International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-1 Issue-6, August 2012

Novel Encoding Scheme in Genetic Algorithms for Better Fitness

Rakesh Kumar, Jyotishree

Abstract*— Genetic algorithms are optimisation algorithms. Every search and optimisation algorithm needs a representation which represents a solution to a specific problem. The major issue is to represent the parameter of the problem in the form of the chromosome. Choosing the right method of encoding chromosome is a crucial task and largely effects solving of optimization problem. This paper studies different encoding techniques and their associated genetic operations and then proposes a new encoding scheme to overcome the limitations of existing encoding techniques.

***Index Terms*— building blocks, encoding, genetic algorithm, schema theorem.**

I. INTRODUCTION

Genetic Algorithms [1] are adaptive heuristic search algorithms based on the evolutionary ideas of natural selection and natural genetics. Genetic algorithms are categorized under evolutionary algorithms. They are mainly used for solving optimisation problems. In real world applications, the search space is defined by a set of objects, each of which has different parameters. The objective of optimisation problem working on these parameters is to optimise them. Every search and optimisation algorithm needs a representation which represents a solution to a specific problem. Likewise, Genetic algorithms work on coding space and solution space alternatively. In nature, coding space refers to the genotypic space and solution space refers to phenotypic space. The genotype represents all the information stored in the chromosomes and allows us to describe an individual on the level of genes [2]. The phenotype describes the outward appearance of an individual. A transformation exists between genotype and phenotype, also called mapping, which uses the genotypic information to construct the phenotype. A chromosome refers to a string of certain length where all the genetic information of an individual is stored. Each chromosome consists of many alleles. Alleles are the smallest information units in a chromosome [3].

Representation can also be termed as encoding. A representation assigns genotypes to corresponding phenotypes. In genetic algorithms, genetic operators work on the level of genotype and evaluation of individuals is performed on the level of phenotype [4]. So, certain mapping or coding function between the phenotypic space and the genotypic space is needed. This can be done by designing the representation as close as possible to characteristics of phenotypic space. The mapping of the object variables to a string code is achieved through an encoding function and the mapping of a string code to its corresponding object variable is achieved through a decoding function.

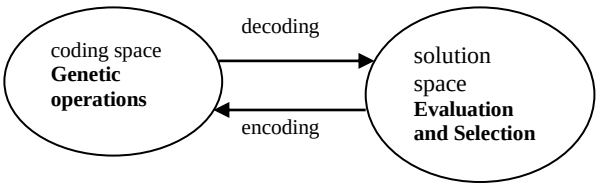

Fig. 1: Encoding – Decoding method

II. TYPES OF ENCODING

Binary Encoding is the most common representation of chromosome in genetic algorithms. A binary string is defined using a binary alphabet {0,1}. Each object variable is encoded in a binary string of a particular length l_i, defined by the user [5]. Thereafter, a complete l-bit string is formed by concatenating all sub-strings together. A binary string li has total of 2 l_i search points. The string length used to encode a particular variable depends on the desired precision in that variable. Sample representation of chromosome is

101100101100101011100101

It contains more schemas than decimal coding [6]. Three different forms of crossover suitable for binary encoding are One point crossover, N point crossover and Uniform crossover. Flip bit is very simple mutation used for binary encoding. In this, bit 0 changes to 1 or bit 1 changes to 0. The number of bits that change depends on the mutation rate.

Floating-point encoding scheme was developed by Deb for continuous variables [5]. In this scheme, separate genes designated by M and E, respectively, represent both mantissa and exponent of a floating-point parameter. For a multi-parameter optimization problem, a typical gene has three elements, as opposed to two in earlier encoding schemes. The three elements are the parameter identification number, mantissa or exponent declaration, and its value. Floating point encoding uses real coded genes. It is capable of incorporating various constraints in its implementation. Chromosomes having Real value or Floating point representation undergo Arithmetic Crossover. For Real value or Floating point representation, there exists uniform mutation, Gaussian mutation and Boundary mutation.

Permutation encoding is used in ordering problems, such as traveling salesman problem or task ordering problem. Given n unique objects, n! Permutations of the object exist. In this scheme, every chromosome is a string of numbers that represent a position (number) in a sequence. In certain cases random numbers between 0 and 1 are also used to encode the problem.

