

GENETIC ALGORITHM

Reinhard Magenreuter

University of Finance, Business and Entrepreneurship – Sofia

Abstract. There are four backbones to analyze time-series in general and forecast time-series for financial markets: Chaos-theory, Fuzzy logic, Neural networks and Genetic algorithms. The first one is considered in (Magenreuter, 2016a), while the second one – in (Magenreuter, 2016c) and the third one – in (Magenreuter, 2016b). The present paper is dedicated to the fourth backbone including promising outcomes in financial markets.

Keywords: chromosome, reproduction, genetic algorithm, neural network, financial market, time-series, forecasting

1. Introduction. Genetic Algorithms (GAs) are adaptive heuristic search algorithms based on the evolutionary ideas of natural selection and genetics. As such they represent an intelligent exploitation of a random search used to solve optimization problems. Although randomized, GAs are by no means random, instead they exploit historical information to direct the search into the region of better performance within the search space. The basic techniques of the GAs are designed to simulate processes in natural systems necessary for evolution, specially those follow the principles first laid down by Charles Darwin of “survival of the fittest”. Since in nature, competition among individuals for scanty resources results in the fittest individuals dominating over the weaker ones.



2. Chromosomes and their reproduction. Cycle of genetic algorithm.² GAs simulate the survival of the fittest among individuals over consecutive generations for solving a problem. Each generation consists of a population of character strings that are analogous to the chromosome that we see in our DNA. Each individual represents a point in a search space and a possible solution. The individuals in the population are then made to go through a process of evolution.

Genetic algorithms were formally introduced in the United States in the 1970s by John Holland (1929 – 2015) at the University of Michigan. Fields of applications for Genetic Algorithms are: Airlines revenue management, Container loading optimization, Control engineering, Filtering and signal processing, Economics, Financial forecasts/classifications, Automotive Design, Engineering Design, Robotics, Evolvable Hardware, Optimized Telecommunications Routing, Biomimetic Invention, Trip, Traffic and Shipment Routing, Computer Gaming, Encryption and Code Breaking, Computer-Aided Molecular Design, Gene Expression Profiling, Optimizing Chemical Kinetic Analysis, Finance and Investment Strategies, Marketing and Merchandising etc.

Famous optimization problems are e.g. maxima and minima in geometry, maxima and minima in algebra. Following is the working principle of a simple Genetic Algorithm:

- formulate initial population;
- randomly initialize population;
- repeat;
- evaluate objective function;
- apply genetic operators;
- reproduction;
- crossover;
- mutation;
- unit stopping criteria.

GA processes a number of solutions simultaneously. Hence, in the first step a population having P chromosomes called *individuals* is generated by pseudo random generators whose individuals represent a feasible solution. This is a representation of solution vector in a solution space and is called *initial solution*. This ensures the search to be robust and unbiased, as it starts from wide range of points in the solution space. In the next step, individual members, chromosomes of the population represented by a string are evaluated to find the objective function value. This is exclusively problem specification. The objective function is mapped into a fitness function that computes a fitness value for each chromosome. This is followed by the application of GA operators. Reproduction or selection is usually the first operator applied on a population. It is an operator that makes more copies of better chromosomes in a new population.

Thus, in reproduction operation, the process of natural selection causes those chromosomes that encode successful structures to produce copies more frequently. To sustain the generation of a new population, the reproduction of the chromosomes in the current population is necessary. For better chromosomes, these should be generated from the fittest chromosomes of the previous population.

There exist a number of reproduction operators in GA literature, but the essential idea in all of them is that the above average fitness value of strings are picked from the current population and their multiple copies are inserted in the mating pool in a probabilistic manner. A crossover operator is used to recombine two chromosomes to get a better one. In the crossover operation, recombination process creates different chromosomes in the successive generations by combining material from two chromosomes of the previous generation. In reproduction, good chromosomes in a population are probabilistically assigned a larger number of copies and a mating pool is formed. It is important to note that no new chromosomes are usually formed in the reproduction phase. In the crossover operator, new chromosomes are created by exchanging information among strings of the mating pool. The two chromosomes participating in the crossover operation are known as parent chromosomes and the resulting ones are known as children chromosomes. It is intuitive from this construction that good sub-strings from parent chromosomes can be combined to form a better child chromosome, if an appropriate site is chosen. With a random site, the children chromosomes produced may or may not have a combination of good sub-strings¹ from parent chromosomes, depending on whether or not the crossing site falls in the appropriate place. But this is not a matter of serious concern, because if good strings are created by crossover, there will be more copies of them in the next mating pool generated by crossover.

