Polibits

ISSN: 1870-9044

polibits@nlp.cic.ipn.mx

Instituto Politécnico Nacional

México

Simões Moreira, Renato; Noura Teixeira, Otávio; Limão de Oliveira, Roberto Célio

Mixing Theory of Retroviruses and Genetic Algorithm to Build a New Nature-Inspired Meta-Heuristic for Real-Parameter Function Optimization Problems

Polibits, vol. 42, 2010

Instituto Politécnico Nacional

Distrito Federal, México

Mixing Theory of Retroviruses and Genetic Algorithm to Build a New Nature-Inspired Meta-Heuristic for Real-Parameter Function Optimization Problems

Renato Simões Moreira, Otávio Noura Teixeira, and Roberto Célio Limão de Oliveira

***Abstract*—This paper describes the development of a new hybrid meta-heuristic of optimization based on a viral lifecycle, specifically the retroviruses (the nature's swiftest evolvers), called Retroviral Iterative Genetic Algorithm (RIGA). This algorithm uses Genetics Algorithms (GA) structures with features of retroviral replication, providing a great genetic diversity, confirmed by better results achieved by RIGA comparing with GA applied to some Real-Valued Benchmarking Functions.**

***Index terms*—Evolutionary computation, genetic algorithm, viruses, retroviruses, hybrid metaheuristic.**

I. Introduction

OVER the years the viruses have been treated as villains in the destruction of organic structures, resulting from the disappearance of entire species and even causing long and fatal epidemics (such as HIV). However, its effectiveness to perpetuate themselves is really impressive, although their acceptance as life forms are still under debate. Retroviruses are the nature's swiftest forms [1] and this retroviral feature could not be discarded to develop some computational structure (specifically in the field of evolutionary computation) that uses them as inspiration.

This manuscript work will describe the development of a new metaheuristic structure inspired on viral structures of family Retroviridae (retroviruses), this algorithm is called as Retroviral Iterative Genetic Algorithm (RIGA). The source of the name comes from the junction of its features: *Genetic Algorithm* for behaving like a GA, *Retroviral* for having retroviruses structures and *Iterative* because occurs every single generation.

Manuscript received May 7, 2010. Manuscript accepted for publication August 29, 2010.

Renato Simões Moreira and Roberto Célio Limão de Oliveira are with PPGEE-ITEC, Universidade Federal do Pará (UFPA), Belém, PA, Brasil (renatosm@gmail.com; limao@ufpa.br).

Otávio Noura Teixeira is with Laboratório de Computação Natural (LCN), Centro Universitário do Pará (CESUPA), Belém, PA, Brasil; Movimento Evolucionário e Cooperativo para a Construção do Artificial (MEC2A), Belém, PA, Brasil; PPGEE-ITEC, Universidade Federal do Pará (UFPA), Belém, PA, Brasil (onoura@gmail.com).

II. Biological Basement: from Viruses to Retroviruses

Viruses are compulsory intracellular parasites with a very simple structure. Their acceptance as life forms is very controversial, since they are very different from the most simple bacteria and they have unique features, like the absence cell membrane, they don't have any known organelles and their size is several smaller, thus, the only possible way to see them is by electronic microscopy. They are also metabolically inert unless they are inside a host cell. It is important to notice also that they cannot contain simultaneously DNA and RNA molecules [3].

The viruses are formed basically by two components: the capsid, consisting of viral proteins, and the core, which contains their genetic information; the combination of these two structures is known as nucleocapsid. The main objective of viruses is to replicate themselves. To achieve this, they need to penetrate a host cell, make copies of themselves and put those copies out of the host cell.

A. Retroviruses

Retroviruses are the only known entities that are able to convert RNA into DNA under normal circumstances. After the adsorption and the injection of their genetic material into the host cell, the process of retrotranscription takes place in the cytoplasm of the infected cell, using the viral reverse transcriptase enzyme. This process will convert a single-stranded molecule of RNA (ssRNA) into a double-stranded DNA (dsDNA) molecule that is larger than the original RNA and has a high error rate, creating DNAs sequences different of which should be [4].

