multi-symbol estimate $\dot{\mathbf{s}}_{\mathrm{ZF}}$, and the corresponding chromosome $\dot{\mathbf{b}}_{\mathrm{ZF}}$ represents the seed individual of the initial population. Except for the seed, all the individuals in the initial population are created at random, as for the basic setup, detailed in section 3.6.1.

The ZF-based signal estimate is obtained as a linear combination of the signals received by different antennas, altered by the ZF array weight matrix $\mathbf{W}_{\mathrm{ZF}}$ as follows:

$$\dot{\mathbf{s}}_{\mathrm{ZF}} = \arg\min_{\breve{\mathbf{s}}} \|\breve{\mathbf{s}} - \mathbf{W}_{\mathrm{ZF}}\mathbf{r}\|^2, \tag{3.8}$$

where $\mathbf{W}_{\mathrm{ZF}} = \left(\breve{\mathbf{H}}^H \breve{\mathbf{H}}\right)^{-1} \breve{\mathbf{H}}^H$, and $\mathbf{W}_{\mathrm{ZF}}\breve{\mathbf{H}}$ gives the identity matrix.

The cost of introducing ZF into the GA-MUD technique is the time and resources required to acquire the seed chromosome, which is expected to be minor in comparison to the benefits it provides.

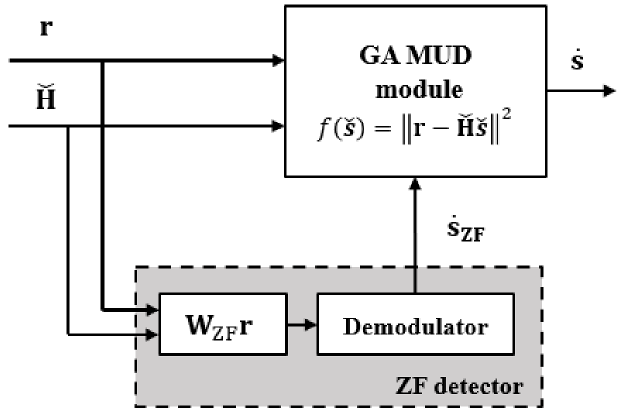
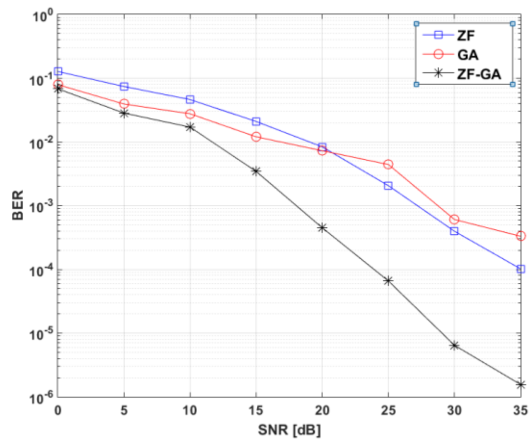


Figure 3. 7 ZF-GA MUD block diagram [KK18, Kha17]

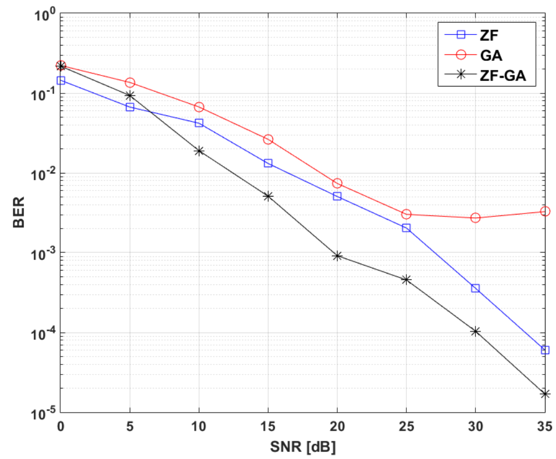
3.6.2.2 Simulation results

This section describes the attainable performance of the concatenated ZF-GA MUD system proposed by the author of the thesis in [KK18, Kha17]. Table 3.1 summarizes the methods and parameters employed in the GA-based MUD module. The simulation results were obtained using (64-QAM, 16-QAM, and QPSK) schemes, assuming that **H** is perfectly known. As can be seen from Figure 3.8, ZF-GA MUD (refer to ZF-GA lines, taking into consideration the results drawn in section 3.6.1.1) promises a decent SNR gain over both the basic GA MUD and regular ZF detector. For QPSK modulation (refer to Figure 3.8 a), the gain is really impressive (*ca.* 11 dB at 10^{-3} BER level). The ZF-GA MUD approach is also robust in the case of 16-QAM – when the GA begins its operation from a better initial population, a 7 dB gain over a regular ZF detector is observed at the BER level of 10^{-3}. For 64-QAM, ZF-GA MUD still exhibits some gain (about 3 dB) over the regular ZF MUD at the BER level of 10^{-3}.

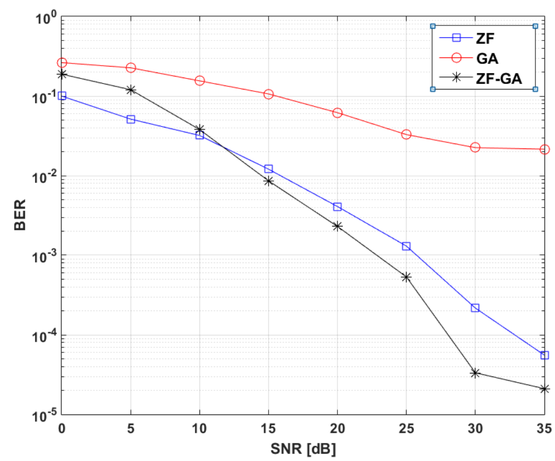
The optimized performance of ZF-GA MUD is achieved at a higher computational complexity than that of the original ZF detector. The complexity of the ZF-GA MUD requires a maximum of ($p \times G$) metric evaluation. The complexity of the plain ZF algorithm is connected with the computation of pseudo-inverse matrix $\mathbf{W}_{\mathbf{ZF}}$. The number of real multiplications for ZF is $\mathcal{O}(N_R^3) + \mathcal{O}(N_R^2 N_T) + \mathcal{O}(N_R N_T^2)$. The number of real additions is also $\mathcal{O}(N_R^3) + \mathcal{O}(N_R^2 N_T) + \mathcal{O}(N_R N_T^2)$, where $\mathcal{O}(\cdot)$ denotes the Big O [LXN13]. For the concatenated ZF-GA routine, the ZF complexity can be ignored since it is used for



a. QPSK modulation



b. 16-QAM modulation



c. 64-QAM modulation

Figure 3. 8 BER vs SNR for the compared MUDs: basic GA, reference ZF, and the proposed GA-ZF [KK18, Kha17]

providing a single initial solution for the GA's initial population and imposes a significantly lower complexity than that of the GA-driven part.

For GA, if the maximum number of generations is used as a termination criterion (as in the considered case), then each generation of the population contains a certain number of individuals; thus, the complexity of GA MUD is proportional to the product of the population size and the number of generations passed, and it reflects the number of times the fitness function calculation is launched.

3.7 Summary

In [KK18, Kha17], the performance of GA-based MIMO MUD was investigated. For a low-order modulation scenario, the GA-based detector performs reasonably good. In the instance of 64-QAM modulation, the GA appears to fail. ZF detection was proposed as an initial processing phase to boost insufficient GA convergence. The proposed combined ZF-GA MIMO MUD managed to outperform the plain ZF reference in terms of SNR, especially for QPSK modulation.

Chapter 4

IC is an attractive solution to the MUD problem [ZH14]. It has become an important aspect that aims to minimize the interference caused by multiple users sharing the same network by demodulating and/or decoding the desired signal and then using this signal, along with channel estimates, to cancel received interference from the received signal [HTD18]. In wireless networks, numerous IC and mitigation strategies have been suggested. This chapter will pay special attention to SIC as a promising technique to improve the efficiency of wireless networks with relatively low additional complexity.

As reported in Chapter 3, the use of GA in the scenario of a 4x4 MIMO system (4 transmit stations, each with 1 antenna and one receiver, equipped with 4 antennas) is inferior to the simple ZF detector for 16-QAM modulation. However, GA offers a significant SNR gain when used together with ZF detector, rather than alone.

In [KK20], the author re-visited the solution to the MUD problem based on the use of the GA. Namely, an alternative population initialization method has been proposed for the combined ZF-GA detector. The concept's strength comes from the principle of SIC. It is described in the forthcoming sections.

The current chapter presents an overview of SIC fundamentals. This chapter also discusses a re-designed method to generate the initial GA population, which improves the performance at no extra computational cost compared to the previous proposal discussed in Chapter 3. Simulation results, published earlier in [KK20], are recalled in this chapter to provide the reader with a comprehensive understanding of the system under investigation.

4.1 Successive Interference Cancellation, The Idea and its Drawback

MAI can be efficiently eliminated by involving IC techniques [PFL00]. In wireless communication networks, IC can be achieved in several approaches. To separate the received signals, the majority of IC methods employ differences in the characteristics of the desired signal and that of the interfering signal. These differences can be exploited in terms of bandwidth, modulation type, amplitude, or power level [BJ09, CS03]. Techniques for cancelling interference are commonly divided into SIC and PIC [KH99].

SIC is a widely used physical layer technology that allows the receiver to receive two or more signals concurrently [SSC10].

The basic principle of SIC is the sequential decoding of information from different users. The decoded users' interference is eliminated from the entire received signal before decoding the signals of other users. It results in a modified received vector with less interference [WQK14].

The general drawbacks of SIC-type detectors are:

1) The problem of error-propagation, in which one erroneous detection case imposes an adverse effect on reliability of subsequent detection tasks [EZ15].

   A solution to this problem consists in pre-ordering the signals that need to be detected according to their reliabilities, such that the more reliable signals are detected before the less reliable ones. Based on the estimated data bits, the impact of the most reliable signal is removed from the mixture of all signals arriving at the receiver antennas. After that, the second most reliable signal undergoes the same procedure, and so on [Yan09]. Wolniansky *et al*. introduced in [WFG98] a technique of ranking symbols known as V-BLAST, which improves the performance of the SIC detector using Signal-to-Interference-plus-Noise Ratio (SINR) values as the measure of reliabilities. V-BLAST has a low complexity yet achieves a lower MAI than their linear counterparts.

   The literature presents a significant amount of research that introduced methods of ranking symbols based on certain reliability measures. Most of these approaches count on ZF or MMSE detectors, *e.g.*, Foschini *et al*. in [Fos96, GFV99] first examined the ZF-based SIC detector designed for Spatial Division Multiplexing (SDM) MIMO systems. In [Yan09], a multistage MS-MMSE detector for MIMO systems is presented with two low-complexity reliability measurement schemes, namely the Type-L and Type-A schemes. The Type-L method uses both the SINR values and the magnitudes of the MMSE detector's decision variables for reliability measurement, whereas Type-A relies only on the magnitudes of the MMSE detector's decision variables to measure the reliabilities. In [EZ15], a low complexity SIC technique, based on a Quadratic Programming detector (QP-SIC), is proposed. This method uses MMSE estimation in both symbol ordering and IC processes. The decision-driven detection algorithm, however, suffers from error propagation and performance degradation.

2) Another (obvious) disadvantage of a SIC is the need to demodulate, decode and re-encode all interfering signals [SSC10].

4.2 Basic SIC Detector

Figure 4.1 demonstrates the principle of SIC in the wireless communication system with *N* users. Assuming that the signal reliability criterion can be assessed from the respective channel gain, all users are ranked by their signal strength, and data detection priority is given to the strongest received signal, then to the next strongest signal and so on [SSC10]. This method entails decoding the signal of the strongest user first, recovering it, and then subtracting it from the whole combined signal to decode the signals of other users, sequentially. It is possible to remove a large part of the total interference once the users' signals are detected. The weakest user will have a significant drop in MAI, which will support its detection. [And05].

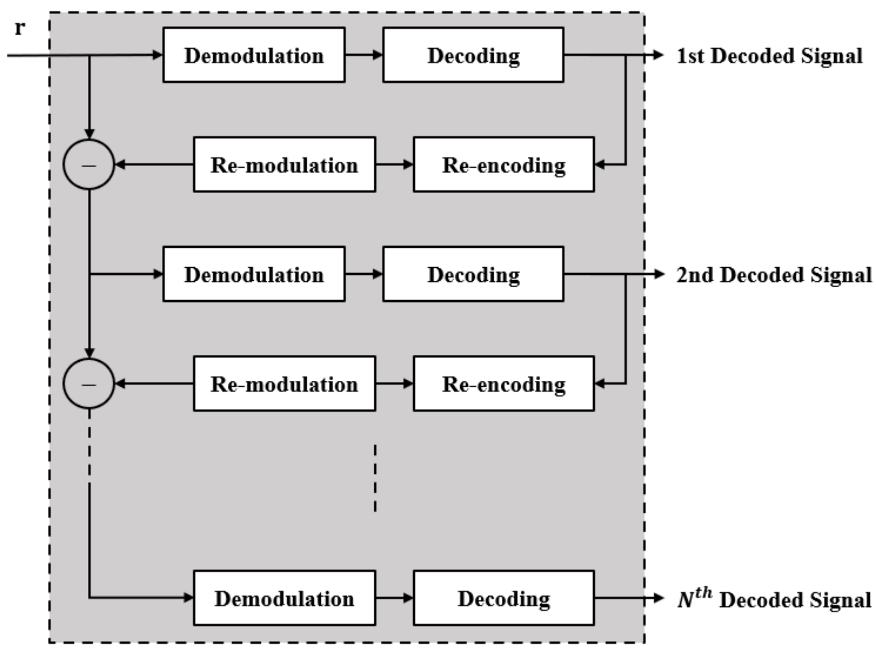


Figure 4. 1 Basic SIC scheme

4.3 The Application of the SIC Concept with GA-MUD

In this section, the author describes a novel SIC-inspired strategy to enhance the GA convergence and the performance of the GA-driven MU-MIMO introduced in Chapter 3.

ZF detector's decision variables are exploited to measure the reliability, thus prioritizing the received signals.

4.3.1 MU-MIMO Model

The author considers the MU-MIMO system, which has been previously described in Section 3.5, as a part of the proposed method. It is an uplink system with N_T independent transmit stations equipped with a single antenna each. The multi-symbol is transmitted through an uncorrelated MIMO Rayleigh flat-fading channel $\mathbf{H} = \left[h_{ij}\right]_{N_R N_T}$, assuming the ideal estimation of **H** at the receiver side.

The signals received through all the receive antennas at a given modulation period can be represented as previously by eq. (3.5). The role of the multi-user detector is to recover a multi-symbol estimate $\dot{\mathbf{s}}$ using the optimum ML-based decision metric, as already mentioned in the previous analysis – refer to eq. (3.6).

4.3.2 Chromosomes Encoding and Fitness Function

The chromosome encoding and the formulation of an efficient fitness function are crucial for the efficiency and convergence of the GA [MQA16]. The employed chromosome representation is a binary encoding.

The author refers to Figure 3.5 as a reference. Considering the case of $N_T = 4$ transmit stations and 16-QAM signalling, the chromosome $\check{\mathbf{b}}$ is described as a concatenation of 4 labels, each having 4 bits. These labels are assigned to candidate data symbols $\check{\mathbf{s}}_i$ that originate from different Tx stations. The labels are assumed to be mapped onto the symbols using the standard Gray labeling map.

The fitness value, given in eq. (3.7), is the distance from a candidate multi-symbol $\check{\mathbf{s}} = \left[\check{s}_1 \ldots \check{s}_{N_T}\right]^T : \check{s}_i \in \eta, \forall_i$, to the received symbol **r,** taking into account the estimated channel state $\breve{\mathbf{H}}$.

4.3.3 GA-MUD Initialization

GA convergence is significantly impacted by its initialization. The initial population of the vast majority of GA applications is created at random. Having many various individuals, the optimization process starts at different locations in the search space, ensuring that it is less likely to become trapped at one location, thus minimizing the risk of getting stuck at a local optimum [OSY11, YHL13]. However, it is clear from the previous study (Chapter 3), that employing random initialization with the GA-MUD problem might prevent the algorithm from achieving any significant convergence, regardless of other GA settings. So, in the previous work, it was proposed to inject one individual that represents the ZF solution into the initial population. This "superior" individual has incomparably better fitness than any other, making it very likely to be chosen as the parent in many crossover operations. As a result, the search concentrates a promising region.

