

Impact of Unique Schemata Count on the Fitness Value in Genetic Algorithms

Rajneesh Pawar¹, Dr. J.S. Saini²

*Assistant Professor, Deptt. of Electrical Engg., D.C.R. Univ. of Science & Tech.,
Murthal, Sonipat, Haryana (INDIA) – 131039; Email: rajneeshdpawar@gmail.com*

*2Professor, Deptt. of Electrical Engg., D.C.R. Univ. of Science & Tech.,
Murthal, Sonipat, Haryana (INDIA) – 131039; Email: jssain@rediffmail.com*

Abstract

Genetic Algorithms (GAs) are stochastic optimization algorithms that are used for minimizing or maximizing a given function, i.e., finding its most suitable solution. This paper describes the schemata and schemata count of Genetic Algorithm (GA) and also a means to examine in a GA how to get the unique schemata count in the population of 'n' strings. With respect to parameter unique schemata count, we also make an experimental analysis on the impact of the number of the unique schemata count on the fitness value of the best individual or solution found in the population.

Key Words: *unique schemata count, similarity bits count, strings, genetic Algorithm (GA), schemata, count of same similarity bits, fitness value, etc.*

1. INTRODUCTION

Genetic Algorithm (GA) works by discovering, reinforcing, and recombining good “building blocks” of solutions contained in the population of chromosomes (strings). In so doing, it uses a fixed-length binary string and the notion of schemas or schemata [1-3].

GA's operate on schemata that give us a powerful and compact way of talking about the similarities among strings [2].

The purpose of this paper is to describe schemata and how to get unique schemata count in a population of binary strings and the impact of the unique schemata count on the fitness value of the best individual found in the population.

The novelty of the work lies first in using the computation of unique schemata count of GA for speedy evolution of diverse population, something that no other literature yet seems to provide. Second, it facilitates an in-depth analysis of the schemata and schemata count, leading to a greater understanding of the schemata and the role they play during evolution of a solution.

The computation of unique schemata count gives the GA user an additional ability to speedily evolve the GA process. The central idea is to obtain the “initial diverse population” for a GA run and

possibly tweaks the population based on schemata count, as the evolution proceeds.

The experimental analysis illustrates the behavior of the different populations as to the fitness value of the best solution within it with the value of unique schemata count and finds a significant impact of unique schemata count on the fitness value.

The paper provides a brief description of GA and theoretical aspects of schemata in Section-2. We then give a description of the synthesis of schemata count for binary coded GA that includes some of the new terms, and expressions coined by us on the subject along with illustrative examples in Section-3. Section-4 reports the experimental analysis of the impact of unique schemata count on the fitness value. Conclusions and future directions are put forth in Section-5.

2. THEORETICAL BACKGROUND

Genetic Algorithms is a versatile stochastic optimization algorithm that is widely applied to challenging engineering problems. This section describes, in brief, the basics of GA and schemata.

2.1 Basics of GA

A simple GA starts with a population of solutions

(also called individuals or strings) encoded in one of the many ways. Binary encodings are quite common and are used in this paper. The GA determines each individual's strength based on an objective function (also called Performance Index or Fitness Function in which case the goal will be to essentially maximize it, however, the goal for an objective function could be either minimization or maximization). The GA then performs one or more of the following three genetic operators on certain individuals in the population [10].

1. Reproduction (also called selection) is simply an orientation towards retaining fitter strings in the following generations, but coupled with a variety of low-fitness and mediocre-fitness solutions too.
2. Crossover involves swapping partial strings of random length between two parent strings.
3. Mutation involves flipping a random bit in a string of binary encoding.

These three operations primarily involve random number generation, copying, and partial string exchange and random alterations [12]. Thus, GA's are simple to implement.

2.2 Theoretical Aspects of Schemata

This sub-section explains the theoretical aspects of schemata and outlines as to why genetic algorithms are so successful in complex optimization problems in terms of the schemata and the effect of genetic operators on them [1].

