ANNALS of Faculty Engineering Hunedoara – International Journal of Engineering
Tome XIV [2016] – Fascicule 1 [February]
ISSN: 1584-2665 [print; online]
ISSN: 1584-2673 [CD-Rom; online]
a free-access multidisciplinary publication
of the Faculty of Engineering Hunedoara

1. Nenad MARJANOVIĆ, 2. Nenad KOSTIĆ,
3. Nenad PETROVIĆ, 4. Miloš MATEJIĆ, 5. Mirko BLAGOJEVIĆ

GENETIC ALGORITHM PARAMETER CONTROL FOR ACHIEVING BETTER OPTIMIZATION PERFORMANCE

1-5. University of Kragujevac, Faculty of Engineering, Kragujevac, SERBIA

ABSTACT: This research is directed towards controlling genetic algorithm operator parameters. Simulations have been done in MatLab on examples taken from literature for genetic algorithm testing. Based on a large number of simulations with different parameter values, algorithm operator values are attained experimentally. An analysis of results has been completed in Statistica, as well as the creation of nonlinear equations for the correlation between operators and results. By optimizing the derived equations it is possible to determine general parameter values of operators which will have beneficial optimization performances, in terms of convergence. One equation which gives the best optimization values is favored. Attained values are again tested on new examples which define achieved performance and benefits of this approach. These results lead to a simplified use of the genetic algorithm for practical optimization with satisfactory results. This approach has a practical engineering optimization use perspective.
Keywords: genetic algorithm, parameter control, operators, MatLab, mathematical mode

1. INTRODUCTION

Optimization, as an alternative solution finding approach, is represented in a wide field of disciplines, and its practical use is most commonly found in engineering. The optimization process implies the use of an optimization method. Due to their good performance heuristic methods are in massive use, which include the Genetic algorithm. The evolutionary character of the genetic algorithm implies operators (operations) of selection, crossover and mutation, and since there is a rapid development of operator algorithm it is not easy to set and control them in order to have an adequate optimization. Bad operator characteristics lead to early convergence, slow convergence, increased time of optimization and an increase of never reaching the optimum. Due to a large possibility of setting operators the user usually has no alternative choice of adequate settings.

Numerous researchers have worked on the quality of genetic algorithm use, convergence characteristics, and possibility of optimization. These researches are generally oriented on specific operator's for real coded genetic algorithm (RCGA).

Author Saber M. Elsayed et al. [1] have oriented their research on the analysis of the work of the crossover as an alternative to new Genetic algorithms useful for optimization. Manoj Thakur et al. [2] oriented their work on crossover and mutation, as segments of the genetic algorithm. Authors used LX-PM method RCGA which had modified crossover and created the BEX approach (bounded exponential crossover). Kusum Deep and Manoj Thakur [3] were oriented on RCGA operators, more specifically on LX. Authors have combined this crossover with already developed mutations MPTM and NUM in order to achieve a new genetic algorithm LX-MPTM and LC-NUM respectively. Values are compared to HX-MPTM and HX-NUM. C. García-Martínezet et al. [4] researched parent-client crossover. Authors suggested steps which they think will increase the efficiency of RCGA. The first step is separating individuals in the population to male and female. The next step they suggest is to have different selections make choices from different parents and determine an adequate model of convergence. Authors [5] Kusum Deep et al. have looked into RCGA for integer and mixed integer variables. The algorithm has a special procedure which is oriented on whole number values. The developed algorithm has good characteristics for this group of problems. Research in [6] by Manoj Thakur presents a new concept of RCGA for nonlinear problems. The author is oriented on the operator crossover and analysis of existing combinations of crossover and mutation.

This approach develops an approach for developing optimization problems where alternative settings for genetic algorithm parameters are suggested. The base motivation of this approach is to rationalize complex use of the genetic algorithm for optimization in practice and finding new optimum parameter values of genetic algorithm operators for practical optimization for the general case. This approach is useful for defining optimization characteristics of the genetic algorithm and determining alternatives for achieving real optimums for cases when solutions cannot be anticipated. The work of the newly achieved solution is tested on examples which confirm and verify correctness of the achieved solution. Testing and analysis is done in MatLab and Statistica software.

