A HUMAN COMMUNITY-BASED GENETIC ALGORITHM MODEL (HCBGA)

NAGHAM AZMI QASIM AL-MADI

UNIVERSITI SAINS MALAYSIA

2009

A HUMAN COMMUNITY-BASED GENETIC ALGORITHM MODEL (HCBGA)

by

NAGHAM AZMI QASIM AL-MADI

Thesis submitted in fulfillment of the requirements for the Degree of Doctor of Philosophy

Nov 2009

DEDICATION

To my Dearest Mother and Father

Abla Jarrar & Azmi AL-Madi

To My Dearest Husband

Khader AL-Rawajfih

To My Lovely Two Sons

Amr & Samer

To my Lovely Two Sisters

Domer & Fanan

To My Lovely Two Brothers

Mohammad & Ahmed

ACKNOWLEDGEMENTS

IN THE NAME OF
ALLAH
THE ALL-COMPASSIONATE, ALL-MERCIFUL

"Read in the name of thy Lord who createth, createth man from a clot. Read and the Lord is the most Bounteous, who teacheth by the pen, teacheth man that which he knew not" (Alalak: 1-5)

*At First, From My Deepest Heart I am Thankfull to almighty **ALLAH** for everything he gave me.*

*My great thanks to my beautiful angles, my cute, loveliest, dearest sons **Amr** and **Samer** who mentioned me encouragement to be able to study for them, and who waited a long time for me to be just with them. **Amr, Samer**, I Love You.*

*From my deepest heart, I proudly would like to express my deepest gratitude and appreciation and thank my Supervisor **Dr. Ahamad Tajudin Khader** for his wonderful and great encouragements and endless supports for this challenge. His patience, guidance, motivation and comments have supported me with a valuable and strong knowledge during this hard mission.*

*I am so grateful to the Dean of Computer Science in University Sains Malaysia, Prof. **Rosni Abdullah**. Her supporting and high quality of managing the school has given us the opportunity to have the best research environment to achieve our goals.*

*Special deeply thanks to my lovely dear parents, my kind hearted father **Prof. Azmi Al-Madi** and my sweet lovely mother **Abla Jarrar** with their eternity love. From my deepest heart I would like to thank them for how they are in eager to reach me to where I am now, I will never be what I am now as in my personality or a practically life without their prayers, supports, love and sincerity. Their confident that they gave it to me makes me more responsible towards what I want to achieve in my life. I am proud to be their daughter.*

*My sincere thanks also goes to my lovely kind hearted husband, **Khader AL-Rawajfih**, for never let me down, for his encouraging and being beside me during my journey. Without his prayers, Love, patience, understanding and supporting I wouldn't be able to achieve my goal.*

*Grateful thanks to my lovely and wonderful dearest siblings, **Domer, Mohammed, Ahmed and Fanan** for their high support and understanding during my studies. With all their encouragements, prayers and love they made me reach to where I am. I am also grateful to my brother in law **Imad** for his encouragements. And to my sweet niece and nephew, lovely **Miran & Ahmed** your smiles cheered me up.*

I am also thankful to my lovely father in law and mother in law ***Khaleel AL-Rawajfih*** *and* ***Samia AL-Faoury*** *for their understanding, loving, and encouraging me. Besides, I would like to express thanks to my wonderful sisters and brothers in law* ***Fatmeh, Aml, Amani, Ammal, Wafa', Rashid, Ashraf, Hiba, Hana and Rakan*** *for encouraging me.*

I would also like to express my deep feelings to ***all relatives*** *and* ***friends*** *especially* ***Khalid Suwais****, for all their prayers, encouragement they gave to me.*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

EAs:	Evolutionary Algorithms
GAs:	Genetic Algorithms
SGA:	Standard Genetic Algorithm
GSGA:	Gendered Standard Genetic Algorithm.
BGSGA:	The Balanced Gender Standard Genetic Algorithm.
HCBGA:	The Human Community Based GA
CGA:	Cellular Genetic Algorithm
TBGA:	Terrain Based Genetic Algorithm
IGA:	Island Genetic Algorithm Model
RBEA:	Religion Based EA Model
TSP:	The Traveling Salesman Problem

KOMUNITI MANUSIA BERASASKAN MODEL ALGORITMA GENETIK (HCBGA)

ABSTRAK

Sebagai satu model gelintaran, Algoritma Genetik (GA), telah membuktikan kejayaannya dalam banyak apikasi. Walau bagaimanapun, beberapa penyelidik menyatakan bahawa GA mempunyai "convergence" yang perlahan. Keperlahanan ini berpunca daripada kerawakan dalam kebanyakan operasinya. Oleh itu, ramai penyelidik terkini telah menggunakan populasi berstruktur dalam GA untuk mengurangkan kerawakan seperti model algoritma genetik pulau (IGA), model algoritma genetik bersel (CGA) dan model lain.

