International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:1, 2007

# “Intuition” Operator: Providing Genomes with Reason

Grigorios N. Beligiannis, Georgios A. Tsirogiannis and Panayotis E. Pintelas

***Abstract*****—**In this contribution, the use of a new genetic operator is proposed. The main advantage of using this operator is that it is able to assist the evolution procedure to converge faster towards the optimal solution of a problem. This new genetic operator is called “intuition” operator. Generally speaking, one can claim that this operator is a way to include any heuristic or any other local knowledge, concerning the problem, that cannot be embedded in the fitness function. Simulation results show that the use of this operator increases significantly the performance of the classic Genetic Algorithm by increasing the convergence speed of its population.

***Keywords*****—**Genetic Algorithms, “Intuition” operator, Reasonable genomes, Complex search space, Non linear fitness functions.

## I. Introduction

GENETIC Algorithms (GAs) are known to be one of the most effective methods for searching and optimization [1]–[14]. By applying genetic operators (reproduction, crossover and mutation) in a population of individuals (sets of unknown parameters properly coded), they achieve the optimum value of the fitness function, which corresponds to the most suitable solution. As a result, they converge to the (near) optimal solution by evolving the best individuals in each generation. The main advantage of the GAs is that they use the parameters’ values instead of the parameters themselves. In this way they search the whole parameter space. However, GAs encounter some serious problems (concerning the convergence speed and the finding of the exact value of the global optimum) when they have to deal with optimization problems including too many local optima.

In this contribution, the use of a new genetic operator, called “intuition” operator, is proposed. The main advantage of including this operator in the well known set of genetic operators (reproduction, crossover and mutation) is that it is able to assist the classic GA to deal with situations where the search space is very complex or the fitness function used has many discontinuities. As shown by experimental results, this new operator helps significantly the evolution procedure of the classic GA to converge faster towards the optimal solution of a problem.

The motivation of using such an operator derives from the need to assist the population of the classic GA in order to be able to evaluate the quality and the importance of each possible solution. This evaluation regards not only the value of the fitness function, that each possible solution achieves in each generation, but also its “perspective” as far as the general form of the search space and the global solution of the problem are concerned.

The paper is organized as follows. In section II the proposed genetic operator is described and analyzed. In section III experimental results are presented in order to prove the significance and the efficiency of the proposed genetic operator. Finally, section IV summarizes the conclusions and suggests future applications and extensions.

## II. “Intuition” Operator

Experimental results have shown that GAs, when applied in order to optimize a complex problem using a strongly non linear fitness function with many discontinuities, are facing many problems especially as far as convergence speed and entrapment in local optima are concerned. Although they are usually able to reach a relative good score (compared to the global optimum) in a small number of generations, it takes them many generations (computational time) in order to refine the solution space and succeed in identifying the exact optimal solution of a problem. As known, classic GAs make use of three basic genetic operators (selection, crossover and mutation) in order to evolve the population of possible solutions to fit to the conditions and the characteristics of each specific problem. However, most of the times these three genetic operators are not capable of including the dynamics and the form of the solution space in order to assist the GA’s convergence.

The evolution procedure of the classic GA wastes a large amount of evaluations of the objective function in order to refine a local optimum solution which is often abandoned in the next generations due to the application of the mutation

G. N. Beligiannis is with the Department of Computer Engineering and Informatics, University of Patras, 26500 Rio, Patras, Greece, with the Department of Computer Applications in Administration and Economics, Technological Educational Institute of Mesologgi, 30200 Mesologgi, Greece and with the Research Academic Computer Technology Institute (R.A.C.T.I.), Riga Feraiou 61, 26221 Patras, Greece (corresponding author, phone: +30-2610-997755; fax:+30-2610-997706; e-mail: beligian@ceid.upatras.gr).

G. A. Tsirogiannis is with the Department of Engineering Sciences, University of Patras, Rio, Patras, Greece (e-mail: g.tsirogiannis@des.upatras.gr).