4.3.4 The novel SIC-inspired GA-MUD

Taking into account advantageous results of injecting the “superior” ZF individual onto the initial population, reported in the previous chapter, the author has been looking for another, even more beneficial strategy for GA initialization. In the new contribution, a method based on the SIC paradigm is studied.

According to the new proposal, ZF detector still plays a significant role in indicating an appropriate search region in the initial stage. The ZF-based multi-symbol candidate $\dot{\mathbf{s}}_{ZF} = \left[\dot{s}_{ZF,1} \ldots \dot{s}_{ZF,N_T}\right]^T$, used for GA initialization, is obtained as a linear combination of the signals received by different antennas, transformed by the ZF weighting matrix $\mathbf{W}_{\mathrm{ZF}}$ as follows:

$$\dot{\mathbf{s}}_{\mathrm{ZF}} = \mathbf{W}_{\mathrm{ZF}}\mathbf{r}, \tag{4.1}$$

where $\mathbf{W}_{\mathrm{ZF}} = \left(\breve{\mathbf{H}}^H\breve{\mathbf{H}}\right)^{-1}\breve{\mathbf{H}}^H$, and $\mathbf{W}_{\mathrm{ZF}}\breve{\mathbf{H}}$ gives the identity matrix.

The ZF outcome is processed in a quite different way than in the case of the previous contribution. Namely, reliability measures of the ZF decisions per every transmit station, $(\dot{s}_{ZF,1}, \ldots, \dot{s}_{ZF,N_T})$, are assessed rather than creating a single "superior" individual in the initial population. The assessment criterion for the ith transmit station is the sum of gains of the sub-channels

$$P_i = \sum_{j=1}^{N_R}\left|h_{ji}\right|^2 \quad ; \quad i \in [1, N_T] \tag{4.2}$$

Without the loss of generality, assume that $P_{i'}$ is the maximum. In such case, the transmit station i' is selected and the ZF decision $\dot{s}_{ZF,i'}$ is demapped onto the binary vector $\dot{\mathbf{b}}_{ZF,i'} = \beta^{-1}(\dot{s}_{ZF,i'})$. The last is passed to the GA machine, as shown in Figure 4.2, where it is placed in appropriate (the i' th) section of all chromosomes in the initial population. The remaining bits for all the individuals are generated randomly. An exemplary initial population consisting of p individuals, given $i' = 3$ is shown in Figure 4.3.

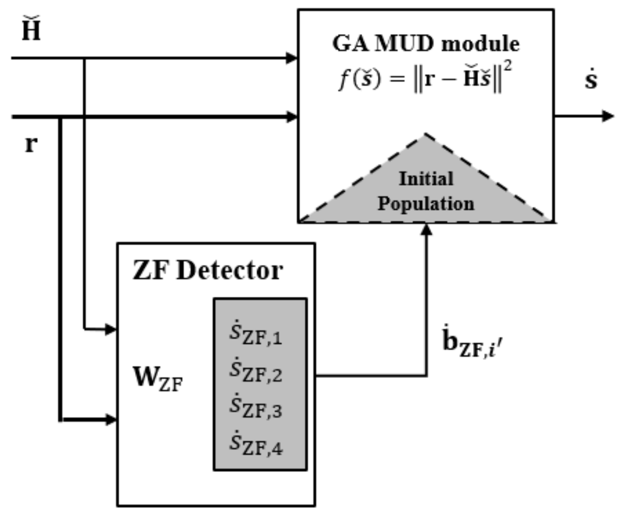


Figure 4. 2 SIC-inspired GA-MUD block diagram

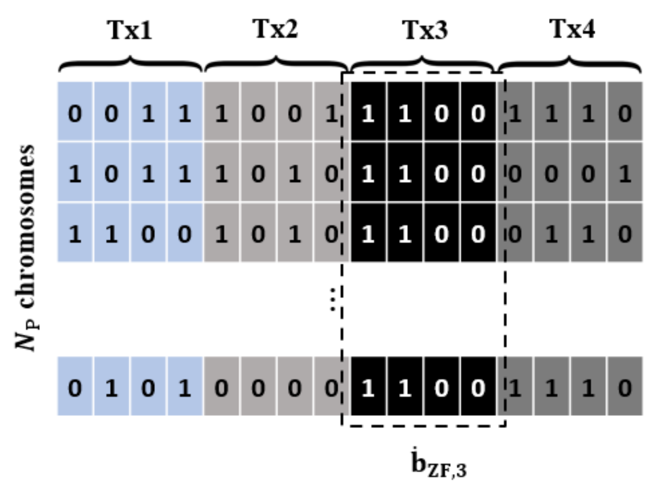


Figure 4. 3 Initial population in the case of 16-QAM modulation, $i' = 3$, and $\dot{b}_{ZF,i'} = (1100)$

Figure 4.4 illustrates the proposed GA cycle. The decision rule of the GA is to find the estimated transmitted symbol vector that minimizes $f(\breve{\mathbf{s}})$ in eq. (3.7).

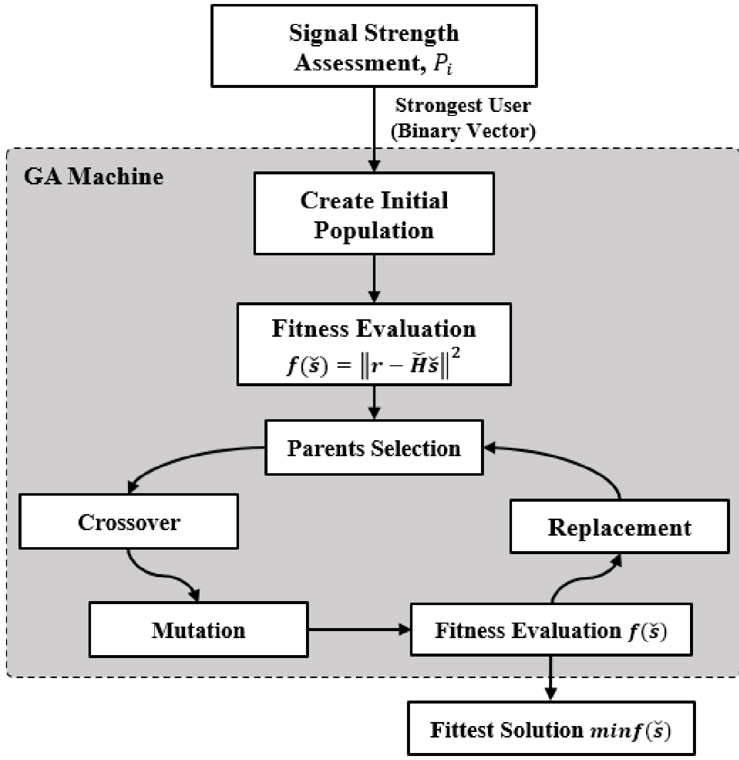


Figure 4. 4 GA machine cycle

4.4 Simulations

4.4.1 Assumptions

To evaluate the performance of the proposed SIC-inspired GA MUD, the simulation experiment is conducted. There are $N_T = 4$ transmit stations, transmitting their 16-QAM signals over the uncorrelated (4×4) MIMO Rayleigh fading channel. It is assumed that the channel state is ideally estimated, so $\breve{\mathbf{H}} = \mathbf{H}$. At the receiver, the ZF detector is used at the first stage, and its decision related to the most reliable transmitted symbol is passed to the GA in the way described in Sec. 4.3.4. GA steps and settings are concisely specified below.

Fitness Evaluation takes into account the fitness measure given in eq. (3.7). The lowest objective function value corresponds with the fittest chromosome in the generation.

Starting from the second generation of the GA algorithm, the worst individuals in the current population are discarded (replacement step) to keep constant population size over the evolutionary processes. The survivors can become the parents of the next generation.

Selection: In the current work, the roulette wheel selection rule, detailed in Sec. 2.3.5.1, is utilized. With this method, all individuals have a chance to be selected with a probability proportional to their fitness [KK18, Kha17, LL12]. The selection probability of the lth individual is given by

$$Pr_l = f_l / \sum_{i=1}^{p} f_i \quad , \tag{4.3}$$

where p is the population size. A small number (elite count, $\vartheta = 2$) of the fittest individuals are guaranteed to be alive among the individuals of the next generation. Elitism ensures that the individual quality gained by the GA will not decrease from one generation to the next.

Crossover: In this work, the single-point crossover method is applied (refer to Section 2.3.5.2). The part of the new population that crossover produces is assumed to be 80% of the old population (Crossover Probability $p_x = 80\%$). To prevent the GA from converging to a local optimum, some random changes are applied to the genes of the individuals by means of the **mutation** step (refer to Section 2.3.5.3). The mutation rate, p_m, is assumed to be 10%. The elite individuals avoid mutation of their genes.

Stopping Criterion: Every GA needs a mechanism to brake the iterative process judging by some features of the current population. In the current work, the GA stops if there has been no improvement in the best fitness value for a specific number of generations (N_S), called stall generations. If the stopping criterion is met, the individual with the best fitness ever is returned as the final solution.

Settings of the above-mentioned parameters applied to the simulation experiment are listed in Table 4.1 for better clarity. They have been carefully chosen after several preliminary runs.

The choice of the population size $p = 2000$ and the number of stall generations $N_S = 20$ is a compromise between the accomplished performance and the computational complexity.

Table 4. 1 GA Parameters

Parameter	Symbol	Value
Population Size	p	2000
Number of stall generation	N_s	20
Crossover Probability	p_x	0.8
Crossover Function		Single-point
Mutation Function		Uniform
Mutation Probability	p_m	0.1
Selection function		Roulette Wheel
Elite count	ϑ	2

4.4.2 Simulation Results

The proposed system is evaluated in terms of BER vs. SNR performance in the case of 16-QAM modulation. It is compared with the following solutions:

- Regular ZF detector, which makes final decisions on all users' data in just one step,
- Simple GA, where all individuals in the initial population are generated randomly,
- ZF-aided GA from the previous chapter, wherein one of the initial population's individuals is the outcome of the ZF detector.

The results are presented in Figure 4.5. From the plot it is clear that the basic GA with the initial population generated randomly (the line with circles) is inferior to the simple ZF (diamond marks) at higher SNRs, and its error floor lies at $ca.$ $3 \cdot 10^{-3}$ BER. The solution proposed in the previous study (represented by the line with stars) brings a significant improvement: the gain of about 7 dB over ZF at the level of 10^{-3} can be observed, but the curve is going to merge or cross the one for ZF near SNR of 35 dB.

The novel approach contributed in the current chapter, represented by the line with squares, offers another 5 dB gain at the level of 10^{-3} BER (in total it gives 12 dB over ZF). What can be also read from the plot, the novel approach is the first to cross the 10^{-4} level.

As a final remark, it must be pointed out that the proposed strategy does not invoke any regular iterative IC routine. In fact, it detects all the users' signals simultaneously, which mitigates possible error propagation effects, observable in the consecutive SIC detection stages. GA could estimate the global optimum for the specific scenario under consideration due to its ability to maintain the best solutions in each generation and use them to improve other solutions. Thanks to that, generation by generation the entire population becomes closer to the optimal solution.

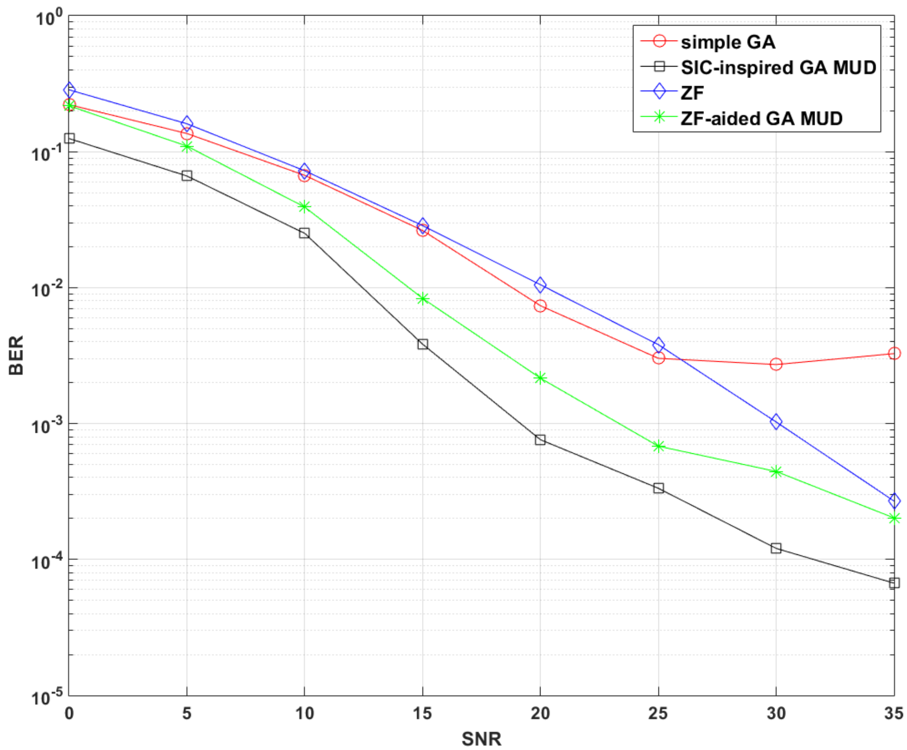


Figure 4. 5 BER vs SNR for the compared multi-user detectors

4.4.3 Computational Complexity

The computational complexity of GA increases linearly with p, G, N_T, and exponentially with the number of bits transmitted per one station per one modulation period N_B (G is the number of populations actually considered in a given algorithm run, it may vary depending on the convergence of the optimization process). To put that in perspective, the computational complexity of ML optimal detector increases exponentially with both the number of users N_T and N_B [Jal04, YHL13].

Obviously, the cost of the GA procedure is increased by the ZF routine from the initialization step. When compared to the solution from the previous chapter, the computational payload of the novel SIC-inspired approach is slightly reduced due to the fact that all individuals in the initial population share the same fraction of genes ($\dot{\mathbf{b}}_{ZF,i'}$). As a consequence, the number of products to be computed for the initial population is $(N_T - 1)/\ N_T$ times the original number.

4.5 Summary

In this chapter, the performance of GA-driven MIMO multi-user detector has been improved. A new method for population initialization has been proposed. It resembles the SIC approach, where the most believable signal is detected and decoded first. On the ground of the GA, the ZF detector is used, and the ZF-based decision related to the most reliable signal is reflected in the chromosome of all individuals in the initial population. Respective part of the chromosome, $\dot{\mathbf{b}}_{ZF,i'}$, is unlikely susceptible to crossover due to the fact that all of the individuals in the initial population share exactly the same genes on the ZF-decided positions. Obviously, it might appear that the ZF decision related to the most reliable signal is wrong. In such cases, the algorithm can still converge thanks to mutation. Nevertheless, in the light of the presented results, it is worth favouring the initial ZF decision to a higher extent than proposed in the previous chapter.

Chapter 5

Adaptation refers to a set of procedures followed by the filter to adjust its parameters (coefficients) according to the input data [Bor13]. Adaptive equalizers, being a form of the adaptive filter, find extensive application in different communication systems. In a communication system transmitting over a wireless channel, the adaptive equalizer is employed to mitigate the detrimental impact of multipath transmission on received signal quality [VR12].

One of the aspects that characterize the adaptive equalizer is adaptation or learning algorithm, which describes how the coefficients are adjusted from one time instant to the next. LMS and RLS exemplify two of the most well-known adaptation algorithms, used to adjust the adaptive equalizer coefficients. The key benefit of LMS over RLS is the computational simplicity at the expense of slow convergence [KNM11]. (Algorithm convergence refers to the number of iterations or adaptation cycles necessary for the algorithm to reach the optimum solution, exhibiting the least Mean Square Error (MSE) [Sim14]).

A couple of years ago, evolutionary algorithms, such as the GA, attracted the interest of researchers as an optimization framework in the field of wireless communication systems [MQA16, JH07]. In the context of the equalization technique, Humaidi *et al*. [HIA19] provided an integrated learning algorithm by hybridizing the GA with the standard LMS; the algorithm can tune both Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters. The GA-based equalizer for a wireless communication system has been introduced in [BZG10], where GA is integrated with a RAKE receiver to overcome ISI caused by the frequency selective nature of Ultra-Wideband (UWB) channels for high data rate transmission. Merabti contributed in [MM14] a new procedure to equalize nonlinear channels using GA. The proposed Volterra Decision Feedback Genetic Algorithm (VDFGA) estimates the so called Volterra kernels to obtain the inverse of the channel response. In [YG10], GA was utilized to improve the weights of neural network, used in a blind equalization task. GA was also utilized to estimate the coefficients of high-order IIR filters by applying a time-frequency fitness function [MP11]. Chen *et al*. [CW98] combined a small population size GA, known as micro-GA with the Viterbi algorithm, to conduct jointly the data estimation and channel estimation. The channel equalizer from [CW98], however, requires periodic injections of training sequences for re-tuning since it does not allow channel tracking.

