SciForum
Mol2Net

# Genetic Algorithms with Fine Tuning

**Marcos Gestal[1,*], Julián Dorado[2]**

[1] RNASA-IMEDIR Group, Faculty of Computer Science, University of A Coruna, 15071 A Coruña, Spain; Emails: mgestal@udc.es, julian@udc.es

* Correspondent author: Marcos Gestal, Information and Communication Technologies Department, Faculty of Computer Science, University of A Coruna, 15071 A Coruña, Spain; E-Mail: mgestal@udc.es; Tel.: +34-981-167-000; Fax: +34-981-167-160.

*Published: 4 December 2015*

**Abstract:** Genetic algorithms are search and optimization techniques which have their origin and inspiration in the world of biology. They provide very good results in different kind of problems, but they are not free of complications. One of the most common problems that may arise with these techniques is that, despite a few generations obtain an approximation to the solution of the problem, they need considerably more to adjust to the final solution. To solve this problem Nature gives us, another time, a valid option. Fine Tuning techniques can model this transmission of knowledge between generations making slight variations in offspring before inserting it into the next generation. For its implementation, a new individual is generated from a solution (non best), changing slightly their genes. It can be performed by means a new Genetic Algorithm, with a lower number of individuals and its own configuration. On this way, solutions avoid local minima and introduce more variability in the global population that increase the possibilities to achieve the best solution. The developed solution uses this approach within a generic tool that makes possible that the user provides their own fitness functions to add any kind of problems. The software will allow to parametrize the execution and will show several graphics to control the evolution. Furthermore, to minimize the time for obtaining solutions the assessment of individuals is made under a distributed scheme. The control of the implementation of Genetic Algorithm will be made from a master computer, which delegated to other slave devices for evaluation and, if necessary, apply fine tuning.

**Keywords:** genetic algorithms, fine adjustment, fitness evaluation, distributed evaluation

## 1. Introduction

Genetic Algorithms (GA) [1,2] are one of the techniques included within the field of Evolutionary Computation. Since they were introduced in the work of Holland, they provided excellent results in very different fields of applications [3-10]. The main advantage of this technique is related with the fact that the user only need to know a way to evaluate the goodness of a solution to rank it instead of the need to know a formal/deterministic way to solve it.

But one of the most common issues of this kind of algorithms is related with the local minima: GAs are so good to make an approximation to the optimal solutions, but need a huge number of generations to reach the best solution.

## 2. Results and Discussion

This paper presents a proposal to prevent the AG can run around a local minimum trying to improve the fitness of genetics individual in each generation

There are different methods to locally improve the fit of an individual [11-14]: hill-climbing, neural networks, fitness sharing... But most of these methods present a similar problem: their dependence with the problem that the GA tries to solve. Our approach is based on perform fine adjustment by a recursive use of GA, so dependency problem is solved. Roughly Fine Tuning technique using GA consists on apply a new (brief) GA evolution to each individual in the population so it attempts to model the transmission of knowledge from one to another generation by means of the acquisition of information of each individual (see Figure 1).

The application of Fine Adjust techniques tries to conquer that during the implementation of GA no stagnation occurs in areas of local minima. Furthermore, these small variations in the offspring attempts to prevent the GA explore with greater profusion to the desired pass these areas and exploring new regions of the search space (increase heterogeneity).

Of course this approach introduces an overhead due the bigger number of evaluations required. To solve this question, a master-slave schema is proposed, as Figure 2 represents.