2. PROBLEM DESCRIPTION

2.1. Genetic Algorithm

Genetic Algorithm (GA) is a heuristic method for optimization whose work is based on simulating natural/evolutionary processes [7]. The algorithm consists of three base operators: reproduction, crossover and mutation. Reproduction is the process of transferring genetic information through generations. Crossover represents the process-operation between two parents, which interchanges genetic information and generates posterity. The mutation operator makes random changes in the genetic structure of some individuals in order to connect the change of convergence. The algorithm is based on the survival of the fittest individuals through evolution which change genetic material. Through selection individuals are ranked in the population using values of the fitness function, which defines the capabilities/qualities of the individual.

The genetic algorithm due to its convergence characteristics has a wide use. Researchers are inspired to use this algorithm for science, industrial applications, basic applications, and to make their use widespread.

Table 1. Test functions

No	Test name	Test function	Constraints	Optimum
1.	Test function 1 [3,6]	$f_1 = -(\exp(-0.5\sum_{i=1}^{N} x_i^2))+1$	[-1,1]	0(min)
2.	Test function 2 [3,4,6,7]	$f_2 = \sum_{i=1}^{N} x_i^2$	[-5.12,5.12]	0(min)
3.	Test function 3 [3,6]	$f_3 = \sum_{i=1}^{N} (x_i^4 + rand(0,1))$	[-10,10]	0(min)
4.	Test function 4 [6]	$f_4 = x_1^2 + 1000000 \sum_{i=2}^{N} x_i^2$	[-10,10]	0(min)
5.	Test function 5 [5]	$f_5 = \left(-y + 2x - \ln\left(\frac{x}{2}\right)\right) - 2.1244$	0.5≤x≤1.5; 0≤y≤1 $-x - \ln\left(\frac{x}{2}\right) + y \leq 0$	0(min)
6.	Test function 6 [6]	$f_6 = \sum_{i=1}^{N} (x_i)^2$	[0,100]	0(min)
7.	Test function 7 [5]	$f_7 = 2x + y - 2$	0≤x≤16; 0≤y≤1 $1.25 - x^2 - y \leq 0$; $x + y \leq 1.6$	0(min)
8.	Test Function 8 [7]	$f_8 = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$	[-2.048,2.047]	0(min)

2.2. Test Function

For the purposes of this research relevant frequently used examples from literature were selected for testing of the Genetic Algorithm. An analysis for eight different examples is presented. Examples are chosen so that they are of various difficulty and multimodality. Most problems are scalable, which means that the decision maker can increase or decrease the number of variables in the function. All examples are combined in table 1. Examples with constraints are presented so the analysis for this research can be relevant. All examples are cited in table 1, and the examples were chosen by most cited in literature. Specific examples are changed from the originals in order to have the optimum for all examples with the same value. The reason for this modification represents the possibility to find an optimal performance operator for the greatest number of functions, which will be presented in more detail in the results.

Result verification is also done on test functions with another analysis, and the results are compared to software suggested values for operators of these functions. All experiments are done on an Intel CORE i3-3240 CPU 3.40 GHz computer with 8GB of RAM on a Windows 7 platform.

3. RESULTS

In this paper an experimental testing of the Genetic Algorithm has been conducted on a large number of adequate examples. For every example simulations were repeated for various operator values and characteristic values of the results were recorded. Based on values of input settings and achieved results nonlinear multi-structural regression equations were developed which can define the dependence of operator parameters for the same case of solving optimization problems with constraints. Basic criteria for creating equations is the correctness of achieved results. Based on this equation an operator parameter setting can be made which provides beneficial characteristics of convergence.

Testing has conducted on the behavior of HX (Heuristic Crossover) operator on a large number of examples. This research is oriented on the influence of the HX operator on optimization results, with a final goal of defining some specific values of this operator. The range of setting values for the HX operator ranges in the interval from 0 to 2, while a specific value can help optimization with problems where it is not possible to determine the best value for this setting. For specific problems the best convergence is derived from adapting HX values from the given range, which in practice is not always possible. With the value which is satisfactory to results it is possible to conduct a quality optimization, which is the goal of this paper, as well as the use of optimization in engineering practice.

In order to determine convergence speed for examples which do not explicitly defined values a large number of variables was used for constraining algorithm time intervals. This way optimal results are not achieved, since the algorithm cannot reach the real optimum, but converge towards those values. The choice and setting of operators defines the speed of convergence based on which the equations were created. Values of the optimized values are presented in table 2.

