Abstract
Today’s one of the most exciting topics in the computer science is learning, i. e. machine learning. Machine learning is included in many topics such as pattern recognition and neuroscience; and so it includes many different aspects having different methods and algorithms. Among these methods artificial neural networks, evolutionary algorithms and different unsupervised learning schemes can be pointed out. A method called genetic algorithms is one of these methods, which is inspired from the theory of evolution. This paper is written in order to introduce this method with its technical aspects.

INTRODUCTION

Genetic algorithms (GA) is a machine learning meta-method in machine learning, which has its roots in the theory of evolution of Darwin and mathematical foundations in simulated annealing. Among the several definitions to GA, stating some of them should be helpful to us: “Genetic Algorithms (is) ... a tool for a search and optimizing methodology.” [MTK99]. “A GA is a mathematical search technique based on the principles of natural selection and genetic recombination.” [HOL75]

A synthesis of these definitions is straightforward, then: Genetic algorithms is a search method based on the theory of evolution and its principles. It simply copies the theory of evolution and principals of the process of reproduction presented by some famous biologists (Darwin, Mendel etc.).

Throughout this study, we will try to inform the reader about the past of GA’s; it’s method and application areas (GA’s has many interesting and widely ranged application areas as will be underlined in detail later). Moreover, we aim to fill the reader with information about the theory of evolution and the logic behind it.

BACKGROUND

Natural Evolution:

Living beings succeed in adaptation (to the nature and the environments) by evolution. This evolution takes place on chromosomes – organic units that specify the structures of living beings. Chromosomes can be imagined as huge-sized strings, which hold all information about us (since we have nothing else observable). They are copied (in a number of different manners including recombination) while reproducing and continue themselves. In the nature this recombination takes place as crossover. Crossover is the way that the new member has both parents’ genes via a combination determined before. Through this process chromosomes are open to mutation, which is an instantaneous change of an element in the chain. Mutations are one of the basic elements of adaptation, hence evolution. They help living beings to survive. There are three types of mutation first two having instantaneous effects; one is lethal and the other is vice versa. These two occurs very rare and have not continuous effects since the population refuses them often. Third type of mutation is the one that does not have any instant effect. This type of mutations fills the chromosomes with hidden changes. When a sudden defect happens in the environment, the immune members (by the help of these hidden characteristics) survive. Hence the hidden property becomes common.

Natural evolution also covers the basic idea of the growth of the populations. It says that a member (of the population) having better
adaptation will have more opportunity to survive. Consider two cheetahs, both hungry. Unluckily there is only a single rabbit around. Faster cheetah will catch the rabbit and the other will stay hungry, and probably will die. This very simple example can be considered as an explanation for the basic system of the nature.

The relationship between the evolution and the learning processes covered by living beings is another important topic for us. This relationship is almost identified by a modal called Baldwin Effect.

**Baldwin Effect:** Baldwin effect depends on the idea that the sudden changes in the environment elicit the members of the population with the capability of learning (about the present change). Thus, after a number of generations, this capability will be present in the chromosomes of the population.

**Brief History:**

Since the beginning of the 1970s, computer scientists deal with genetic algorithms. The founder of genetic algorithms is said to be a scientist named John Holland. “The field would not exist if he had not decided to harness the power inherent in genetic processes in the early 1970s and functioned as the technical and political leader of the genetic algorithm field from its inception to the present time.” [DAV91].

The value of the field increased a lot after its appropriateness to the real life problems was observed. We will cover some basic application areas later.

**METHOD**

**Overview of the Concept:**

Genetic algorithms, as we discussed before, is a method that simulates the natural evolution mechanism. It is an abstraction of events such as mutation, encoding (the problems), reproduction and concepts such as Baldwin effect. The relation between the biology and the computer program can be enumerated as follows:

- The position of a gene in a chromosome is its *locus*, the position of a feature in the string is its string position
- *Genotype* of a living being is the encoding of a problem in the real-life.
- *Phenotype* of a living being is the solution to that problem.
- Fitness of the string is the objective [GOL89] of the living being, hence the chromosome.
- In the natural reproduction, events (mutations, crossovers) occur randomly; on the other hand, in genetic algorithms they are stochastic.
- Natural evolution is on the population of living beings, genetic algorithms act on sets of strings.
- Both evolutions know only about the current configurations. No information about the past can be gathered.