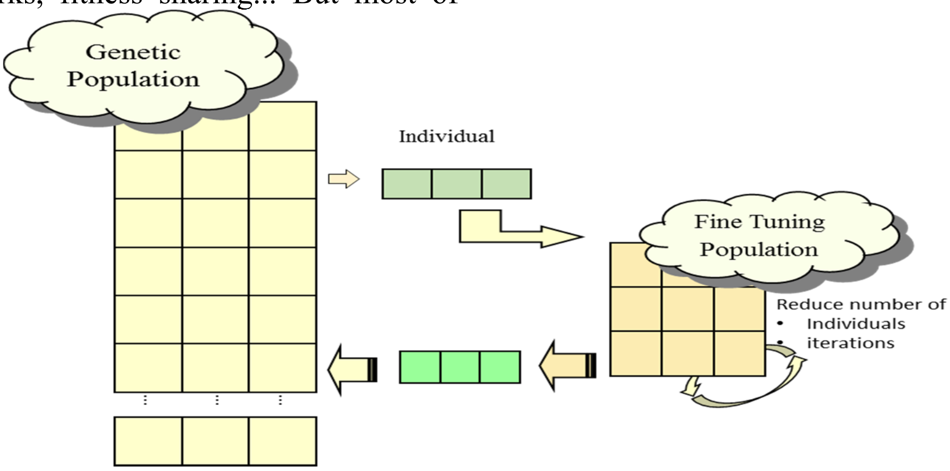


**Figure 1.** General schema for Fine Tuning

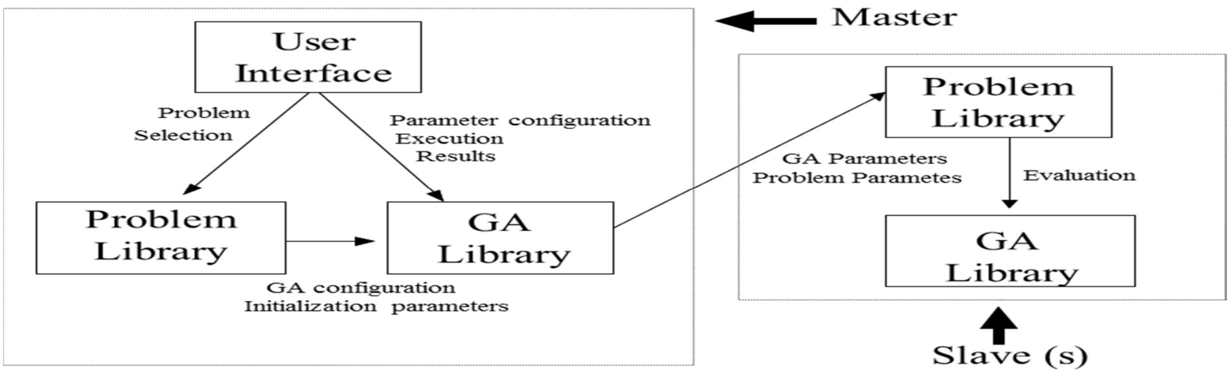


**Figure 2.** Communication schema between subsystems under distributed evaluations

## 3. Materials and Methods

The approach explained in this short paper is part of a Phd. Thesis [11] where it was used as support for several methods developed to address the feature selection process under multimodal search spaces [12-15]. The tool was implemented within a library in VisualC++ [16] to leverage its high performance and the user interface was developed in Delphi [17] to produce a good user experience when setting the method, check the evolution, configure the distribution or reviewing the results (see Figure 3)

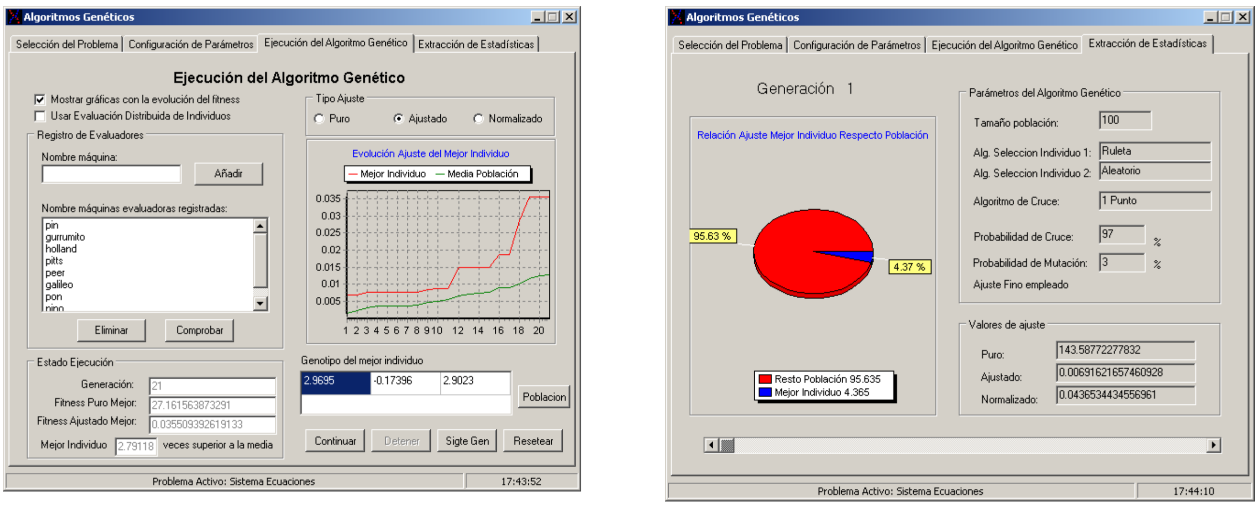


**Figure 3.** Screenshots to check the execution (left) and results (right) phases

One of the advantages of the system is the possibility to add new problems in an easy way.

If a user want to optimize this problem using GAs (and optionally using the Fine Tuning approach) he only needs to provide a library problem (a DLL file) that describes the way how the fitness is evaluated in several functions to allow the communication with the user interface. These functions are the following:

- void FAR PASCAL Inicializar()

  *This function is called from the user interface to initialize the problem, so it should read the datasets, initialize the genetic individuals*

*(number of genes, limits for each one of them…)*

- void FAR PASCAL Finalizar()
  *Releases memory and other locked resources.*

- void FAR PASCAL GetNumeroVariables()
  *It is used to pass the number of genes required graphical interface*

- __declspec(dllexport) float Evaluar(float* genotype, int nGenes)
  *The most important function. The genetic library will call it, so it is necessary to use this nomenclature to allow that remote call. It will return the fitness of the individual passed as argument*

- __declspec(dllexport)void SetParametros(char **lArgs, float *argsNum)
  *This function, called from the distributed evaluator, will indicate to the slave equipments the problem parameters.*

## 4. Conclusions

This short communication presents a simple modification over the Genetic Algorithm standard functioning to reduce the number of evaluations required to achieve the final solution by means of a fine tuning approach. This new approach is coded within a tool that also allows to distribute the computation requirements along different machines in an asynchronous way to reduce the computational time needed. This tool include a library with the evolutionary algorithm, where the user can add their own functions with a minimum effort and an elemental programming skills. Finally, the tool offers a graphical interface that provides an easy way to parametrize the problem, check the GA evolution and review the results of the execution.