Manuscript published on 30 August 2012.
***** Correspondence Author (s)
Rakesh Kumar*, Department of Computer Science and Applications, Kurukshetra University, Kurukshetra, India.
Jyotishree, Department of Computer Science and Applications, Guru Nanak Girls College, Yamunanagar, India.

These values are used as sort keys to decode the solution. Sample representation for two chromosomes is:

Chromosome A 1 5 3 2 6 4 7 9 8
Chromosome B 8 5 6 7 2 3 1 4 9

Permutation problems cannot be processed using the same general recombination and mutation operators that are applied to parameter optimization problems. Special crossover operators are need to perform crossover operation on ordered chromosomes [7] having permutation encoding such as Edge Crossover, Partially mapped (PMX) crossover, Order crossover and Cycle crossover. For permutation representation, there are variety of mutation operators like Swap, Scramble and Inversion [7]. Inversion is the most important and widely used mutation operator that is mainly used in ordered chromosome. It selects two positions on the chromosome randomly and reverses the order of values between these two positions. Inversion operation can change the location of character in a string.

In the value encoding, every chromosome is a sequence of some values that can be anything connected to the problem, such as real numbers, characters or any objects [MEN96]. Sample representation is:

Chromosome A 1.2324 5.3243 0.4556 2.3293 2.4545
Chromosome B ABDJEIFJDHDIERJFDLDFLFEGT

Encoding having integer values in their representation also perform the same set of crossover operations as performed by binary encoding such as One point crossover, N point crossover, Uniform crossover. Special mutation – Creep occurs in case of integer representation.

III. SCHEMATA THEOREM AND BUILDING BLOCK HYPOTHESIS

Schemata were first proposed by Holland in 1975 to model the ability of Genetic algorithms to process similarities between bitstrings [3]. A schema h = (h1, h2. . . hl) is defined as a ternary string of length l, where hi ∈ {0, 1, *}. * denotes the "don't care" symbol and tells us that the allele at this position is not fixed. The size or order o(h) of a schema h is defined as the number of fixed positions (0s or 1s) in the string. A position in a schema is fixed if there is either a 0 or a 1 at this position. The defining length δ(h) of a schema h is defined as the distance between the two outermost fixed bits. The fitness of a schema is defined as the average fitness of all instances of this schema and can be calculated as:

$$f(h) = \frac{1}{||h||} \sum_{\boldsymbol{x} \in h} f(\boldsymbol{x})$$

Where ||h|| is the number of individuals x ∈ Φg that are an instance of the schema h. The instances of a schema h are all genotypes where xg ∈ h [9].

Based on the notion of schemata, Holland formulated the schema theorem which describes how the number of instances of a schema h changes over the number of generations t:

$$m(h, t+1) \geq m(h, t) \frac{f(h,t)}{\bar{f}(t)} (1 - p_c \frac{\delta(h)}{l-1} - p_m o(h))$$

where

- m(h, t) is the number of instances of schema h at generation t,
- f(h, t) is the fitness of the schema h at generation t,
- f (t) is the average fitness of the population at generation t,
- δ(h) is the defining length of schema h,
- pc is the probability of crossover,
- pm is the probability of mutation,
- l is the string length,
- o(h) is the order of schema h

Genetic algorithms follow two basic principles for choosing the encoding method namely:

- The principle of meaningful building blocks: The schemata should be short, of low order, and relatively unrelated to schemata over other fixed positions. The principle of meaningful building blocks is directly motivated by the schema theorem. If schemata are highly fit, short and of low order, then their numbers exponentially increase over the generations. If the high-quality schemata are long or of high order, they are disrupted by crossover and mutation and they can not be propagated properly.
- The principle of minimal alphabets: The alphabet of the encoding should be as small as possible while still allowing a natural representation of solutions. It states that the user should select the smallest alphabet that permits an expression of the problem so that the number of exploitable schemas is maximized [1]. The principle of minimal alphabets tells us to increase the potential number of schemata by reducing the cardinality of the alphabet. When using minimal alphabets the number of possible schemata is maximal.

These two principles of representations are based on the assumption that genetic algorithms process schemata and building blocks. The building block hypothesis states that Genetic Algorithms have the ability to propagate building blocks. By combining schemata of lower order which are highly fit, overall good solution can be constructed.

