Performance Evaluation of Methods for handling Premature Convergence in GA - Case of Grammar Induction

Nitin S. Choubey, Ph.D
Professor & Head,
Department of Computer Engineering,
SVKM's NMIMS, MPSTME, Shirpur

Madan U. Kharat, Ph.D
Professor & Head,
Department of Computer Engineering
MET's IOE, BKC, Nashik

ABSTRACT

Genetic algorithms are meta-heuristic algorithms based on the biological evolution. These algorithms are found to be useful for finding near to optimum results for the NP-category of problems. GA suffers with the disadvantage of premature convergence. The paper focuses on the implementation of various techniques of handling premature convergence and the statistical evaluation of the obtained results to identify the optimal method to the problem of grammar induction.

General Terms

Evolutionary Computations, Grammar Inference, Learning.

Keywords

Genetic algorithm, grammar induction, Grammar Inference, Statistical evaluation, t-test, F-test.

1. INTRODUCTION

Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest amongst string structures with a structured yet randomized information exchange to form a search algorithm with some sort of innovative flair of human search [1]. GA's have been developed by Holland, his colleagues and his students at the University of Michigan, for adopting the process of natural selection to design artificial systems with mechanism of natural evolution [2]. GA's are theoretically and empirically proven to provide robust search in the complex spaces. GA's works with the coding of parameters and search from the populations of point instead of single point. They require the natural parameter set of the optimization problem to be coded as a finite length string, chromosome, over some alphabet. The population of such coded strings is subjected to the natural process of selection and reproduction till they converge to a common solution or reach to a threshold value satisfying the criteria of optimization.

Grammar induction is the process of inferring grammar from the set of corpus. The generalizations sought in this research are languages. The focus is on grammatical inference (i.e. the inference of formal languages such as those of the Chomsky hierarchy from positive (and negative) sample strings). There are two different approaches for inducing grammar. The process of inferring grammar from the set of positive example is referred as Text Learning; whereas the process involving set of negative example is referred as Informant Learning Inductive inference involves making generalizations from examples [3]. Grammar induction has several practical applications outside the field of theoretical linguistics, such as structural pattern recognition [4] (in both visual images and more general patterns), automatic computer program synthesis and programming by example, information retrieval, programming language and bio-informatics. Syntactic processing has always been paramount to a wide range of applications, such as machine translation, information retrieval, speech recognition and the like. It is therefore natural language syntax has always been one of the most active research areas in the field of language technology [5]. All of the typical pitfalls in language like ambiguity, recursion and long-distance dependencies, are prominent problems in describing syntax in a computational context. The field of evolutionary computing has been applying problem-solving techniques that are similar in intent to the Machine Learning recombination methods

GA's suffer from the difficulty of local optimum convergence. It is the case when an extraordinary individual take over significant proportion of the finite population and leads towards the undesirable convergence. There are various techniques to avoid the premature convergence, such as Pygmy Algorithm, use of Incest Prevention, Crowding [9]. Introducing a Random Offspring in every generation adaptive mutation rate [8], immoderate crossover greediness and low influence of random factors [7], Social Disaster Technique, the population Partial Re-initialization, Dynamic Application of Crossover and Mutation Operators [8]. The authors have suggested a hybrid tool in the form of enhancement to Genetic algorithms process to handle the premature convergence [10]. The paper focuses on the performance evaluation of the proposed Hybrid system for handling premature convergence in GA.

2. GRAMMAR INDUCTION METHOD

The grammar encoding method [11] maps the binary chromosome in to its equivalent grammar by using three step processes. First step involves biasing the binary chromosome to produce Variable on the left side of every grammar rule to match with the Backus Naur Form (BNF). The second step involve the splitting of the equivalent symbolic form to get the production rules with desired length followed by simplification of the received grammar in third step. Figure 1. Shows the grammar encoding for the language L = $\{(10)^+\}$. The binary chromosome with size 300 is first divided in to the blocks of fifteen each. The block of three digits is then replaced with equivalent symbolic chromosomes.

BC	011 100 101 100 011	000 111 101 010 101	001 100 010 100 011	000 111 001 111 111	000 101 010 111 000	Variables S – 000 A – 001 B – 010 C – 011 Terminals 0 – 100 1 – 101 ? - 110 ? – 111 BC- Binary Chromosome SC – Symbolic Chromosome ULP/UFP - Use Less/ Use Full Productions
SC	C010C	S?1B1	A0B0C	S?A??	S1B?S	
ER	C→010C	S→1B1	A→0B0C	S→A	S→1BS	
ULP/UFP	ULP	ULP	ULP	ULP	ULP	
Final	DISCARDED	DISCARDED	DISCARDED	DISCARDED	DISCARDED	
BC	000 010 110 000 010	011 001 010 101 100	000 111 101 100 000	000 111 101 100 111	010 001 010 011 001	
SC	SB?SB	CAB10	S?10S	S?10?	BABCA	
ER	S→BSB	C→AB10	S→10S	S→10	B→ABCA	
ULP/UFP	ULP	ULP	UFP	UFP	ULP	
Final	DISCARDED	DISCARDED	S→10S	S→10	DISCARDED	
BC	000 001 011 000 001	001 100 101 010 110	010 000 010 110 000	010 011 010 010 111	000 010 000 000 010	
SC	SACSA	A01B?	BSB?S	BCBB?	SBSSB	
ER	S→ACSA	A→01B	B→SBS	B→CBB	S→BSSB	
ULP/UFP	ULP	ULP	ULP	ULP	ULP	
Final	DISCARDED	DISCARDED	DISCARDED	DISCARDED	DISCARDED	
BC	010 100 101 010 111	001 010 100 110 000	001 100 110 011 011	000 011 010 011 000	001 110 111 001 010	
SC	B01B?	AB0?S	A0?CC	SCBCS	A??AB	
ER	B→01B	A→B0S	A→0CC	S→CBCS	A→AB	
ULP/UFP	ULP	ULP	ULP	ULP	ULP	
Final	DISCARDED	DISCARDED	DISCARDED	DISCARDED	DISCARDED	

Figure 1. Grammar Encoding Scheme

The first symbol is regarded as left hand side variable while the remaining four symbols are taken as the Variable/terminals on the right hand side of the production. The grammar received is then processed through the simplification of grammar to get resultant grammar. The availability of "?" (Epsilon) symbol ensures the variable length productions

3. GA METHODS USED

Simple Genetic Algorithm (SGA) works by creating a random initial population of fixed length chromosomes. Each iteration (generation), the population evolves by means of the use of selection, crossover and mutation, which are the main genetic operators in GAs. Individuals are chosen based on their fitness measure to act as parents of offspring which will constitute the new generation. A GA is typically iterated for anywhere from 50 to 500 or more generations. The entire set of generations is called a run. At the end of a run there are often one or more highly fit chromosomes in the population. SGA uses a single crossover and single mutation operator for reproduction. An version suggested by Nicora[8] uses more than one crossover and mutation operator by allocating the crossover/mutation combination to reproduce the portion of child population based on their contribution in last generation.

One of the most important issues in GA referred as premature convergence. In Premature convergence a non-optimal genotype takes over the population which results in such a way that every individual in the population is either same or identical to the other. The population does not have sufficient diversity to evolve further. Increase in the population increases the cost of extra computation, the key to solve this problem is to bring sufficient diversity in the population on every stage so that the premature convergence does not takes place or to take steps in advance to avoid the premature convergence itself. Authors have suggested an enhanced version of Nicora's[8] approach in the form of Dynamic Allocation of Reproduction Operator (DARO) and an approach of maintaining an additional mating pool of Elite Individual (EMP) in every generation for handling premature convergence[17]. DARO found to be taking the efficient use of multiple reproduction operators at the cost of slightly delayed convergence where as EMP was costlier on account of creating and maintaining the additional Mating pool of the elite individual in every generation. An enhanced version which takes the used for the better part of both DARO and EMP is suggested in the form of Hybrid System for handling premature convergence in GA [10].

Various approaches are found in literature for avoiding premature convergence. The methods used for the purpose of comparison with proposed Hybrid method are Incest Prevention algorithm (IPA), Random Offspring Generation Algorithm (ROGA), The Social Disaster Algorithm (SDA), Partial Re-initialization Algorithm (PRA), Pygmy Algorithm (PA) and Crowding Algorithm (CA).

In IPA, two parents are selected with the difference of a predetermined threshold value with the view of not losing the sufficient information within both parents. The parents are chosen only if their hamming distance is greater than the threshold value [12]. In ROGA, It is ensured that the both parents are having different genetic material. If the selected parents have same fitness value either one or both randomly generated chromosomes are considered for the reproduction process. If one of the parents is randomly generated, the process is referred as 1-RO, it is 2-RO otherwise. The idea behind this is to check the genetic material before the reproduction process for its diversity [8]. SDA technique works with the entire population. In SDA, the entire population is processed with the warp operator, when any individual is found to taking over the entire population. The use of catastrophic operator is suggested in SDA to create diversity [7]. In PRA, a portion of the population is re-initialized in every generation. PA was originally applied to the problem of evolving minimal sorting networks, which must not only be able to sort numbers, but also in as few steps as possible. In the pygmy algorithm, two separate lists of the parents are maintained with two fitness function. The fitness functions are chosen based on the different criteria of the problem. The approach of two different fitness functions is likely to work better than approach having fitness function with the combination of different criteria [13]. The criteria in another fitness function in the experiment were minimum number of rules in the resultant grammar.

CA was introduced by De Jong [14] as a technique for preserving population diversity and preventing premature convergence. Crowding is applied in the survival selection step of GA. in order to decide which individuals among those in the current population and their offspring individuals will pass to the next generation. The De Jong's scheme of replacing offspring to the most similar parent was modified by Mahfoud [15, 16] by the scheme which can efficiently preserve diversity in the population [10].

4. EXPERIMENTAL SETUP AND RESULT ANALYSIS

The experiment is conducted with JDK 1.6D on Intel(R) Core(TM) i3-2350M CPU @ 2.30 GHz with 2.00 GB RAM. For selecting corpus, the strings of terminals from the given language are generated for the length, L, starting with L = 0 and gradually increasing L to get the required size to represent the language features. A corpus having twenty five positive strings and twenty five negative string sufficient to represent each language is chosen as input for grammar induction process. The fitness function is directly proportional to the number of accepted positive string and rejected negative strings where as inversely proportional to rejected positive string, accepted negative strings and number of rules available in Genetic algorithm. Equation (1) represents the fitness function.

$$fitness\ function = \frac{n(SA_p) + n(SR_n)}{n(SR_p) + n(SA_n) + n_r} * 100 \qquad (1)$$

Where, n(SA_p) &n(SR_n) is number of accepted positive & rejected negative strings respectively from the corpus whereas n(SR_p), n(SA_n) & n_r are number of rejected positive, Accepted negative strings from the corpus and number of rules in the resultant grammar. The fitness function is used to evaluate the fitness of the grammar equivalent to the binary chromosome. The crossover rate and mutation rate of 80% and 10 % respectively, the population size of 360 and the chromosome size of 300 is used for Genetic algorithm.

Table 1. Languages used for experimentation

Lang-uage	Language description	Standard set
L1	0* over (0+1)* over (0+1)*.	DuPont set
L2	Odd binary number over (0+1)*.	--
L3	Even binary number over (0+1)*	--
L4	(10)* over (0+1)*.	Tomita/DuPont set
L5	{0^n1^n} over (0+1)*.	Keller & Lutz set
L6	{$0n1^{2n}$} over (0+1)*.	--
L7	0*1 over (0+1)* over (0+1)*.	DuPont set

The experiment is repeated for fifteen randomly chosen language set for each of the methods discussed in section is II. Number of runs required to obtain a resultant grammar accepting all the positive samples and rejecting the entire negative sample were recorded. F-Test is conducted for the collected sample by assuming hypothesis that the difference within the sample is not significant at 5% level of confidence. F-test is also known as Fisher test. It is used to find whether two samples may be regarded as drawn from the normal population having same variance. Since F-test is based on the variance it is also known as variance ration test. F f-test in the analysis of variance (ANOVA) is used to assess whether the expected values of a quantitative variable within several pre-defined groups differ from each other. F-test is used here to verify whether samples collected from the results of various methods fall within the same group or is there any possibility of getting one or few of the members different from the group.

The formula used [18][19] for finding the table value and the calculated value of F is given in equation (2) & (3).

$$F\ Table = \frac{Number of\ Methods used - 1}{Number of\ Methods used * (Number of\ Samples - 1)} \qquad (2)$$

$$F\ Calculated \ = \ Number of\ Samples\ * \frac{Variance of\ \ Average\ for\ each\ Method}{Average of\ Variance\ for\ each\ Method} \qquad (3)$$

The total of fifteen samples was drawn from each method (Total of Ten methods). The Calculated value and the table value of F found to be 6.2816 and 0.06428 respectively. Since the calculated value is greater than the table value, the hypothesis is rejected. It indicates that one of the samples is better than other ones.

For further evaluation of the individual samples and testing the difference between them, t-test for the independent samples is conducted. The t-test is used when sample size is 30 or less and the population standard deviation is unknown. Since the collected samples are from the results of different methods for handling premature convergence, t-test for testing the means different between two independent samples is used for the study. To carry out the test the statistics is calculated as given below.

$$t = \frac{\overline{X_1} - \overline{X_2}}{S} \times \sqrt{\frac{n_1 n_2}{n_1 + n_2}} \qquad (4)$$

Where

$\overline{X_1}$ = Mean of the first sample,

$\overline{X_2}$ = mean of the second sample,

n_1 = number of observations in the first sample,

n_2 = number of observations in the first sample,

S = Combined standard deviation.

The value of S is calculated by the following formula

$$S = \sqrt{\frac{\sum (X_1 - \overline{X_1})^2 + \sum (X_2 - \overline{X_2})^2}{n_1 + n_2 - 2}} \qquad (5)$$

‘t’ test analysis for the methods used in the Experiment

DARO	-0.6835								
EMPA	-14.269	-19.208							
SGA	12.6749	13.4859	19.0858						
PIGMI	22.9360	23.4685	26.8845	13.1671					
CA	19.9453	20.5256	243331	9.5228	-3.7943				
IPA	17.3767	17.8687	21.506	7.74391	-4.9097	-1.2644			
PRA	21.2942	22.3615	27.6607	7.09842	-7.7202	-3.6935	-2.1317		
SDA	9.70968	10.2213	14.8572	-0.7286	-12.953	-9.4764	-7.8309	-7.1176	
ROGA	17.2679	18.0405	22.8345	5.02306	-8.7417	-4.9189	-3.3800	-1.6341	5.25839
Methods	HYBRID	DARO	EMPA	SGA	PIGMI	CA	IPA	PRA	SDA

Table Value for ‘t
5% Level of Significance
Degree of Freedom = 15-1 =14
‘t’ Value = 2.145 (Two Tailed)

Calculated value for ‘t’

Figure 2. t-test results for the samples(GA runs required) collected from various methods

The results of t-test conducted on the individual samples are shown in figure 2. If the calculated value of t be > table value, the difference between the sample mean is said to be significant at 5% level of significance otherwise the data is considered to be consistent. Figure 2 indicates that the results obtained for the methods DARO, IPA, PRA, SDA & ROGA are consistent with that Hybrid, CA, IPA, SGA & PRA methods respectively. The Bar chart showing Mean and the standard deviation of the samples collected is shown in figure 3 where as the Histogram for the same is shown in figure 4. It is found from the figure 3 & figure 4 that EMPA, Hybrid & DARO methods have outperformed the other methods in obtaining the optimal grammar induction result.

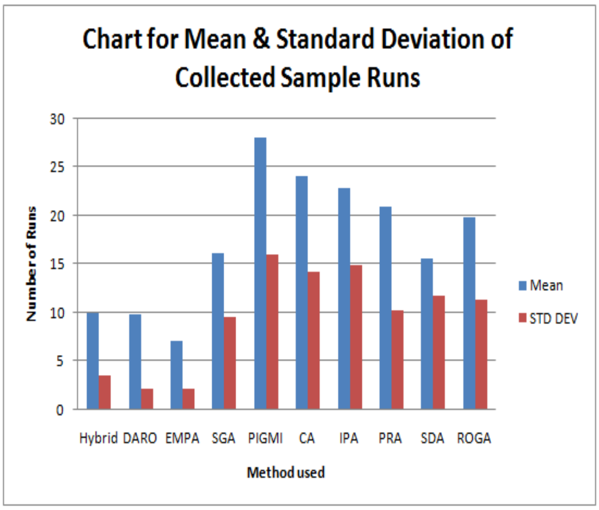

Figure 3. Bar Chart for Mean and Standard Deviation

The optimal Grammar results for the various languages from Table-1 are shown in Table-2.

5. CONCLUSION

The proposed methods, when compared with other approaches, found optimal in the support of the number of runs required to find the resultant grammar. Although EMPA, outperforms all the other methods it is found that it has higher tendency of getting local optimal convergence. DARO & Hybrid Approaches on the other hand found to be more successful in obtaining the results without local optimum convergence. Though such execution time varies from language to language, it follows the way that it is larger in case of EMP and lesser in DARO for each individual language. It is thereby found that the Hybrid method is better choice for grammar induction process in order to get optimum results without getting in to local optimum convergence.

Table 2. Resultant Best Grammar

Language	Language description
L1	<{S}, {0, 1}, {S→0S, S→?}, S >.
L2	< {S, M}, {0, 1}, { S→1M, S→0SM, M→SM, M→? }, S>.
L3	<{S, L}, {0, 1}, {S→1S, S→0L, L→S, L→?}, S>.
L4	<{S}, {0, 1}, { S→10S, S→?}, S >.
L5	<{S}, {0, 1}, {S→?, S→0S1}, S >.
L6	<{S, M}, {0, 1}, {S→?, S→0S11 }, S >.
L7	<{S}, {0, 1}, {S→0S, S→1}, S >.

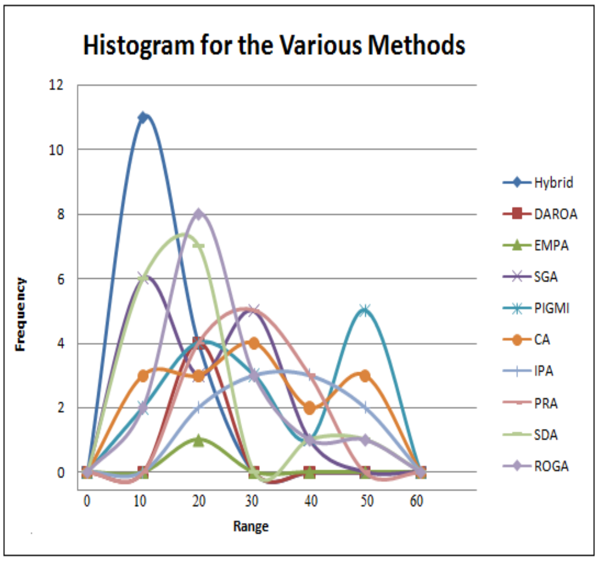

Figure 4. Histogram for the samples collected

6. ACKNOWLEDGMENTS

Authors thank to Dr. V. M. Thakare, P. G. Department of Computer Science, Sant Gadge Baba Amravati University, Amravati, Maharashtra, for his kind support in providing laboratory infrastructural facility required for the conduction of the experiment. Authors also thank to Mr. Vivek Sharma, MPSTME, Shirpur, for his kind support in preparation of this paper.

7. REFERENCES

[1] David E. Goldberg, 2007, "Genetic Algorithms- in search, optimization & Machine Learning", Pearson Education.

[2] John Holand (1992), "Adaption in Natural and Artificial Systems- An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence", The MIT Press.

[3] Wyard, P. 1994. "Representational Issues for Context-Free Grammar Induction Using Genetic Algorithm". Proceedings of the 2nd International Colloquim on Grammatical Inference and Applications, Lecture Notes in Artificial Intelligence. 862:222-235.

[4] De la Higuera, C. 2005. "A Bibliographical Study of Grammatical Inference". Pattern Recognition. 38(9):1332-1348.

[5] De Pauw, G. 2003. "Evolutionary Computing as a Tool for Grammar Development". CNTS – Language Technology.

[6] Sivanandam, Deepa, "Introduction to Genetic Algorithm", Springer Verlag Publication,New Delhi, 2008.

[7] V.M. Kureichick, A.N. Melikhov, V.V. Miagkikh, O.V. Savelev, A.P. Topchy, "Some new features in genetic solution of the travelling salesman problem", in: Proceedings of ACEDC'96 PEDC, University of Plymouth, UK, 1996.

[8] E.S. Nicoara, "Mechanisms to avoid the premature convergence of genetic algorithms", Bulletin of P. G. University of Ploiesti, Mathematics-Informatics-Physics Series LXI (1/2009) (2009) 87–96.

[9] M. Rocha, J. Neves, "Preventing premature convergence to local optima in genetic algorithms via random offspring generation", in: Proceedings of the 12th international conference on Industrial and engineering applications of artificial intelligence and expert systems: multiple approaches to intelligent systems, Cairo, Egypt IEA/AIE, 1999.

[10] Choubey N. S. , Kharat M. U. (2013), "Hybrid System for Handling premature convergence in GA- Case of Grammar induction", Applied Soft Computing 13 (2013) 2923–2931.

[11] Choubey N. S., Kharat M. U. (2010). "Sequential Structuring element for CFG Induction using Genetic Algorithm", International Journal of Computer Applications (0975 – 8887), Volume 1 – No. 1.

[12] L.J. Eshelman, "Preventing premature convergence in the genetic algorithms by preventing incest", in: R. K. Belew, L.B. Booker (Eds.), Proceedings of the Fourth International Conference on Genetic Algorithms, San Diego, July 1991, 1991, pp. 115–122.

[13] C. Ryan, "Racial harmony in GA", in: Proceedings of KI94 Workshop, 1994.

[14] K.A. de Jong, "An analysis of the behavior of a class of Genetic Adaptive Systems", Ph.D. Thesis, Department of Computer and Communication sciences, University of Michigan, Ann Arbor, MI, 1975.

[15] S.W. Mahfoud, "Niching methods for genetic algorithms", Ph.D. Thesis, Department of General Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, 1995.

[16] S.W. Mahfoud, "Crowding and preselection revisited", in: R. Manner, B. Manderick (Eds.), Proceedings of the 2nd International Conference on Parallel Problem Solving from Nature (PPSN II), Brussels, Belgium, Elsevier, Amsterdam, The Netherlands, 1992, pp. 27–36.

[17] Choubey N. S., Kharat M. U. (2011), "Approaches for Handling Premature Convergence in CFG Induction Using GA", in A. Gaspar-Cunha et al. (Eds.): Soft Computing in Industrial Applications, AISC 96, pp. 55–66.Springer-Verlag Berlin Heidelberg.

[18] Kothari C. R. (2013), "Research Methodology- Methods and Techniques", New Age International (P) Limited, Publisher, New Delhi.

[19] Gupta S. P. (2012), "Statistical Methods", Sultan Chand & Sons Educational Publisher, New Delhi.

8. AUTHOR'S PROFILE

M. U. Kharat, BE, MS, Ph.D. was educated at SGBA University. Presently he is working at MET's IOE, Nashik, Maharashtra, India, as Professor & Head Computer Engineering Department. He has presented papers at National and International conferences and also published papers in National and International Journals on various aspects of Computer Engineering and Networks. He has worked in various capacities in academic institutions at the level of Professor, Head of Computer Engineering Department, and Principal. His areas of interest include Digital Signal Processing, Computer Networks, and the Internet

N. S. Choubey, BE, ME, MBA, Ph.D. (Management) was educated at SGBA University, India and also holds a Diploma in TQM & ISO 9000. He is pursuing Ph.D. program in faculty of Computer Science & Engineering from SGBA University, Amravati, Maharashtra, India. Presently he is working at M.P.S.T.M.E. at S. V. K. M.'s N.M.I.M.S. Deemed-to-be-University, Shirpur Campus, Dhule, Maharashtra, India, as Professor & Head of the Computer Engineering Department. He has presented papers at National/International conferences and also published papers in National/International Journals on various issues of Computer and Management. He has published books in Computer and Management. His areas of interest include algorithms, Theoretical Computer Science, and Computer Networks and Internet.