P. E. Pintelas is with the Department of Mathematics, University of Patras, Rio, Patras, Greece, (e-mail: pintelas@math.upatras.gr).

This contribution was financially supported by the Greek Ministry Of Education and Religious Affairs under grant “PYTHAGORAS: Supporting Research Groups in Universities”, b.365.018 (EPEAEK II).

operator. This is because the use of the mutation operator can often lead to much better solutions compared to the ones found so far, directing the evolution of the algorithm to another area of the search space, in which a local optimum with higher objective value is located. So, there is need to incorporate into classic GAs evolution procedure a function or an operator able to succeed better exploration of the search space and avoid premature convergence and entrapment in local optima.

In this contribution, a new genetic operator, to be included in the classic GA's armory, is proposed, so as to assist the evolution procedure to achieve better global exploration of the solution space while executing the minimum possible number of functional evaluations. In order to achieve this goal, except for the standard global exploration mechanism, used by classic GAs (selection, crossover, mutation), we apply this new operator so as to preserve the global search procedure. This technique alleviates the enormous computational burden introduced by entrapment in local optima, which is quite often useless in finding the optimal solution.

As known, a well designed and constructed fitness function is able to describe in a very satisfactory way the environment (solution space) to which the population of a GA is trying to adapt. However, in most cases, the classic GA's evolution procedure is totally dependent on the dynamics of the specific GA. The genetic operators (selection, crossover and mutation) comprise the sole evolution mechanism having the responsibility to adapt the genomes of the GA so as to be as close as possible to the optimal solution of each problem.

As stated before, for each specific problem, the classic GA uses the form of the fitness function in order to describe the solution space. It would be very helpful to provide the genomes of the population with the ability to evaluate themselves not only as far as their appropriateness is concerned (that is adequate only when they are already very close to the optimal solution), but also regarding the intuition of how close they are to the optimal solution. This can be accomplished by including a new operator in the evolution procedure. This operator will favor those genomes that are believed to be (or heading to) near the optimal solution.

Someone may argue at this point that this criterion can be included in the fitness function of a GA. However, this may not lead to certain convergence to the optimal solution. That is because, this criterion although it is capable of sending genomes away from local optima (not letting them spending calculations in order to find a useless local optimum) it has limited knowledge concerning the whole search space and as a result it is not necessarily able to lead to the optimal solution by itself. At this point, we have to mention that using "intuition" operator aims not to get rid of all local optima, since they comprise a significant part of the problem's definition, but to assist genomes in order not to be trapped in them. Local optima are important in order the description of the search space made by the fitness function to be as complete as possible.

The "intuition" operator that is proposed in this contribution is playing exactly this role. It is able to identify and prevent situations in which although the value of the fitness function is smaller in the current generation compared to the previous one (when minimization is concerned), the optimal genome found till that generation proves to be not so "optimal" because it leads the population of the GA in entrapment in a local optimum. So, the "intuition" operator is the one providing "reason" to the genomes of the GA, so as not to reproduce based solely on the evolution dynamics of the genetic operators but also based on the evaluation that they are able to do on themselves. Based on this self-evaluation they can either impose their existence in the next generations or retire if they are not appropriate according to their reason.