IV. INVERSION

It is the most important and widely used mutation operator that is mainly used in ordered chromosome. It selects two positions on the chromosome randomly and reverses the order of values between these two positions. Inversion operation can change the location of character in a string. Any kind of arrangement of characters in a string can be obtained by using related inversion operations. It randomly selects two locations and inverses the elements between the two locations to form the new offspring. The benefit of inversion is that it brings certain alleles together or closer. It also helps in establishing linkage between better alleles in the chromosome. In terms of schema, the main influence of inversion operation on schema H is to randomly change the length of schema H and the relationship among effective characters in schema H [8]. Inversion operator is suitable to chromosomes represented by permutation encoding. But when inversion is applied to other representations, alleles must accompanied by markers and decoding of strings is needed at the end [10]. In case of binary encoding, inversion is not applicable as it would change the basic principle of binary encoding.

V. ANALYSIS OF ENCODING TECHNIQUES

Depending on the structure of encoding, it can be classified into two categories, namely, one dimensional and two-dimensional.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Binary, Value, Real value and Permutation encoding is one dimensional and Tree encoding is two dimensional encoding technique [9]. Studying these encoding schemes, one can infer that characters represented by permutation encoding are position dependent. In Binary encoding, real value encoding, the characters are value oriented. The two factors identified by studying different encoding schemes are locus and value of character in the chromosome. So, factors like locus and value should be kept in mind while encoding a solution for a particular problem. Binary encoding is the simplest representation and supports various types of crossover operations. It does not support inversion operator as the chromosome is not ordered and implying inversion on binary representation would result in disruption of building blocks and change in fitness value. Integer and Floating point representations have limited application depending on the problem. Permutation encoding is used to represent certain order in chromosomes. It supports inversion operator but does not support simple crossover operations like one point crossover, n point crossover. Special crossover operators like PMX, order or cycle crossover is used for this representation making its use quite complex.

Binary encoding is the simplest representation and supports various types of crossover operations. It does not support inversion operator as the chromosome is not ordered and implying inversion on binary representation would result in disruption of building blocks and change in fitness value. Integer and Floating point representations have limited application depending on the problem. Permutation encoding is used to represent certain order in chromosomes. It supports inversion operator but does not support simple crossover operations like one point crossover, n point crossover. Special crossover operators like PMX, order or cycle crossover is used for this representation making its use quite complex.

Goldberg has also stated in his work that fitness function for a specific encoding scheme is dependent on two factors – value and order [1]. Three different categories of encoding can be grouped depending on fitness evaluation factors such as:

- Encoding schemes where fitness depends on value only: f(v). Eg: Value encoding
- Encoding schemes where fitness depends on value and order : f(v,o).Eg: Binary Encoding
- Encoding schemes where fitness depends on order only: f(o). Eg: Permutation encoding.

It can be stated that the existing encoding schemes fall under these three categories and they are dependent on value or order or both factors for evaluation of fitness function. So, there arises a need to unearth a new encoding scheme that is independent of these two factors. In this paper, a naive encoding scheme is put forward that evaluates fitness of a chromosome on individual contribution of gene and not depending upon its value or order.

VI. PROPOSED ENCODING

Seeing the limitations of binary encoding and permutation encoding, the authors propose a new encoding scheme that aims to overcome these limitations and converge the advantages of both the schemes into one scheme. This novel approach of encoding is based on the concept to retain good building blocks in the chromosome regardless of their locus. The novel encoding scheme also has its biological justification. As in case of human beings, the presence of genes on the chromosomes is independent of its location. Genes in human chromosomes can be shifted from locus to other. The proposed encoding scheme also focuses on retaining good building blocks in the chromosome regardless of their position. The gene or set of genes is considered to be good building block on the basis of its contribution to fitness value. In binary encoding, the genes are positional in nature and contribute to fitness according to its position in the chromosome. For example, in chromosome 1 0 1 1 0, the fitness of chromosome is 22. But if any of the bit changes its position, the fitness value also changes accordingly. For example, fitness of 1 1 0 1 0 is 26. This clearly shows that the genes in binary encoding have positional significance.

The novel encoding scheme proposes to represent each gene in the chromosome by its fitness contribution instead of 1 or 0 as in binary encoding. Fitness contribution of each gene is computed as the 8421 scheme used in binary number system. For example, 1 0 1 1 0 in binary encoding could be represented as 16 0 4 2 0 as per the new encoding scheme.