## Acknowledgments

The authors acknowledge the support provided by the Galician Network of Drugs R+D REGID (Xunta de Galicia R2014/025) and by the "Collaborative Project on Medical Informatics (CIMED)" PI13/00280 funded by the Carlos III Health Institute from the Spanish National plan for Scientific and Technical Research and Innovation 2013-2016 and the European Regional Development Fund / GAIN (FEDER - CONECTAPEME - INTERCONECTA). This work was partially supported by the Galician Network for Colorectal Cancer Research (Red Gallega de Cáncer Colorrectal - REGICC, Ref.: CN 2012/217), Institute for Biomedical Informatics of A Coruña (INIBIC), and Center for Research of Information and Communication Technologies (CITIC).

## Conflicts of Interest

The authors declare no conflict of interest.

**References and Notes**

1. Goldberg, D. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley Longman Publishing Co.: Boston, MA, 1989.
2. Holland, J. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press: 1975.
3. Fernandez-Lozano, C.; Canto, C.; Gestal, M.; Andrade-Garda, J.M.; Rabunal, J.R.; Dorado, J.; Pazos, A. Hybrid model based on genetic algorithms and svm applied to variable selection within fruit juice classification. *Scientific World Journal* **2013**.
4. Gomez-Carracedo, M.P.; Gestal, M.; Dorado, J.; Andrade, J.M. Chemically driven variable selection by focused multimodal genetic algorithms in mid-ir spectra. *Anal. Bioanal. Chem.* **2007**, *389*, 2331-2342.
5. Picado, H.; Gestal, M.; Lau, N.; Reis, L.P.; Tome, A.M. Automatic generation of biped walk behavior using genetic algorithms. In *Bio-inspired systems: Computational and ambient intelligence, pt 1*, Cabestany, J.; Prieto, A.; Sandoval, F.; Corchado, J.M., Eds. 2009; Vol. 5517, pp 805-812.
6. Li, S.; Mehra, R.; Smith, R.; Beard, R. In *Multi-spacecraft trajectory optimization and control using genetic algorithm techniques*, Aerospace Conference Proceedings, 2000 IEEE, 2000, 2000; pp 99-108 vol.107.
7. de Croon, G.C.H.E.; O'Connor, L.M.; Nicol, C.; Izzo, D. Evolutionary robotics approach to odor source localization. *Neurocomputing* **2013**, *121*, 481-497.
8. Gomez-Carracedo, M.P.; Gestal, M.; Dorado, J.; Andrade, J.M. Linking chemical knowledge and genetic algorithms using two populations and focused multimodal search. *Chemom. Intell. Lab. Syst.* **2007**, *87*, 173-184.
9. Ramadan, Z.; Jacobs, D.; Grigorov, M.; Kochhar, S. Metabolic profiling using principal component analysis, discriminant partial least squares, and genetic algorithms. *Talanta* **2006**, *68*, 1683-1691.
10. Cantú-Paz, E. In *Feature subset selection, class separability, and genetic algorithms*, Genetic and Evolutionary Computation Conference, Seattle, Washington, USA, 2004; Springer-Verlag: Seattle, Washington, USA, pp 959–970.
11. Gestal, M. Computación evolutiva para el proceso de selección de variables en espacios de búsqueda multimodales. Universidade da Coruña, A Coruña, 2009.
12. Gestal, M.; Dorado, J. Genetic algorithms in multimodal search space. In *Encyclopedia of information science and technology*, Information Science Reference: Hershey, USA, 2009.
13. Gestal, M.; Gómez-Carracedo, M. Finding multiple solutions with ga in multimodal problems. In *Encyclopedia of artificial intelligence*, Information Science Reference: Hershey, USA, 2009; pp 647-653.
14. Gestal, M.; Vázquez-Naya, J.M.; Ezquerra, N. Genetic algorithms and multimodal search. *Advancing Artificial Intelligence Through Biological Process Applications* **2008**, 231.
15. Rabunal, J.R.; Dorado, J.; Gestal, M.; Pedreira, N. Diversity and multimodal search with a hybrid two-population ga: An application to ann development. In *Computational intelligence and*

*bioinspired systems, proceedings*, Cabestany, J.; Prieto, A.; Sandoval, F., Eds. 2005; Vol. 3512, pp 382-390.
16. Kruglinski, D.J.; Wingo, S.; Sheperd, G.W. *Programming microsoft visual c++*. Microsoft Press: 1998; p 1153.
17. Pacheco, X.; Teixeira, S. *Delphi 5 developer's guide*. Sams: 2000.