2.2.1 Schema

Holland [2] [5] invented the idea of schema to formally conceptualize the notion of 'building blocks'. The theoretical foundations of traditional genetic algorithms rely on using binary string representation of individuals, and on the notion of schemas. A schema is defined as similarity template for describing a subset of chromosomes or strings displaying similarities at certain string positions. The schemata consist of the symbol alphabet $\{0, 1, *\}$ where '*' is a special wild card or don't care symbol that matches both, 0 and 1. For example, the schema $[1*0*]$ is a template that matches the

following four strings: [1000], [1001], [1100], and [1101].

Other traits of a schema include the defined bits, non-defined bits, defining length, and schema order etc. The defined bits are the number of bits defined (i.e., other than don't care '*s') within the schema and their positions in the string are called the defining positions [4, 6]. For example, consider the schema $(0\ 1\ * \ 1)$ where 0 and 1 are called the defining alleles. In this example, there are 3 defined bits. The non-defined bits are the number of bits not defined within the schema. In the above example, there is 1 non-defined bit. The defining length is the distance between the two outer-most defined bits [3]. The schema order is simply the number of fixed or defined positions present in the schema [3].

2.2.2 Processing of Schemata

In general, there are 2^m possible binary strings, where m is the length of the string but 3^m unique schemata to represent them. A particular binary string of length ' m ' inside a population of ' n ' strings into one of the 2^m schemata can be obtained from this string [13]. Thus, in the entire population, the number of schemata lie between 2^m and n . 2^m , depending upon the diversity [4, 6] in the population. J. Holland estimated that in a population of ' n ' chromosomes, the GAs process $O(n^3)$ schemata in each generation. This process is called as implicit parallelism.

The search heuristics of GA are based upon Holland's fundamental schema theorem. Holland derived an expression that estimates the number of copies a particular schema would have in the next generation after undergoing selection, crossover and mutation. It should be noted that particularly good schemata will propagate in the subsequent generations. Thus, schema that are low-order, of short defining length and of above average fitness are preferred and are termed as 'building blocks'.

3. SYNTHESIS OF UNIQUE SCHEMATA COUNT

3.1 Terms associated with Schemata Count

This subsection comprises of some definitions needed in the context of schemata count. We coin

and define the important terms needed for the synthesis of schemata count as:

3.1.1 Similarity Bits Count

The number of matching alleles between any two individual strings is called the similarity bits count [7-9]. It is denoted by *csb*.

3.1.2 Diversity Bits Count

The number of non-matching alleles between any two individual strings is called the diversity bits count [7-9]. It is denoted by *cdb*.

3.1.3 Count of same similarity bits

The number of times the same similarity bits count occurs in the population of 'n' strings constitutes the count of same similarity bits. It is denoted by *cb*.

3.1.4 Count of same maximum similarity bits

The number of times the same maximum similarity bits count occurs in the population of 'n' strings is termed as the Count of same maximum similarity bits. It is denoted by *cbm*, where *m* is the length of the string.

3.1.5 Count of same minimum similarity bits

The number of times the same minimum similarity bits count occurs in the population of 'n' strings constitutes the Count of same minimum similarity bits. It is denoted by *cb0*.

3.2 Unique schemata count

This subsection comprises of the developed empirical mathematical expressions for computation of unique schemata count and illustrative examples on unique schemata count.

3.2.1 Computation of Unique Schemata Count

The total number of different schemata in the population of 'n' strings is termed as unique schemata count. It is denoted by *Cs*. The generalized mathematical expression developed by us [11] that

computes unique schemata count (*usc*) for Genetic Algorithms is as follows:

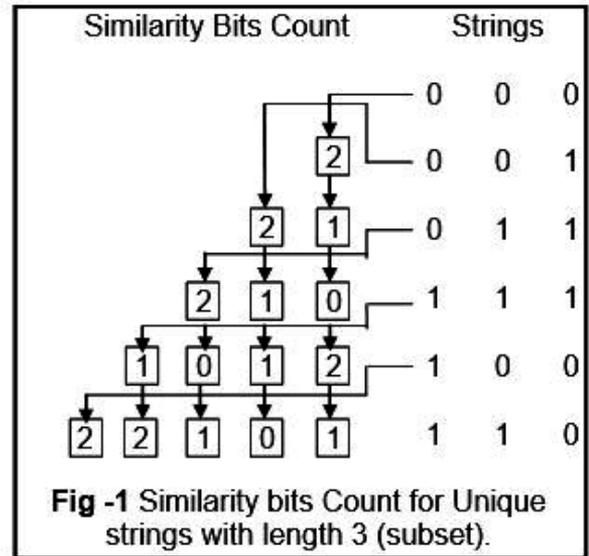
Unique schemata count (*Cs*) = $n \cdot 2^m -$

$$\sum_{j=1}^m (\min((n-1), C_{sb(m-j)})) \cdot 2^{(m-j)} + \sum_{j=1}^{(m-1)} (\min((n-2), C_{sb(m-j-1)})) \cdot 2^{(m-j-1)} - \sum_{k=1}^{(n-1)} (C_{sb(m-k)} - (n-1))$$

11

3.2.2 Illustrative Examples

Fig. 1 presents illustrative examples on the similarity bits count between strings, count of same similarity bits and finally manually constructed structures for schemata count for population subsets of two cases. Case 1: strings 'n' = 6 with length, *m* = 3 and Case 2: first five strings, i.e., 'n' = 5 with length, *m* = 3.



The similarity bits count indicated in a square is between the row from which the arrow flows and the row in which the square appears.

From Fig. 1, count of same similarity bits for of case 1 is: *cb*(*m*-1) = *cb*2 = 6, *cb*1 = 6, and *cb*0 = 3.

Similarly, count of same similarity bits for case 2, i.e., first five strings of Fig. 1: *cb*(*m*-1) = *cb*2 = 4, *cb*1 = 4, and *cb*0 = 2.

Sr. No.	Schemata	No. of different, Cs Schemata (Cumulative)
1.	0 0 0 0 0 1 0 1 1 1 1 1 1 0 0 1 1 0	6
2.	0 0 * 0 0 * 0 1 * 1 1 * 1 0 * 1 1 *	10
3.	0 * 0 0 * 1 0 * 1 1 * 1 * 1 * 0 1 * 0	14
4.	0 * * 0 * * 0 * * 1 * * 1 * * 1 * *	16
5.	* 0 * * 0 * * 1 * * 1 * * * 0 * * 1 * *	18
6.	* 0 0 * 0 1 * 1 1 * 1 1 * * 0 0 * 1 0	22
7.	* * 0 * * 1 * * 1 * * 1 * * 1 * * 0 * * 0	24
8.	* * * * * * * * * * * * * * * * * *	25

Fig -2 Schemata represented by population of Fig. 1

Figs. 2 & 3 outline the manual procedure for counting of unique schemata for the population of both the cases of Fig. 1.

Sr. No.	Schemata	No. of different Schemata (Cumulative)
1.	0 0 0 0 0 1 0 1 1 1 1 1 1 0 0	5
2.	0 0 * 0 0 * 0 1 * 1 1 * 1 0 *	9
3.	0 * 0 0 * 1 0 * 1 1 * 1 * 1 * 0	13
4.	0 * * 0 * * 0 * * 1 * * 1 * *	15
5.	* 0 * * 0 * * 1 * * 1 * * * 0 * *	17
6.	* 0 0 * 0 1 * 1 1 * 1 1 * * 0 0	20
7.	* * 0 * * 1 * * 1 * * 1 * * 1 * * 0	22
8.	* * * * * * * * * * * * * * * * *	23

Fig -3: Schemata represented by population of first five strings of Fig. 1

The common schemata are enclosed in rectangles (solid or dotted) on the same line. The unique schemata count, Cs, via manual counting procedure as in Figs. 2 & 3 comes out to be 25 and 23 respectively.

However, in a far simpler manner, the unique schemata count using our developed Eqn. 1 is:

For case 1: $C_s = 6.23 - (5.22 + 5.21 + 3.20) + \{(5 - 1).21 + (5 - 1).20\} - \{(6 - 5) + (6 - 5)\} = 25,$ (2)

For case 2: $C_s = 5.23 - (4.22 + 4.21 + 2.20) + \{(4 - 1).21 + (4 - 1).20\} = 23,$ (3)

which, of course, gives the same result for both the cases.

4. EXPERIMENTAL RESULTS

Based on the theoretical framework of unique schemata count of genetic algorithms, we make an analysis of the impact of the number of unique schemata count of the different populations (meaning thereby, varying the diversity of the different populations) on the fitness value of the best solution found in each case. In this experimental analysis, a uni-modal function given by Eqn. 4 is employed to illustrate the performance of the proposed schemata count.

Function: Spherical Model

$$f(x) = \sum_{i=1}^3 x_i^2 = (x_1^2 + x_2^2 + x_3^2) \quad (4)$$

where x ranges from 0 to 7.

In this test, the best fitness found by varying unique schemata count is recorded after 25 generations. Initially, for the program execution, we take a population of 10 strings, each variable (x1, x2 and x3) composed of 3 genes. We employ 1-point crossover, bit mutation and Stochastic Universal Sampling (SUS) selection method. Other parameters of the GA are: pc = 0.8, pm = 0.001. The global fitness value is 147. As GA's are stochastic in nature, the results are averaged over 25 independent runs.

Fig. 4 shows the analysis of the impact of unique schemata count on the fitness value. For further clarity, the results obtained are also tabulated in Table-1.

Table -1: Number of Unique Schemata Count v/s Fitness Values

No. of strings	Number of Generations	usc value	Fitness value
10	25	1016	59
10	25	1968	97
10	25	2861	121
10	25	3440	134
10	25	4248	147

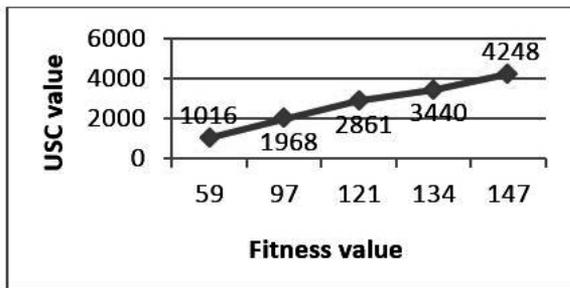


Fig -4:

Depicting the influence of the usc value on the fitness value

Table-1 illustrates the behavior of the different populations as to the fitness value of the best solution as the value of schemata count changes (i.e., increase in diversity of the population). It is seen that when the number of unique schemata count is very small, the best solution fitness value is significantly lower than those obtainable with good unique schemata count values. It may be concluded that increasing value of unique schemata count in the population significantly affects the fitness value of the best solution found.

5. CONCLUSIONS & FUTURE DIRECTIONS

In this paper, the traditional schema processing arguments for GAs, as developed initially by Holland, have been described. We have also considered some of the extensions in the case of terms related to schemata count and evaluation of unique schemata count of binary coded GAs that have been coined by us from this basic framework of binary coded GAs.

From the experimental analysis, we conclude that when the number of schemata count is more, (which implies more diverse population initially) it significantly affects the fitness value of the best solution found. In other words, the more usc we have (i.e., more diverse population initially) the more likely it is that we will find a speedier evolution of GA process. A future prospect of the findings is that schema count may provide an aid to the reproduction and recombination operators with maintenance of diversity in the population.

The binary coded GAs unique schemata count impact on the fitness value will attract the researchers to extensively test our ideas onto various applications. Moreover, the binary encodings for

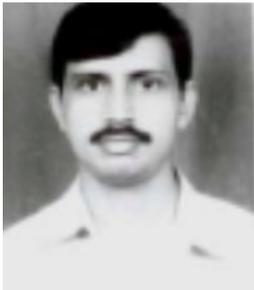
computation of unique schemata count can be further extended to real encodings.