It is clear from this discussion that the effect of crossover may be detrimental or beneficial. Thus, in order to preserve some of the good chromosomes that are already present in the mating pool, all chromosomes in the mating pool are not used in crossover. When a crossover probability, defined here as pc is used, only $100 \times pc$ per cent chromosomes in the population are used in the crossover operation and $100 \times (1 - pc)$ per cent of the population remains as they are in the current population. A crossover operator is mainly responsible for the search of new chromosomes even though mutation operator is also used for this purpose sparingly. Many crossover operators exist in the GA literature.

One site crossover and two site crossover are the most common ones adopted. In most crossover operators, two strings are picked from the mating pool at random and some portion of the strings is exchanged between the strings. Crossover operation is done at string level by randomly selecting two strings for crossover operations. Mutation adds new information in a random way to the genetic search process and ultimately helps to avoid getting trapped at local optima. It is an operator that introduces diversity in the population whenever the population tends to become ho-

homogeneous due to repeated use of reproduction and crossover operators. Mutation may cause the chromosomes to be different from those of their parent. Mutation in a way is the process of randomly disturbing genetic information. They operate at the bit level. When the bits are being copied from the current string to the new chromosomes, there is probability that each bit may become mutated. This probability is usually a quite small value, called as mutation probability p_m . The need for mutation is to create a point in the neighborhood of the current point. The mutation is also used to maintain diversity in the population. These three operators are simple and straightforward. The reproduction operator selects good chromosomes and the crossover operator recombines good sub-strings from good strings together, hopefully, to create a better sub-string chromosome. The mutation operator alters a string locally expecting a better chromosome. Even though none of these claims are guaranteed and/or tested while creating a chromosome, it is expected that if bad chromosomes are created they will be eliminated by the reproduction operator in the next generation and if good chromosomes are created, they will be increasingly emphasized.

Application of these operators on the current population creates a new population. This new population is used to generate subsequent populations and so on, yielding solutions that are closer to the optimum solution. The values of the objective function of the chromosomes of the new population are again determined by decoding the strings. These values express the fitness of the solutions of the new generations. This completes one cycle of genetic algorithm called a generation. In each generation if the solution is improved, it is stored as the best solution.

3. Different Recombinations. The first step in the reproduction process is the recombination (crossover). In it the genes of the parents are used to form an entirely new chromosome. The typical recombination for the GA is an operation requiring two parents, but schemes with more parents, as mentioned above, are also possible. Two of the most widely used algorithms are Conventional (Scattered) Crossover and Blending (Intermediate) Crossover.

3.1. Conventional (Scattered) Crossover. In this recombination type, the parents exchange the corresponding genes to form a child. The crossover can be single- or multipoint. For the recombination a bit Mask is used. The equations describing the process are:

$$C_1 = Mask_1 \& P_1 + Mask_2 \& P_2$$

$$C_2 = Mask_2 \& P_1 + Mask_1 \& P_2,$$

where P_1 and P_2 are parent's chromosomes; C_1 and C_2 are children's chromosomes (offspring individuals); $Mask_1$ and $Mask_2$ are bit masks ($Mask_2 = NOT(Mask_1)$); & is bit operation "AND".

Example: $Mask_1 = [1\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 0]$; $Mask_2 = NOT(Mask_1) = [0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 1]$;
 $P_1 = [2\ 7\ 5\ 8\ 0\ 3\ 1\ 5\ 9]$; $P_2 = [8\ 8\ 4\ 5\ 1\ 6\ 9\ 7\ 1]$;

The coefficient α allows the user to change the area in which the value of the resultant (offspring) gene can appear. When $\alpha = 0$ it is guaranteed that the value of the resultant gene is between the values of the corresponding genes of the parents. When the value of α is above 0, neighbor areas could be explored. In Nature in a similar way is transferred the information about skin pigmentation, body structure, etc.

