Adv. Studies Theor. Phys., Vol. 7, 2013, no. 8, 355 - 358
HIKARI Ltd, www.m-hikari.com

Genetic Algorithms with Exons and Introns for the Satisfiability Problem

Vladimir Popov

Department of Intelligent Systems and Robotics
Ural Federal University
620083 Ekaterinburg, Russia
Vladimir.Popov@usu.ru

Abstract

In this paper we propose a new model of genetic algorithms. This model uses notions of exons and introns. We consider the satisfiability problem as a testbed for a genetic algorithm with exons and introns.

PACS: 02.70.Rr

Keywords: genetic algorithms, exons, introns, the satisfiability problem

DNA is a polymer, constructed of subunits known as nucleotides. Respectively, proteins are polymers, constructed of subunits known as amino acids. The sequence of each protein is a function of some DNA subsequence which serves as the gene for that protein. The cellular expression of proteins proceeds by the creation of some message copy from the DNA template into a closely related molecule known as RNA. This RNA is then translated into a protein. It should be noted that one of the most unexpected findings in molecular biology is that large pieces of the RNA are removed before it is translated further. The retained sequences are known as exons (see [1]), while the removed sequences are known as introns. In this paper, we consider genetic algorithms in which code are interrupted by intervening, non-coding sequences.

Let $W = \{u[1], u[2], \ldots, u[n]\}$, $u[i] \in \{0,1\}^+$, $1 \leq i \leq n$. We can consider W as a population of chromosomes. We assume that

$$u[i] = x[i,1]y[i,1]x[i,2]y[i,2]\ldots x[i,k[i]]y[i,k[i]]x[i,k[i]+1],$$

$k[i] \geq 1$, $x[i,j] \in \{0,1\}^+$, $y[i,j] \in \{0,1\}^+$, $1 \leq j \leq k[i]$, $x[i,k[i]+1] \in \{0,1\}^*$. For any $1 \leq j \leq k[i]+1$, we consider $x[i,j]$ as exons. Respectively, for any $1 \leq j \leq k[i]$, we consider $y[i,j]$ as introns.

Let $V = \{v[1], v[2], \ldots, v[n]\}$, $v[i] = x[i,1]x[i,2]\ldots x[i,k[i]]x[i,k[i]+1]$, $1 \leq i \leq n$. We can consider V as a population of chromosomes for simple genetic algorithm. In particular, we can consider the following process. During each successive generation, a proportion of the existing population is selected to breed a new generation: $\mathcal{P} : \mathbf{V} \to \{1, 2, \ldots, n\}$ where $\mathbf{V}$ is the set of all possible values of V. Individual chromosomes are selected by a fitness function: $\mathcal{F} : V \to R$ where R is the set of real numbers. The next step is to generate a second generation population of chromosomes. To solve this problem we can use genetic operators: crossover $\mathcal{C}$ and mutation $\mathcal{M}$. Usually, crossover defines a part of parent chromosome which used for construction of child chromosome. Mutation is a final modification of child chromosomes. This generational process is repeated until a termination condition $\mathcal{T}$ has been reached.

Usually, we have some natural reasons to determine the values of $\mathcal{P}$, $\mathcal{F}$, and $\mathcal{T}$. On the other hand, generally, we we do not have such reasons to determine the values of $\mathcal{C}$ and $\mathcal{M}$. So, we usually use random functions as $\mathcal{C}$ and $\mathcal{M}$. It is easy to see that we can use introns to define $\mathcal{C}$ and $\mathcal{M}$ (see e.g. [2, 3]). Note that different satisfiability problems and applications of these problems for investigations of methods of solution of other problems received a lot of attention recently (see e.g. [4] – [9]). In this paper we use the satisfiability problem as a testbed for a genetic algorithm with exons and introns.