The retroviruses replication process can be described basically in follow steps [1]:

- Viral recognition by the receptors present in the host cell surface;
- Penetration into the host cell;
- Reverse transcription (RNA to DNA)
- Viral integration to the host's genome, where it will replicate;

- Viral DNA translation (produces viral mRNA that will translated in viral proteins)
- Viral assembling
- Viral shedding, when the new viruses leave the host cell.

One of the proteins of the virus is the integrase, which is still associated with provirus. This enzyme cuts the chromosomal DNA of the host cell and inserts the viral-converted DNA, integrating the provirus into the host cell chromosome (Fig. 1.). The next time this infected cell divides, the provirus will be replicated to the daughter cells [1]. After the viral genome is integrated in the host cell genome, the virus will be totally dependent of the cellular metabolism to continue its process of transcription, translation, genome replication, viral assembling and shedding.

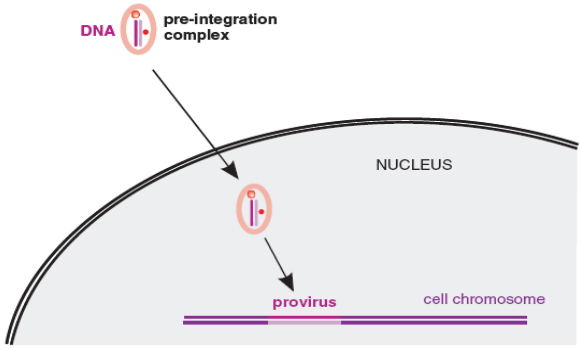

Fig.1. Provirus creation and reverse transcriptase.

The concepts of Charles Darwin about reproduction and Natural Selection applied to organics forms are also applied to viruses. Even though their acceptance as an organic form is very questionable, the viruses have genes that are striving to perpetuate the species. The main mechanisms used for viral evolution are mutation, recombination, reassortment and acquisition of cellular genes [1].

III. Genetic Algorithms

Genetic algorithms are part of probabilistic techniques and may find different solutions in different executions with the same parameters, even with the same population [5]. Some of the main advantages of GA are [2]:

- Optimization of discrete and continues values.
- Simultaneous search.
- Possibility to work with many variables.
- Provide a set of solutions, instead of only one specific solution.

The fundamental principle is to explore a space of search by a population of chromosomes, whose evolution depends on mutation and crossover operations, as in the natural evolutionary process [5].

The basic GA steps are:

- Initialize random population.
- Evaluate chromosomes and check solution.
- Selection to crossover.
- Effect crossover.
- Mutation.

Evaluate the chromosomes and check solution, if the solution was not found then go back to step 2.

A. Viral Infections in Genetic Algorithm

The viral infection is not an innovation in GA. It was discussed other times like in VEGA [10] and GAVI [6]. In both methods is used another population composed by virus, called viral population and infection of chromosomes called transcription [6][10].

In VEGA the viral population is a subset of chromosomes, created from initial hosts [10].

During the process of infection, a virus is selected by rank process. However, as the virus has no fitness value, because it doesn't have a complete solution to be evaluated, it is used a parameter of infection, called *fitvirus* [10]. This is an indicator of how well the virus has acted. After selecting the virus that will effect the infection (in a particular chromosome), that time is made the transcription, which consists in the modification of the infected chromosome by the viral information that contains a section similar to the one represented by the infecting agent [6, 10].

When a virus is created or modified, its infectivity level is set at a fixed initial value. If the virus infects a chromosome (increase chromosome's fitness) their level of infectivity is increased by 1 (one), otherwise (if turn down the fitness) this value is reduced by 1 (one). If the virus infectivity value reaches 0 (zero) the virus discard it parts and copy a portion of chromosome to itself [6].

The main difference between VEGA and GAVI is because GAVI uses viral infection as operator, ignoring the operator of mutation, and VEGA is a complete GA [6, 10].