Tesis ini menyediakan satu pendekatan baru untuk populasi berstruktur dalam Algoritma Genetik, berdasarkan kelaziman, tingkah laku dan corak komuniti manusia. Antaranya termasuklah gender, umur, generasi, perkahwinan, kelahiran dan kematian. Oleh itu, model ini dinamakan model Komuniti Manusia Berasaskan Alogritma Genetik (HCBGA). Model ini merupakan satu evolusi daripada Alogitma Genetik mudah (SGA). Genderisasi diaplikasikan pada model ini, diikuti dengan imbangan gender, dan akhirnya dimasukkan komuniti manusia berasaskan peraturan. Siri eksperimen dijalankan pada tiga masalah yang berbeza:: masalah Knapsack, fungsi pertama De

Jongs' (F1) dan masalah jurujual yang berjalan (TSP). Masalah ujian ini meliputi masalah permutasi dan bukan-permutasi, berserta dengan masalah selanjar dan bukan-selanjar, yang memberikan keputusan yang lebih tepat. Prestasi HCBGA didapati lebih baik daripada SGA dan dua yang lain.

A HUMAN COMMUNITY BASED GENETIC ALGORITHM MODEL (HCBGA)

ABSTRACT

As a general search model, Genetic Algorithm (GA) has proved its success in many applications. However, several researchers argue that GA has slow convergence. This shortfall is due to the randomness in most of its operations. Hence, recently researches have employed structured populations in GA to reduce this randomness, such as in the island genetic algorithm model (IGA), cellular genetic algorithm model (CGA) and other models.

This thesis provides a new approach to the structured population in Genetic Algorithm, based on the custom, behavior and pattern of human community. This includes gender, age, generation, marriage, birth and death. As such, this model is named the Human Community Based Genetic Algorithm (HCBGA) model. This model is an evolution of the simple Genetic Algorithm (SGA). Genderization is applied to this model, followed by gender balancing, and finally the human community based rules are included. Series of experiments were carried out on three problems of different nature: the Knapsack problem, De Jongs' first function (F1) and traveling salesman problem (TSP). These test problems cover permutation and non-permutation problem, together

with continuous and non-continuous problem, hence give better results. The performance of the HCBGA is found to be far better than the SGA and two advanced models, which are the island genetic algorithm model (IGA) and cellular genetic algorithm model (CGA) as this proposed model obtains better optimal maxima or minima, besides maintaining the diversity.

أعوذ با   من الشيطان الرجيم

بسـم اللـه الرحمـن الرحيـم

"Prohibited to you (for marriage) are: your mothers, daughters, sisters; father's sisters, mother's sisters; brother's daughters, sister's daughters; foster-mothers (who gave you suck), foster-sisters; your wives' mothers; your step-daughters under your guardianship,

born of your wives to whom ye have gone, no prohibition if ye have not gone in; (those who have been) wives of your sons proceeding from your loins; and two sisters in wedlock at one and the same time, except for what is past; for Allah is Oft-Forgiving, Most Merciful." (The Holy Qura'an, Soura (4), Aya (23)).

CHAPTER ONE

INTRODUCTION

1.1 Background

The complexity of nature makes many philosophers argue about how it was created. However, the remarkable theory that Darwin came up with in 1872 is able to explain the existence by the means of Evolution from Natural Selection (Darwin, 1872).

"I have called this principle, by which each slight variation, if useful, is preserved, by the term Natural Selection." (Darwin, 1859, 1872).

Darwin further clarified that the survival of individuals depends on better fitness among them, as they succeed to adapt themselves to their environment better than others (Darwin, 1859, 1872).

The Darwinian theory of evolution has inspired many researchers to develop various biology-related models and approaches. These approaches and models provide new ways to view and solve problems. In the field of artificial intelligence (AI), a new subfield was created to accommodate all these models and approaches (Madar, Abonyi & Szeifert, 2003). It is called the evolutionary computation (EC) and it mimics the Darwinian evolutionary process (Bäck, Hammel & Schwefel, 1997; Eiben & Smith, 2003).