Table 2. Heuristic crossover parameter results

Name		Opt. value	Time limit	# of simulation	# of variable	Constraint func.	Crossover ratio: 0.0	Crossover ratio: 0.5	Crossover ratio: 1.0	Crossover ratio: 1.5	Crossover ratio: 2.0
Test func. 1	best	0	20 sec.	30x5=150	30	no	0.90284510600	0.03475812100	0.06545024100	0.00009662390	0.00080700300
	avg.						0.99398457400	0.13263802200	0.30741712600	0.00035142000	0.00470082000
	worst						1.00000000000	0.48650960500	0.93291158900	0.00073420200	0.03692853300
Test func. 2	best	0	20 sec.	30x5=150	30	no	7.48627990500	0.02929606900	0.00384625100	0.00001623250	0.00000737860
	avg.						15.58617341000	0.28167728200	0.05151431700	0.00003529620	0.00004414630
	worst						24.54464403000	1.53026820300	0.25344547300	0.00007868040	0.00036697800
Test func. 3	best	0	20 sec.	30x5=150	30	no	15.99353917000	4.64463766100	1.70020871500	1.17270382000	3.59471880400
	avg.						30.50509300000	12.07185489000	6.68366857900	4.35938410800	7.29124250800
	worst						58.14283255000	28.34496009000	20.03638112000	9.27870054000	18.77810629000
Test func. 4	best	0	20 sec.	30x5=150	30	no	7458184.753	292.0400701	40.122263140	0.006418277	0.027187654
	avg.						15604862.240	47015.79392	477.999024300	0.019190627	0.033753973
	worst						33297561.100	177557.11980	4157.200569000	0.042293008	0.043896129
Test func. 5	best	0	20 sec.	30x5=150	2	yes	0.00000304508	0.00001188030	0.00000670010	0.00002491400	0.00004989480
	avg.						0.00156263000	0.00216017100	0.00063731900	0.00068906200	0.00248713200
	worst						0.00725642400	0.02735261900	0.00276832200	0.00274241000	0.01848527700
Test func. 6	best	0	20 sec.	30x5=150	30	no	6.53542756900	0.05673746500	0.00005763840	0.00000711918	0.00000904067
	avg.						14.81218252000	1.16646987800	0.00044592100	0.00002199010	0.00002208570
	worst						32.10100418000	4.87615733300	0.00148654600	0.00004140040	0.00004803390
Test func. 7	best	0	20 sec.	30x5=150	2	yes	0.00055712100	0.00067695000	0.00063446900	0.00045233100	0.00063334600
	avg.						0.12653839500	0.19055044100	0.13721519100	0.04527148100	0.03402193200
	worst						0.24849232800	0.31783833000	0.23996942600	0.23662597900	0.23550767000
Test func. 8	best	0	20 sec.	30x5=150	2	no	0.00000003234	0.00000000000	0.00001865140	0.00000000000	0.00000000040
	avg.						0.22940481100	0.00002041490	0.11226248500	0.00000000028	0.00000000619
	worst						1.07668034400	0.00020638500	0.90096766300	0.00000000402	0.00000002925

Based on these results using a statistical regression analysis using multi-structural regressions an equation was reached for the dependency of parameters to the optimal result.

$$f(HX) = g(t_i) + c$$

In the equation $g(t_i)$ is the partial influence of the test function on the operator value, and c is a constant. The regression analysis is done in Statistica software. Since examples were chosen so that the optimal value of all test functions should be zero, the following equation is true.

$$f(HX) = c$$

Where c represents a constant with a number value which the HX operator should be set to in order to have the best possible results in the general case regarding convergence speed. In order to verify the results a comparative analysis was conducted comparing suggested values in MatLab with the statistically attained value.

The simulations were repeated an additional 30 times for all eight examples for both cases, with the same constraints. Results for each function are partially presented in figure 1, for the HX statistic and suggested values.