IV. Retroviral Iterative Genetic Algorithm

The main reason for the use of viral structure in the algorithm is the fact that these viruses are associated with a source of genetic innovation, which is influenced by the rapid rate of replication and changing [9].

For biological inspiration of RIGA, the family of retroviruses was chosen. These viruses do not possess correction mechanisms to undo possible genetic mutations that occur naturally during viral multiplication, which causes a high mutation rate, arising genetically modified individuals at each generation, what is considered an important characteristic during the processing time of GA [2, 8]. The acquisition of cellular genes was chosen as a method to viral evolution, since it is quite common in retroviruses [8].

There are so many differences between RIGA and GAVI and VEGA, some of them are:

- RIGA doesn't change any GA component, GAVI remove the mutation operation.
- In the GAVI the worst viruses has them genetic material changed, in the RIGA they are completely changed, thereby, the viral population is constantly remade, increasing the possibility of infection.

- The biological basement of RIGA is very specific for the use of retroviral structure.
- VEGA creates virus only from host chromosomes, RIGA creates virus from host chromosomes and a new virus (providing high diversity of genetic material viral).
- VEGA and GAVI handle a virus as a sequenced subset of chromosome, in the other hand, RIGA handles virus with dispersed subset.
- The viral lifecycle and parameters in RIGA are well-defined.

In the next sessions are presented the components of RIGA.

A. *Viruses*

Viruses in RIGA are structures that have the same size of a chromosome, however, with some empty spaces, because the idea is to share genetic material and avoid other population of chromosomes working in parallel. The amount of empty spaces and its provisions are determined randomly. Thus for a problem that requires a binary representation of eight positions, some viruses have information as seen in Fig 2.

(A)		1		0		1	0	
(B)			1		1			1
(C)	0			1		1		

Fig.2. Possible viruses for a binary chromossome with eight positions.

B. *Viral Population and Creation of New Viruses*

For creation of the new viruses that will compose the viral population, RIGA was inspired by the natural process common in retroviruses called reverse transcription. The process consists basically of the following steps (Fig 3):

(A)		1		1		1	0	
(B)	0	1	1	1	1	0	0	1
(C)	0	1	1	1	1	1	0	1
(D)	0	1		1				1

Fig. 3. Creating a new virus process (A) Random virus (B) Chromossome from population (C) Auxiliar chromossome containing a mix of both genetic materials (D) New virus containing genetic material from virus (A) and chromossome (B).

C. *Infection*

Infection is the process of inclusion of the viral genetic material into the host chromosome, which is required a virus and a chromosome. The target chromosome will have changed their genetic material in the same positions where the genes are arranged on the virus, so all the viral genetic information, will be copied to the target chromosome, excepting the empty spaces, which will be filled by the host chromosome. The RIGA infection is represented in Fig 4.

(A)		1		1		1	0	
(B)	0	1	1	1	1	0	0	1
(C)	0	1	1	1	1	1	0	1

Fig. 4. Chromossome infection (A) Virus (B) Chromossome (C) Infected Chromossome.

The infection process depends exclusively on one single factor: increasing of chromosome fitness. The infection is successful when an infected chromosome has an increase on its fitness and unsuccessful when it has a decrease on its fitness. This is an important factor because it determines which viruses will infect the next generation. The worst viruses (with less infection) will be extinct. The reason of the infectious process is restrict to the increase in fitness, because if was considered any kind infection, good chromosomes could become bad. Thus, only the successful infections are important to RIGA.