3.4. Mutation. The newly created by means of selection and crossover population can be further applied to mutation. Mutation means, that some elements of the DNA are changed. Those changes are caused mainly by mistakes during the copy process of the parent's genes. In the terms of GA, mutation means random change of the value of a gene in the population. The chromosome, which gene will be changed and the gene itself are chosen by random as well. Mutation in a chromosome and mutation places in the population are shown below, respectively:

[4 8 5 7 0 3 1 6 9]
 ↓
 [4 8 5 7 0 3 8 6 9]



4. Mathematical background. To maximize a solution space:

S (set of possible solutions) and *target function* $f : S \rightarrow R$. The target function (also referred to as fitness function) dedicates to each solution $s \in S$ the quality to s . Good solutions have thereby a bigger value than worse solutions.

Thus, a solution is searched $s \in S$, so that $f(s) = \max \{f(x) | x \in S\}$.

The optimization is to maximize the target function (fitness function). To minimize a solution space:

The minimizing problem is reducible to a maximizing problem: $f: S \rightarrow R, f(x) = -f(x)$.

In many cases there are more than one criterion which have to be optimized. In general there are k optimization criteria and $k \in N$ associated target functions $f_i : S \rightarrow R (1 \leq i \leq k)$.

One approach to solve such a multiple criteria optimization problem is to summarize the target function to a fitness function

$$g : S \rightarrow \mathbb{R} \text{ mit } g(x) = \sum_{i=1}^k g_i \cdot f_i(x)$$

and to maximize it. Here the variables $g_i \in R$ are to be determined weights of the valuation functions.

Example (Harbich, 2007):

The classical optimization problem is the TSP, the travelling salesman problem, which may not be solved within an acceptable amount of time with common optimization algorithms. The TSP can be described like following: may $M = \{s_1, \dots, s_n\}$, the set of cities, and $d : M \times M \rightarrow R$ a function which models the distance between the cities where from each city to the other a way shall exist. Then we can state the search space and the fitness function as

$$S = \left\{ (s'_1, s'_2, \dots, s'_n) \mid s'_i \in M, s'_i \neq s'_k, k \leq n, k \neq i \right\}$$

$$f(s'_1, s'_2, \dots, s'_n) = \sum_{i=1}^{n-1} d(s'_i, s'_{i+1})$$

Thus, the valuation function calculates the total distance traveled at the circular tour. The target is, to minimize it.

For our application, the ‘PPS-Prediction System’, we implement the Genetic Algorithms to search the best Neural Network parameter during thousands of iterations, simultaneously and at the same time to find the best Neural Network, which could generalize over a given time best. It is a multiple optimization problem. So the Genetic Algorithm is not used as analytic tool to value e.g. stocks-markets, its task is just to find the best network parameters and network-topology.

*“The needs of the individuals of a species are co-responsible,
that they evolve specific characteristics.”*

(Antje Hansen, 2003)

4. Postscript. In four cosecutive papers: (Magenreuter, 2016a), (Magenreuter, 2016c), (Magenreuter, 2016b) and the present one, we gave a brief overview to the principles of four ‘backbones’, which could be used for analyzing financial time-series and for performing good results over years. Although each method itself is a powerful analytic tool, we suggest to combine them to a hybrid analytic and analysis-tool, where all methods interact and result in promising outcomes.

NOTES

1. www.intechopen.com 306 Real-World Applications of Genetic Algorithms
2. Samuel Lukas, Arnold Aribowo and Milyandreana Muchri, ”Solving Timetable Problem by Genetic Algorithm and Heuristic Search Case Study: Universitas Pelita Harapan Timetable”, Faculty of Computer Science, Universitas Pelita Harapan, Indonesia

REFERENCES

- Magenreuter, R. (2016a). Forecasting of time-series for financial markets. *Mathematics and Informatics*, 59 (5).
- Magenreuter, R. (2016c). Fuzzy Logic. *Mathematics and Informatics*, 59 (6)
- Magenreuter, R. (2016b). Neural networks. *Mathematics and Informatics*, 59 (5).
- Popov, A. (2005). *Genetic algorithmes for optimization*. Sofia: TU.
- Harbich, S. (2007). *Einführung genetischer Algorithmen mit Anwendungsbeispiel*.
- Grozdev, S. (2007). *For high achievements in mathematics. The Bulgarian experience (Theory and practice)*. Sofia: ADE.

✉ **Mr. Reinhard Magenreuter, PhD student**
University of Finance, Business and Entrepreneurship
1, Gusla St.
1618 Sofia, Bulgaria
E-mail: rm@pariserplatz.eu