Neurocomputing 127 (2014) 104–113

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Transgenic: An evolutionary algorithm operator

Laurence Rodrigues do Amaral [a,b,*], Estevam Rafael Hruschka Jr.[b]

[a] Federal University of Uberlandia, Faculty of Computation, Av. Getulio Vargas - 230, CEP 38.700-103, Patos de Minas/MG, Brazil
[b] Federal University of Sao Carlos, Computer Science Department, Rod. Washington Luiz KM 235, CEP 13.565-905, Sao Carlos/SP, Brazil

ARTICLE INFO

Article history:
Received 7 February 2013
Received in revised form
6 June 2013
Accepted 5 August 2013
Communicated by Swagatam Das
Available online 22 October 2013

Keywords:
Transgenic operator
Genetic algorithms
Genetically modified organisms
Classification task
Elitism

ABSTRACT

Traditionally, many evolutionary algorithm operators have biological inspiration. Genetics has contributed to the proposal of a number of different evolutionary operators, such as haploid crossover, mutation, diploid, inversion, gene doubling, deletion, and others. In the present study, we propose a new genetic-inspired evolutionary operator, named *Transgenic*, which was specially designed for Genetic Algorithms (GA). The proposed operator is inspired by genetically modified organisms (GMOs), where important features are artificially introduced into their genome. *Transgenic* can be used to artificially insert relevant characteristics in the chromosome of individuals, thus converging to better results faster than traditional GAs. When relevant characteristics are known *a prior*, then, *Transgenic* simply forces the presence of such characteristics in part of the population (in an elitism-based approach). Whenever there is no *a priori* knowledge available, *Transgenic* automatically identifies relevant features (based on historical information) to perform the elitism approach. The GA, used in this study was designed to allow the discovery of concise, yet accurate, high-level rules (from synthetic and real biological databases) which can be used as a classification system. The empirical results have shown that *Transgenic* is capable of generating better results than traditional rule classification methods, such as J48, Single Conjunctive Rule Learner, One R and PART, using synthetic datasets.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Genetic engineering is a research area focused on the performance of direct human manipulation of an organism's genetic material in a way that does not occur under natural conditions. It involves the use of recombinant DNA (deoxyribonucleic acid) techniques, but does not include traditional animal and plant breeding or mutagenesis. Any organism that is generated using these techniques is considered to be a genetically modified organism (GMO). Genetic engineering or recombinant DNA technology, transcends classical plant and animal breeding by permitting the rapid transfer of genetic traits among entirely different organisms [1].

Horizontal Gene Transfer (HGT) is a form of genetic engineering. In general, there are three major procedures applied in HGT ([2–4]):

1. *Transformation* is the genetic alteration of a cell resulting from the introduction, uptake and expression of foreign genetic material. This procedure is commonly applied to bacteria and it is often used to insert novel genes into bacteria for experiments, or for industrial or medical applications. In Molecular Biology, transformation is the genetic alteration of a cell resulting from the uptake, genomic incorporation, and expression of foreign genetic material. The term transformation is also used, as a more general way, to describe mechanisms of transference of DNA. For example the production of a transgenic product, like as transgenic corn requires the insertion of new genetic information in the genome of the corn using the appropriate mechanism of DNA transfer.
2. *Transduction* is the process in which bacterial DNA is moved from one bacterium to another by a bacterial virus. Packaging the desired genetic material wished in a suitable virus (*V*) that is able to infect vegetables, allows *V* to modify the target bacterium. If the genetic material is DNA, it can recombine the chromosomes to produce recombinant cells. Nevertheless, the vegetable genomes of most viruses consist of RNA (ribonucleic acid), that replicates in the cytoplasm of the infected cell. The lineage of these free infected plants will be free from the virus, as well as from the inserted gene.
3. *Bacterial conjugation* is a process in which a living bacterial cell transfers genetic material through cell-to-cell contact [5].

Benefits presented by the proponents of Genetically Modified (GM) technology include improvement in fruit and vegetable shelf-life

* Corresponding author at: Federal University of Uberlandia, Faculty of Computation, Av. Getulio Vargas - 230, CEP 38.700-103, Patos de Minas/MG, Brazil.
Tel.: +55 34 9990 0662.
E-mail addresses: laurence@facom.ufu.br, lramaral@yahoo.com.br (L. Rodrigues do Amaral).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.08.037

and organoleptic quality, improved nutritional quality and health benefits in foods, improved protein and carbohydrate content of foods, improved fat quality, improved quality and quantity of meat, milk and livestock. Other potential benefits are the use of GM livestock to grow organs for transplant into humans, increased crop yield, improvement in agriculture through breeding insect, pest, disease, and weather resistant crops and herbicide tolerant crops, use of GM plants as bio-factories to yield raw materials for industrial uses, use of GM organisms in drug manufacture, in recycling and/or removal of toxic industrial wastes [6].

Based on the biological inspiration for new computational methods, several computer scientists studied evolutionary systems with the idea that evolution could be used as an optimization tool for engineering problems. Many authors propose methods using the biology inspiration, such as Rechenberg introduced "evolution strategies", Fogel, Owens, and Walsh developed "evolutionary programming" [7], among others. To evolutionary-computation researchers, the mechanisms of evolution seem well suited for some of the most pressing computational problems in many fields. Many computational problems require searching through a huge number of possibilities for solution [8].