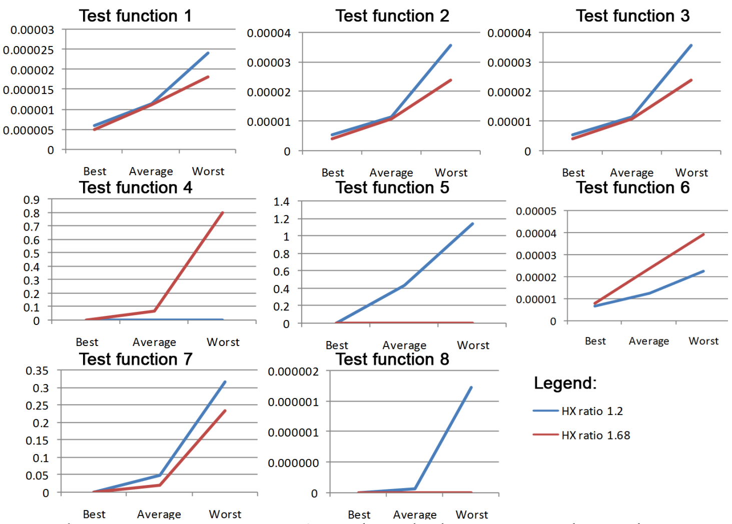

Figure 1. Structure of Real coded Genetic Algorithm

4. CONCLUSIONS

Results confirm that for the chosen test functions the attained value for the HX ratio of 1.68 gives better results than the software suggested value of 1.2. Obviously the achieved value cannot be the best for absolutely all examples, which can be seen in figure 1. Generally speaking in the vase where the decision maker has no alternative for the choice of HX, this value will surely be beneficial for use.

Values derived from optimization using this approach with the attained setting do not necessarily give the best results, since they cannot be the best at the same time for all examples. These solutions can be considered optimal for the general case if the algorithm cannot be adopted to the mathematical model.

The attained value of the setting is just one of the possible solutions and is considered to be the suggested value, however there is a possibility of finding other solutions if the test functions were different, if the statistic values were differently processed, etc.

Note: This paper is based on the paper presented at The 12th International Conference on Accomplishments in Electrical and Mechanical Engineering and Information Technology – DEMI 2015, organized by the University of Banja Luka, Faculty of Mechanical Engineering and Faculty of Electrical Engineering, in Banja Luka, BOSNIA & HERZEGOVINA (29th – 30th of May, 2015), referred here as[8].

References

[1] Saber M. Elsayed, Ruhul A. Sarker, Daryl L. Essam, A new genetic algorithm for solving optimization problems, Engineering Applications of Artificial Intelligence, Volume 27, January 2014, Pages 57-69, ISSN 0952-1976

[2] Manoj Thakur, Suraj S. Meghwani, Hemant Jalota, A modified real coded genetic algorithm for constrained optimization, Applied Mathematics and Computation, Volume 235, 25 May 2014, Pages 292-317, ISSN 0096-3003

[3] Kusum Deep, Manoj Thakur, A new crossover operator for real coded genetic algorithms, Applied Mathematics and Computation, Volume 188, Issue 1, 1 May 2007, Pages 895-911, ISSN 0096-3003

[4] C. García-Martínez, M. Lozano, F. Herrera, D. Molina, A.M. Sánchez, Global and local real-coded genetic algorithms based on parent-centric crossover operators, European Journal of Operational Research, Volume 185, Issue 3, 16 March 2008, Pages 1088-1113, ISSN 0377-2217

[5] Kusum Deep, Krishna Pratap Singh, M. L. Kansal, C. Mohan, A real coded genetic algorithm for solving integer and mixed integer optimization problems, Applied Mathematics and Computation, Volume 212, Issue 2, 15 June 2009, Pages 505-518, ISSN 0096-3003

[6] Manoj Thakur, A new genetic algorithm for global optimization of multimodal continuous functions, Journal of Computational Science, Volume 5, Issue 2, March 2014, Pages 298-311, ISSN 1877-7503

[7] Kedar Nath Das, Rajashree Mishra, Chemo-inspired genetic algorithm for function optimization, Applied Mathematics and Computation, Volume 220, 1 September 2013, Pages 394-404

[8] Nenad Marjanović, Nenad Kostić, Nenad Petrović, Miloš Matejić, Mirko Blagojević,Genetic algorithm parameter control for achieving better optimization performance, Proceedings of the 12th International Conference on Accomplishments in Electrical and Mechanical Engineering and Information Technology – DEMI 2015, Banja Luka, Bosnia & Herzegovina