Although the GA has been known for years, this method should not be dismissed. On the contrary, the author believes it is still worth developing and improving. Because of its search strategy, GA can navigate broadly through search spaces and quickly reach the globally optimal solution to the optimization problem [ES03].

This chapter presents details of a different, computationally efficient approach to an adaptive equalizer so-called UCGA. The novelty of the current contribution is the dependence of a single algorithm time-step (Generation), which reduces the computational cost of the GA. The simulation results that have been presented by the author in [KK21, KK22] are addressed in this chapter to display a reasonable performance gain of UCGA over RLS and LMS for most of the wireless channel models under consideration.

5.1 Equalization Technique

ISI is one of the factors that affect the data transmission performance in wireless communication. This phenomenon occurs because of the physical properties of the transmission medium, which causes the transmitted signal to reach the receiver via multiple paths with different path delays [Wes09, PSB12]. These delayed echoes are the reason for time dispersion observed in the transmitted signal. As a result, the channel responses to subsequent symbols overlap.

It is possible to solve the ISI problem by designing a receiver that employs a means of compensating or reducing the ISI. The general approach is the application of channel equalization. Thus, the term equalization refers to any signal processing operation that mitigates ISI in time-dispersive channels [Hay13, Pro01].

A channel equalizer is a filter that attempts to match the propagation channel response, in which

$$h(t) * h_{eq}(t) = \delta(t), \tag{5.1}$$

where $h(t), h_{eq}(t)$ is the impulse response of the channel and the equalizer, respectively. Figure 5.1 shows the channel and equalizer general scheme.

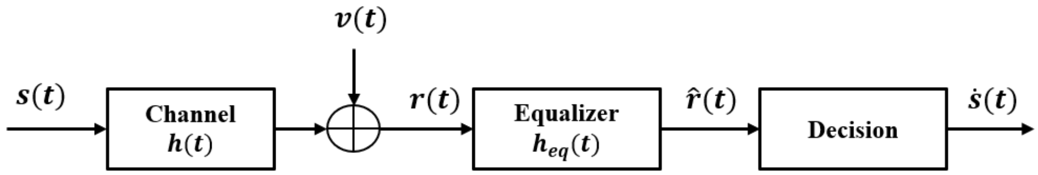


Figure 5. 1 Basic idea of equalization

5.2 Equalizer Structures and Algorithms Classification

The classification of equalization structures is shown in Figure 5.2. It is possible to perform channel equalization using linear or nonlinear approaches. Among linear receivers, equalizers based on FIR transversal filters are of great importance and they are implemented employing symbol-spaced or fractionally spaced taps [Wes09]. The linear equalizer employing a lattice filter has also received a lot of attention in the literature. Although more complicated than the transversal filter, the latter enables fast convergence of the adaptation algorithm. Despite this, FIR equalizers are the most common because of their implementation simplicity [Wes09, Pro01].

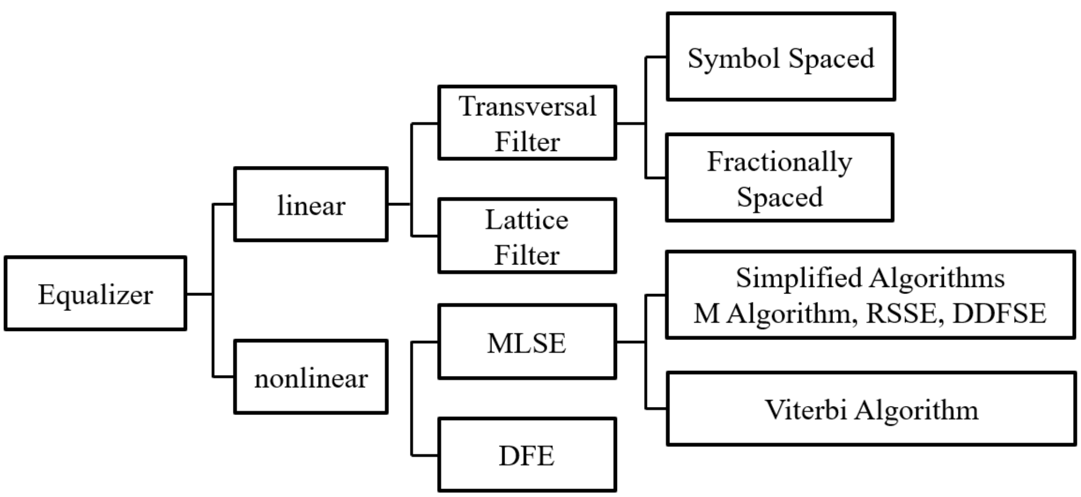


Figure 5. 2 Classification of the equalization structures

Nonlinear receivers are utilized for channels characterized by the occurrence of deep notches. Decision Feedback Equalizer (DFE) is the most basic type of a nonlinear receiver in which the ISI in the currently estimated symbols is suppressed basing on the previous decisions [BP79]. The Maximum Likelihood Sequence Estimation (MLSE) equalizer is optimum, but its computational complexity increases exponentially with the span of ISI. It usually uses the Viterbi algorithm to detect an entire sequence of data symbols [Pro01, For72]. If ISI is caused by a long channel impulse response, the MLSE equalizer becomes impractical due to excessive computational complexity. Thus, several suboptimal structures and approaches, such as the M algorithm [AM84], Delayed Decision Feedback Sequence Estimation (DDFSE) [HH89], and Reduced State Sequence Estimation (RSSE) [EQ88], are suggested as an alternative.

The ability to perform initial adaptation and track channel characteristics in time is the key feature of all equalization structures. In order to attain the initial adaptation, an optimization criterion has to be specified. One criterion, resulting in ZF equalizer, was the minimization of the maximum value of ISI. Minimization of the mean square error, which leads to the MSE equalizer, is the most common adaptation criterion; it assumes minimization of the squared error signal at the equalizer output [Wes09]. The Least Squares (LS) error minimization is the criterion utilized in the fastest adaptation algorithms [Pro01].

In typical equalizers, the adaptation process takes place in two phases. A training data sequence, known to the receiver, is transmitted in the first phase. This sequence is used by the adaptation algorithm to adjust the equalizer coefficients. This mode of operation is called training mode. Having achieved the equalizer parameters exhibiting a sufficiently low error probability, the second adaptation phase starts, in which the equalizer operates in the decision-directed mode [Wes09, Hay13].

5.3 Linear Equalizers

In a channel with ISI, the computational complexity of the MLSE grows exponentially with the length of the channel time dispersion. Assuming the symbol alphabet size is γ and there are I_s interfering symbols, the Viterbi algorithm computes γ^{I_s+1} metrics for each new symbol received. It is prohibitively expensive for most channels of practical interest [Pro01].

In this section, the author describes a suboptimum channel equalization approach to compensate for the ISI. This technique utilizes a linear transversal filter. The computational complexity of this filter structure is a linear function of the channel dispersion length.

In channels with unknown frequency response but time-invariant, the channel characteristics might be measured and then used to adjust the equalizer parameters. Once tuned, the parameters remain unchanged during the data transmission. Such an equalizer is known as a **preset equalizer**. In contrast, **adaptive equalizers** update their parameters periodically during data transmission, allowing them to track a slowly time-varying channel response [PSB12].

A system that employs a linear filter as a channel equalizer is shown in Figure 5.3., The design characteristics of a linear equalizer from a frequency domain viewpoint have been considered by Proakis (refer to [Pro01] for details). From there, one might find out that the desired condition for zero ISI in a non-ideal channel frequency response is:

$$H_T(f)H_c(f)H_R(f)H_{eq}(f) = S_{rc}(f), \tag{5.2}$$

where $S_{rc}(f)$ refers to the desired raised-cosine spectral characteristic. By design $H_T(f)H_R(f) = S_{rc}(f)$, so the equalizer's frequency response, ideally compensating for channel distortion, is:

$$H_{eq}(f) = \frac{1}{|H_c(f)|} e^{-j\theta_c(f)}, \tag{5.3}$$

where $|H_c(f)|$ and $\theta_c(f)$ refer to amplitude and phase components of the channel frequency response, while $1/|H_c(f)|$ and $\theta_{eq} = -\theta_c(f)$ are the equalizer amplitude and phase response. It can be said that the equalizer acts as an inverse channel filter, totally eliminating ISI. This equalizer is called a **zero-forcing equalizer** since it forces the ISI to be zero at sampling instant $t = nT$ for $(n = 0,1, \dots)$.

ZF equalizer has the drawback, which consists in ignoring additive noise. Thus, its use may lead to a significant increase in noise. This is obvious by noting that the equalizer compensates for the weak amplitude frequency channel response by setting a large gain in that frequency range. This significantly increases the noise in that frequency range.

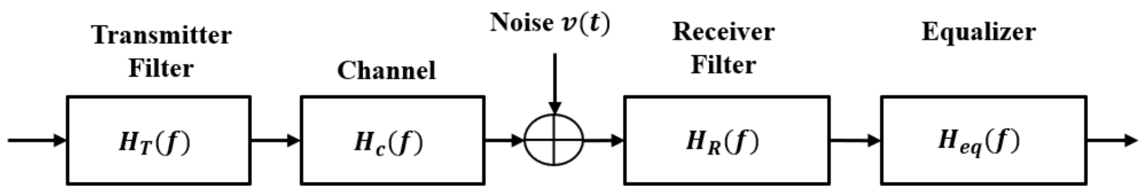


Figure 5. 3 An equalization system block diagram

On the basis of the literature studies, it can be concluded that the linear equalizer is ineffective for compensating inter-symbol interference on channels with spectral nulls, which may occur in radio transmission. The channel spectral nulls often cause a significant noise amplification at the equalizer output. The interested reader is referred to [Pro01, AB93, DMM74] for details.

5.3.1 Adaptive Linear Equalizer

Throughout the development of the equalization methods, it is implicitly assumed that the receiver is aware of the channel characteristics. However, in most communication systems, the channel characteristics are very often unknown or time-variant. Thus, it is worthwhile making the equalizer adaptive so that its parameters are periodically updated to compensate for the distorting channel characteristics; thus, it can track the changes of channel characteristics in time [Wes09, AO15]. LMS and RLS represent two basic adaptation algorithms, which can be used for optimizing the coefficients of an adaptive equalizer [Hay14, KNM11].

Linear filters with adjustable coefficients, illustrated in Figure 5.4, are the most common channel equalizers used in practice [Wes09]. Its input is the sequence **r** that can be expressed as

$$r(n) = \sum_{l} \chi(l) \cdot s(n-l) + v(n), \tag{5.4}$$

where $\{\chi\}$ is a set of tap coefficients of an equivalent discrete-time transversal filter that models the channel, **s** refers to the transmitted symbols, and $\boldsymbol{v}$ is a sequence of white Gaussian noise samples.

The equalizer output $\hat{\mathbf{r}}$ at the *n*th signalling interval depends of the input signal samples **r** and the 2L+1 coefficients of the equalizer $\mathbf{c}(n)$ as per equation

$$\hat{r}(n) = \sum_{l=-L}^{L} c_l(n) \cdot r(n-l). \quad (5.5)$$

Several studies have been conducted on the criteria for optimizing the filter coefficients [Hay13]. In digital communication systems, the average probability of error is the most meaningful measure of performance. Thus, choosing the coefficients that reduce this performance measure is desirable. The probability of error is a nonlinear function of $\mathbf{c}(n)$. This makes the probability of error an inefficient performance indicator for optimizing the tap weight coefficients of an equalizer [Bor13, Pro01].

It has been widely found that two criteria can be used to optimize the coefficients of an equalizer. One is the mean-square-error criterion, while the other is the peak distortion criterion [Hay14].

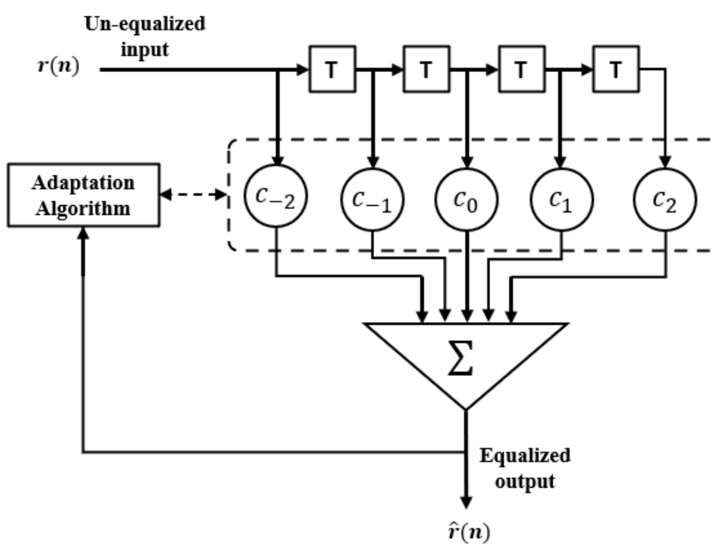


Figure 5. 4 Basic structure of linear adaptive equalizer

Basically, **peak distortion** refers to worst-case inter-symbol interference at the equalizer's output. The peak distortion criterion involves minimization of the performance index. **MSE** criterion involves adjusting the tap weight coefficients of the equalizer to minimize the mean square error [Luc65, Pro01, Wes09]

$$\varepsilon(n) = s(n) - \hat{r}(n). \tag{5.6}$$

For complex-valued information symbols, the MSE performance index, I_{MSE}, is defined as

$$I_{MSE}(n) = E|\varepsilon(n)|^2 = E|s(n) - \hat{r}(n)|^2. \tag{5.7}$$

The solution of I_{MSE} with respect to the equalizer coefficient $c(n) = [c_{-L}(n), \cdots, c_L(n)]^T$ leads to a set of linear equations that may be expressed in the general matrix form (see [Wes09] for details):

$$\boldsymbol{\Lambda}\mathbf{c}_{opt} = \psi. \tag{5.8}$$

It results in the Wiener-Hopf equation that gives the optimum equalizer coefficient, where $\boldsymbol{\Lambda}$ denotes the $(2L+1) \times (2L+1)$ Hermitian matrix, **c** is the column vector of $2L+1$ equalizer coefficients, and ψ is a column vector of channel filter coefficients in $2L+1$ dimensions. Thus, the solution for the optimum coefficients involves inverting the matrix $\boldsymbol{\Lambda}$:

$$\mathbf{c}_{opt} = \boldsymbol{\Lambda}^{-1}\psi. \tag{5.9}$$

5.3.1.1 Least Mean Square Algorithm

For practical equalizer implementations, an efficient method of achieving the optimum coefficients and the minimum MSE is usually obtained through an iterative procedure, so that the inverse of matrix $\boldsymbol{\Lambda}$ in eq. (5.9) is not explicitly calculated [Wes09, PSB12]. The steepest descent method is the simplest iterative procedure, in which one starts by randomly selecting the coefficient vector, *e.g.*, $\mathbf{c}_0$. As such, the coefficient vector $\mathbf{c}_0$ corresponds to a point on the criterion function that is being optimized. With the MSE criterion, the initial guess $\mathbf{c}_0$ corresponds to a point on the quadratic MSE surface in the space of coefficients. The gradient vector g_r is computed at this point. Using the computed gradient vector, each tap weight is adjusted in the direction opposite to its gradient component. The changes in tap weight are proportional to gradient component size. Accordingly, successive values of the coefficient vector **c** can be obtained by applying the relation [PSB12]

$$\mathbf{c}(n+1) = \mathbf{c}(n) - \alpha\, \boldsymbol{g}_{\boldsymbol{r}}(n) \qquad n = 1,2,\cdots, \tag{5.10}$$

where α is a positive number representing the step-size parameter for the iterative procedure, small enough to assure convergence of the iterative procedure. It is difficult to determine the optimum tap weights using the steepest descent method due to the lack of knowledge about the gradient vector g_r, which is dependent on both $\boldsymbol{\Lambda}$ and ψ [Pro01]. It may be possible to overcome this difficulty by using estimates of the gradient vector. Thus, the algorithm for adjusting tap weight coefficients can be expressed as:

$$\hat{\mathbf{c}}(n+1) = \hat{\mathbf{c}}(n) - \alpha\, \hat{\boldsymbol{g}}_{\boldsymbol{r}}(n), \tag{5.11}$$

in which $\hat{\boldsymbol{g}}_{\boldsymbol{r}}$ represents an estimate of gradient vector $\boldsymbol{g}_{\boldsymbol{r}}$ and the estimate of the coefficients vector is denoted by $\hat{\mathbf{c}}$. A gradient vector estimates $\hat{\boldsymbol{g}}_{\boldsymbol{r}}$ in the *n*th iteration is calculated as follows (see Proakis *et al*. [PSB12] for details)

$$\hat{\boldsymbol{g}}_{\boldsymbol{r}}(n) = -\varepsilon(n) \cdot r^*(n). \tag{5.12}$$

The adaptive algorithm for optimizing the tap coefficients (based on the MSE criterion) can be obtained by substituting (5.12) into (5.11)

$$\hat{\mathbf{c}}(n+1) = \hat{\mathbf{c}}(n) + \alpha\, \varepsilon(n) \cdot r^*(n). \tag{5.13}$$

This algorithm is called a stochastic gradient algorithm because it uses an estimate of the gradient vector, also known as the LMS algorithm. The essentials of LMS algorithm are described by Widrow (see [Wid66] for details). The computational simplicity is one of the LMS algorithm's main advantages. However, slow convergence is a trade-off for this simplicity. This algorithm has only one adjustable parameter, Λ, which controls the convergence rate. This fundamental limitation causes the convergence rate to be slow [Pro01].