The encoding part and the evaluation functions depend on the structure of the problem. There are a lot of encoding techniques in the area. One of them is using bit strings; following the idea of Holland and his colleagues. Throughout this study I will use bit strings but I have to mention that none of the techniques present works best for all problems.

With the encoding technique and the evaluation function(s) in hand, we can simulate the evolution of an abstract population. This simulation is based on a loop of mating chromosomes, deleting some of the old generation and inserting the new ones. Definitely, we have to initialize the population somehow. An overview of the process is given in Figure 1.

As Figure 1 shows, the evolution process involves high-level steps such as fitness evaluation, creating new chromosomes and updating the population, which constitutes the following topics to be discussed in detail.

**Evaluation Functions & Fitness:**

An evaluation function is a mechanism that is used to obtain the value\(^2\) of a chromosome.

---

\(^2\) ‘status’ is also used instead of ‘value’.
### Evolution Process

1. Initialize the population
2. Evaluate the fitness
3. Create new members
4. Update the population
5. If it is the last generation return the highest fitness value, Else return 2

**Figure 1:** Main loop of the evolution of a population of chromosomes

An evaluation value is the solution given to the problem of the chromosome. There are many methods of obtaining the evaluation values. These methods vary from problem to problem; they are problem dependent. Fortunately, there are some common evaluation functions in the field.

An example: **f6:** Binary f6 is a mathematical function used for this purpose. It decodes a chromosome of length (44 for this case) and produces a real value between 0 and 1. It first partitions the string into two equal parts, converts them into real numbers and multiplies with 0.00004768372718899898. This numbers then substracted from 100 and two real numbers in range [-100,100] are obtained. Then, primary step of f6 is applied to these real numbers (x,y):

\[
0.5 - \frac{(\sin(\sqrt{x^2 + y^2})^2 - 0.5)}{(1.0 + 0.001 \times (x^2 + y^2))^2}
\]

This symmetric function results in an oscillating graph, which is open to hill-climbing techniques. Further examples can be found in Lawrence Davis’ book [DAV91].

Fitness is the response of the environment to the individual (chromosome). There are many methods of specifying the fitness:

- **Fitness is Evaluation:** Fitness is the same as the evaluation value.
- **Windowing:** Assign a boundary value to the individuals with lower evaluation values than this boundary.
- **Linear Normalization:** Create fitnesses beginning with a constant value and decrease linearly.

The same words said to the evaluation functions can be repeated here also. Fitness techniques differ from system to system and problem to problem. For example, in the learning systems the training process evaluates fitnesses by itself according to the evaluation value and the environment.

Evaluation functions and fitnesses are atomic characteristics of genetic algorithms and are used in other important processes such as selection.

**Reproduction:**

Reproduction concept involves mainly two procedures: crossover and mutation. Although reproduction should be analyzed with the parent selection techniques, this analysis will be done later. In this topic, only the genetic operations will be discussed.

**Crossover:**

Crossover is simply the combination of the information of the parents. Crossover can be in two different ways: One-point, Multi-point crossovers. One-point crossover has basic disadvantages. In one-point crossover of two arbitrary parents, the number of the possible schemata is very low. For example:

P₁: 11101101  
P₂: 00110000

In this case P₁ and P₂ cannot produce 1**1***1. This can be seen as a deficiency.

Multi-point crossover, on the other hand, has more than one cut-point. In Akşamçı’s [3] I have stored the information of the cut-points in the chromosomes also. This let me perform a pseudo-random implementation.

Crossover is the most useful operation for a population in order to generate a diverse gene pool.

**Mutation:**

Mutation is a sudden change in the chromosome in the reproduction process. It simply means a negation of a bit in the sequence assuming a bit-string implementation of chromosomes.

---

3 The name of the learning agent implemented by us to play card games.
The number of mutations is again implementation dependent. This number, in common, is not evaluated directly but the mutations occur rather probabilistically. This probabilistic value ranges between 0.0005 and 0.005. Of course, this range can be edited according to the field.

Mutation is the basic operation for Baldwin Effect (Refer to Background section). It is still the most enthusiastic part of the genetic algorithms.

Growth of the Population:

Organization of the population requires mating controls, parent selections and handling of evolutionary characteristics such as elitism and robustness. I will discuss every element in detail one-by-one.

The size of the population however should be discussed firstly. There are actually two common ways for the population size. One is to keep the size constant. This method is called generational replacement (steady-state reproduction). Steady-state models include approaches such as killing (eliminating) the worst, killing the oldest or killing linearly. It helps the evolution be discrete in terms of time, i.e. generations keep evolving in similar periods. On the other hand some methods do not require a strict rule about size. Specifying the very-top limit of the population size, they only evaluate the reproduction rate inversely proportional to the current size. As a remark, using this type of a method should lead to a termination of the population and hence the end of the evolution. The environmental effects (and defects) should be analyzed before such approaches.

Selection:

Selection of the individuals to reproduce is a major topic in population control. There are several methods of selection, which commonly use evaluation values and fitnesses basically. Random mating techniques are rarely used.

**Fitness Proportionate Selection:** The individuals are selected according to their fitnesses and the fitnesses of the rest of the population.

\[
P(h_i) = \frac{\text{fitness}(h_i)}{\sum_{all} \text{fitness}(h_i)}
\]

This method is sometimes called the roulette wheel selection. It was first introduced by Holland. He decided to select an individual with this method and (if necessary) select its partner at random. Today, researchers prefer to select both parents by selection methods.

This method has some problems such as stagnating, which means the cause of decreases in the variations. A similar problem is elitism. To handle these problems, fitness evaluation methods can be re-considered again because windowing and such approaches may differentiate the fitnesses accordingly. Moving to other selection methods is also a solution.

**Rank Selection:** Individuals’ fitnesses are given by interpolation. The best individual is given a value \( s \), between 1 and 2. The worst one is given \( 2-s \). So, the evaluation will look like:

\[
P(i) = s - \frac{s - (2 * i * (s - 1))}{N - 1}
\]

According to Baker, \( s = 1.1 \) gives 94% involvement; on the other hand \( s = 2.0 \) leads to 75% [BAK85].

**Tournament Selection:** With \( n \) individuals chosen from the population at random, a tournament is held, which require better fitness to win. The winner is given the chance to reproduce. Fortunately, the best chromosome will win all the tournaments, which it is involved. An average member will probabilistically win half of the tournaments that it is involved in.

The idea of tournament selection comes from the nature itself where we can observe such ‘fights’ for the same purpose.

**Elitism:**

As I have mentioned before, elitism is an improvement in the evolution to the best-member-defeat problem. This problem occurs when the best member of the population produces worse off springs (according to
probabilistic fails and environmental defects). In elitism, the best member of the population is directly copied to the next generation. This technique has many advantages to the performance of the system.

Robustness:

Steady-state reproduction has unexpected results when the system is open to noise. It is said to be ‘less robust’, then. An algorithm is said to be robust when its performance is well in different type of problems. Robustness is lost when specializing begins, which is a basic requirement in the GA field. As a last word, the aim of the robustness is to find a single algorithm, which works for both deterministic and noisy reproductions.

Summary of the Method:

An implementation of a genetic algorithm is an abstraction of the real-world into three modules: Evaluation module, Population module, and Reproduction module. These modules can further be examined:

<table>
<thead>
<tr>
<th>Evaluation Module:</th>
<th>Evaluation Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Module:</td>
<td>Representation technique</td>
</tr>
<tr>
<td></td>
<td>Initialization Technique</td>
</tr>
<tr>
<td></td>
<td>Deletion Technique</td>
</tr>
<tr>
<td></td>
<td>Parent-Selection technique</td>
</tr>
<tr>
<td></td>
<td>Reproduction Technique</td>
</tr>
<tr>
<td></td>
<td>Fitness Technique</td>
</tr>
<tr>
<td></td>
<td>Population Size</td>
</tr>
<tr>
<td>Reproduction Module:</td>
<td>Operators</td>
</tr>
<tr>
<td></td>
<td>Operator Rates</td>
</tr>
</tbody>
</table>

Figure 2: Overall Structure of a genetic algorithm.

To summarize; with all these information and the methods in hand, the developer should make his/her decisions about selection by considering the following topics:

- Using direct fitness measures or applying ranking and/or windowing.
- Using generational models or evolutionary (incremental) models.
- Handling the noise and diversity of the gene-pool.

APPLICATION AREAS

In their papers about genetic algorithms [MIT93], Melanie Mitchell and Stephanie Forrest point out the application areas of the field by giving a list of them. The list explains itself, so I directly copy this part:

- Optimization: Genetic algorithms have been used in a wide variety of optimization tasks, including numerical optimization, and combinatorial optimization problems such as circuit design and job shop scheduling.
- Automatic Programming: Genetic algorithms have been used to evolve computer programs for specific tasks and to design other computational structures, e.g., cellular automata and sorting networks.
- Machine and robot learning: Genetic algorithms have been used for many machine-learning applications, including classification and prediction tasks such as the prediction of dynamical systems, weather prediction, and prediction of protein structure. Genetic algorithms have also been used to design neural networks, to evolve rules for learning classifier systems or symbolic production systems, and to design and control robots.
- Economic models: Genetic algorithms have been used to model processes of innovation, the development of bidding strategies, and the emergence of economic markets.
- Immune system models: Genetic algorithms have been used to model various aspects of the natural immune system, including somatic mutation during an individual's lifetime and the discovery of multi-gene families during evolutionary time.
- Ecological models: Genetic algorithms have been used to model ecological phenomena such as biological arms races, host-parasite co-evolution, symbiosis, and resource flow in ecologies.
Genetic Algorithms – an Introduction

- Population genetics models: Genetic algorithms have been used to study questions in population genetics, such as “under what conditions will a gene for recombination be evolutionarily viable?”

- Interactions between evolution and learning: Genetic algorithms have been used to study how individual learning and species evolution affect one another.

- Models of social systems: Genetic algorithms have been used to study evolutionary aspects of social systems, such as the evolution of cooperation, the evolution of communication, and trail-following behavior in ants.

The list continues day-by-day, as they said, unfortunately it is impossible to mention about all of them in this document. So I will point out some well-known fields and leave the research to the reader.

Speech Recognition Systems:

Learning has an important role in the speech recognition systems. There are two basic models in the field: Dynamic Time Warping (DTW) and Hidden Markov Model (HMM).

A block diagram of a speech recognition system is as follows:

![Speech Recognition System Diagram]

More information about the system, models, the test results of input data, used hardware and implementation of operators are discussed in [MTK99].

Communication systems:

There are searching problems in the communications such as the capacity and delay constraints, routing assignment, topology and cost. Optimization techniques are being developed in the field using genetic algorithms. The techniques are applied on ATM and WLA networks most popularly.

CONCLUSION & NOTES

The abstraction of the real life, which is named as genetic algorithms in this context, has a main structure as shown in the figure 2.1. This structure has no distinct cut-offs and it certainly allows us to construct our ways independently. On contrary, there exist also common methods to help us in the way. These properties make the field enjoyable and interesting.

A field of science has both rules and methods. Rules are more than the methods if the field is old, such as physics and chemistry. On the other hand methods are more than the rules if the field is young, sociology and psychology are examples to this. The same situation holds for genetic algorithms, which can be considered as a young field. What I mean is simply this: In genetic algorithms there are a lot of methods. That is true for all sub-topics, such as selection, evaluation, chromosome representation and reproduction. I underlined just a few of the methods in the area. The reader should examine all the methods and make a (statistical) decision according to the problem in hand.

As a last word, genetic algorithms is a field in the computer science in which the scientist plays the role of the nature. In this game of life the goal is to survive at all.

REFERENCES

[DAV91]: Handbook of Genetic Algorithms, 1991 – Lawrence Davis
[HOL75]: Adaptation in Natural and Artificial Systems, 1975 – John Holland
[MIT93]: Genetic Algorithms and Artificial Life, 1993 – Melanie Mitchell, Stephanie Forrest