1.2 Evolutionary Computation (EC)

Evolutionary computation (EC) is a class of algorithms within computer science that attempts to solve complex problems (De Jong, 2006). EC uses a simulated evolution in some degree, that these algorithms are able to evolve the population of potential solutions in a manner such that weaker solutions are removed and replaced with stronger and better solutions (Eiben & Smith, 2003). This process is done by mimicking the processes of Darwinian evolution (Back et al., 1997; Eiben & Smith, 2003; Blum & Roli, 2003).

Different approaches have appeared and developed as subclasses of EC named evolutionary algorithms (EA) (Eiben & Smith, 2003; Blum & Roli, 2003). EA is a general term for various computational techniques, mostly based on biological life of natural world. Although these approaches are similar in terms of the basic assumptions where they are inspired by the same principles of natural evolution; they differ in their strategies. These include:

(a) Evolutionary programming (EP) developed by Lawrence Fogel (Eiben & Smith, 2003; Blum & Roli, 2003) which focuses on optimizing continuous functions without recombination,

(b) Evolutionary strategies (ES) developed in Germany by Ingo Rechenberg and Hans-Paul Schwefel (Eiben & Smith, 2003; Blum & Roli, 2003) which focuses on optimizing continuous functions with recombination,

(c) Genetic algorithms (GA) developed in USA by J. H. Holland, which focuses on optimizing general combinatorial problems (Holland, 1975; Eiben & Smith, 2003; Blum & Roli, 2003) and

(d) Genetic programming (GP) championed by Koza (Koza, 1990, 1994; Banzhaf, Nordin, Keller & Francone, 1998; Eiben & Smith, 2003; Blum & Roli, 2003; Kicinger, Arciszewski & De Jong, 2005) which which focuses on evolving programs.

Various problems have found solutions by using the evolutionary approach. These include applications in telecommunication (data, image compression and noise filtering), financial and market forecasting (stock market and exchange rates) and optimization problems (such as wire routing, scheduling, traveling salesman, image processing, engineering design, parameter fitting, computer game playing, knapsack problems, and transportation problems) (Fogel, Back & Michalewicz, 2000; Liao & Sun, 2001; Eiben & Smith, 2003; Bagheri & Deldari, 2006).

1.3 Genetic Algorithm (GA)

GA is one of the approaches developed from EA in the 1960's (Holland, 1975). It has attracted many researchers due to its general purpose algorithm (Whitley, 1994; Miller & Todd, 1995; Bäck et al., 1997; Krink, Mayoh & Michalewicz, 1999; Nobel, 1999; Hemelrijik, 1999; Thomsen, Rickers & Krink, 2000).

GA mimics the natural biology process particularly in the human genes. It is based on the survival of the fittest the better genes have a higher chance to survive (Liao & Sun, 2001; Bagheri & Deldari, 2006). So, the solution will improve all the time in which the better ones stay and the worst are removed. A solution to a problem in GA is represented as a genome (or chromosome) (Holland, 1975, 1992; Zheng & Kiyooka, 1999; Liao & Sun, 2001). This genome is a string with a fixed bit-length. Figure 1.1 represents the simple standard genetic algorithm (SGA) evolution flow.

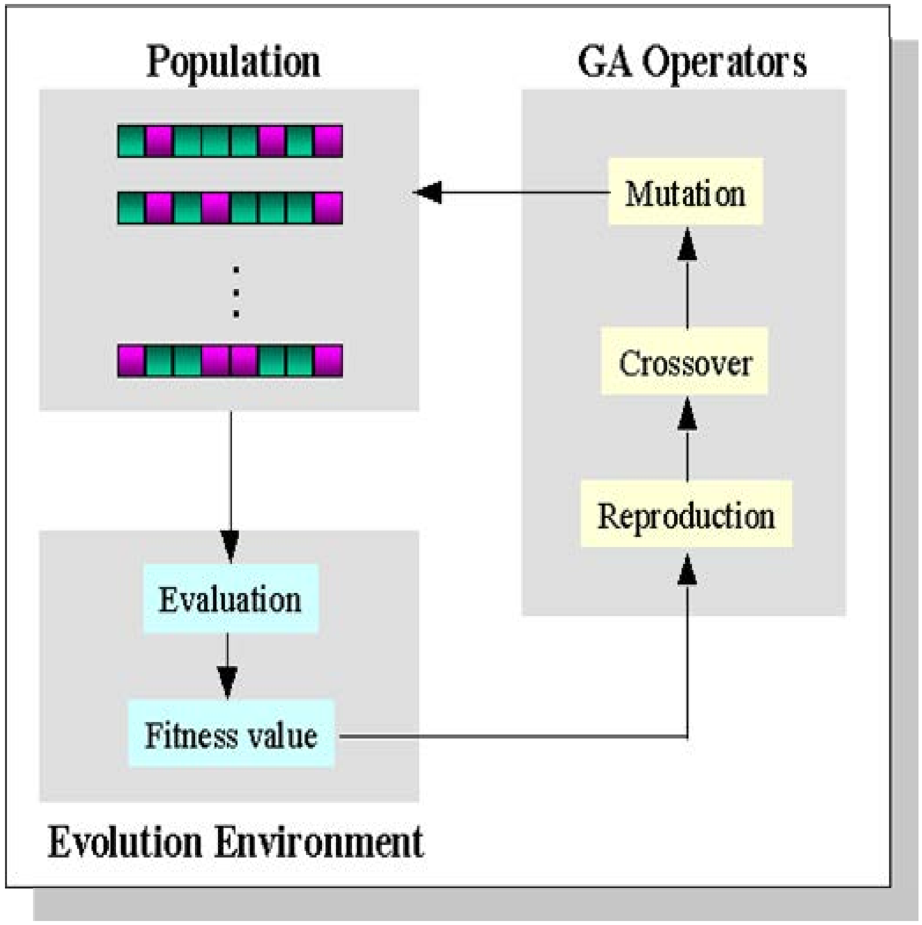

Figure 1.1: The evolution flow of the SGA (Liao & Sun, 2001)

Initially, a population size of the first generation is generated randomly. Next a fitness function is used to evaluate each solution in this first generation. Better solutions will have higher evaluations. Then, some genetic operations reproduction, crossover and mutation are employed to generate the next generation based on these evaluations. This procedure is performed iteratively until the optimal solution(s) is (are) found or the time allotted for computation ends.

The usage of GA began by solving classical problems such as the traveling salesman problem and the n queens' problem (Bäck, 1996; Zheng & Kiyooka, 1999; Liao & Sun, 2001). Years later, GA grew rapidly as it increased its applications to optimize complex scheduling problems and spatial layout (Dalton, 2007). GA is mostly used to approximate solutions for NP-hard problems, to solve minimization and maximization problems, and to evaluate difficult classical optimization problems (Holland, 1975; Beasley, Bull & Martin, 1993; Zheng & Kiyooka, 1999). Furthermore, GA is efficient when applied on combinatorial optimization problems including several common computer science problems such as the knapsack problems (Zheng & Kiyooka, 1999).

In spite of the successes of GA; it also suffers from some weaknesses such as the premature convergence and loss of diversity (Affenzeller & Wagner, 2003; Sanchez-Velazco & Bullinaria, 2003; Ursem, 2003; Gustafson & Burke, 2006; Vrajitoru, 2008). Researchers have been trying to overcome these drawbacks in many different ways. This thesis discusses a new strategy to overcome some of the weaknesses.

1.4 Problem Statement

The simple SGA works randomly in selecting parents which means there are no constraints in choosing two individuals to mate together (Sanchez-Velazco & Bullinaria, 2003). This has made the SGA to fall into what is called a premature convergence, in which the population of a problem converges too early which means finding a good solution too fast, and this results in suboptimal solution. This leaves the possibility of searching for other solutions that could be better (Affenzeller & Wagner, 2003, 2004; Wagner & Affenzeller, 2005). Furthermore, the randomness of SGA may generate similar or new identical individuals; hence, the new generations do not span the entire search space effectively for optimal solutions. This affects the SGA and leads to a loss of diversity which causes the algorithm to fall into local minima or local maxima when searching for optimal solutions (De Jong, 1975; Goldberg, 1989; Krink et al., 1999; Ursem, 2002; Riget, & Vesterstrøm, 2002; Wagner & Affenzeller, 2005; Skolicki & De Jong, 2005).

Previous works that were intended to reduce this randomness by structuring the population with some control on how individuals interact, have come out with better performance than SGA (Thomsen et al., 2000; Goldberg, 2002; Thomsen & Krink, 2002; Gustafson & Burke, 2006; Vrajitoru, 2008). The main works include cellular genetic algorithms (CGA) (Whitley, 1994; Alba & Dorronsoro, 2004; Alba, Dorronsoro, Giacobini & Tomasini, 2006; Nebro, Durillo, Luna, Dorronsoro & Alba, 2007), island genetic algorithms (Back et al., 1997; Enrique, Mario, Marco & Sergio, 2002; Skolicki & De Jong, 2005; Gustafson & Burke, 2006), patchwork genetic algorithms (Krink et al., 1999; Krink & Ursem, 2000), terrain based genetic algorithms (TBGA) (Krink & Ursem, 2000; Gordon & Thein, 2004), religion-based genetic algorithms (RBGA) (Thomsen et al., 2000; Thomsen & Krink, 2002) and others.

In line to these works, this thesis presents an approach of structuring the population based on human communities, to further improve the weakness of SGA.

1.5 Motivation

The motivation to this thesis is the human communities which are governed by rules and regulations. These rules and regulations could be religions, customs and norms of the community or laws of the country. These rules govern most human communities and they are quite common across the communities. The most important relationship in the human communities is the relationship between males and females. The union between males and females is normally formalized as marriage in most communities. Some examples of these rules and regulations are: marriage is normally between a male and a female; there is no marriage between siblings and there is no marriage with uncles or aunts. These rules and regulations have been in existence since the beginning of mankind and the success of these rules and regulations are self evident.

The mankind is the most successful example of the evolutionary process and their numbers are rapidly increasing. This human community with its rules and regulations between them has inspired us to construct a structured population for GA. This motivation provides the basis for the new model.

1.6 Research Scope

This work in general involves humanizing the GA's population. In particular we are concerned with the process of marriage in human community. As such, the research scope will include those elements related to marriage. This includes genderization, relationship, marriage and aging. Genderization will include portioning the population into males and females. Relationship between family members will be maintained. This will include relationship between parent and child, between siblings, and between child and their uncle and aunt. Basic rules of marriage are also implemented. Finally, aging is implemented in three major blocks: youth, parents and grandparent.

The addition of rules of marriage, on top of the genderization and balancing, causes the algorithm to search more effectively, however, it takes longer computation time. Hence, the effectiveness of this algorithm is achieved at the expense of speed and time, which are worth compromising.

1.7 Research Questions

In trying to tackle the SGA problems stated previously, the following research questions were formulated:

1. How to construct and build a structured population model to enhance the performance of SGA by relying on the functions of a community?
2. What are the constraints that control the selection and crossover of individuals, in order to preserve the genetic strength throughout the generation?
3. Does the enhanced model perform better than other genetic algorithm models?

1.8 Research Objectives

Accordingly, our research objectives based on the research questions are as follows:

1. To structure the SGA's population to mimic the human communities towards maintaining diversity and preventing premature convergence:
   a) To divide the population into male and female while keeping the balance between them.
   b) To partition the age of a population into three blocks: child, parent, and grandparent.
   c) To set rules and regulations governing marriage.
2. To compare the performance towards optimal solutions of the structured model against the standard model (SGA), and against several advanced models such as the cellular genetic algorithm (CGA) model and the island genetic algorithm (IGA) model by using several test problems. This can be observed through preserving the diversity in the population which leads to preventing the search to fall in a premature convergence.

1.9 Research Contributions

In this thesis our main contribution is constructing and building an enhanced and structured population model for SGA to increase its performance. The proposed HCBGA model will adapt the Islamic rules as a case study to measure the effectiveness of this model in achieving better solutions for the problems mentioned in section 1.4.

The contributions can be detailed as follows:

1. A new structured population model based on human communities.
2. Parent selection is modeled after the selection of mates in a marriage.
3. A balanced gender based population.

1.10 Research Methodology

The research is carried out in a series of steps, beginning with SGA and ending with the new advance model.

Firstly, the SGA was tested with the three chosen test problems: the knapsack problem, the De Jong's first test function, and the traveling salesman problem. This forms the basis for the rest of the experiments.

In the second stage, genderization is introduced to SGA. The population is divided into male and female. This is called the genderized SGA (GSGA). The behavior of GSGA is tested with the same three test problems and results are compared against the ones of the simple SGA.

In the third stage, a balance between the male and female in the population is introduced to enhance GSGA. This mechanism will preserve the diversity of the SGA's population. This is called balanced gender SGA (BGSGA). The behavior of this mechanism is also tested with the same three test problems and the obtained results are compared against the results of GSGA, as well as simple SGA.

