GENETIC ALGORITHMS USED IN FINANCIAL OPTIMIZATION

Lecturer Florin MOLNAR Ph.D
Faculty of Computer Sciencies,
„Ioan Slavici" University, Timişoara,
România

Lecturer Vasile ROTĂRESCU Ph.D
Faculty of Economics,
„Ioan Slavici" University, Timişoara,
România

REZUMAT. Algoritmii genetici aplică concepte de evoluţie pentru a rezolva probleme de matematică. Această idee a fost pentru prima dată exploatată de J. Holland în 1975 şi a fost aplicată în domenii cum ar fi inginerie, informatică, biologie şi muzică.

Cuvinte cheie: algoritmi genetici, optimizare financiară

ABSTRACT. Genetic Algorithms apply the concepts of evolution to the solving of mathematical problems. This idea was first exploited by J. Holland in 1975 and has been applied to areas such as engineering, computing, biology and music.

Keywords: genetic algorithms, financial optimization

1. INTRODUCTION

Genetic algorithms have been used successfully in many contexts, including meteorology, structural engineering, robotics, econometrics, and computer science. The genetic algorithm is particularly appealing for financial applications because of its robust nature and the importance of the payoff in guiding the process

Genetic Algorithm is a heuristic function for optimization, where the extreme of the function (i.e.,minimal or maximal) cannot be established analytically. A population of potential solutions is refined iteratively by employing a strategy inspired by Darwinist evolution or natural selection. Genetic Algorithms promote "survival of the fittest". This type of heuristic has been applied in many different fields, including construction of neural networks and finance.

2. A FIRST QUALITATIVE APPROACH AND QUASI – EMPIRIC OF GENETIC ALGORITHMS BASED ON AN EXAMPLE

Basic problem is considered to begin with x2 = 64, seeking the solution for x; it is proposed to find ways of solving the problem through specific methods AG; addressing the issue in this way requires basic knowledge about numeration systems, namely the use and conversion between dates 2, 10, 16.

Potential solutions to the problem will be represented by sequences or strings of binary numbers, so for example the number 10011, may represent a potential solution (in decimal is 19 = 1 * 24 + 0 * 23 + 0 * 22 + 1 * 21 + 1 * 20). In terms of genetic organization of features, this number 10011 can be assimilated as a type chromosome structure, the values 0 and 1 correspond to genes within the chromosome. It will still go through specific stages of a genetic algorithm:

Step 1 - create a random population of solutions corresponding potential problem, to begin to assume a population of four, you can still use a random number generator, which generated such parts of the population following four strings: 01000, 10101, 01010 and 11000, equivalent to 4, 21, 10 and 24. all possible solutions to the problem. Stated once again that was chosen for reasons of simplicity the example analyzed, potential solutions are actually thousands of billions, each population consisting of hundreds of individuals not only in four;

Step 2 - calculating the performance of each individual of the population, at this point we introduce the notion of matching value (fitness value, score) assigned to each individual (string), fitness = 1000 minus the absolute value of the string decimal - 64; particularity AG, is to maximize value Centre: in the example considered interactive sequence of calculations, you have to take the difference x2 - 64 tending to zero at least indirectly, this trend is equivalent to the maximum value trend 1000 - (x2 - 64). The following table presented several phases of the iterative calculation for different developments in the string (the default state of the individual and population).

Binary string
Decimal equivalent value (x)
X2-64
Fitness (depending on performance)
= 1000 - (x2 - 64)

Table 1

(1)	(2)	(3)	(4)
		Column	Performance column
Binary string	Decimal equivalent	Two square minus 64	(1000 minus column 3)
00100	4	-48	952
10101	21	377	623
01010	10	36	964
11000	24	512	488

Step 3 - Select individuals potential to become parents of the next generation, is still considered one of the possible schemes of choice for parents <-> remove the worst performers and replacing them with the best performing string, in the example considered, the most weak result it has the string 4, which generated a 488 fitness, so the string will be replaced by 3;

Step 4 - create a second generation of parents previously selected, from the multitude of tasks specific to this step, one of the most powerful is the crossover, the first two rows are chosen randomly from parents selected in step 3, for example rows 1 and 3 and namely 00100 and 01010, in the second phase also choose a random point in the two strings will be cut - in the example considered is assumed chosen point after the third bit, shown separated by a line intermediate 001-00 and 010-10, in the third phase is also considered a random number that will indicate whether or not efficient crossover, in the example considered we assume that the phenomenon of crossover to occur in approximately 60% the time considered; crossover phenomenon will occur in our case by changing the headers of the two strings together so that the strings are now considered structure: 00110 and 01000, the optimal sequence is given by 01000. There is three times successively using the process of randomization (random choice), this characteristic of genetic algorithms AG.