Genetic Algorithms (GAs) were proposed by John Holland in the 1960s. In contrast with evolution strategies and evolutionary programming, Holland's original goal was not to design algorithms to solve specific problems, but rather to formally study the phenomenon of adaptation as it occurs in nature and to develop ways in which the mechanisms of natural adaptation might be imported into computer systems. Holland presents in the book *Adaptation in Natural and Artificial Systems*, the genetic algorithm as an abstraction of biological evolution and gave a theoretical framework for adaptation under the GA [8].

Originally, Holland [9] presented four operators for the GAs: selection, crossover, mutation and inversion. The selection operator chooses those chromosomes in the population that will be allowed to reproduce, and in general, the chromosomes having best fitness will be chosen to produce more offspring than the other ones. Crossover operators are responsible for exchanging subparts of two selected chromosomes, roughly mimicking biological recombination between two single-chromosome haploid organisms. Mutation operators, on the other hand, randomly changes the allele values in specific locations in the chromosome. The inversion operator reverses the order of a contiguous section of the chromosome, thus rearranging the order in which genes are arrayed [8].

Many new ideas were incorporated into GAs from Genetics, such as haploid crossover, mutation, diploid, inversion, gene doubling, deletion, dominance, translocation, sexual differentiation [10], and introns [11]. In recent years a huge amount has been learned in the genetics community about how genes regulate one another – how they turn one another on and off in complicate ways so that only the appropriate genes get expressed in a given situation. It is these regulatory networks that make Genome a complex but extremely adaptive system. Capturing this kind of genetic adaptivity will be increasingly important as GAs are used in more complicate and changing environments [8].

Following along these ideas, we are proposing[1] an operator for evolutionary algorithms inspired in genetically modified organisms (GMOs). More specifically, we propose a transformation mechanism (Section 1), where important features are introduced into their genome artificially. The proposed *Transgenic* operator is described in more details in Section 2. In the sequel, the evolutionary environments used to implement *Transgenic* is presented and discussed in Section 3. Finally, the experiments and results with critical analysis are presented in Sections 4 and 5 respectively.

2. Transgenic operator

The main contribution of this paper is to propose, discuss and empirically evaluate the *Transgenic* operator, for Evolutionary Algorithms, inspired by genetic engineering. The new operator (*Transgenic*) allows the manipulation of the genetic material of individuals, adding important and relevant features to a chromosome. In this sense, our approach can be seen as an elitism strategy focused on specific genes which are known to be important or relevant to a target behavior. When knowing in advance which are the most relevant genes, *Transgenic* keeps these values in the next individuals' target genes. Whenever the most relevant genes are not known in advance, the proposed operator identifies relevant genes having specific values and then, keeps these values in the next individuals' target genes. In other words, if a specific value S of a gene G is identified as having strong influence in the fitness function (of the GA), then, *Transgenic* will force some individuals in the next generation to maintain the value S in gene G.

As aforementioned, whenever the most relevant genes are not known in advance, *Transgenic* works selecting the best genes in the population using historical information from previous generations. Fig. 1 shows the historical structure used for this task.

The historical structure (shown in Fig. 1) is composed by N positions, where $N - 1$ positions are integer-valued and one position is represented by a float number. The float value is used to store the best fitness function value present in the population and the $N - 1$ integers are the number of genes in the individual. Fig. 2 shows an example of the historical structure for a real case.

In the specific case represented in the example given in Fig. 2, the number of occurrences for genes *bln*01 and *bln*02 are bigger than 0 (zero) because they appear in individuals with fitness function value equals to 0.59 (best value founded in the population). Gene *bln*01 is present in 4 (four) individuals and *bln*02 is present in 2 (two) individuals. Considering that there are other individuals (with other genes) in which the fitness function equals to 0.59, the number of occurrences for these other genes are incremented. If a new individual is analyzed and this new individual has fitness function value higher than 0.59, then all historical information is reset based on the new individual. In Fig. 2, gene *bln*20 does not appear in any individual in which the fitness function value equals to 0.59. Furthermore, gene *bln*01 has the value sub-unit set to "true" and *bln*02 has the value sub-unit set to "false".

As can be seen in Fig. 3, *Transgenic* is composed of 4 modules (M_1, M_2, M_3 and M_4), and works in a strategy similar to the one applied in [13]. Considering there is no *a priori* information on the most relevant genes for a specific problem, all the four modules use the historical information to change the current population individuals as follows. Based on the values present into the historical information, a roulette wheel is generated. Genes having the highest number of occurrences (in historical data) will receive a larger portion of the roulette wheel, proportional to the their number of occurrences. For a more concrete example, consider using the historical example given in Fig. 2, in that situation, gene *bln*01 will receive four parts of the roulette wheel, whereas gene *bln*02 will receive two parts and so on. Thus, based on the historical example shown in Fig. 2, the probability of selecting gene *bln*01 is twice the probability of selecting gene *bln*02.

[1] Initial results of this proposed approach were published in [12]. In this paper we extend the ideas and discussions, as well as, present more robust empirical evaluation. In addition, in this paper we are able to empirically show the expected behavior of the proposed method in specific situations.

دریافت فوری ← متن کامل مقاله

ISIArticles
مرجع مقالات تخصصی ایران

✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
✓ امکان دانلود نسخه ترجمه شده مقالات
✓ پذیرش سفارش ترجمه تخصصی
✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
✓ دانلود فوری مقاله پس از پرداخت آنلاین
✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات