2015 International Conference on Information Technology

Decision Points in Application of Genetic Algorithm

Biplab Mahapatra
PhD Scholar,
Xavier University,
Bhubaneswar, India
Biplab_Mahapatra@yahoo.com

Sanjay Mohapatra
Information System
Xavier University
Bhubaneswar, India

***Abstract*—Selecting run time parameters while applying Genetic Algorithm is a crucial decision. Numerous theoretical and empirical researches are performed and guidelines are suggested on this, researches are conducted but still an overall acceptable guideline is lacking. In this work the parameters are discussed with reference to the existing literature and suggested a guideline to follow for initial setup based on literature.**

Keywords—Genetic Algorithm, Optimization, Application, Key Decision Points, crossover, mutation

I. Introduction

Optimization problems, ranging from simple to complex, are ubiquitous in reality. Many of these problems are NP hard problems. Due to the complexities of these problems in the presence of constraints and limit on the processing time, traditional mathematical models fail to give satisfactory results. The failure of these traditional mathematical models inspired researchers to look for alternative techniques and Evolutionary algorithms, inspired by natural selection, came to the front. Since then Evolutionary Algorithms, specifically Genetic Algorithms(GA), are used for optimization in different areas like Designing of efficient buildings[1], Land use planning, Autopilot controller design, Airlines Crew rostering[2] etc. The problem domain information independence nature, makes GA less susceptible to the shape of the objective function.

Even though Genetic Algorithm (GA) is being used in different fields, the practitioner needs to take judicious decisions related to the parameters to be used in the process and also needs to go through the control parameters tuning process to make this optimization technique effective. Without proper tuning process the results will not be optimal or near-optimal. In this paper an attempt is made to create reference points on the parameters to be tuned based on the corresponding references available in the literature. The motivation is to create a reference point, which will help the researchers as well practitioners with a starting point in the process. To the best of the knowledge of the author, there is no such article available.

The remainder of this paper is organized in the following order. Section II provides short introduction on the genetic algorithm, Section III detailed out the decision parameters and the corresponding references available in the literature. Finally Section IV provides conclusion.

II. Background

Genetic Algorithm (GA), part of Evolutionary algorithm, was made popular by Holland [3]. Subsequently many other researchers worked on this area and contributed to the development of the discipline. Like Evolutionary Algorithms, search in GA starts with a randomly generated population. Then each individual's fitness is determined by the objective function formulated based on the problem being solved. Once the fitness scores are determined, next step is to select the best solutions from the population. These solutions then go through the process of crossover to generate the next generation of solutions as in nature. Some individuals, based on mutation rate, will also go through the mutation process. Again the fitness scores of the new generations are calculated and the process is repeated till the predefined number of generations, specified by the user, are processed or objective function reached a solution where further improvement is not possible.

In reality the optimization problems are not single-objective as there always are conflicting objectives, which need to be balanced based on user's preferences or further qualitative knowledge. Therefore the purpose of the multi-objective evolutionary algorithm is to find the pareto-front of the solution set, created by varying degrees of tradeoffs between the various conflicting objective functions[4][5]. Pareto optimal solutions are those solutions where one objective function cannot be improved further without degrading the other objective functions. Also that is not the end as there will always be constraints. In case of multi-objective optimization there will never be a solution where both the conflicting functions are minimized/maximized to the fullest [6].

978-1-5090-0487-4/15 $31.00 © 2015 IEEE
DOI 10.1109/ICIT.2015.38

III. DECISION PARAMETERS

While applying Genetic algorithm to a problem to create optimum or near optimum solution, there is a need to take decisions at every point starting from the encoding scheme, Population size, crossover mechanism, crossover rate, mutation rate, to the stopping criteria. Other equally important points to be considered are software package to be used, algorithm to be used specifically in case of multi-objective Genetic Algorithm. Judicious decisions related to these aspects needed to be taken for successful application of GA. The below diagram depicts the decision points to be considered:

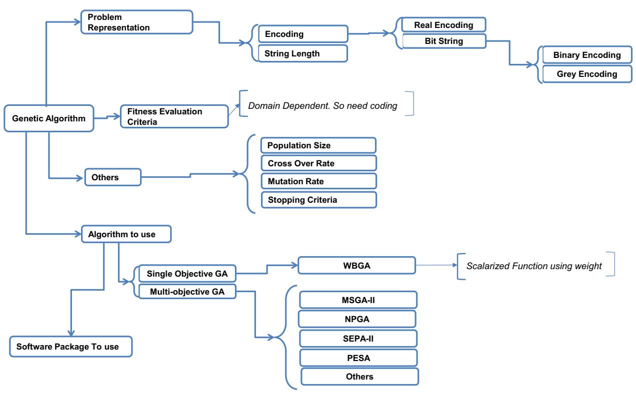

Figure 1: Decision Points

In the subsequent sections the individual points are explained in more detail.

A. Paramater Representation

In Genetic Algorithm process the first step is to represent the parameters of the problem. The practitioners faced with two options and a judicious decision needed to be taken based on the knowledge of the solution space. In Real-coding parameters are represented using real numbers where as in Bit-String coding the parameters are represented by binary digits (i.e. 0, 1). In Bit String the entire string is made of individual substrings representing the individual parameters. Again the bit string coding is classified into two categories based on the way the substrings are arranged within the string. In case of Binary String there is no specific order to maintain where as in case of Grey coding binary substrings are arranged in such a way that adjacent numeric values differ by one bit only. In case of Grey coding, the mutation process is likely to induce small changes in the real parameter value. As per Carunana and David [7], Grey Encoding can remove the hidden bias in the Binary Encoding. Also the large *Hamming Distance* i.e. number of bit positions that differ in adjacent bit strings of equal length, resulting from mutation in the binary coding can disturb the search space to the extent that locating global optima efficiently might not be possible. Real-coded GA, using real numbers as parameter, is mainly suitable for problem spaces which are continuous [2]. Binary coding can be also be used to handle both with right string length. As per Back et al. real coding [8] is expected to be superior to Binary coding. Due to simplicity and ease of handling the Binary encoding is much more popular. Also Binary coding is more closer to the basic principle of genetic algorithm as this is pseudo-chromosomal representation of a solution [2].

B. String Length

Choosing the right string length needs knowledge of the solution space and the requirement related to the degree of precision. Deb [2] gave the following guide line while choosing the right length:

$$x_i = x_i^{min} + \frac{\left(x_i^{max} - x_i^{min}\right)}{2^{l_i} - 1} DV(s_i)$$

Where,
l_i = String length used to code the i-th variable and
$DV(s_i)$ = decode value of the string
This represents the challenge of having the knowledge of the solution space.

But this is a guide line only and the judicious decision needs to be taken while applying the formula and deciding on the string length.

C. Population Sizing

Population size i.e. the number chromosomes or encoded strings, used to represent population plays an important role in the efficient processing of the algorithm. Smaller population size can cause the process to converge pre-maturely on a suboptimal solution whereas larger population size will take more time to converge as it needs to process more population for fitness. Different studies [9], [10] attempted to give a guidance on population size. Harik et al. [11] proposed a model to help user to determine the adequate population size to reach a solution of a given quality. But as indicated by Deb [2] Population size depends on the complexity of the problem being solved. As per Gothsall and Raylander [12] the population size depends on the nature of the problem and should be analyzed for determining the population size. Increase in population size will increase the number of generation needed for convergence but it also increases the probability of having an optimal solution in the initial population. So there is a need to tune the population size to get optimal solution.

D. Selection

Selection is the implementation of the natural selection process i.e. "*Survival of the form that will leave the most copies of itself in successive generations*" [13]. This is commonly known by the term "*survival of the fittest*", coined by Herbert Spencer. Darwin used this term in his book "On the Origin of Species", fifth edition [14]. The same process of selecting the best solutions and creating multiple copies of the same in the next generation is the aim of the selection process. There are different ways of implementing the selection process like Proportionate Selection, where the solutions are assigned copies proportional to the fitness values, tournament selection, where the fitness of the two solutions are compared with each other and the better one is chosen. The Proportionate Selection method is implemented through Roulette Wheel Selection method or Stochastic Remainder Roulette-wheel Selection (SRRWS) or Stochastic Universal Sampling (SUS). Computationally SUS is better than Roulette Wheel Selection or SRRWS as there is no need to generate random numbers multiple time. But proportionate selection has scaling issue due to large computational complexity. Goldberg [15] suggested an alternate method to overcome this but tournament selection does not suffer from scaling issue. Other method like Ranking Selection also does not suffer from scaling issue. Zhong et al. [16] experimented to compare Roulette Wheel Selection with Tournament Selection and found that Tournament Selection is more efficient in convergence. Goldberg and Deb concluded that Ranking and Tournament selection outperform proportionate selection. And out of these two they suggested binary tournament selection because of more efficient time complexity. Razali and Geraghty[17] tested with the TSP problem and concluded that Tournament selection is more appropriate for small problem but for larger Rank based selection is suggested. Also suggested that if solution quality is of more importance compare to computation time then Rank Based selection strategy is best.

In this article, the common methods of selections are mentioned but the list of the selection mechanism available in the literature is quite long. In case on tournament selection different varieties of the selection mechanism are available like Larger Tournament Selection, Boltzmann tournament selection, Restricted Tournament Selection, Correlative Tournament Selection .Similarly Ranking Selection mechanism can be of two types Linear Ranking Selection, Truncate Selection

As the number of options increase the decision making process becomes complex too.

E. Crossover

In the process of crossover two randomly selected parents exchange some portion of the genetic information with each other and create new offspring. The offspring might be better than the parents. In GA two important decision points need to be considered and they are: i) Crossover rate and ii) number of cross over points.

In the genetic algorithm process the crossover happens based on the user defined crossover rate. Also multiple crossover mechanisms are depicted in the literature like Single Point Cross over (where cross over happens at a single point), N-point cross over (where cross over happens at N points), Uniform cross over (where cross over happens at alternate positions), Intermediate cross over, heuristic crossover, Arithmetic cross over, Flat Crossover. Among these Single Points Cross over produces best result [18]. In case of order based problem, where order of the solution is important like Travelling Salesman Problem, ordered crossover method proposed by Goldberg [9] needs to be used. In case of ordered based crossover, two random crossover points are selected and designated as left, middle and right. In the crossover process the middle portion is swapped between the two parents to create children. There are different variants of order crossover also like One Point Crossover, Two Points crossover, Linear Ordered crossover etc. to name a few.

So while choosing the crossover operator, nature of the problem, coding mechanism chosen need to be considered

F. Mutation

Mutation operator helps in maintaining the diversity of the population and helps in escaping local optima. [23] There are different types of mutation based on type of encoding. The most common mutation operators are Polynomial Mutation and Gaussian Mutation. Deb et al. [19] used different combination of mutation operators and 5 different mutation schemes namely Usual Scheme (one variable every time), mutation clock (proposed by Goldberg) , one mutation per solution, fixed strategy mutation and diversity based mutation and compared the result with no mutation operator for real coded genetic algorithms. Conclusion is that mutation is better than not doing mutation and Mutation clock scheme is fastest. There are other forms of mutation operators like Cauchy Mutation, Adaptive Levy Mutation [20].

For binary encoding simple mutation operators are required. The probability of mutation i.e. the mutation rate is taken as 1/L where L is the length of the string.

The two existing mutation operators for real coded genetic algorithms are non-uniform Mutation, MPTM i.e. Makinen, Periaux and Toivanen Mutation.

Different from of mutation for order based problems proposed in the literature are Twors, Centre Inverse Mutation, Reverse Mutation, Thoras Mutation, Thrors Mutation, Partial Shuffling Mutation [24]. Abdoun et al. [24] tested with different Mutation operators using Travelling Sales Man problem and reached a conclusionthat RSM i.e. Reverse Sequence Mutation and PSM i.e. Partial Shuffle Mutation are better option for this type of problems.

G. Stopping Criteria

In case of optimization problem, most popular approaches are to run the algorithm for a pre-decided the time or to run for pre-decided no of iterations. As the optimal solution is not known beforehand there always is a possibility of not reaching the optimal solution using the pre-decided values of iteration or time. Also determining these values poses challenges. Aytug et al. [21] tried to provide stopping criteria as well as estimation of stopping time. Safe et al.[22] did critical analysis of the various aspects related to the termination conditions specifications. So always there is a need to do some experiment before finalizing the criteria.

H. Factors to decode in combination of other factors

The decisions related to the factors need to be considered in combination with other decision factors. Decision in standalone mode will not be an appropriate mode. Literature does not deal this facets in details.

I. Other Factors

There are other factors also for example elitism size, algorithms to use for Multi-objective GA, software library to use.

IV. Conclusion

Parameter tuning is an important part of the application of GA. In the literature different methods are described but no conclusive guideline is available for easy reference. Due to this issue while trying to attempt to apply the Genetic Algorithm the practitioner will face lots of decision points and judicious decisions need to be taken to get the optimal solution based on the domain and problem. Also the decisions are interlinked. An initial guideline which can be derived from the literature is to use Real Coding with ranking selection for selection mechanism, Single point crossover and mutation. But this is guideline. The validity of this will be checked in future with data points and further research will be carried out to come to a conclusion about the applicability of the reference points derived. It is evident from the literature that a need to do parameter tuning and experiment with different combinations based on the problem domain cannot be eliminated.

V. References

[1] A. Alajmi and J. Wright, "Efficient optimisation of buidling design using a genetic algorithm," *Loughborough University.*

[2] K. D. Deb, Multi-Objective Optimization using Evolutionary Algorithms, John Wiley and Sons Ltd, ISBN: 0-471-87339-X, 2001.

[3] J. Holland, Adaptation in Natural and Artificial Systems, Ann Arbor,MI: The University of Michigan Press, 1975.

[4] E. Zitzler, K. Deb and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results.," *Evolutionary Computation,* vol. 8, no. 2, pp. 173-195, 2000.

[5] E. Zitzler, M. Laumanns, L. Thiele, C. M. Fonseca and V. Grunert da Fonseca, "Why quality assessment of multiobjective optimizers is difficult.," in *Proceedings of the genetic and evolutionary computation conference (gecco'2002)*, San Francisco, California, 2002.

[6] X. Gandibleux and M. Ehrgott, 1984–2004 20 Years of multiobjective metaheuristics. But what about the solution of combinatorial problems with multiple objectives? Evolutionary multi-criterion optimization. Lecture notes in computer science, vol. 3410/2005, Berlin, Heidelberg: Springer-Verlag., 2005, pp. 33-46.

[7] R. Caruana and S. J. david, *Gray Vs. Binary Coding for Genetic Algorithm Function Optimizers,* Newyork: Philips Laboratories, North American Philips Corporation, 1987.

[8] T. Back and H. p. S. U. Hammel, "Evolutionary computation: Comments on the history and current state," *IEEE Transactions on Evolutionary Computation,* vol. 1, no. 1, pp. 3-17, 1997.

[9] D. E. Goldberg and M. Rudnick, "Genetic algorithms and the variance of fitness.," *Complex Systems,* vol. 5, no. 3, pp. 265-278, 1991.

[10] D. E. Goldberg, K. Deb and J. H. Clark, "Genetic algorithms, noise, and the sizing of populations," *Complex Systems,* vol. 6, pp. 333-362, 1992.

[11] G. Harik, E. C. -Paz, D. E. Goldberg and B. L. Miller, "The Gambler's Ruin Problem, Genetic Algorithms, and the Sizing of Populations," *Evolutionary Computation,* vol. 7, no. 3, pp. 231-253, 1999.

[12] S. Gotshall and B. Rylander, "Optimal population size and the genetic algorithm," *Population,* vol. 100, no. 400, p. 900, 2002.

[13] C. R. Darwin, On The Origin of Species By means of Natural Selection, John Murray, Albemarle Street, 1859.

[14] R. B. Freeman, The Works of Charles Darwin: An Annotated Bibliographical Handlist (2nd ed.), Cannon House, Folkestone, Kent, England: Wm Dawson & Sons Ltd, 1977.

[15] G. D. E, Genetic Algorithms for Search, Optimization and Machine Learning, Reading, MA: Addison-Wesley, 1989.

[16] J. Zhong, X. Hu and J. Z. M. Gu, "Comparison of Performance between Different Selection Strategies on Simple Genetic Algorithms," in *Proceeding of the International Conference on Computational Intelligence for Modelling, Control and automation, and International Conference of Intelligent Agents, Web Technologies and Internet Commerce*, 2005.

[17] N. M. Razali and J. Geraghty, "Genetic Algorithm Performance with Different Selection Strategies in Solving TSP," in *Proceedings of the World Congress on Engineering 2011 Vol II*, London, U.K., 2011.

[18] J. Magalhaes-Mendes, "A Comparative Study of Crossover Operators for Genetic Algorithms to Solve the Job Shop Scheduling Problem," *WSEAS Transactions on Computers,* vol. 12, no. 4, p. 164–173, 2013.

[19] K. Deb and D. Deb, "Analyzing Mutation Schemes for Real-Parameter Genetic Algorithms," *Int. J. Artificial Intelligence and Soft Computing,* vol. 4, no. 1, pp. 1-28, 2014.

[20] A. Koenig, "A Study of Mutation Methods for Evolutionary Algorithms.," *Advanced Topics in Artificial Intelligence, CS 447.,* 2002.

[21] H. Aytug and G. J. Koehler, "New stopping criteria for Genetic Algorithms," *European Journal of Operational Research,* vol. 126, pp. 662-674, 2000.

[22] M. Safe, J. A. Carballido, I. Ponzoni and N. B. Brignole, "On stopping criteria for genetic algorithms," in *Proc. of Advances in artificial intelligenceSBIA 2004, 17th Brazilian symposium on artificial intelligence*, S. L. E. A. L. C. Bazzan, Ed., Berlin, Springer,, 2004.

[23] E. E. E. Ali, "A Proposed Genetic Algorithm Selection Method," King Saud University, ccis,, 2006.

[24] O. Abdoun, J. Abouchabaka and C. Tajani, "Analyzing the Performance of Mutation Operators to Solve the Travelling Salesman Problem," *International Journal of Emerging Sciences,* vol. 2, no. 61, 2012.