D. *Parameters*

The RIGA uses the same parameters from classic GA (number of individuals, rate of mutation, crossover and elitism and the type of selection and crossover). However, to apply the concepts of viral infection by retroviruses, the defined parameters are:

1. Infection population rate: the rate of chromosses that will be infected in generation. This parameter will cause a slow execution if the value is higher than 50.
2. Viral elitism rate: the rate of viruses that will be kept in the next generation of viral population, if the value is more than 50, the evolution is compromized.
3. Number of viruses: the number of viruses of viral population, the higher that number, the lower is the time of execution.
4. Weakest infection: this parameter forces the infection of weakest chromossome, cause a fast execution, but results in more generations.
5. Single infection: this parameter forces a unique infection per chromossome, cause a fast execution, but results in more generations.
6. Internal infection rate: this parameter indicates the maximum percentual of genetical material from any chromossome that will form a new virus.

E. *The Algorithm*

RIGA has an additional step in the traditional algorithm (step 6), above:

- Initialize random population.
- Evaluate chromosomes and check solution.
- Selection to crossover.
- Effects crossover.
- Mutation.
- Viral application
  - If it is the first time, generate a random viral population.
  - Generate new virus based in chromosome population.

- Infect chromosomes.
  - Infect each randomly chosen chromosome with each existent virus.
  - For each successful infection, increment 1 to virus, however, decrement 1.
  - Check the virus with highest infection rate and keep according with viral elitism rate.

– Evaluate the chromosomes and check solution, if the solution wasn't found then go back to step 2.

V. Results

To the test of RIGA, the implementation was based to support the functions: F1 (Shifted Sphere Function), F2 (Shifted Schwefel's Problem), F3 (Shifted Rotated High Conditioned Elliptic Function) and F5 (Schwefel's Problem 2.6 with Global Optimum on Bounds) [7].

Considering all experiments performed (a total of 80 executions for 40 using RIGA and 40 using GA with same configurations per function and initial population), the RIGA was superior to GA in all the experiments, getting closer to the optimal result in all of them (Fig. 5 to Fig. 8).

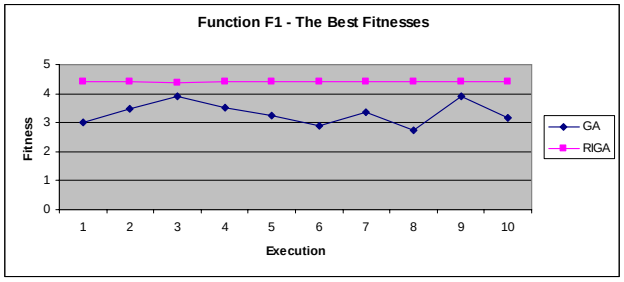

Fig. 5. The best fitnesses for function F1.

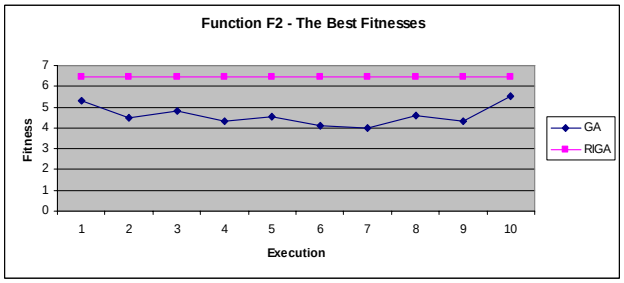

Fig.6. The best fitnesses for function F2.

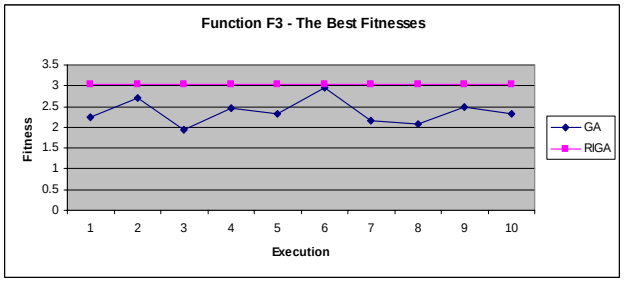

Fig. 7. The best fitnesses for function F3.