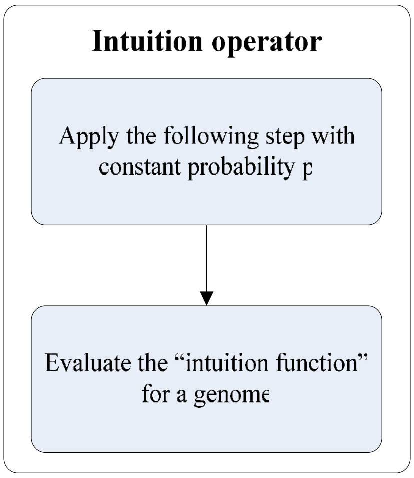


FIGURE 1(A)
THE "INTUITION" OPERATOR

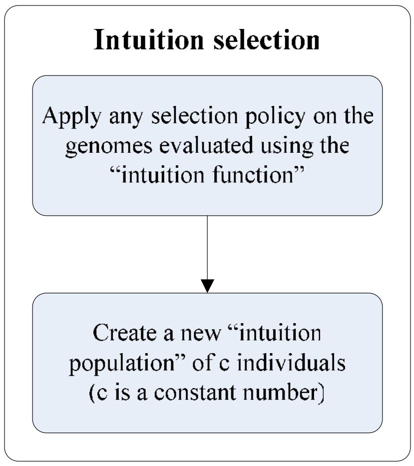


FIGURE 1(B)
THE "INTUITION" SELECTION

We propose this new genetic operator to be applied in all genomes in the same way that the existing ones (selection, crossover and mutation) are applied, that is, with a probability *p* (Fig.1(a)). The operator will evaluate each genome based on the "intuition" function which will be specific to each different optimization problem. Next, in the selection phase, some (a specific percentage of the population size) genomes of the population of the current generation will be reproduced to the next one based on the value of the "intuition" function achieved (Fig.1(b)). This procedure is exactly the same as the selection procedure based on the value of the fitness function achieved by each genome. So, any known selection method (for example roulette wheel selection) can be used in order to implement "intuition" operator. The genomes selected by "intuition" function and the genomes selected by the reproduction operator used, are merged and passed to the next generation. This procedure is repeated until the termination criterion is satisfied. The structure of the evolution procedure of the classic GA including the proposed "intuition" operator is presented in Fig.2.

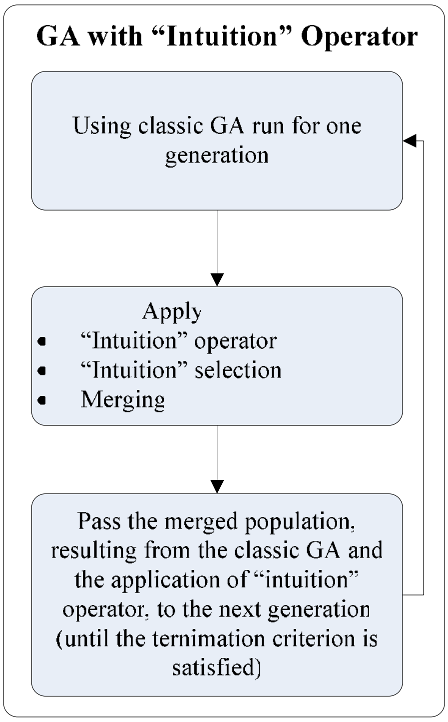


FIGURE 2
THE EVOLUTION PROCEDURE OF THE CLASSIC GA INCLUDING THE PROPOSED "INTUITION" OPERATOR

As someone can easily conclude, by using this new operator we affect positively only those genomes that are believed to be good approximations of the optimal solution without affecting negatively or impeding the evolution procedure of the GA. The main advantage of this approach is that it assists the GA's evolution procedure to deal successfully with complex problems, where the solution space cannot be satisfactory described by the fitness function or the fitness function used is strongly non-linear having many discontinuities. In such cases, based on experience, someone has most of the times useful clues concerning the evaluation of possible solutions. These clues due to the complexity of the solution space can even be contradictory and as a result it is not possible to be included in the same fitness function. This valuable knowledge can be included in the evolution procedure only by using the "intuition" operator presented in this contribution. This operator is responsible for providing with reason the genomes themselves and as a result it assists them to avoid local optima and useless evaluations.

## III. EXPERIMENTAL RESULTS

In order to demonstrate the efficiency and the performance improvement introduced by using the proposed genetic operator, several simulation experiments were carried out. All the experiments were carried out 100 times (100 Monte Carlo runs). In this section, we present the results of the application of both the classic GA with and without the proposed genetic operator to two well known optimization problems. The functions selected to be optimized are the first two functions of the De Jong test suite [15]. These functions are quite popular in GAs' literature, so it is possible to make direct comparisons.

The first De Jong test function is the sphere model:

$$f_1(x_1, x_2, x_3) = \sum_{i=1}^{3} x_i^2 \,, \; -5.12 \le x_i \le 5.12 \tag{1}$$

It is smooth, unimodal and symmetric. The goal is to find the global minimum $\min(f_1) = f_1(0,0,0) = 0$.

The second De Jong test function is the Rosenbrock's valley:

$$f_2(x_1, x_2) = 100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2 \,, \; -2.048 \le x_i \le 2.048 \tag{2}$$

It has a very narrow ridge. The tip of the ridge is very sharp and it runs around a parabola. The goal is to find the global minimum $\min(f_2) = f_2(1,1) = 0$.

In most minimization problems, the fitness function used by classic GAs is the function, which needs to be minimized, itself. The "intuition" operator proposed for this problem aims in including reason, in the evolution procedure, which will enable genomes to detect whether they may lead in the next generations in finding the optimal (minimum) solution. The proposed "intuition" operator for these problems is the use of one step of the steepest descent method [16], that is

$$\mathbf{x}^{k+1} = \mathbf{x}^k - a^k \nabla f\left(\mathbf{x}^k\right) \tag{3}$$

where $\mathbf{x}$ is the solution vector, $k$ is the current generation, $a$ is a constant number (equal to 0.01 in our experiments) and

$$\nabla = \begin{bmatrix} \dfrac{\partial f}{\partial x_1} \\ \dfrac{\partial f}{\partial x_2} \\ \vdots \\ \dfrac{\partial f}{\partial x_m} \end{bmatrix} \tag{4}$$

The value of the "intuition" function, used by the "intuition" operator to select useful genomes, is the following:

$$\left\| \mathbf{x}^{k+1} - \mathbf{x}^k \right\| \tag{5}$$

It is obvious that, computing this value enable us to track whether it is possible to improve a specific solution. Although this value constitutes a local evaluation criterion and cannot be included in the fitness function, it is a "good" clue of whether the evolution procedure is heading to the global optimum or not.

For both the classic GA with and without the proposed genetic operator the same set of GA's operators and parameters were used in order to have a fair comparison of their efficiency and performance. The representation used for the genomes of the genetic population is real number representation. As far as the reproduction operator is concerned, the classic biased roulette wheel selection was used. The crossover operator used is uniform crossover [17] (with crossover probability equal to 0.9), while the mutation operator is gaussian mutator [17] (with mutation probability equal to 0.001). The size of the population both for the classic GA with and without the proposed genetic operator was set to 50, while the percentage of the genomes selected by "intuition" operator to pass to the next generation equals 30% (in our case 15 genomes). Except for that, both GAs used linear scaling and elitism.

Both GAs were implemented using the C++ Library of Genetic Algorithms GAlib [17] and especially the GASimpleGA class for the implementation of the GAs (non-overlapping populations) and the GARealGenome class for the real valued genomes (an implementation of an array of real values). All the experiments were carried out on a Intel Pentium IV 2.7GHz PC with 256 MB RAM.

The comparison of the algorithms is based on two criteria. For both test functions two specific quantities are taken into consideration. The first one is the value achieved by the fitness function of each algorithm. We measure the number of fitness function evaluations made by each algorithm in order the value of the fitness function to overcome a predefined threshold. The second quantity is the number of fitness function evaluations. We measure the best value of the fitness function achieved by each algorithm for a specific number of fitness function evaluations.

In the following table the performance and efficiency of both the classic GA with and without the proposed genetic operator is shown for the first De Jong function.

TABLE I
EXPERIMENTAL RESULTS FOR THE FIRST DE JONG FUNCTION

| Performance Criterion | Classic GA | Classic GA with "intuition operator" |
|---|---|---|
| Fitness function value | Number of evaluations | |
| <1.0e-10 | 20768 | 19567 |
| <1.0e-16 | 35290 | 33724 |
| Number of evaluations | Fitness function value | |
| 10000 | 1.69e-06 | 1.15e-06 |
| 20000 | 8.99e-10 | 2.3e-10 |

In the following table the performance and efficiency of both the classic GA with and without the proposed genetic operator is shown for the Second De Jong function.