In the next step, the final model is introduced. It includes relationship, marriage, and aging besides the previous enhancements. Its behavior is tested, once again with the same three test problems and the obtained results are compared against all the results of previous mechanisms – results of GSGA, BGSGA and simple SGA.

Then finally, we compare the completed model against two advanced models namely CGA and IGA.

1.11 Outline of this Thesis

The rest of the thesis is organized in accordance with the objectives mentioned above:

Chapter Two covers some basic information presented to interested readers for a better understanding of this thesis. It also introduces basic knowledge of different techniques in EA. In addition, GA will be described more specifically in this chapter. This is followed by a literature review of a number of different types of GA models and approaches developed to improve the SGA.

Chapter Three presents the research methodology and the major steps involved in developing the methodology. This chapter will also cover how the model evolved from the original SGA to the HCBGA.

In Chapter Four, three different test problems: the knapsack problem, the De Jongs' first function and the traveling salesman problem applied on the enhanced model will be presented and discussed.

In Chapter Five implementation results are presented and analyzed. In addition, comparisons between the enhanced model and existing genetic algorithm models which are the SGA, CGA and the IGA will also be presented. Finally, Chapter Six contains the conclusion and discussion of future work.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter begins with an overview of EA. This is followed by a brief introduction to the main techniques of EA. A more detailed coverage of GA is included due to the importance of the material to this thesis. This is followed by a review on the major works in GA. In finalizing this chapter a discussion on the development of the HCBGA model is included.

2.2 An Overview of Evolutionary Algorithm (EA)

From the Darwinian evolution theory, many researchers have taken the opportunity to invent many optimization methods to provide solutions to complicated problems (Uresm, 2003). Early ideas relating to EA were proposed during the 40’s (Blickle, 1996; Uresm, 2003). Later, Fogel (EP), Rechenberg and Schwefel (ES), John Holland (GA) and Koza (GP) were considered as the fathers of modern EA (Blickle, 1996; Eiben & Smith, 2003; Ursem, 2003). In the early 90’s, EA and EC were introduced as unifying terms among the optimization techniques which mimic the biological evolution (Blickle, 1996; Ursem, 2003).

EC is the process of computing evolution where it models the processes of natural evolution. The subset of EC is called an EA, which is the algorithm or what is called the general scheme of the EC. An EA is a stochastic search for an optimal solution to a given problem. Kicinger et al., (2005) emphasize that EA is a search algorithm which has a population as a base for its search, and simulates natural organisms by iterative processes of selection, reproduction and variation (Grosan, Abraham & Nicoara, 2005; Kicinger et al., 2005).

These days, EA is considered the most popular techniques for complex problems which do not need a single solution, but needs a population of potential solutions to choose the best among them (Whitely, 1994; Grosan et al., 2005). General steps of EA will be detailed in the following section.

2.3 General Steps of the Evolutionary Algorithm (EA)

Figure 2.1 illustrates the general schema of an EA as a flowchart. There are several major steps in EA. At the beginning, the population is initialized only once. Then the algorithm passes through a loop. This loop is repeated until a termination condition is reached, and each repetition is known as a generation. There are several steps in each loop: selection, recombination, mutation, and replacement.