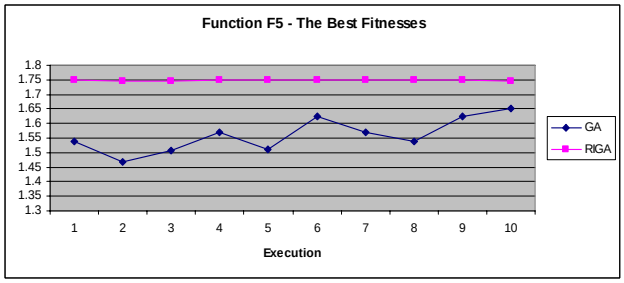

Fig. 8. The best fitnesses for function F5.

According to Table 1, it is possible verify that the average fitness of the RIGA was even higher (F1, F2, F3 and F5) than the best fitnesses found in some executions by the GA. Thus, we can affirm that the RIGA, with the same population, elevates the fitness of individuals considered bad, getting closer to an optimal result in only 100 generations generated by RIGA which were lower in all cases.

TABLE I
THE BEST EXECUTIONS FOR FUNCTIONS F1, F2, F3AND F5

CEC-LIB		GA		RIGA	
F(x)	*Max(F(x))*	*BF*	*AF*	*BF*	*AF*
F1	$4{,}4 \times 10^4$	**3,9214**	1,0015	**4,3893**	3,9341
F2	$6{,}5 \times 10^4$	**5,4998**	1,4292	**6,4784**	5,6315
F3	$3{,}8 \times 10^{10}$	**2.9530**	0.5893	**3.0421**	2.7696
F5	$1{,}8 \times 10^4$	**1.6513**	0.6740	**1.7486**	1.6474

(BF) Best Fitness (AF) Average Fitness

Thereby, it is correct to affirm that the method (RIGA) was more efficient for mathematical problems exposed.

VI. Conclusions

This work presented the creation of a genetic algorithm based on the structure of the retroviruses family. The biological application in conjunction with genetic algorithm took this algorithm to be called Retroviral Iterative Genetic Algorithm.

After the present work, it was found that the retroviral biological inspiration had a beneficial effect in the application of RIGA compared to the GA, since all the RIGA results were significantly better when compared to GA results.

Thus, with positive results obtained by RIGA, we conclude that RIGA proves to be more efficient when compared to GA for the functions exposed in the work. Still, this should not be taken as a rule for any problem involving resolutions of GA's problems.

References

[1] J. Carter and V. Saunders, *Virology Principles and Applications*. John Wiley & Sons ltd., England, 2007.
[2] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*. John Wiley & Sons ltd., England, 1998.
[3] S. Hogg, *Essential Microbiology*. John Wiley & Sons ltd., England, 2005.
[4] A. Agut, "Um Sistema Estratégico De Reprodução," *Scientific American Brasil*, Edição Especial, N. 28, p. 14-19, São Paulo, 2009
[5] R. Linden. *Algoritmos Genéticos 1*. Ed. Rio De Janeiro: Brasport 2006.

[6] A. Guedes, J. Leite, D. Aloise, “Um Algoritmo Genético Com Infecção Viral Para O Problema Do Caixeiro Viajante,” *Revista PublICa*,Ano IV, vol 4, 2005.
[7] P. Suganthan, N. Hansen, J. Liang, K. Deb, Y. Chen, A. Auger, S. Tiwari, “Problem Definitions and Evaluation Criteria For The Cec 2005 Special Session On Real-Parameter Optimization,” *Technical Report*, Nanyang Technological University, Singapore and Kangal Report Number 2005005 (Kanpur Genetic Algorithms Laboratory, Iit Kanpur), 2005.
[8] M. Mitchell, *An Introduction to Genetic Algorithms*. MIT Press, 1999.
[9] L. Villarreal, “Virus São Seres Vivos?” *Scientific American Brasil*, Edição Especial, n. 28, p. 21-24. São Paulo, 2009.
[10] N. Kubota and T. Fukuda and K. Shimojima, “Virus-evolutionary genetic algorithm for a self-organizing manufacturing system,” *Computers & Industrial Engineering*, vol 30, pp. 1015-1026, 1996.