TABLE II
EXPERIMENTAL RESULTS FOR THE SECOND DE JONG FUNCTION

| Performance Criterion | Classic GA | Classic GA with "intuition operator" |
|---|---|---|
| Fitness function value | Number of evaluations | |
| <1.0e-4 | 1238271 | 110693 |
| <1.0e-8 | Not able after 4000000 | 236556 |
| Number of evaluations | Fitness function value | |
| 50000 | 3.51e-02 | 3.27e-04 |
| 100000 | 2.09e-02 | 5.39e-05 |
| 200000 | 1.55e-02 | 6.12e-07 |

## IV. CONCLUSIONS AND FUTURE WORK

As experimental results show, the proposed genetic operator manages to significantly enhance the performance of the classic GA, especially in solving complex problems with strongly non linear fitness functions having many discontinuities. It would be very interesting to check the efficiency and performance of the proposed "intuition" operator to other difficult test functions and NP-Hard problems like the TSP problem. These issues will be the main scope of our future work.

## REFERENCES

[1] T. Bäck, D. B. Fogel and Z. Michaelewicz, Eds., *Handbook of Evolutionary Computation*, Institute of Physics Publishing Ltd and Oxford University Press, 1997.
[2] W. Banzhaf, C. Reeves and C. R. Reeves, Eds., *Foundations of Genetic Algorithms 1999 (FOGA 5) (Foundations of Genetic Algorithms)*, Morgan Kaufmann, 1999.
[3] L. Chambers, Ed., *Practical Handbook of Genetic Algorithms: Applications: Volume I*, Boca Raton, FL: CRC Press, 1995.
[4] L. Chambers, Ed., *Practical Handbook of Genetic Algorithms: New Frontiers*. *Volume II*, Boca Raton, FL: CRC Press, 1995.
[5] L. Chambers, Ed., *Practical Handbook of Genetic Algorithms: Complex Coding Systems*: *Volume III*, Boca Raton, FL: CRC Press, 1999.
[6] D. A. Coley, *An introduction to Genetic Algorithms for Scientists and Engineers*, Singapore: World Scientific, 1999.
[7] L. Davis, Ed., *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold, 1991.
[8] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Reading, Mass.: Addison-Wesley, 1989.
[9] R. Haupt, S. E. Haupt, *Practical Genetic Algorithms*, John Wiley & Sons Inc, 1998.
[10] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed., N. Y.: Springer-Verlag, 1996.
[11] K. F. Man, K. S. Tang, S. Kwong and W. A. Halang, *Genetic Algorithms for Control and Signal Processing*, London: Springer-Verlag, 1997.
[12] K. F. Man, K. S. Tang, S. Kwong and W. A. Halang, *Genetic Algorithms: Concepts and Designs*, London: Springer-Verlag, 1999.
[13] M. Mitchell, *An Introduction to Genetic Algorithms (Complex Adaptive Systems)*, Cambridge, Massachusetts, London, England: A Bradford Book, The MIT Press, 1998.
[14] M. Vose, *The Simple Genetic Algorithm: Foundations and Theory*, Cambridge, MA: The MIT Press, 1999.
[15] K. De Jong, "*A*n analysis of the behaviour of a class of genetic adaptive systems", PhD thesis, University of Michigan, 1975.
[16] E. W. Weisstein, "Method of Steepest Descent", MathWorld – A Wolfram Web Resource. Available: http://mathworld.wolfram.com/MethodofSteepestDescent.html
[17] GAlib - A C++ Library of Genetic Algorithm Components, Matthew Wall, Massachusetts Institute of Technology (MIT). Available: http://lancet.mit.edu/ga/