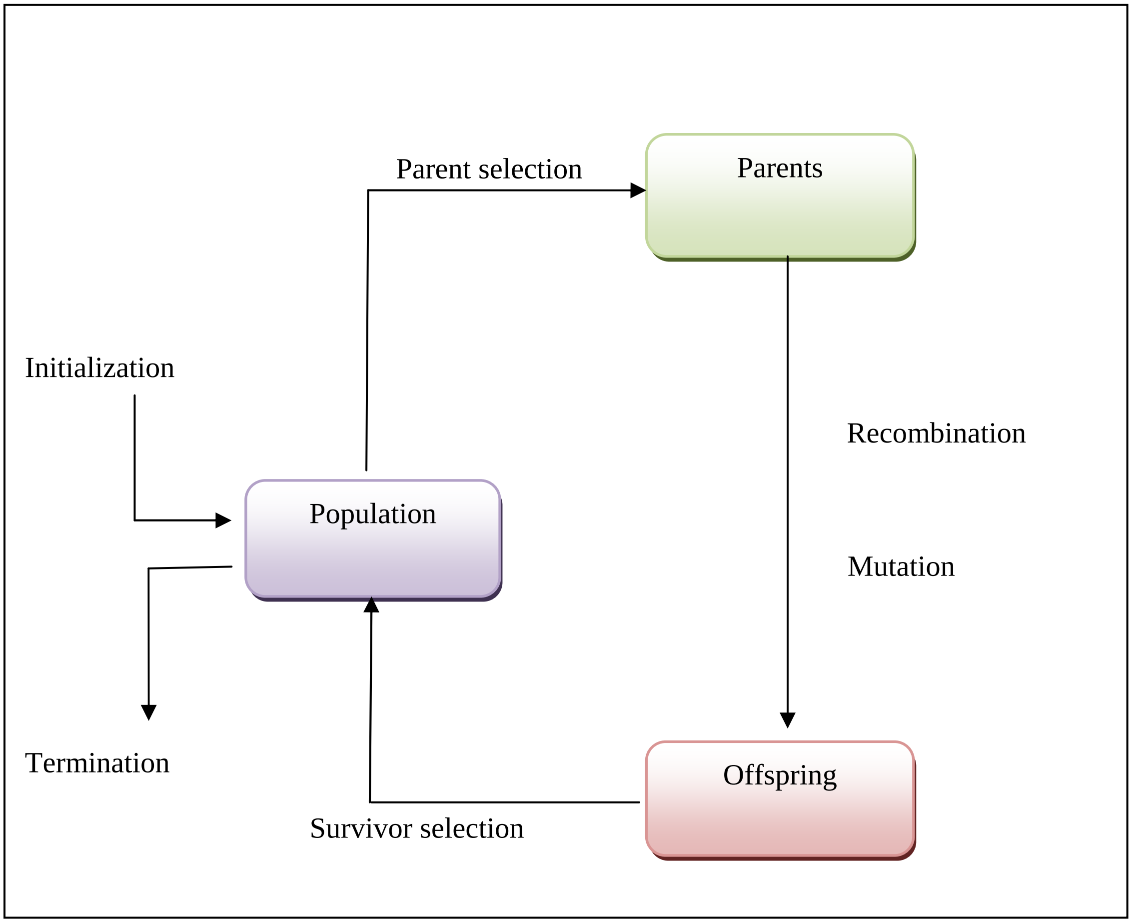

Figure 2.1: The general scheme of an EA as a flow chart (Eiben & Smith, 2003)

The evolutionary search process is influenced by the following main components of EA (Eiben & Smith, 2003):

- Representation: The link between the real world and the EA world. It is a bridge between the original problem context and the problem-solving space where evolution takes place. Representation encodes the objects forming possible solutions to a certain problem.

- Evaluation function: it is commonly known as the fitness function. It is a function or procedure that assigns a quality measure to individuals.

- Population: it is a set of individuals. The role of the population is to hold the possible solutions of a certain problem.

- Parent selection mechanism: it is also called mating selection, which is to allow better individuals to become parents of the next generations. It is a process whereby parents are selected for recombination purposes, based on their fitness. Fitter individuals get higher chances to be selected. This step is considered the most important step of the EA steps due to the fact that choosing the most suitable individuals which has a high fitness is very important. There are several different methods in the selection step the stochastic uniform sampling, the tournament selection method, the fitness proportional selection and the fitness ranking selection method (Madár et al., 2004).

- Variation operators: to create new individuals from the older ones. They are divided into two types based on the number of objects they take as inputs (arity).

- Recombination: often called as crossover. It is a binary variation operator which applies two objects as inputs. Some information from each individual is interchanged between them, and then merges them to produce one or two new individuals (offspring).

- Mutation: a unary variation operator which applies one object as input. It is applied to an individual who then delivers a slightly modified child or offspring of it. Just as in real life, no individual inherits all the genes exactly as they are from its parents. There should be some kind of differences between the individuals. A slight change on each offspring may occur. Accordingly, part of the code of the new offspring's chromosome is randomly changed. By this, new different individuals are produced. This operator is known as mutation.

- Survival of the fittest (replacement): it is often called the replacement. In this step, selection of the best quality individuals is made, similar to the selection mechanism. The only difference is, this survival selection mechanism is called after the creation of the offspring. The selected individuals form the new generation.

- Initialization: it is simple in most EA applications. The first population is initialized by generating individuals randomly.