REFERENCES

- [1] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, MA, 1989.
- [2] J.H. Holland, Adaptation in Natural and Artificial Systems, MIT Press, Cambridge, MA, 1992. First edition: University of Michigan Press, 1975.
- [3] M. Mitchell, An Introduction to Genetic Algorithms, PHI Pvt. Ltd., New Delhi, 1998.
- [4] Jasbir Singh, "Intelligent Control Techniques & Applications," Ph.D. Thesis, M.D. University, Rohtak, Haryana (India) - 2002.
- [5] J.H. Holland, "Outline for a logical theory of adaptive systems," J. Assoc. Comput. Mach., vol.3. pp.297-314, 1962.
- [6] J.S. Saini, M. Gopal and A. P. Mittal, "An alternative to schemata Count for Genetic Algorithms," All-India Seminar on Power & Energy for Sustainable Growth (PESG-2003) organized by IE(I) Hr. Centre & CRSCE, Murthal, Sonapat, Haryana (India), Pp. 41-48, Feb. 20-21, 2003.
- [7] Rajneesh Pawar, J.S. Saini, M. Gopal and A.P. Mittal, "Towards generalized expression for schemata count". Applied Soft Computing, Elsevier, 11, 2134-2141, 2011.
- [8] J.S. Saini, M. Gopal, A.P. Mittal, Rajneesh Pawar and Seema Sharma, "A New Technique for Schemata Count", Asian Conference on Intelligent Systems and Networks (AISN-2006), Centre for advanced technology, Jagadhari, Haryana, India, vol.3. pp. 37-44, February 24-25, 2006.
- [9] Rajneesh Pawar, "Recent advances in genetic algorithms," M.E. dissertation, MDU, Rohtak, Haryana, India, 2006.

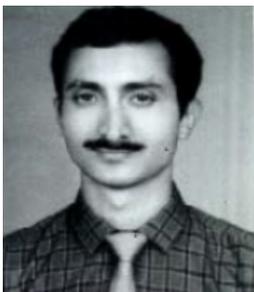
- [10] L. Davis, Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York, 1989.
- [11] Rajneesh Pawar and J.S. Saini, "Generalized Expression for Unique Schemata Count Parameter of Genetic Algorithms", submitted to Journal of the Franklin Institute, Oct. 2015.
- [12] Rajneesh Pawar and Abhishek Sharma, "Designing of PID Controller using genetic algorithms," Third National Conf. on Computing, INDICOM New Delhi, 2007.
- [13] S.N. Sivanandam and S.N. Deepa, Introduction to Genetic Algorithms. Springer-Verlag, Berlin, 2008.

papers in journals & Proc. of National & International conferences. His current interests include GAs, Chaotic Systems, Fuzzy Logic, & ANN applications in Control, Instrumentation & Optimization.



Rajneesh Pawar received his B.E. and M.E. degree in Electrical Engg. from C.R. State college of Engg., Murthal, Sonapat, India in 1999 and 2006. He is presently, Assistant Professor in Dept. of Electrical Engg., D. C. R. University of Science & Technology, Murthal, Sonipat,

India. His industrial, teaching and research experience stints at organizations such as BSES Rajdhani Power Limited Delhi, BVCOE Delhi and DCRUST Murthal. His current research interests are in the areas of Soft Computing, Energy Management System, Power Distribution Audit and planning.



Dr. J. S. Saini received his B.Sc. (Engg.) degree (Hons.) in Electrical Engg. from Punjab Univ., Chandigarh, India in 1983. He did his M.Tech. in Control Engg. & Instrumentation from I.I.T., New Delhi (India) in 1989. He was awarded Ph.D. degree in Elect. Engg. by M.D.

University, Rohtak (Haryana, India) in 2003. He is currently Professor in Dept. of Electrical Engg., D. C. R. University of Science & Technology, Murthal, Sonipat, India. He has published about 50 research