Now we define a genetic algorithm with exons and introns (EIGA) for 3SAT. We assume that $k[i] = 1$, $1 \leq i \leq n$. Let $f(z[1], \ldots, z[m])$ be a Boolean function. Let $x[i,1] = a[i,1]\ldots a[i,m]$, $a[i,j] \in \{0,1\}$, $1 \leq i \leq n$, $1 \leq j \leq m$. We can consider $a[i,j]$ as a value of $z[j]$. Let $\mathcal{P} = \frac{n}{2}$. Let $\mathcal{F}(u[i])$ be the number of true clauses for $x[i,1]$.

We assume that $y[i,1] = b[i,1]\ldots b[i,m]$, $b[i,j] \in \{0,1\}$, $1 \leq i \leq n$, $1 \leq j \leq m$. We can use $b[i,j]$ to define values of $\mathcal{C}$. In particular, let

$$\mathcal{F}(u[i[1]]) \geq \ldots \geq \mathcal{F}(u[i[n]]).$$

In this case, we select $u[i[1]], \ldots, u[i[\frac{n}{2}]]$. Let

$$\mathcal{C}(\{u[i[1]], \ldots, u[i[\frac{n}{2}]]\}) = \{w[1], \ldots, w[n]\}$$

where

$$w[j] = c[j,1]\ldots c[j,m]d[j,1]\ldots d[j,m],$$

$$c[j,k] = \begin{cases} a[j - \frac{n}{2}, k], j > \frac{n}{2}, \\ a[j,k], j \leq \frac{n}{2}, b[j,k] = 1, \\ a[\frac{n}{2} - j + 1, k], j \leq \frac{n}{2}, b[j,k] = 0, \end{cases}$$

$$d[j,k] = \begin{cases} b[j - \frac{n}{2}, k], j > \frac{n}{2}, \\ b[j,k], j \leq \frac{n}{2}, \end{cases}$$

$1 \leq k \leq m$. Note that in a simple genetic algorithm we need a new population for new Boolean function. In EIGA, we use simple genetic algorithm for $y[i,1]$ after each generation. Also, we use old values of $y[i,1]$ for new Boolean function. It is easy to see that we obtain a simple genetic algorithm from EIGA if we consider a random function for $\mathcal{C}$ instead of just specified function. Selected experimental results are given in Figure 1.

number of generations for y	10^3	10^4	10^5	$5 \cdot 10^5$	10^6	$5 \cdot 10^6$
number of true clauses	73 %	85 %	91 %	97 %	98 %	99 %

Figure 1: Experimental results for EIGA.

ACKNOWLEDGEMENTS. The work was partially supported by Analytical Departmental Program "Developing the scientific potential of high school" 8.1616.2011.

References

[1] W. Gilbert, Why genes in pieces?, *Nature*, 271 (1978), 501.

[2] A. Gorbenko and V. Popov. Planning for Modular Robots. LAP GmbH. & Co. KG, Saarbrucken, Germany, 2012. 242 p.

[3] A. Gorbenko, V. Popov, and A. Sheka, Robot Self-Awareness: Exploration of Internal States, *Applied Mathematical Sciences*, 6 (2012), 675-688.

[4] A. Gorbenko and V. Popov, Task-resource Scheduling Problem, *International Journal of Automation and Computing*, 9 (2012), 429-441.

[5] A. Gorbenko and V. Popov, On the Longest Common Subsequence Problem, *Applied Mathematical Sciences*, 6 (2012), 5781-5787.

[6] A. Gorbenko and V. Popov, On the Problem of Sensor Placement, *Advanced Studies in Theoretical Physics*, 6 (2012), 1117-1120.

[7] A. Gorbenko and V. Popov, The Problem of Selection of a Minimal Set of Visual Landmarks, *Applied Mathematical Sciences*, 6 (2012), 4729-4732.

[8] A. Gorbenko and V. Popov, The set of parameterized k-covers problem, *Theoretical Computer Science*, 423 (2012), 19-24.

[9] A. Gorbenko, V. Popov, and A. Sheka, Localization on Discrete Grid Graphs, *Lecture Notes in Electrical Engineering*, 107 (2012), 971-978.

Received: February 12, 2013