5.3.1.2 Recursive Least-Squares Algorithm

In order to achieve faster convergence, it is required to create more complex algorithms that take into account additional parameters [Pro01]. Applying the Least Squares adaptation criterion can lead to a fast initial equalizer convergence. The coefficients of a linear equalizer are selected to minimize the cost function [Wes09]

$$\varepsilon_{LS}(n) = \sum_{l=0}^{n} \Gamma^{n-l} \, |\mathbf{c}(n)^T \mathbf{r}(l) - s(l)|^2, \tag{5.14}$$

Γ^{n-l} $(\Gamma \leq 1)$ is a weighting factor applied to follow changes in the channel characteristics. For each time instant *n*, the algorithm minimizes the weighted summed squared error by calculating the current coefficient vector from the initial moment up to moment *n*. The calculation of (5.14) leads to the equation:

$$\mathbf{c}(n) = \mathbf{c}(n-1) + \phi^{-1}(n)\mathbf{r}(n)\varepsilon(n), \tag{5.15}$$

where ϕ is the autocorrelation matrix. The algorithm expressed by (5.15) is named the RLS direct form or Kalman algorithm [God74]. Proakis [Pro01] provides the complex version of this algorithm.

Kalman algorithm has two disadvantages, despite its superior convergence performance. The first is its complexity. Second, it is sensitive to roundoff noise accumulated as a result of the recursive computations. This may lead to algorithm instability [Pro01].

5.4 System Model

In this section, the author presents a different approach to adaptive equalizer based on a computationally efficient uni-cycle GA. In this proposal, the GA operates on sets of coefficients encoded as a string of real variables. The novelty of the current contribution is the dependence of a single algorithm time-step (Generation), hence the name UCGA, which reduces the computational cost of the GA. Equalizer decisions upon the data symbols are made on a symbol-by-symbol basis; it does not involve any interaction with a decoder, contrary to the procedure in [CW98].

The performance of the proposed UCGA is evaluated using a computer-based simulation experiment. The GA solver, which is part of the MATLAB packet, was employed during the study. Figure 5.5 presents a block diagram of the considered system. The developed equalizer receives the samples of Binary Phase-Shift Keying (BPSK) modulated signal, transmitted over a time-varying Rayleigh fading channel. A transversal filter, *i.e.*, an adaptive linear equalizer, processes the resultant channel output **r** to provide estimates (denoted by $\hat{\mathbf{r}}$) of the transmitted symbols. In detail, the estimate of the *n*th symbol of a data frame is defined as

$$\hat{r}(n) = \sum_{l=1}^{L} c_l(n) \cdot r\left(n + l - \left\lfloor \frac{L}{2} \right\rfloor\right), \tag{5.16}$$

where L is the equalizer length and the complex-valued numbers $c_1(n) \cdots c_L(n)$, *i.e.*, the elements of vector $\mathbf{c}(n)$ from Fig. 5.5 represent the tap coefficients of the equalizer in respective signalling interval.

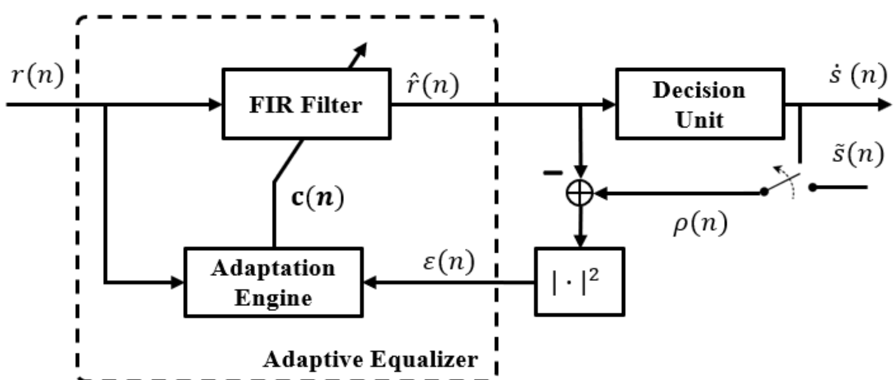


Figure 5. 5 Adaptive equalizer block diagram

The adaptation is performed by observing the error, *i.e.*, the difference between the desired sequence elements and the equalizer output. The error volume controls the direction of the coefficients adjustment from one iteration to the next in order to approach the optimum set of values. The Mean Square Error criterion serves as the basis for the optimization procedure that the adaptive algorithm uses to update the coefficient values [Hay13]; thus, the error signal is defined as:

$$\varepsilon(n) = |\rho(n) - \hat{r}(n)|^2. \tag{5.17}$$

During a training phase, a training sequence $\tilde{\mathbf{s}}$ is transmitted. The sequence is known in advance at the receiver side ($\rho(n) = \tilde{s}(n)$), so that the signal received in the training phase can be taken as a reference. The optimization criterion for the adaptive equalizer is to minimize the error given by eq. (5.17) according to the MSE criterion. In other words, the equalizer looks for the best vector $\mathbf{c}(n)$ of filter taps in the *n*th signalling interval.

Once the training phase is completed, the adaptive equalizer begins to operate using the decision-directed rule. The decision unit regenerates the signal outputted by the filter, so that it becomes an ideal BPSK symbol, *i.e.*,

$$\rho(n) = \dot{s}(n) = \arg\min_{s\in\{-1,+1\}} |s - \hat{r}(n)|^2. \tag{5.18}$$

Thanks to the reference samples provided by the decision unit, the adaptive equalizer can track the channel statistical variations without wasting the bandwidth for passing any midamble over the channel. However, the equalizer might fail if the decision is made in favour of a BPSK symbol opposite to the actually transmitted symbol.

5.4.1 Uni-Cycle Genetic Algorithm

The classical GA has been designed to address optimization problems that do not evolve over time [SD08, Abr07]. The population structure continuously evolves through the application of different GA operators, the individuals with the least fit chromosomes die, the newborns constitute a new generation; some random mutations are also applied to the chromosomes to introduce some minor random changes. Each new generation advances the algorithm toward the optimal (fixed) solution.

In the case of channel equalizer task, the role of the equalizer is to continuously track the current channel state, so there is no one fixed optimization goal. Conceptually, it is achievable to consider several GA generations to seek a solution (the channel taps or, equivalently, the decision on one transmitted symbol) that is optimal for a single signalling interval and then start from scratch to consider the next signalling interval. Such a solution has been presented in [CW98].

In the proposed UCGA strategy, only one generation per one signalling interval is assumed to limit the computational payload. The population results from one signalling interval is considered to be the initial population for the next UCGA instance, associated with the next signalling interval, and so on, as shown in Figure 5.6.

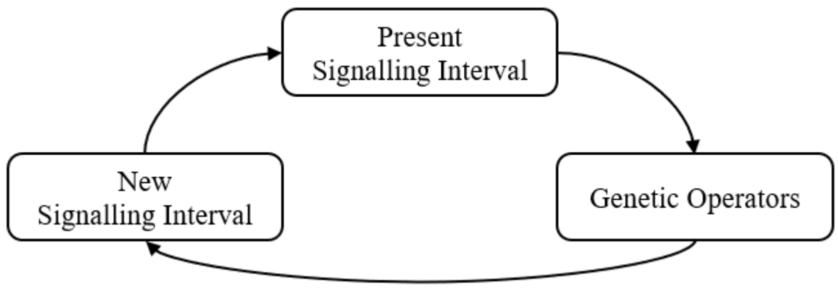


Figure 5. 6 UCGA framework. The GA operators are the base for the transition from one iteration to the next

Since the channel evolution is rather slow, it can be considered as if the global optimum was fixed over a few successive signalling intervals, which allows for UCGA convergence. Obviously, the algorithm cannot identify the exact optimum, which is slinking, but a rough estimation of the channel coefficients is often adequate to compensate for the channel effect effectively. The cycle of the UCGA operations provides the opportunity to adapt the current set of suggested solutions to the evolving channel state.

The implementation details of the UCGA are as follows:

- **Problem Representation**

In the proposed GA scheme, the chromosomes reflect the equalizer coefficients to be optimized. They are represented as vectors of real numbers. The data type used during the experiment is floating-point [SD08]. It is worth noting that, in general, one individual may include more than one chromosome; in the current system, the individual is determined by exactly one chromosome.

In detail, the chromosome represents both real and imaginary parts of the candidate coefficients:

$$\hat{\mathbf{c}}(n) = \left[\hat{c}_1^{\Re} \cdots \hat{c}_L^{\Re}\ \hat{c}_1^{\Im} \cdots \hat{c}_L^{\Im}\right], \tag{5.19}$$

where $\hat{c}_l^{\Re}, \hat{c}_l^{\Im} \in \mathbb{R}$ define the candidate coefficient $\hat{c}_l$ of the *l*th equalizer tap (or more precisely, its real and imaginary part, respectively).

- **Initialization**

The UCGA initialization is performed in the first signalling interval. Two strategies have been considered. The first approach assumes that all the chromosomes are generated randomly. The values of the real or imaginary components are constrained to the range between -1 and 1. The second strategy assumes random generation of all chromosomes, as above, with the exception of the first individual, for which $\hat{c}_{\lfloor L/2 \rfloor}^{\Re} = 1, \hat{c}_{l \neq \lfloor L/2 \rfloor}^{\Re} = 0$ and $\hat{c}_{l=1 \cdots L}^{\Im} = 0$. Such settings would be ideal for a single-path transmission, but the author anticipates they enhance GA convergence even in the case of a multipath Rayleigh fading channel.

Immediately after receiving the initial sample from the channel, the equalizer starts adjusting the equalizer taps.

- **Fitness Evaluation**

In every *n*th iteration (signalling interval), the UCGA population, $\Omega\,(n)$, comprises p individuals. As mentioned in Section 5.4, the optimization criterion for the proposed scheme is based on minimizing the mean-square value of the error signal, given by eq. (5.17), the MSE criterion. Consequently, to evaluate the fitness of a given individual, represented by chromosome $\hat{\mathbf{c}}$, in iteration *n*, the following cost function is used

$$J_{\hat{\mathbf{c}}}(n) = \left| \rho(n) - \sum_{l=1}^{L} \hat{c}_l(n) \cdot r(n + l - \lfloor L/2 \rfloor) \right|^2, \tag{5.20}$$

where $\hat{c}_l(n) = \hat{c}_l^{\Re} + j\hat{c}_l^{\Im}$. Let us denote the chromosome exhibiting the best fitness in iteration *n* as

$$\hat{\mathbf{c}}^{best}(n) = \arg\min_{\hat{\mathbf{c}}} J_{\hat{\mathbf{c}}}(n). \tag{5.21}$$

In every *n*th iteration, the UCGA searches for the minimum estimation error

$$J^{min}(n) = \min_{\hat{\mathbf{a}} \in \Omega(n)} \left(J_{\hat{\mathbf{a}}}(n) \right) \tag{5.22}$$

across the entire population $\Omega(n)$ to obtain a reference fitness value for comparisons between individual chromosomes. The chromosome with the best fitness is selected as the outcome of the algorithm in the given iteration; starting from the 2nd iteration, the worst chromosomes are replaced with newborns, resulting from genetic operations like selection, crossover, and mutation.

- **Selection**

In the current work, the chromosome selection is based on a roulette wheel selection method, *i.e.*, the probability that a given individual becomes a parent is proportional to its fitness, as detailed in Section (2.3.5.1). Accordingly, the selection probability in the *n*th iteration for an individual with chromosome $\hat{\mathbf{c}}$ can be expressed as:

$$Pr_{\hat{\mathbf{c}}}(n) = \frac{J_{\hat{\mathbf{c}}}(n)}{\sum_{\hat{\mathbf{a}} \in \Omega(n)} J_{\hat{\mathbf{a}}}(n)}\,. \tag{5.23}$$

The elite strategy is employed to avoid the loss of the best genetic material. According to it, some high-ranked individuals, in terms of fitness value, co-exist with the offspring generation.

- **Crossover and Mutation**

The intermediate crossover is utilized for the UCGA model to combine genes from two parents' chromosomes in the current iteration and form a new child, or individual, for the next iteration.

A random complex zero-mean Gaussian number ϖ is added to the chromosome value throughout the mutation operation. After some preliminary experiments, the standard deviation of ϖ has been set to an appropriate value, specified in Table 5.1. Thanks to the mutation, which brings some random changes to the chromosomes, the UCGA is unlikely to stick at one of the local optima. It must be noted that elite individuals and crossover children are not subject to the mutation procedure.

Table 5. 1 GA settings

Parameter	Value
Population Size	200
Number of Generations	1
Crossover Type	Intermediate
Crossover Probability	0.85
Mutation Method	Gaussian
Std. Deviation of Mutation Probability	0.06
Selection Method	Roulette Wheel
Elite count	72

5.5 Results

To evaluate the performance of the proposed UCGA-based Adaptive Equalizer (UCGA-AE), the measures of the following characteristics are taken: BER against SNR and MSE vs iteration number. Afterwards, they are compared with the reference results obtained by means of the LMS-based Adaptive Equalizer (LMS-AE) and RLS-based Adaptive Equalizer (RLS-AE). The UCGA initialization is also investigated. To ensure comparison fairness, the same channel conditions are considered for the proposed and reference methods, including the same random generator seed when simulating the channel performance.

5.5.1 Simulation Scenarios

Two simple but realistic GSM propagation models with 6 and 12 delay taps, respectively, accurate for an urban area, are considered [TS17]. The power delay profiles of such channels are presented in Table 5.2 and Table 5.3, respectively. In order to get a wider scope of the UCGA-AE capabilities, a 6-tap hilly-terrain model, characterized by the power delay profile from Table 5.4, is also taken into account. The sampling rate of 1.5 MHz is assumed. Respective sample time of *ca.* 0.67 μs, compared with all power delay profiles, guarantees a selective kind of fading for every considered channel profile. The impact of the Doppler effect on the equalizer performance is studied for every channel model; numerous Doppler shift values are assumed one by one: f_d= 0, 3, 15, 50, 200 Hz.

Table 5. 2 Typical urban area 6-Tap channel power delay profile

Tap number	Relative time (μs)	Value
1	0.0	-3.0
2	0.2	0.0
3	0.5	-2.0
4	1.6	-6.0
5	2.3	-8.0
6	5.0	-10.0

Table 5. 3 Typical urban area 12-Tap channel power delay profile

Tap number	Relative time (μs)	Value
1	0.0	-4.0
2	0.1	-3.0
3	0.3	0.0
4	0.5	-2.6
5	0.8	-3.0
6	1.1	-5.0
7	1.3	-7.0
8	1.7	-5.0
9	2.3	-6.5
10	3.1	-8.6
11	3.2	-11.0
12	5.0	-10.0

Table 5. 4 Typical hilly terrain 6-Tap channel power delay profile

Tap number	Relative time (μs)	Value
1	0.0	0.0
2	0.1	-1.5
3	0.3	-4.5
4	0.5	-7.5
5	15.0	-8.0
6	17.2	-17.7

The receiver in any real communication system is not capable of predicting exact channel parameters, *e.g.*, time dispersion. Hence, the number of equalizer taps is assumed to be 15 for the UCGA-AE, LMS-AE and the RLS-AE, regardless of the considered channel model. To ensure the accuracy of the results, the data frame contains 100,000 uncoded BPSK symbols, from which 500 initial symbols belong to the training sequence.

Table 5.1 presents the most important settings of the MATLAB GA solver used to evaluate the performance of UCGA-AE. Regarding the reference LMS and RLS algorithms, some preliminary tests have been carried out to find the most accurate value of the forgetting factor. The forgetting factor of LMS-AE was set to 0.045 and 0.95 in the case of RLS-AE.

5.5.2 Simulation Results

The performance results for the adaptive equalizers to be compared are presented in this section. Fig. 5.7 and Fig. 5.8 give the performance results of UCGA, LMS, and RLS for 6-tap urban channel model. The performance results of the compared equalizers in the case of 12-tap urban channel model are shown in Fig. 5.9 and Fig. 5.10, while the results for 6-tap hilly-terrain channel model are illustrated in Fig. 5.11 and Fig. 5.12, respectively. Each time, a variety of Doppler shift (f_d) values are considered. The measure points at the SNR scale are distributed every 5 dB, which, in the author's opinion, is enough to display the trend and the curves' locations to each other (beware that the markers do not represent data points). The 95% confidence intervals are presented at data points to prove accuracy of the simulation outcomes; they are obtained according to the method presented in [JBS92].

In general, for all channel models and most Doppler shifts, the UCGA-AE (solid lines in Figures 5.7, 5.8, 5.9, 5.10, 5.11, and 5.12) is superior to both the LMS-AE and the RLS-AE (dashed lines).

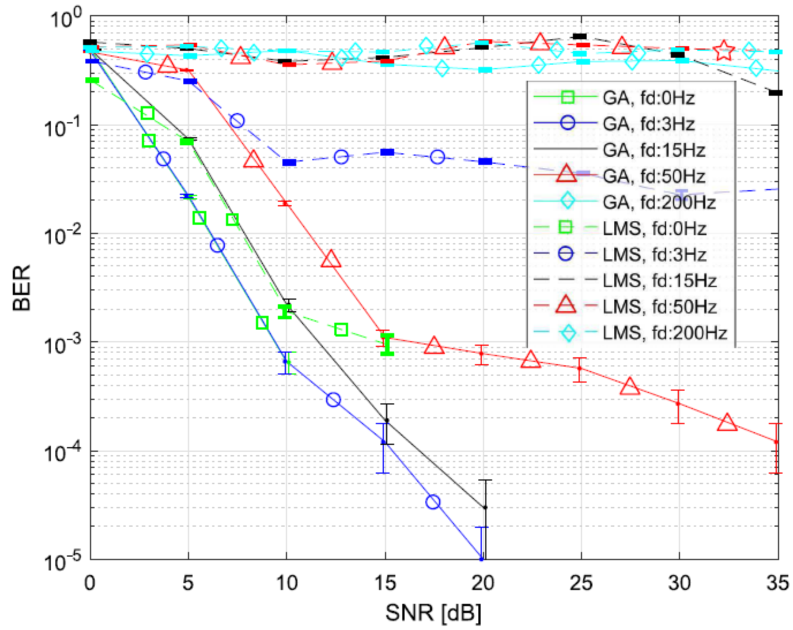


Figure 5. 7 BER vs SNR performance for the considered adaptive equalizers: GA and LMS under different Doppler shifts (f_d = 0, 3, 15, 50, and 200 Hz) over a typical urban area 6-tap channel

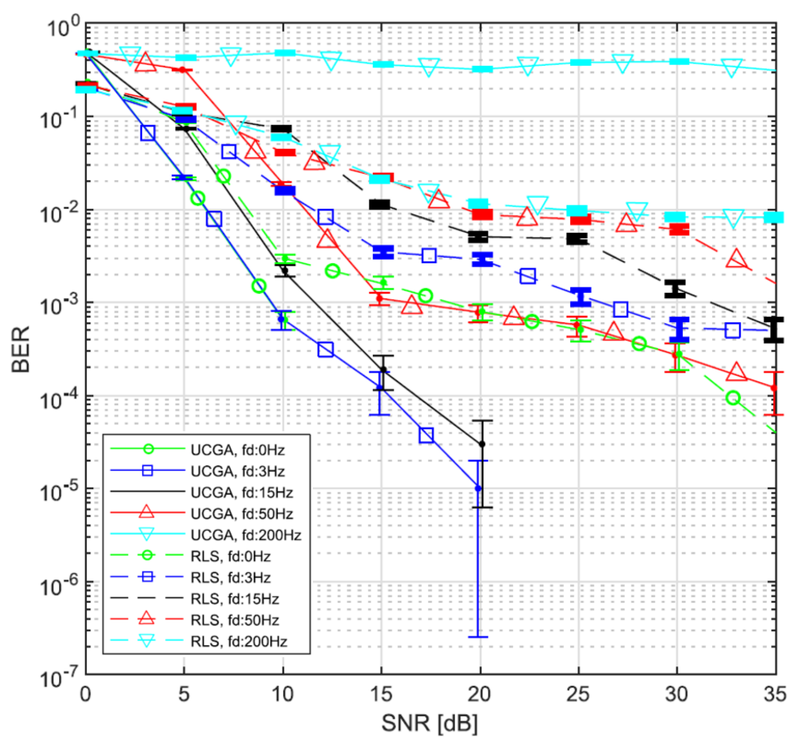


Figure 5. 8 BER vs SNR performance for the considered adaptive equalizers: GA and RLS under different Doppler shifts (f_d= 0, 3, 15, 50, and 200 Hz) over a typical urban area 6-tap channel

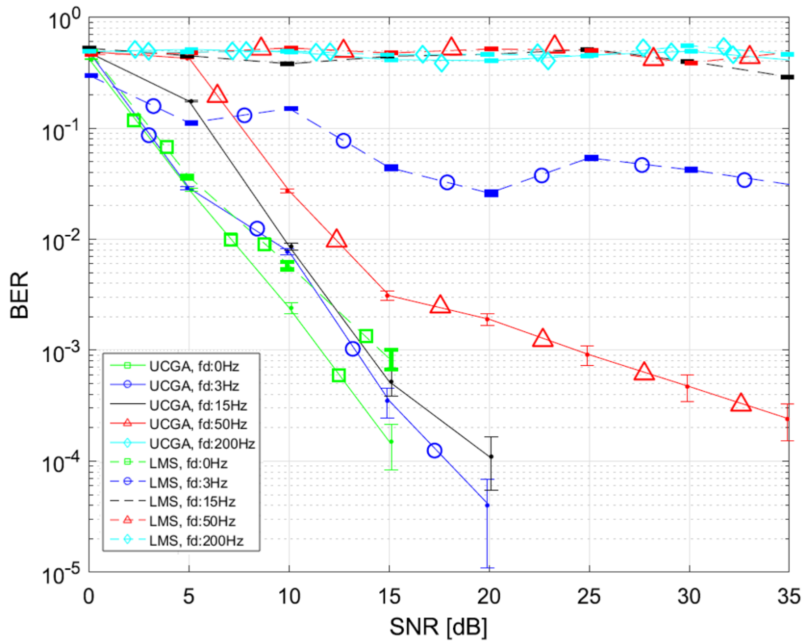


Figure 5. 9 BER vs SNR performance for the considered adaptive equalizers: GA and LMS under different Doppler shifts (f_d = 0, 3, 15, 50, and 200 Hz) over a typical urban area 12-tap channel

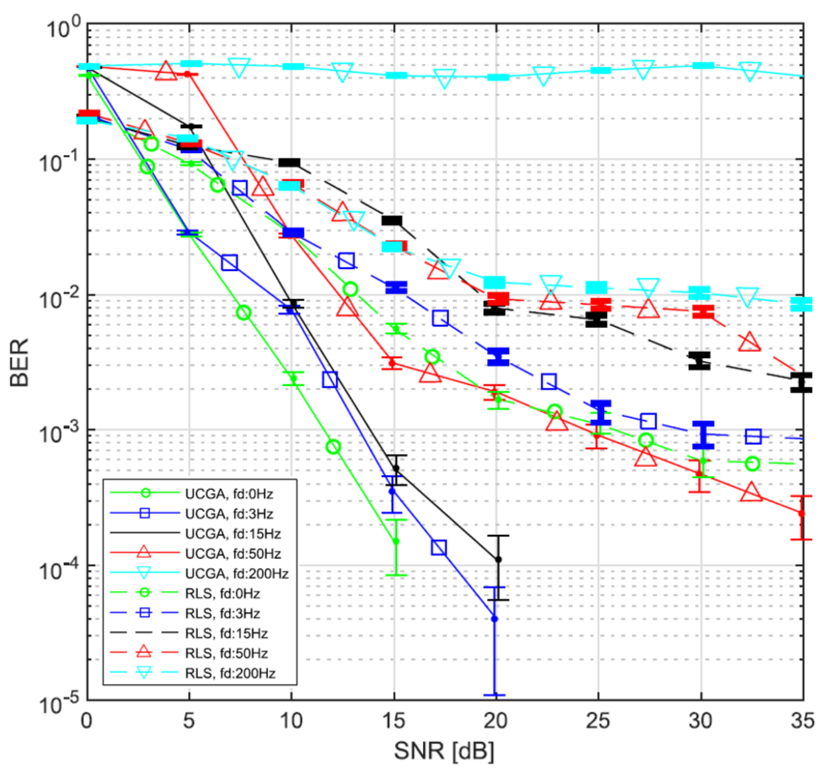


Figure 5. 10 BER vs SNR performance for the considered adaptive equalizers: GA and RLS under different Doppler shifts (f_d = 0, 3, 15, 50, and 200 Hz) over a typical urban area 12-tap channel

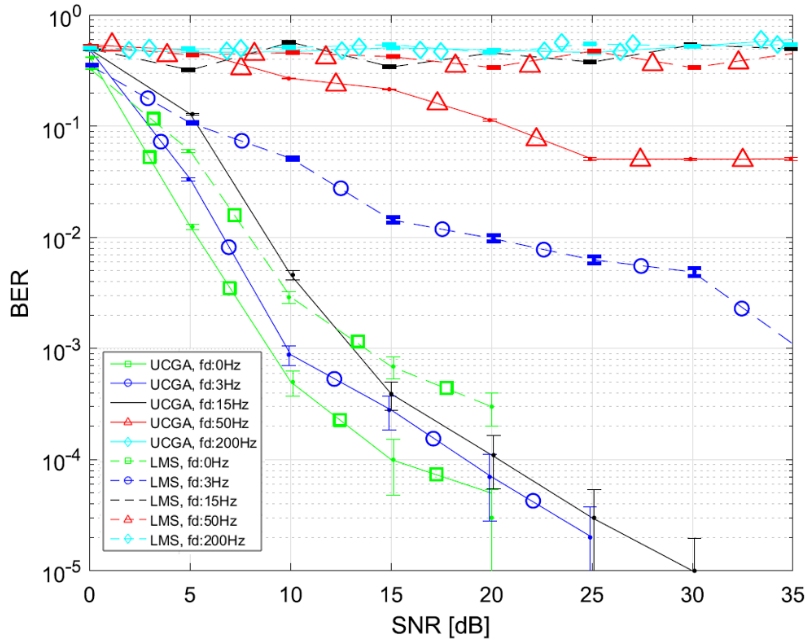


Figure 5. 11 BER vs SNR performance for the considered adaptive equalizers: GA and LMS under different Doppler shifts (f_d = 0, 3, 15, 50, and 200 Hz) over a typical hilly-terrain 6-tap channel

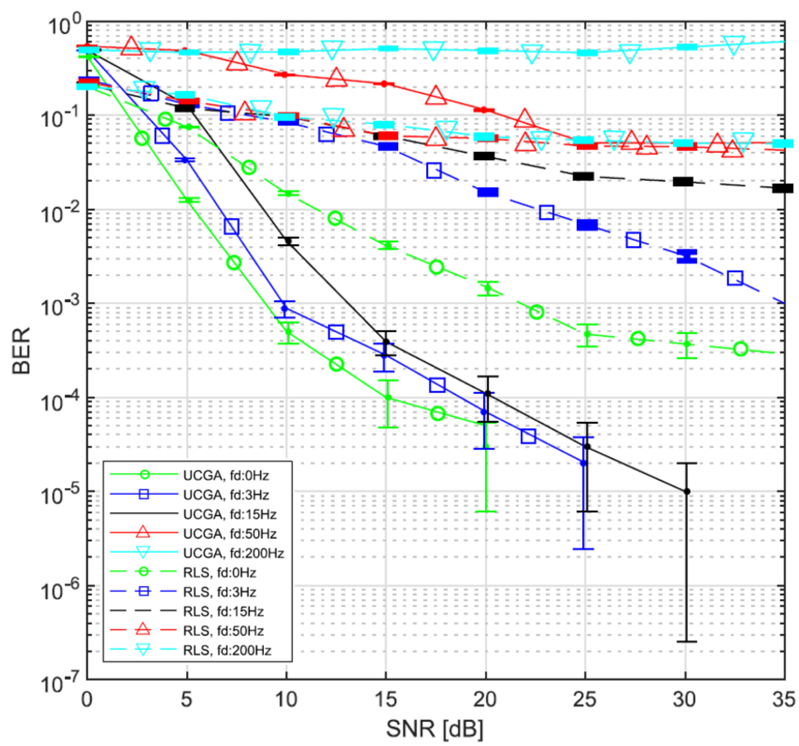


Figure 5. 12 BER vs SNR performance for the considered adaptive equalizers: GA and RLS under different Doppler shifts (f_d = 0, 3, 15, 50, and 200 Hz) over a typical hilly-terrain 6-tap channel

In detail, having assumed f_d = 0 Hz, one can observe a 6 dB SNR gain of the UCGA-AE over the LMS-AE at 10^{-3} BER level, in Figure 5.7. Note the overlapping GA, f_d = 0 Hz and GA, f_d = 3 Hz curves. In the case of 12-tap channel model, refer to Figure 5.9, the gain of *ca.* 3 dB is spotted, instead. In the case of 6-tap hilly-terrain channel, Figure 5.11, it is as much as 5 dB. For f_d = 3 Hz, the reference LMS-AE system experiences a slow decline with increasing SNR over any considered channel (in the case of UCGA-AE it is much more significant). For any higher f_d, LMS-AE fails at all, regardless of considered channel model. It must be noted that the results display poor performance for both UCGA-AE and LMS-AE with f_d = 200 Hz. Additionally, the proposed UCGA-AE cannot cope with the 6-tap hilly-terrain channel model when f_d = 50 Hz.

In comparison with RLS-AE, a particularly high SNR gain (as much as 17 dB) can be observed at 10^{-3} BER level with either 6-tap or 12-tap urban channel models, if one assumes f_d = 3 Hz (refer to Figures 5.8 and 5.10; note the overlapping curves for UCGA, f_d = 0 Hz and UCGA, f_d = 3 Hz). Equally remarkable is that with a 6-tap hilly-terrain channel, Figure 5.12, RLS-AE curves decline much slower with increasing SNR, in comparison with respective UCGA-AE curves. A weak point of the UCGA-AE is its performance with f_d = 200 Hz. Moreover, the efficiency of the proposed UCGA-AE is not satisfactory if the transmission runs over the hilly-terrain 6-tap channel exhibiting a 50 Hz Doppler shift.

For both comparisons, the presented 95% confidence intervals become notably wide at high SNRs as the number of observed errors is quite low in relation with the estimated BER level. It makes the results below 10^{-4} quite worthless.

Figures 5.13, 5.14, 5.15, and 5.16 compare different population initialization methods, mentioned in the previous Section. One is fully random initialization, in which all chromosomes in the initial population in the 1st signalling interval are generated randomly, and single-1 initialization is the second method. The typical urban area 6-tap channel model was employed in this part of the experiment. From all presented figures it can be seen that the single-1 initialization brings some gain over fully random initialization, *i.e.,* the UCGA-AE is capable of achieving a given BER level at a reasonably lower SNR.

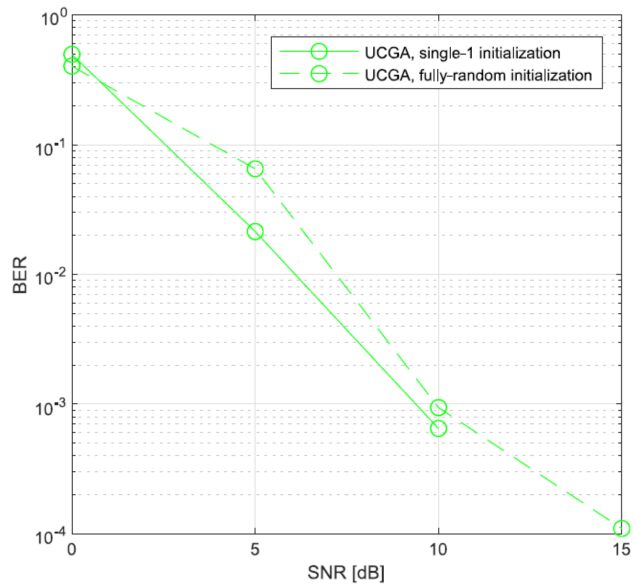


Figure 5. 13 Comparison of different population initialization methods: fully-random initialization and single-1 initialization; assumed typical urban area 6-tap channel model with a specific Doppler shift: f_d = 0 Hz

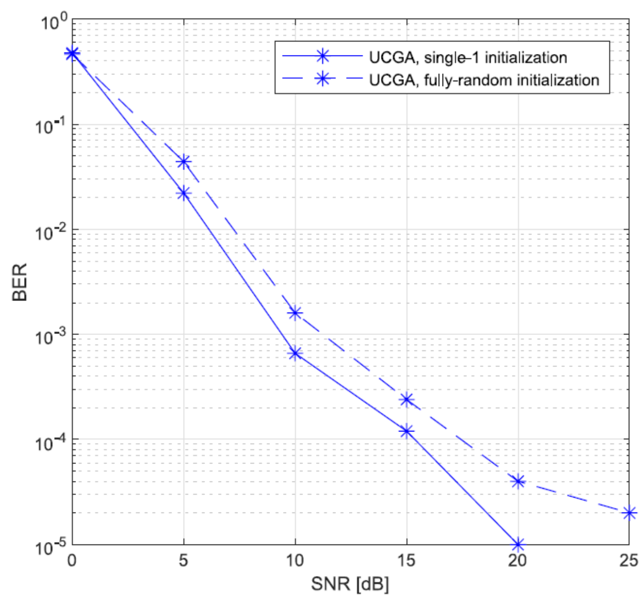


Figure 5. 14 Comparison of different population initialization methods: fully-random initialization and single-1 initialization; assumed typical urban area 6-tap channel model with a specific Doppler shift: f_d = 3 Hz

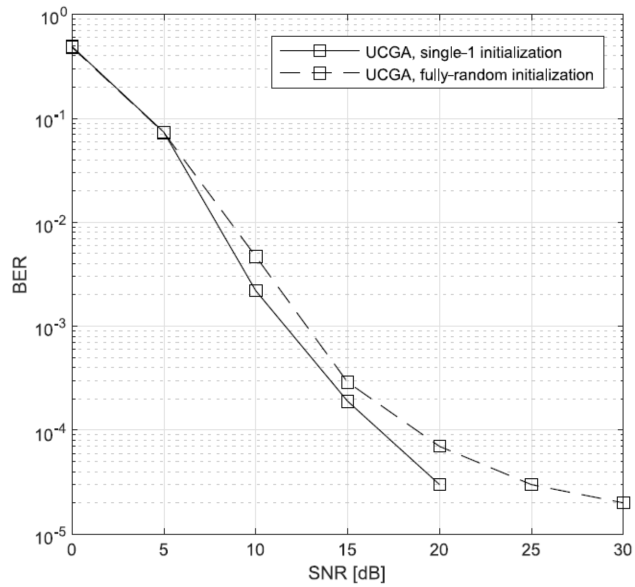


Figure 5. 15 Comparison of different population initialization methods: fully-random initialization and single-1 initialization; assumed typical urban area 6-tap channel model with a specific Doppler shift: f_d = 15 Hz

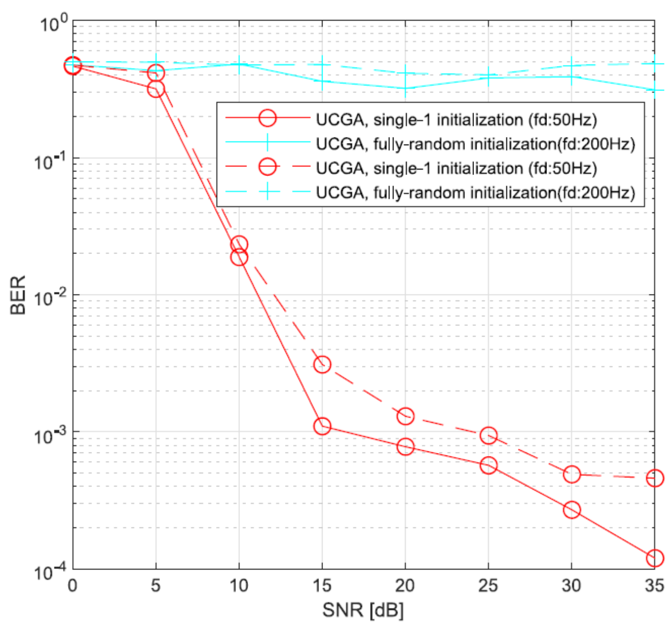


Figure 5. 16 Comparison of different population initialization methods: fully-random initialization and single-1 initialization; assumed typical urban area 6-tap channel model with specific Doppler shifts: f_d = 50 Hz (red curves) and f_d = 200 Hz (blue curves)

The training-phase convergence of UCGA-AE, LMS-AE and RLS-AE is also studied for the same channel model, *i.e.*, the curves in Figures 5.17, 5.18, and 5.19 represent the MSE vs iteration number, given SNR = 35 dB and $f_d = 0$ Hz for the considered transmission of a single data frame. More reliable data regarding the training phase convergence are presented in the form of a box plot [Doc22] for every 20th iteration (signalling interval). The box plots, shown in Figures 5.20 and 5.21 deliver MSE statistics based on the results collected in 500 trial runs (in every trial run, only the training sequence is transmitted). The red lines on the boxes represent the expected values of the median in the considered iteration, the notches delimit 95% confidence intervals of the median. The bottom and the top boxes' ends have the meaning of the 1st and the 3rd MSE quartile, respectively. The whiskers, set to be 1.5 times the interquartile range, cover as much as *ca.* 99.3% ("almost all") entries. The remaining outliers are shown by (+) markers.

Judging by the presented plots, it must be pointed out that UCGA converges faster than both LMS and RLS as its starting point of the MSE vs iteration curve is placed significantly lower (*ca.* 10^{-4} in comparison with 10^{0} for both LMS and RLS). The average steady-state MSE exhibited by UCGA (*ca.* 10^{-6}) is unattainable for the LMS (*ca.* 10^{-2}) and the RLS (*ca.* 10^{-1}).

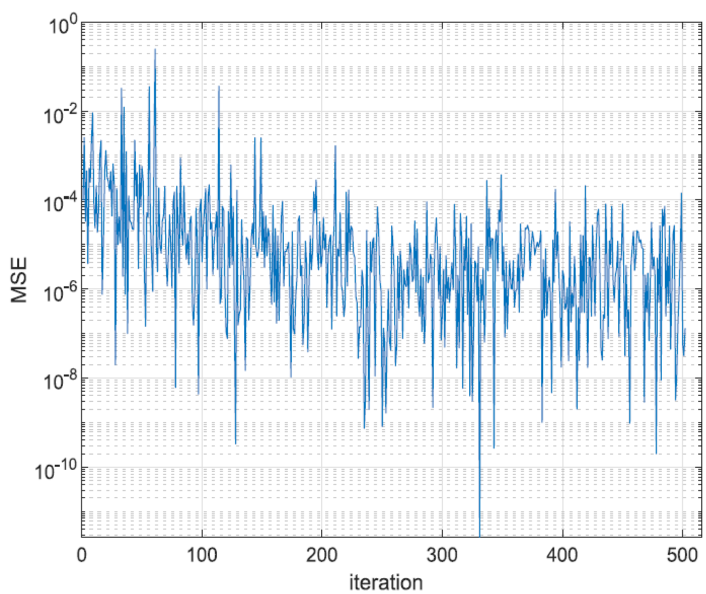


Figure 5. 17 Convergence characteristics of UCGA at SNR = 35 dB, f_d = 0 Hz in the case of typical urban area 6-tap channel

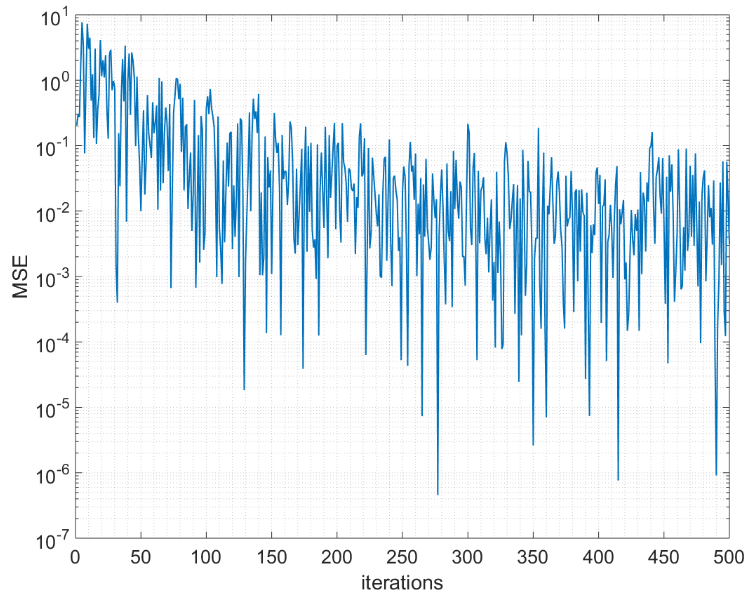


Figure 5. 18 Convergence characteristics of the LMS at SNR = 35 dB, f_d = 0 Hz in the case of typical urban area 6-tap channel

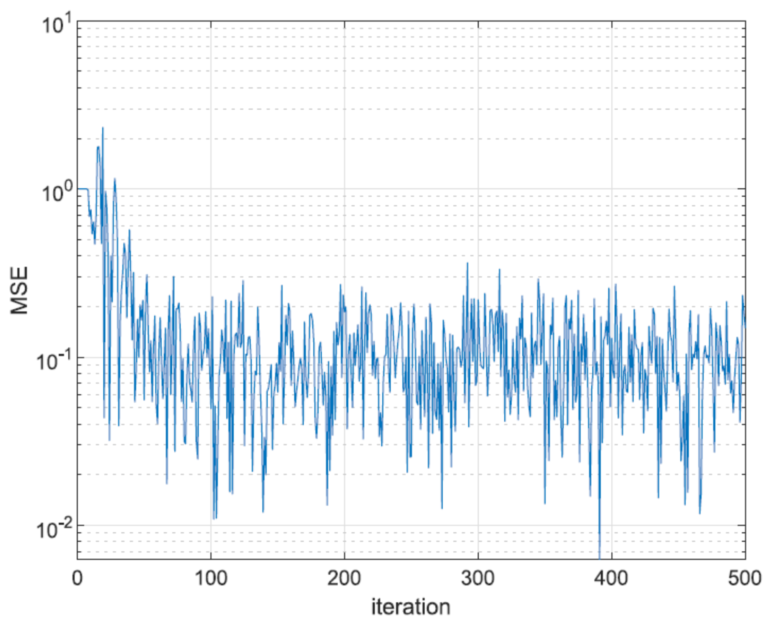


Figure 5. 19 Convergence characteristics of the RLS at SNR = 35 dB, f_d = 0 Hz in the case of typical urban area 6-tap channel

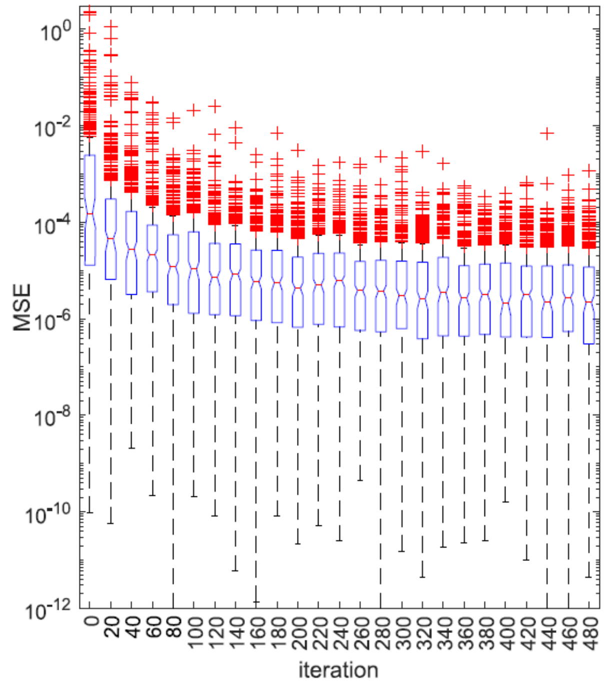


Figure 5. 20 Box plots of convergence characteristics of UCGA at SNR = 35 dB, f_d = 0 Hz in the case of typical urban area 6-tap channel

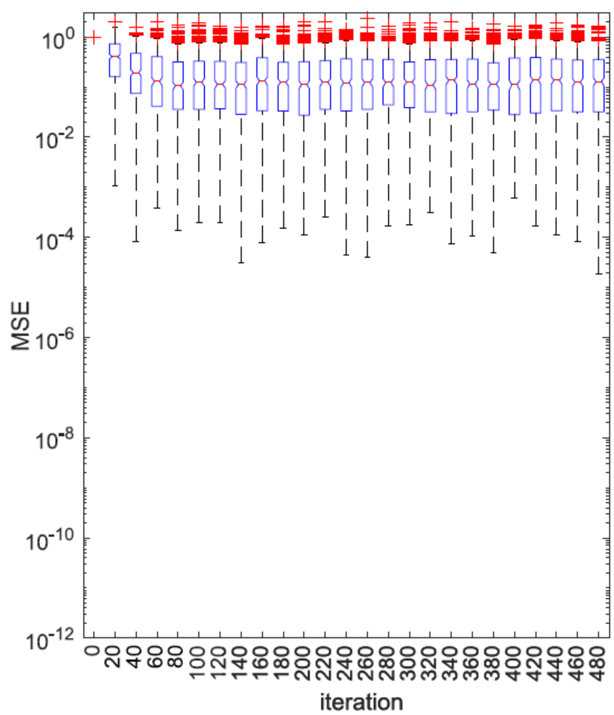


Figure 5. 21 Box plots of convergence characteristics of the RLS at SNR = 35 dB, f_d = 0 Hz in the case of typical urban area 6-tap channel

Figures 5.22 and 5.23 illustrate the tracking capability of UCGA and LMS for the time-varying channel. The transmitted BPSK symbols are subject to Doppler shift effect, $f_d = 3$ Hz in the case of the 6-tap urban propagation model.

Fig. 5.22 proves the UCGA ability to track the channel state for a long period of time without any training sequence in the middle of the frame. Meanwhile, the LMS algorithm does not exhibit the capability to track the channel variations over time, *i.e.*, the MSE begins to increase as soon as the training phase (500 samples) has finished, as presented in Fig. 5.23.

The crucial UCGA assumption that just one GA generation is considered per one signalling interval has a substantial impact on the computational complexity of the GA. This approach makes the cost of the learning part linearly proportional to the population size. Regarding the filtering part, in order to calculate $\hat{r}$ according to (5.16), given L filter taps, the system requires L multiplications and $L - 1$ additions.

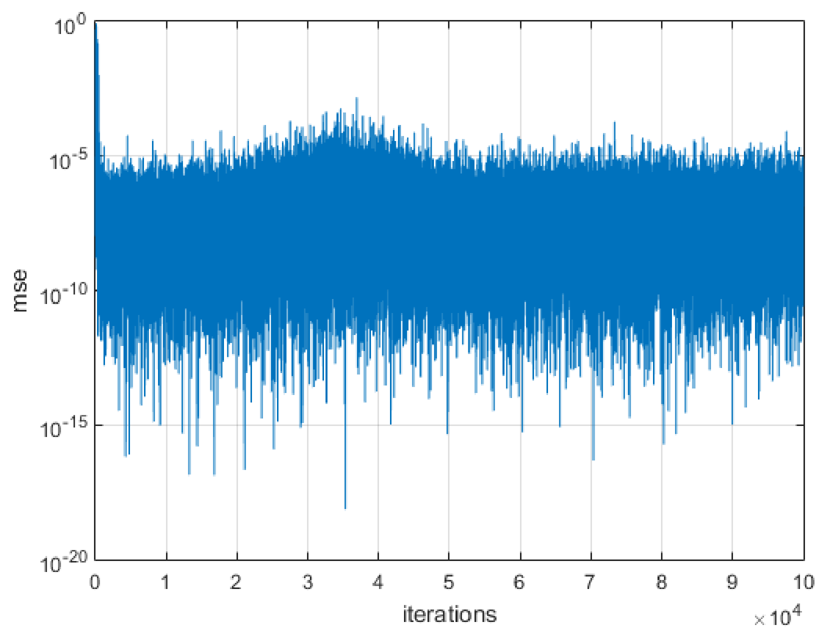


Figure 5. 22 MSE vs iteration of UCGA at SNR = 35 dB and f_d = 3 Hz in the case of 6-tap urban propagation model

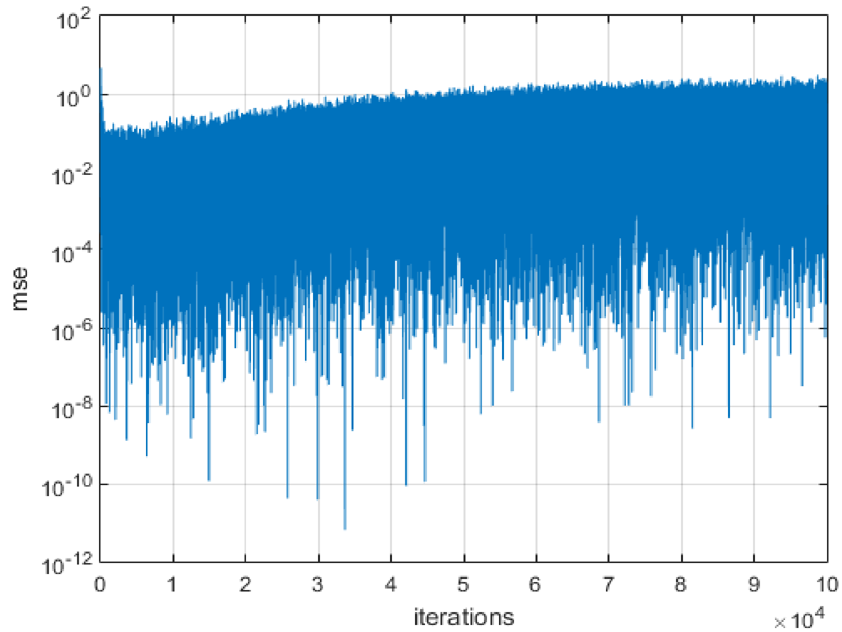


Figure 5. 23 MSE vs iteration of the LMS at SNR = 35 dB and f_d = 3 Hz in the case of 6-tap urban propagation model

5.6 Summary

GA is a powerful tool for solving many optimization problems. The UCGA technique enables channel tracking, which has been verified in several simulation experiments, reported in this chapter. In the majority of the considered simulation setups, the UCGA-AE remarkably outperforms the reference RLS-AE and LMS-AE. As a consequence, the UCGA can be considered a robust channel equalizer engine for future wireless systems, including 6G telecommunications. As for any problem solved by means of the GA, it is crucial to properly set the solver parameters and initialize the population. It has been verified experimentally that the single-1 initialization gives faster UCGA convergence than the random assignment of initial values to all chromosomes.

Chapter 6

6.1 Conclusions

In this thesis, the GA-aided applications in the wireless network have been investigated, namely MU-MIMO Detection and Adaptive Linear Equalization.

The GA-based MU-MIMO detector exhibits an outstanding performance for low order modulation scenario but in comparison with ZF performance, it imposes a higher computational complexity, proportional to the population size and/or the number of generations.

As an evolutionary algorithm, GA can try to find the optimal solution starting from scratch. However, the author observed that loading a single well-fitted seed individual into the initial population brings faster convergence and better optimization result in general. At the first attempt, he proposed to run the simple ZF MUD before launching GA to obtain the seed individual. The resultant combined ZF-GA strategy appeared to outperform both pure ZF reference and basic GA-driven MUD in terms of BER, especially for low-order modulations.

The current author's contribution is a new population initialization method that resembles the SIC approach. In brief, it consists in retaining the ZF detector's decision with respect to the most reliable out of all interfering signals. The novel approach exhibits the ability to improve the performance of GA-driven MIMO multi-user detector at no cost in comparison with the previously considered ZF-GA MUD.

A different GA application, considered in the thesis, is the problem of wireless channel equalization. Several simulation experiments have proven that the proposed UCGA-driven channel equalizer is able to converge quickly and track the channel state, significantly outperforming the benchmarks: RLS-AE and LMS-AE. The UCGA may, therefore, be considered as a reliable channel equalization engine for the next wireless systems.

Taking into account the above conclusions, the author believes that the dissertation thesis has been proven, *i.e.*, the performance of the GA-based MU MIMO detector has been significantly enhanced at no extra computational cost through the application of a new GA initialization technique based on the SIC approach. Besides, the performance of an adaptive channel equalizer has been improved by employing a low-cost GA-driven adaptation engine that boosts the convergence rate and improves the channel tracking ability.

6.2 Future and unsolved problems

On the basis of the solutions discussed throughout this thesis, some important advantages of the GAs for wireless communications were illustrated. There are numerous ways in which GA can be further improved. For instance, taking into account that attaining a small BER improvement at the cost of much higher effort is impractical, the population size can be adapted as per requirement, thereby accelerating search and saving computation time, which makes the algorithm faster.

The use of higher-order modulations, such as QPSK, 16-QAM, 64-QAM, etc., could also be considered in the future in the context of UCGA channel equalizer. Chromosome representation is another item worth particular interest – fixed point representation of equalizer taps is beneficial since fixed-point arithmetic can be easily implemented on popular logic structures, such as Field-Programmable Gate Arrays (FPGAs), or microprocessors, as opposed to floating-point operations.

Some other interesting topics are also worth further studying, for example, the MUD problem with a higher number of users transmitting their signal concurrently.

Another interesting aspect would be the test of other evolutionary strategies, *e.g.*, Ant Colony Optimization (ACO) [DBS06], Particle Swarm Optimization (PSO) [EK95], in the same context.

Bibliography

[AB93] S. A. Altekar, and N. C. Beaulieu. “Upper Bounds on the Error Probability of Decision Feedback Equalization,” IEEE Trans. Inform. Theory, vol. IT- 39, 1993: pp. 145–156.

[Abr07] M. A. Abramson. “Genetic algorithm and direct search toolbox user’s guide.” The MathWorks Inc (2007).

[ACM03] E. Amaldi, A. Capone, and F. Malucelli. “Planning UMTS base station location: Optimization models with power control and algorithms”. IEEE Transactions on Wireless Communications, 2, 2003: pp. 939-952.

[AH94] H. Adeli, and S. L. Hung.” Machine Learning: Neural Networks, Genetic Algorithms and Fuzzy Systems”. New York: Wiley, 1994.

[AM84] J. B. Anderson, and S. Mohan.” Sequential Decoding Algorithms: A Survey and Cost Analysis”. IEEE Trans. Commun. 32, 1984: pp.169–176.

[And05] J. G. Andrews. “Interference cancellation for cellular systems: A contemporary overview.” IEEE Wireless Communications, vol.12, no.2 ,2005: pp.19-29.

[AO15] S. Akaneme and G. N. Onoh, “Algorithms for evaluating adaptive equalization in wireless communication systems,” Eur. J. Eng. Technol., vol. 3, no. 2, 2015: pp. 1–8

[APA15] I. A. Alimi, J. J. Popoola, K. F. Akingbade, and M. O. Kolawole,” Interference Management in MIMO-OFDM- Based Emerging Wireless Systems”, American Journal of Information Science and Computer Engineering, Vol. 1, No. 1, 2015.

[Bak85] J. E. Baker. “Adaptive selection methods for genetic algorithms.” Proceedings of an International Conference on Genetic Algorithms and their applications”. vol. 1. 1985.

[BCC07] E. Biglieri, R. Calderbank, A. Constantinides, A. Goldsmith, A. Paulraj, and H.V. Poor. “MIMO wireless communications”. Cambridge university press, 2007.

[BJ09] J. Blomer, and N. Jindal. “Transmission capacity of wireless ad hoc networks: Successive interference cancellation vs. joint detection." 2009 IEEE International Conference on Communications. IEEE, 2009.

[BLM12] J. R. Barry, E.A. Lee, and D. G. Messerschmitt, "Digital communication". Springer Science & Business Media, 2012.

[BNK98] W. Banzhaf, P. Nordin, R.E. Keller, and F.D. Francone. "Genetic programming: an introduction: on the automatic evolution of computer programs and its applications". Morgan Kaufmann Publishers Inc., 1998.

[BO13] C. Bruno, and C. Oestges. "MIMO wireless networks: channels, techniques and standards for multi-antenna, multi-user and multi-cell systems". Academic Press, 2013.

[Bor13] B. Farhang-Boroujeny. "Adaptive filters: theory and applications". John Wiley & Sons, 2013.

[BP79] C. A. Belfiore and J. H. Park, "Decision feedback equalization," Proc. IEEE, vol. 67, no. 8, 1979: pp. 1143–1156.

[BZG10] N. Surajudeen-Bakinde, X. Zhu, J. Gao, A.K. Nandi, and H. Lin. "Genetic algorithm Based equalizer for ultra-wideband wireless communication systems." IEICE transactions on communications vol.93, no.10 ,2010: pp. 2725-2734.

[CFP94] A. J. Chipperfield, P.J. Fleming, and H. Pohlheim. "A Genetic Algorithm Toolbox for MATLAB". In: Proc. Int. Conf. Sys. Engineering, 1994: pp. 200-207.

[Che13] M. Chen. "Iterative Detection for Overloaded Multiuser MIMO OFDM Systems". Diss. University of York, 2013.

[CKY10] Y. S. Cho, J. Kim, W.Y. Yang, and C.G. Kang." MIMO-OFDM wireless communications with MATLAB". John Wiley & Sons, 2010.

[CLV07] C. A. Coello, G. B Lamont, and D. A. Van Veldhuizen. "Evolutionary algorithms for solving multi-objective problems". Vol. 5. New York: Springer, 2007.

[CS03] D. Chen, and T. Saito. "A new method to reduce the complexity of joint detection algorithm." GLOBECOM'03. IEEE Global Telecommunications Conference (IEEE Cat. No. 03CH37489), vol. 4, IEEE, 2003.

[CW98] S. Chen, and Y. Wu. “Maximum likelihood joint channel and data estimation using genetic algorithms.” IEEE Transactions on Signal Processing vol. 46, no.5, 1998: pp. 1469-1473.

[CW98] S. Chen, and Y. Wu. “Maximum likelihood joint channel and data estimation using algorithms.” IEEE Transactions on Signal Processing vol. 46, no .5, 1998: pp. 1469-1473.

[CY10] H. Cheng, and S. Yang. “Multi-population genetic algorithms with immigrants’ scheme or dynamic shortest path routing problems in mobile ad hoc networks.” European conference on the applications of evolutionary computation. Springer, Berlin, Heidelberg, 2010.

[DBS06] M. Dorigo, M. Birattari, and T. Stutzle. “Ant colony optimization.” IEEE computational intelligence magazine vol.1, no.4 ,2006: pp.28-39.

[DC03] Y. Du and K. T. Chan. “BFeasibility of applying genetic algorithms in space-time block coding multiuser detection systems, [in Proc.” IASTED Int. Conf. Wireless and Optical Communications, Banff, AB, Canada. Vol. 3. 2003.

[DC05] Y. Du, and K. T. Chan. “Feasibility of applying genetic algorithms in space-time block coded multiuser detection systems.” Digital Signal Processing vol.15, no .2 ,2005: pp. 161-170.

[DMM74] D. Duttweiler, J. Mazo, and D. Messerschmitt. “Error Propagation in Decision-Feedback Equalizers,” IEEE Trans. Inform. Theory, vol. IT-20, 1974: pp. 490–497.

[Doc22] Documentation on the Use of Boxplots in MATLAB. Available online: https://www.mathworks.com/help/stats/boxplot.html

[DSA20] A. Davahli, M. Shamsi, and G. Abaei. “Hybridizing genetic algorithm and grey wolf optimizer to advance an intelligent and lightweight intrusion detection system for IoT wireless networks.” Journal of Ambient Intelligence and Humanized Computing vol. 11, no.11, 2020: pp. 5581-5609.

[EH00] C. Ergu¨n, and K. Hacioglu. “Multiuser detection using a genetic algorithm in CDMA communications systems, IEEE Trans. Commun., vol. 48, no. 8, 2000: pp. 1374–1383.

[EK95] R. Eberhart, and J. Kennedy. “Particle swarm optimization.” Proceedings of the IEEE international conference on neural networks, vol. 4. 1995.

[EKK95] A. E. Eiben, C. H.M Kemenade, and J. N. Kok. “Orgy in the computer: multi-parent reproduction in genetic algorithms.” European conference on artificial life. Springer, Berlin, Heidelberg, 1995.

[Eng07] A. P Engelbrecht. “Computational intelligence: an introduction”. John Wiley & Sons, 2007.

[EQ88] M. V. Eyuboglu, and S.U. Qureshi. “Reduced-state sequence estimation with set partitioning and decision feedback.” IEEE Transactions on Communications vol.36, no.1 ,1988: pp. 13-20.

[ES03] E. Eiben, and J. E. Smith.” Introduction to evolutionary computing”. vol. 53. Berlin: springer, 2003.

[EZ15] A. Elghariani, and M. Zoltowski. “Successive interference cancellation for large-scale MIMO OFDM.” 2015 IEEE international conference on electro/information technology (EIT). IEEE, 2015.

[FG98] G. J. Foschini and M. J. Gans. “On limits of wireless communications in a fading environment when using multiple antennas”. Wireless personal communications vol. 6, no.3, 1998: pp. 311-335.

[Fog98] D. B. Fogel. “Evolutionary Computation: The Fossil Record”. IEEE Press, Piscataway, NJ, 1998.

[For72] G. D. J. R Forney. “Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference.” IEEE Transactions on Information theory vol.18, no.3 ,1972: pp. 363-378.

[Fos96] G. J. Foschini. “Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas”. Bell Labs Technical Journal, vol. 1, no. 2, 1996: pp. 41–59.

[FOW65] L. J. Fogel, A. J. Owens, and M. J. Walsh. “Artificial intelligence through a simulation of evolution”. Proc. of the 2nd Cybernetics Science Symp, 1965.

[FOW66] L. J. Fogel, A. J. Owens, and M. J. Walsh. “Artificial Intelligence through Simulated Evolution”. Wiley, Chichester, UK, 1966.

[GB89] J. J. Grefenstette and J. E. Baker. “How genetic algorithms work: A critical look at implicit parallelism.” Proceedings of the 3rd International Conference on Genetic algorithms, 1989.

[GC00] M. Gen and R. Cheng. “Genetic Algorithms and Engineering Optimisation, Engineering Design and Automation”. New York: Wiley, 2000.

[GD91] D. E. Goldberg, and K. Deb. “A comparative analysis of selection schemes used in genetic algorithms, Foundations of Genetic Algorithms”. vol. 1. Elsevier, 1991: pp. 69-93.

[Gen96] M. Gen. “Genetic algorithms and engineering design”. Wiley-IEEE,1996, ISBN 0471127418.

[GFV99] G. D. Golden, G. J. Foschini, R. A. Valenzuela, and P. W. Wolniansky. “Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture,” Electronics Letters, vol. 35, no. 1, 1999: pp. 14–16.

[God74] D. Godard. “Channel equalization using a Kalman filter for fast data transmission.” IBM journal of Research and Development vol.18, no.3,1974: pp. 267-273.

[Gol89] D. E. Goldberg. “Genetic Algorithms in Search, Optimization & Machine Learning”. Reading MA: Addison-Wesley (1989).

[Gre86] J. J. Grefenstette, “Optimization of control parameters for genetic algorithms,”. IEEE Transactions on systems, man, and cybernetics, vol. 16, no. 1, 1986: pp. 122–128.

[Gre93] J. J. Grefenstette. “Genetic Algorithms for Machine Learning”. Norwell, MA: Kluwer Academic, 1993.

[GYG19] H. Ganame, L. Yingzhuang, H. Ghazzai, and D. Kamissoko. “5G base station deployment perspectives in millimeter wave frequencies using meta-heuristic algorithms.” Electronics 8, no.11, 2019: pp. 1318.

[Hay13] S. Haykin. “Digital Communication Systems”. New York, NY, USA: Wiley, 2013.

[Hay14] S. Haykin. “Adaptive Filter Theory”. 5th ed. London, U.K.: Pearson, 2014.

[HH04] R.L. Haupt, and S.E. Haupt.” Practical genetic algorithms”. John Wiley & Sons, 2004.

[HH89] A. Duel-Hallen, and C. Heegard. “Delayed decision-feedback sequence estimation.” IEEE Transactions on Communications vol. 37, no .5 ,1989: pp. 428-436.

[HIA19] A. J. Humaidi, I. K. Ibraheem, and A. R. Ajel. “A novel adaptive LMS algorithm with genetic search capabilities for system identification of adaptive FIR and IIR filters.” Information vol.10, no.5 ,2019: p. 176.

[HK06] L. Hanzo, and T. Keller. “An OFDM and MC-CDMA Primer”. Piscataway, NJ: IEEE Press/Wiley, 2006.

[Hol73] J. H. Holland. “Genetic algorithms and the optimal allocation of trials”. SIAM J. of Computing, no. 2, 1973: pp. 88–105.

[Hol75] J. h. Holland. “Adaptation in natural and artificial systems.” Ann Arbor (1975).

[Hol92] J. Holland. “Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence”, MIT press, 1992.

[Hol92] J. H. Holland. “Adaptation in natural and artificial systems Cambridge” ,1992.

[HTD18] J. He, Z. Tang, Z. Ding, D. Wu. “Successive interference cancellation and fractional frequency reuse for LTE uplink communications”. IEEE Trans vol. 67, no. 11, 2018: pp.10528–10542.

[IEE12] “IEEE standard for local and metropolitan area networks – part 11: Wireless LAN medium access control (MAC) and physical layer (PHY),” IEEE Std. 802.11, 2012.

[IEE12] “IEEE standard for air interface for broadband wireless access systems,” IEEE Std. 802.16, 2012.

[Jal04] J. Jalden. “Maximum likelihood detection for the linear MIMO channel”. Diss. 2004.

[JBS92] M. C. Jeruchim, P. Balaban, K. S. Shanmugan. “Simulation of Communication Systems”. Plenum Press: New York, NY, USA, 1992; pp. 500–501.

[JH07] M. Jiang, and L. Hanzo, "Multiuser MIMO-OFDM for next generation wireless systems", Proceedings of the IEEE, vol. 95, no. 7, 2007: pp. 1430 -1469.

[JM91] C. Z. Janikow, and Z. Michalewicz. "An experimental comparison of binary and floating-point representations in genetic algorithms." ICGA, 1991.

[Jon75] K. A. De Jong. "An Analysis of the Behavior of a Class of Genetic Adaptive Systems". PhD thesis, University of Michigan, 1975.

[JSL97] M. J. Juntti, T. Schlosser, and J.O. Lilleberg. "Genetic algorithms for multiuser detection in synchronous CDMA." Proceedings of IEEE international symposium on information theory. IEEE, 1997.

[KCK00] S. H. Kim, K.N. Chang, and S. Kim. "A channel allocation for cellular mobile radio systems using simulated annealing". Telecommunication Systems 14, 2000: pp. 95-106.

[KD21] V. Kapoor, and S. eds. Dey. "Genetic Algorithms and Applications for Stock Trading Optimization". IGI Global, 2021.

[KH99] E. L. Knan, and L. Hanzo. "Comparative study of joint-detection and interference cancellation-based burst-by-burst adaptive CDMA schemes." Gateway to 21st Century Communications Village. VTC 1999-Fall. IEEE VTS 50th Vehicular Technology Conference (Cat. No. 99CH36324). vol. 2. IEEE, 1999.

[Kha17] M. J. O. Khafaji "Genetic-Algorithm-Aided MIMO Receiver". MSc Thesis. Poznan University of Technology, 2017.

[KK18] M. J. Khafaji, and M. Krasicki. "Genetic-Algorithm-Driven MIMO Multi-user Detector for Wireless Communications." International Conference on Dependability and Complex Systems. Springer, Cham, 2018.

[KK20] M. J. Khafaji, and M. Krasicki. "Successive-Interference-Cancellation-Inspired Multi-user MIMO Detector Driven by Genetic Algorithm." International Conference on Dependability and Complex Systems. Springer, Cham, 2020.

[KK21] M. J. Khafaji, and M. Krasicki. "Uni-Cycle Genetic Algorithm to Improve the Adaptive Equalizer Performance." IEEE Communications Letters 25, no.11 ,2021: pp. 3609-3613.

[KK22] M. J. Khafaji, and M. Krasicki. “Uni-Cycle Genetic Algorithm as an Adaptation Engine for Wireless Channel Equalizers.” Electronics 11, no.2, 2022: p. 171.

[KNK14] K. Kinoshita, M. Nakagawa, K. Kawano, and K. Murakami. “An efficient spectrum Sharing method based on genetic algorithm in heterogeneous wireless network.” International Journal of Computer Networks & Communications vol. 6., no. 5, 2014: p. 145.

[KNM11] S. Katwal, R. Nath, and G. Murmu. “A simple Kalman channel Equalizer using adaptive algorithms for time variant channel.” 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies. IEEE, 2011.

[Koz94] J. R. Koza. “Genetic Programming II. MIT Press, Cambridge, MA, 1994.

[KW19] M. J. Kochenderfer and T. A. Wheeler. “Algorithms for optimization”. Mit Press, 2019.

[Lev06] G. ed. Levitin.” Computational intelligence in reliability engineering: evolutionary techniques in reliability analysis and optimization”. Springer Science & Business Media, vol. 39 (2006).

[LL12] A. Lipowski, and D. Lipowska. “Roulette-wheel selection via stochastic acceptance.” Physica A: Statistical Mechanics and its Applications 391, no.6 ,2012: pp. 2193-2196.

[LLF11] P. Li, R.C. De Lamare, and R. Fa. ‘Multiple feedback successive interference cancellation detection for multiuser MIMO systems.” IEEE Transactions on Wireless Communications vol.10, no.8 ,2011: pp. 2434-2439

[Lou91] L. S. Louis. “Statistical signal processing: detection, estimation, and time series Analysis”. Addision-Wesley Publishing company (1991).

[Luc65] R.W.Lucky. “Automatic equalization for digital communication.” Bell System Technical Journal 44, no. 4 ,1965: pp. 547-588.

[LXN13] H. Lu, T. Xu, H. Nikookar, and L.P. Ligthart. “Performance analysis of the cooperative ZP-OFDM: Diversity, capacity and complexity.” Wireless personal communications vol. 68, no.3 (2013): pp.587-608

[MAG07] M. Jiang, J. Akhtman, F. Guo, and L. Hanzo. "Iterative joint channel estimation and multi-user detection for multiple-antenna aided OFDM systems." IEEE Transactions on Wireless Communications vol. 6, no.8 ,2007: pp. 2904-2914.

[Mir19] S. Mirjalili. "Evolutionary algorithms and neural networks." Studies in computational intelligence. Vol.780. Berlin/Heidelberg, Germany: Springer, 2019.

[MM14] H. Merabti and D. Massicotte. "Nonlinear adaptive channel equalization using genetic algorithms." 2014 IEEE 12th International New Circuits and Systems Conference (NEWCAS). IEEE, 2014.

[MP11] T. Mostajabi, and J. Poshtan. "IIR filter design using time and frequency responses by genetic algorithm for system identification." 2011 1st International eConference on Computer and Knowledge Engineering (ICCKE). IEEE, 2011.

[MQA16] U. Mehboob, J. Qadir, S. Ali, and A. Vasilakos. "Genetic algorithms in wireless networking: techniques, applications, and issues". Soft Computing, vol. 20, no. 6 (2016): pp. 2467-2501.

[MWW19] T. Mao, Q. Wang, Z. Wang, and S. Chen. "Novel index modulation techniques: A survey." IEEE Communications Surveys & Tutorials 21, no.1, 2019: pp. 315-348.

[NYH02] S. X. Ng, K. Yen, and L. Hanzo. "TTCM assisted genetic-algorithm aided reduced-complexity multiuser detection". Electron. Lett., vol. 38, no. 14, 2002: pp. 722–724.

[OEB16] T. Oda, D. Elmazi, Barolli, A. Barolli, S. Sakamoto, L. Barolli, and F. Xhafa. "A genetic algorithm-based system for wireless mesh networks: analysis of system data considering different routing protocols and architectures". Soft Comput 20, 2016: pp. 2627–2640.

[OGS05] S. S. Owais, P. Gajdos, and V. Snásel. "Usage of Genetic Algorithm for Lattice Drawing." CLA, 2005.

[OHY07] P. S. Oliveto, J. He, and X. Yao. "Time complexity of evolutionary algorithms for combinatorial optimization: A decade of results." International Journal of Automation and Computing vol. 4, no .3, 2007.

[OSY11] K. Obaidullah, C. Siriteanu, S. Yoshizawa, and Y. Miyanaga. "Evaluation of genetic algorithm-based detection for correlated MIMO fading channels." 2011 11th International Symposium on Communications & Information Technologies (ISCIT). IEEE, 2011.

[Özk18] O. ÖZKARACA. "A review on usage of optimization methods in geothermal power generation." Mugla Journal of Science and Technology vol. 4, no.1 (2018): pp.130-136.

[PFL00] D. P. Palomar, J.R. Fonollosa, and, M.A. Lagunas. "MMSE Joint Detection in frequency-selective wireless communication channels for DS-CDMA systems." 2000 IEEE Sixth International Symposium on Spread Spectrum Techniques and Applications. ISSTA 2000. Proceedings (Cat. No. 00TH8536). vol. 2. IEEE, 2000.

[PGN04] A. J. Paulraj, D. A. Gore, R. U. Nabar, and H. Bölcskei, "An overview of MIMO communications - A key to gigabit wireless," Proc. IEEE, vol. 92, no. 2, 2004: pp. 198-218.

[Pro01] J. G. Proakis. "Digital Communications", 4th ed. New York, NY, USA: McGraw-Hill, 2001.

[PSB12] J. G. Proakis, M. Salehi, and G. Bauch. "Contemporary Communication Systems Using MATLAB". Scarborough, ON, Canada: Nelson Education, 2012.

[RC98] G. G. Raleigh and J. M. Cioffi, "Spatio-temporal coding for wireless communication," IEEE Trans. Commun., vol. 46, no. 3, 1998: pp. 357–366.

[RCA16] S. Ruchi, T. J. Choi, and C. W. Ahn. "A genetic algorithm with location intelligence method for energy optimization in 5G wireless networks". Discrete Dynamics in Nature and Society, 2016.

[Rec73] I. Rechenberg. "Evolutions strategies." Optimierung technischer Systeme nach Prinzipien derbiologischen Evolution, 1973.

[Sch95] H. P. Schwefel. "Evolution and Optimum Seeking". Wiley, New York, 1995.

[SD08] S. N. Sivanandam, and S. N. Deepa." Introduction to genetic algorithms. Springer, Berlin, Heidelberg, 2008: pp. 15-37.

[Sim13] D. Simon. "Evolutionary optimization algorithms". John Wiley & Sons, 2013.

[Sim14] H. Simon. “Adaptive Filter Theory”. 5th ed.; Pearson Education Ltd.: Harlow, UK, 2014.

[SK15] E. N. Singh, and E.V. Kumar. “Linear and Non-Linear decoding Techniques in Multi-User MIMO Systems.” A Review 4, no.5 (2015).

[SOZ10] A. Sibille, C. Oestges, and A. Zanella. “MIMO: from theory to implementation” Academic Press, 2010.

[SSC10] S. Sen, N. Santhapuri, R.R. Choudhury, and S. Nelakuditi, “Successive interference cancellation: A back-of-the-envelope perspective.” Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks. 2010.

[Tad11] F. O. Tade. “Receiver Architectures for MIMO Wireless Communication Systems based on V-BLAST and Sphere Decoding Algorithms”. Diss. 2011.

[Tel99] I. E. Telatar. “Capacity of multi-antenna Gaussian channels”. EuropeanTrans. Telecomm., vol. 10, no. 6, 1999: pp. 585-595.

[THL10] T. H. Tan, Y.F. Huang, and F.T. Liu. “Multi-user detection in DS-CDMA systems using a genetic algorithm with redundancy saving strategy.” Int J Innov Comput Inf Control 6, no.8, 2010: pp. 3347-3364.

[TS17] 3GPP, TS 45.005; GSM/EDGE Radio Transmission and Reception, Release 14. European Telecommunications Standards Institute: Sophia Antipolis Cedex, France, 2017.

[TV05] D. Tse, and P. Viswanath. “Fundamentals of wireless communication”. Cambridge university press, 2005.

[UKC15] M. Umair, M. A. Khan and M. A. S. Choudry. “Island genetic algorithm-based MUD for MC-CDMA system”. International Conference on Information and Communication Technologies (ICICT), 2015, pp. 1-6.

[Ver98] S. Verdu. “Multiuser Detection”. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[Vik16] P. A. Vikhar. “Evolutionary algorithms: A critical review and its future prospects.” 2016 International conference on global trends in signal processing, information computing and communication (ICGTSPICC). IEEE, 2016.

[Vos99] M. D. Vose. “The Simple Genetic Algorithm: Foundations and Theory (Complex Adaptive Systems)”. Cambridge, MA: MIT Press, 1999.

[VPE01] P. Vandenameele, L. V. D. Perre, and M. Engels. “Space Division Multiple Access for Wireless Local Area Networks”. London, U.K.: Kluwer, 2001.

[VR12] L. R. Vega, and H. Rey. “A rapid introduction to adaptive filtering”. Springer Science & Business Media, 2012.

[WAC04] A. Wolfgang, N. N. Ahmad, S. Chen, and L. Hanzo. “Genetic algorithm assisted error probability optimisation for beamforming”. Inst. Electr. Eng. Electron. Lett., vol. 40, no. 5, 2004: pp. 320–322.

[Wes01] K. Wesolowski.” Mobile communication systems”. John Wiley & Sons, Inc., 2001.

[Wes09] K. Wesolowski. “Introduction to digital communication systems”. John Wiley & Sons, 2009.

[WFG98] P. W. Wolniansky, G.J. Foschini, G.D. Golden, and R.A. Valenzuela.” V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel.” 1998 URSI international symposium on signals, systems, and electronics. Conference proceedings (Cat. No. 98EX167). IEEE, 1998.

[Whi89] L. D. Whitley. “The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best”. *Icga*. vol. 89. No. Jun: pp 116-121.

[Wid66] B. Widrow. “Adaptive filters I: fundamentals”. Systems Theory Laboratory, Stanford Electronics Laboratories, Stanford University, 1966.

[WLA98] X. F. Wang, W.S. Lu, and A. Antoniou. “A genetic-algorithm-based multiuser detector for multiple-access communications.” 1998 IEEE International Symposium on Circuits and Systems (ISCAS). Vol. 4. IEEE, 1998.

[WQK14] M. Wildemeersch, T.Q. Quek, M. Kountouris, A. Rabbachinand, and C.H. Slump. “Successive interference cancellation in heterogeneous networks.” IEEE Transactions on Communications, vol. 62, no. 12, 2014: pp.4440-4453

[Yan09] L. L. Yang. "Using multi-stage MMSE detection to approach optimum error performance in multiantenna MIMO systems." 2009 IEEE 70th Vehicular Technology Conference Fall. IEEE, 2009.

[Yan10] X. S. Yang. "Engineering optimization: an introduction with metaheuristic applications". John Wiley & Sons, 2010.

[YG10] L. Yuan and C. Zhi-Gang. "The research of blind equalization algorithm based on clonal genetic algorithm and neural network." 2010 International Conference on Computer Application and System Modeling (ICCASM 2010). vol. 13. IEEE, 2010.

[YHL13] C. Yang, J. Han, Y. Li, and X. Xu. "Self-adaptive Genetic algorithm-based MU-MIMO scheduling scheme." 2013 15th IEEE International Conference on Communication Technology. IEEE, 2013.

[ZH14] X. Zhang, M. Haenggi. "The performance of successive interference cancellation in random wireless networks". IEEE Trans. Inf. Theory vol. 60, no. 10, 2014: pp. 6368–6388.

[ZTB09] E. Zitzler, L. Thiele, and J. Bader. "On set-based multiobjective optimization." IEEE Transactions on Evolutionary Computation vol.14, no. 1 (2009): pp. 58-79.

[ZYC19] Z. H. Zhou, Y. Yu, and C. Qian." Evolutionary learning: Advances in theories and algorithms". Singapore: Springer Singapore, 2019.