ACM PAPER FORMAT SAMPLE

Memetic Algorithms

MARIA WILLIAMS, Institute of computing
PAMELLA ROBERTS, University of technology
STEVE THOMPSON, Institute of science

Evolutionary algorithms that include genetic algorithms, memetic algorithms, evolution strategies, and genetic programming are used to find the solution space by stochastic, using a population of solutions where they simulate the principles of evolution. The effectiveness of the evolutionary algorithm depends on the process of coding the problem of chromosome problems, the function of evaluating the goodness of the individuals, the implementation and the frequency of the application of the recombination and mutation operator, the process of selection of individuals and others...

KEYWORDS
Evolutionary algorithms, Memetic, Matlab

1 INTRODUCTION

Evolutionary Programming is a branch of computer science that deals with algorithms based on the principles of Darwin's theory of natural selection, and also finds inspiration in molecular genetics. Many species throughout the history of the world have evolved to better accommodate the environment they live in. In the same way, we want to devise an evolutionary algorithm that will adapt the initial population to each new environment.

Often, but not always, the environment will be a problem that we want to solve, while the ones are solutions. Intuitive, how good the individuals are adapted to their surroundings, so much is a good solution to the problem. Memetic algorithms, which are the main topic of this paper, are actually evolutionary algorithms combined with other techniques or upgraded by some other method or data structure. Their use is very successful in practice and possesses great potential for further research. Below we will first describe the evolutionary algorithms and their known cages. After that, we move to the memetic algorithms, their characteristics, properties and examples. Finally, we approach an example of computing in MATLAB, the analysis of the algorithm used and the results obtained.

2 EVOLUTIONARY ALGORITHMS

2.1 Motivation

There are various variants of evolutionary algorithms, but the basic idea in the background of all techniques is the same: we have a given population of individuals in an environment that has a limiting energy source; resource constraint leads to boredom provokes natural selection (survivors), and in turn we gain better capability of the entire population. For the beginning, let's consider the evolution in the biological framework. We have an environment populated by the population of some individuals trying to survive and play. The fitness of these individuals is determined by the environment in which they live and relates to how well each of them is successful in achieving their goals. In other words, the eligibility of the representative of the subject is depraved in the surrounding environment and spreads. The basic way of evolutionary solution to the problem relies on the method of trial and error. In that context, our task is actually redfitnuced to determining a set of candidacy decisions. Their quality, or coins, solves the given problem, determines the likelihood that they will survive and participate in the process of finding future candidates for solutions.

So, if we look at the problem within the algorithm, then the quality entity is actually a good approximation of the desired solution, which in each subsequent iteration (the offspring of that individual) becomes more and more closely related to the correct solution. The analogy between biological evolution and its use in solving the problem is given in Table 1.

Table 1 The basic view of linking evolution to problem solving in the context of evolutionary programming

EVOLUTION		PROBLEM SOLVING
Environment	≥	Problem
Individual	≥	Candidate for solution
Fitness	≥	Quality

2.2 What is an evolutionary algorithm?

So, we have a set of individuals in some space and we want each of them to be separately evaluated in terms of its quality. Therefore, the sought after quality of each individual can be viewed as the function that we want to maximize. In the context of the seg problem, we take the fitness measure as a function to measure the quality of an individual, that is, the suitability of a visa, that is the quality better. So, the quality function we have just described and for which domain we are taking individuals from a certain generation, we actually choose the best candidates for solutions that will participate in the next generation. Creating a new generation takes place by applying recombination and / or mutation to selected candidates. Recombination or Crunch is an operator applying to two or more candidates (the so-called parents) of one or more new candidates (children). The mutation is applied to one candidate and results again with a new candidate. So, by recombining and mutating parents, we get a set of new candidates for succession (offspring). New candidates, considering their eligibility and age, are competing with old candidates for a new generation. This process is repeated as long as there is no hope of satisfying the quality (correct solution) or until some pre-set request (e.g. maximum number of iterations, etc.) is met. There are two main factors for evolutionary systems:

- Variation operators (recombination / mutation) - create the necessary diversity within the population and thus enable the emergence of new markings
- Selection - acts as a force within a population that increases the average quality of new candidates for solutions

The combination of these two processes leads to the improvement of generic gene sufficiency. Thus, the evolutionary process results in a population that is capable of best adapting to the environment, reflecting the number of descendants (which in this case increases). It is important to emphasize that many components of the evolutionary process are stochastic [1]. During natural selection, the best individuals are not selected deterministic [2] and therefore it is possible that even the weaker individuals become parents or survive. In the recombination process, we also randomly select which parts of the parent we will encircle. Similar to the mutation, i.e. the selection of the parts to be changed, and the new parts that will replace them, are selected in a random manner.
It is obvious that such a scheme falls into the category of so called. generate-and-test algorithms, or as we have already mentioned, are based on experiment and error method. The function of the evaluation gives a heuristic assessment of the quality of the solution, and the variation and selection operators manage the search process. Evolutionary algorithms contain a number of properties that can be classified into a generic-and-test method, some of which are:

- Evolutionary algorithms treat a whole set of candidates simultaneously because they are based on the overall population
- Most evolutionary algorithms use recombination, or crunch information from two or more candidates for solutions to create a new one
- Evolutionary algorithms are stochastic [3]

2.3 Components of the evolutionary algorithm

In this section we will elaborate in detail the evolutionary algorithms. There are numerous components, procedures and operators that we must state and explain their meaning and role so that we can define certain evolutionary algorithms. Let's start with the pseudocode and list the most important components:

- Representation of individuals
- Evaluation function, i.e. evaluation (or fitness function)
- Population
- Parent selection mechanism
- Variation operators - recombination and mutation
- The unit selection mechanism for the next generation (selection of heirs)

In order to create a complete algorithm that works well and executes, it is necessary to specify each component and define the way it is initiated. If we want the algorithm to stop executing at some point, we must determine the stopping condition.2.4 Ground-State Magnetization Determination and DMM Micro magnetic Simulations

2.3.1 Representation of individuals

The first and fundamental step in defining the evolutionary algorithm is to create a link between the original problem and the evolutionary way of solving the problem. This often means that we need to simplify or set up a look at the real world to get a well-defined 'tangible' problem within the context of the context. There are solutions that can be evaluated. First of all, we need to decide what to do about the possible solutions and how to store them in the computer so that we can manipulate them. We will say that objects that make possible solutions to the original problem are related to the phenotype, and their recovered data within the evolutionary algorithm will be termed a genotype. This step is called a representation because it joins the phenotype with the genotype that represents it (represent). For illustration, let's take some optimization problem with the possible solutions of the whole number. So, a set of whole numbers is actually a set of phenotypes. If someone decides to write those numbers in their binary record, then genotype 10010 would represent number 18 of a set of phenotypes. It is important to note that the space of phenotypes can be greatly distinguished from the genotype space and that the entire evolutionary search takes place in the genotype space. The solution, or a good phenotype, is obtained by decoding the best genotype after the completion of the algorithm. Therefore, it is desirable that optimal problem solving (phenotype) is shown in the given genotype space. In fact, since we cannot generally know in advance what the solution seems to be, we are saying that all of the solutions can be presented at this gathering.

2.3.2 Evaluation function (fitness function)

The role of the evaluation function is to provide the requirements that the population should satisfy. It makes the basis for selection and thus facilitates the improvement. From perspective

solving problems in an evolutionary sense, it is a task that needs to be solved. Technically speaking, it is a function or process that assigns value to a genotype. Usually, this function consists of an inverse representation (we want to get a suitable phenotype from the genotype) measured by quality measurement in the phenotype area. In the context of our example, if the task finds an integer x that maximizes x2, then the genotype 10010's suitability can be determined by demo matching the phenotype (10010 → 18) and squaring it: 182 = 324. In evolutionary programming, the function of evaluation is commonly called function similarities.

REFERENCES

[1] Rocco, De Nicola, Diego, Latella, Michele, Loreti, and Mieke, Massink. 2012. A uniform definition of stochastic process calculi. CSA Technical Report #01/2012 IMT Institute for Advanced Studies Lucca

[2] Thomas, Müller, and Tomasz, Placek. 2015. Defining Determinism. British Journal for the Philosophy of Science, 69 (1), 1 - 43

[3] J. P., Rennard, (Ed.) 2006. Handbook of Research on Nature Inspired Computing for Economics and Management, Hershey, IGR.

[4] V., Barnett. 2002. Sample survey - Principles and methods. 3rd edition. Hodder Arnold, London.

Received September 2018; revised November 2018; accepted January 2019