- Termination: There are two termination conditions – either the problem has a known optimal fitness level or a condition is selected to stop the algorithm. In the first condition, when the problem has a known optimal fitness, attaining this level becomes the stopping condition. However, EA is stochastic, thus there is no guarantee that EA will reach an optimum, which may lead to an infinite loop and the algorithm may never stop. In the second option, a condition is chosen to stop the algorithm. The commonly used conditions are: the maximally allowed CPU time elapses, the total number of fitness evaluations reaches a given limit, or given number of generations or fitness evaluations, the fitness remains below a threshold value, or the population's diversity drops under a given threshold.

2.4 Evolutionary Algorithm (EA) Techniques

In the following subsections, a brief description of each technique in EA will be discussed.

2.4.1 Genetic Algorithm (GA)

GA has become the most popular type among EA techniques in terms of its simple framework in solving complex search problems (Eiben & Smith, 2003; Blum & Roli, 2003). In GA, the recombination operator is highlighted over the mutation operator. This algorithm uses fitness proportionate selection, bitstrings of a fixed length to represent the individuals and one-point crossover.

Taking into consideration that GA is the scope of this thesis, the GA technique will be presented in more detail in Section 2.5.

2.4.2 The Genetic Programming (GP)

GP can be viewed as a GA with tree representation to represent the individuals. This technique was developed by Koza (Koza, 1990; Banzhaf et al., 1998; Eiben & Smith, 2003; Blum & Roli, 2003). It uses tournament selection as a preferred selection scheme, and crossover with no mutation. For example, the population of GP consists of computer programs (Wright, 2002), hence GP evolves the computer programs (Walker, 2001).

As GP is represented by a tree representation, this makes it much more flexible, and it has the same operators as GA. The main difference between GP and GA is in the representation scheme used. GA uses string representations whereas GP represents individuals as executable programs. In GP, each evolved program is executed for each generation, to measure its performance within the problem domain. The results are then used to determine the fitness of that program.

2.4.3 Evolution Strategies (ES)

ES is considered a search algorithm which basically focuses on gene mutation, where the recombination role in the search is mainly in adapting mutation (Pérez-Fructuoso, Garcia, Berlanga & Molina, 2007). This strategy was developed in Germany by Rechenberg and Schwefel (Whitley, 1994; Bäck et al., 1997; Jones, 1998; Eiben &

Smith, 2003; Blum & Roli, 2003; Kicinger et al., 2005). ES have been traditionally used for optimization problems with real-valued vector representations (Blickle, 1996; Jones, 1998). ES tends to use more direct representations (Bäck, Hoffmeister & Schwefel, 1991), where each individual is represented by a one-dimensional vector. In this algorithm, the mutation is highlighted over recombination (Whitley, 1994). The encoding used in an individual is a list of real numbers.

2.4.4 The Evolutionary Programming (EP)

Evolutionary Programming (EP) is similar to ES (Minhat, Musirin & Othman, 2008). Nelson, (1995) stated that EP is a robust optimization technique. EP was first introduced by Fogel, Owens, and Walsh in 1966 (Eiben & Smith, 2003; Blum & Roli, 2003Minhat et al., 2008). The difference between the EP, GA and ES is that no recombination operator appears in the EP. It relies on mutation as variation operator. The selection mechanism is a mixture of tournament selection and truncation selection.

Since this research focuses on the GA, further detail of this algorithm is discussed in the following sections.

2.5 GA Overall Process

The overall process of GA is very similar to the overall process of EA. GA is an iterative procedure. As a population based approach, GA deals with a group of solutions.

A solution candidate is referred as a chromosome and a population consists of a group of chromosomes. This solution will be evaluated by a fitness function (Marczyk, 2004). These chromosomes will then undergo crossover and mutation to produce new offspring which will then form a new population. Then the GA process is repeated until a termination condition stops this iteration (Beasley et al., 1993; Sipper, 1996). The general procedure of GA is outlined in Figure 2.2.

```
procedure GA;

{
        t = 0;
        initialize population P(t); // initialize population with random
                                // solutions
        evaluate P(t);              // evaluate each individual
        until (done) )   // repeat the following processes until termination
                      // condition is satisfied
        {
                t = t + 1;
                parent selection P(t);   // select parents randomly
                recombine P(t);          // pairs of parents
                mutate P(t);             // the resulting offspring
                evaluate P(t);           // new individuals
                survive P(t);            // individuals for the next generation
        }
        }
```

Figure 2.2: GA (Spears, De Jong, Back, Fogel & De Garis, 1993; Jones, 1998)