Crossover is a process that expands research, search in many directions, to purely mathematical description is laborious beyond the context of this paper, in summary we can say that it made new combinations in order to obtain optimal values (in this case the variable fitness) in a very efficient manner.

A second process is specific mutations and is also a process of biological inspiration, in this sense a bit of a string (individual) is selected bit value that will be complementary - yet this rare phenomenon occurs (frequency of 1 / 1000) even in real biological sphere.

The main goal of the two processes, but also others who will be introduced later is to transform the population over generations in order to improve certain characteristics of the vital, the analysis of the sample considered is obvious the importance of certain bits that deal a certain position (obviously in this situation further to the left bits that most significantly associated rendering fui 2 higher rank); croosoverr creates the possibility that these bits just a higher form to be transported to other strings ("beneficial characteristics of individuals and villagers to send their").

Step 5 - is proposed to return to step 2 and repeat steps 2, 3, 4 until the populations converge, according to the procedures of the previous steps, substrings of a string of bits that were actually generated worse replaced by other substrings that generate greater effects good.

("Performance attributes generated weaker individuals are replaced by better ones, so that future generations inherit a priority to what is good"), or in other words, good triumphs over evil. By repeating sequential and iterative (multiple repeats) of these procedures population becomes increasingly homogeneous, and when the strings are perfectly identical in all positions, it is said that the population is totally convergent (Fully Converged), the phenomenon now the crossover has no effect, and any improvements can be produced only by mutations, but if you already have optimum ranges possible mutations can only worsen the situation, taking us what to improve.

Previously noted the randomness of the processes in nature AG, but they can sometimes produce favorable effects, and sometimes adverse effects, sometimes people can be on the way to a good result, sometimes not.

Fine-tuning of the AG, is one of the sensitive topics specific ad, which will be detailed in the following paragraphs, there is a compromise between speed of convergence to the solution proposed and the quality of the solutions obtained, obviously a long search will lead to better solutions ("many attributes of humanity has been perfected in huge intervals ");

Table 2

(1)	(2)	(3)	(4)
		Column	Performance column
Binary string	Decimal equivalent	Two square minus 64	(1000 minus column 3)
00100	4	-48	952
10101	21	377	623
01010	10	36	964
11000	24	512	488

Classification systems based on AG

The problem with classification systems AG is one of the most fascinating applications AG, the power of this technique is extraordinary.

It is considered an example of a system management of equity portfolios, decisions taken refers to packages appropriate actions to be sold, and that is purchased, obviously these decisions will be made according to several criteria specified on the first line of the table the next. Inputs are used in assessing the three codes, namely one for the situation in terms of acceptable criterion, 0 situation unacceptable and # means irrelevant as criteria considered.

If there are only two codes corresponding output validation option to buy, sell, respectively, for the first example is considered, leading to the conclusion that it is appropriate to purchase.

Each example considered, together with the assessment logic output is actually a rule, the AG development is proposed to begin compiling a set of 100 rules on a random, each usually receives even a factor of importance, power (level of strength) also from a simulation of a potential bid, to quantify the number of bids (bids) for each rule, such a rule may have important factor X and supply in time is proposed to reduce the coefficient of importance that the tenders rule that are declining.

In the second step, the set of rules will be faced with a new set of data and a new set of offers, the process will be repeated so that good will win in the important rules, and the weak will fall or be removed.

In this moment begins the specific applying principles AG, it can introduce new rules on operators AG, selection, mutation and crossing, the difference is remarkable in this context object to use the main AG, where he proposed finding an optimal solution in this is proposed to find a set of rule, which in turn optimal rules to be applied to another system, in this case and AG procedures in this situation will have certain characteristics.

3. CONCLUSIONS

Genetic algorithm (GA) optimization, a global technique, searches for a design that minimizes an objective function subject to constraints. As an example of using numerical optimization to help automate the design process, GA optimization was used to determine parameter values for an active vehicle suspension that minimized a performance criterion while satisfying a number of other design requirements. The response of the active design that best minimizes the objective function shows that the road disturbance has little effect on the seat acceleration when compared to passive designs. In practice, however, realistic implementation of the active skyhook dampers must be addressed by obtaining actuators that can deliver the required power, if possible.

REFERENCES

[1] **David E. Goldberg**. Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, 1989.

[2] **John H. Holland**. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. MIT Press, 1992.

[3] **Andone**, Inteligenta artificiala si sisteme expert in contabilitate, Moldava Publishing House, Iasi, 1993

[4] **Liliana Dorneanu, Mircea Untaru, Doina Darvasi, Vasile Rotarescu, Lavinia Cernescu** - Using Artificial Neural Networks in Financial Optimization, Fundatiei pentru cultura si invatamant "Ioan Slavici' Publishing House, Timisoara, 2011.

[5] **T. Slavici**, (2009) Inteligenta artificiala, Fundatiei pentru cultura si invatamant "Ioan Slavici' Publishing House, Timisoara.