The use of fitness contribution itself as gene instead of allele value 0 or 1 aids in locating good building blocks in the chromosome easily. Genes with higher fitness contribution are more likely to be selected as good building blocks and carried forward to next generations by any of the genetic operation. This form of encoding allows inversion of genes without affecting the fitness of chromosome which would help in grouping or bringing closer good building blocks and develop linkage between them. The scheme also retains the capability of binary encoding to undergo one-point crossover or N-point crossover. In case, there occurs some redundancy of genes in chromosome during crossover, then any one of the redundant gene could be chosen and the other is replaced by 0. The position of genes is independent of fitness of chromosome.

For example, 16 0 4 0 1,
4 16 0 0 1,
0 0 4 1 16 and
1 0 16 0 4

These are various representation of chromosomes having same fitness value i.e. 21 but the locus of genes is different in each case.

VII. IMPLEMENTATION

To test the viability and usage of the proposed encoding, it was tested on fitness function used in Standard Genetic Algorithm (SGA) i.e. f(x) = x2. Matlab code has been developed to test the performance of genetic algorithm using roulette wheel selection in four cases.

- Case 1: Binary encoding and one point crossover
- Case 2: Proposed encoding and one point crossover
- Case 3: Proposed encoding and PMX crossover
- Case 4: Proposed encoding and Inversion

Case 1 tests the performance of binary encoding with one point crossover just like SGA. In exisiting encoding schemes, it has been observed that they are not suitable (apt) for all types of crossover operators. The proposed encoding has been tested with one point crossover (Case 2) that can be used with binary encoding and has proved to be suitable. Further, the proposed encoding has been used with the PMX crossover (Case 3) which is specifically used with permutation encoding.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The proposed encoding has been found apt for PMX crossover as well and the test runs confirmed the proposition. To add an extra advantage, proposed encoding was tested with inversion operator also (Case 4). Inversion operator was confined to specific encoding schemes earlier. Test runs have confirmed the suitability of the proposed encoding with inversion operator too.Demonstration of proposed encoding is illustrated for four different cases in this section.

Case 1: Binary encoding with one point crossover

P1 0 1 1 1 1 Fitness value = 15
P2 1 0 1 1 0 Fitness value = 22
C1 0 1 1 1 0 Fitness value = 14
C2 1 0 1 1 1 Fitness value = 23

Child chromosomes are created after one point crossover.

Case 2: Proposed encoding with one point crossover

P1 16 0 4 2 0 Fitness value = 22
P2 0 8 4 2 1 Fitness value = 15
C1 16 0 4 2 1 Fitness value = 23
C2 0 8 4 2 0 Fitness value = 14

As per proposed encoding, the alleles contain the value corresponding to the fitness contribution. After one point crossover, the two child chromosomes are created as illustrated. This shows that proposed encoding supports one point crossover as in binary encoding and can be used with n-point crossover also.

Case 3: Proposed encoding with PMX crossover

P1 16 0 4 2 0 Fitness value = 22
P2 0 8 4 2 1 Fitness value = 15
C1 16 8 4 2 0 Fitness value = 30
C2 0 0 4 2 1 Fitness value = 7

Highlighted parts of the parent chromosomes represent the swab portion of chromosome which is copied as such in PMX crossover. After PMX crossover, the two child chromosomes are created as shown. This shows that proposed encoding supports PMX crossover as in permutation encoding because fitness value of chromosome as per proposed encoding is not dependent on position of allele.

Case 4: Proposed encoding with Inversion

P1 0 8 4 2 1 Fitness value = 15
P2 16 0 4 2 0 Fitness value = 22

After Inversion

P1 0 4 2 1 8 Fitness value = 15
P2 16 4 2 0 0 Fitness value = 22

After one point crossover

C1 0 4 2 0 0 Fitness value = 6
C2 16 4 2 1 8 Fitness value = 31

Inversion is beneficial genetic operator which allows to group together good building blocks in a chromosome. Inversion operation cannot be applied to binary encoding as it is value dependent and cannot be applied to permutation encoding as it is order dependent. Since, the proposed encoding is independent of value and order, inversion operator can be applied to the chromosome. This would lead to grouping of good alleles which would lead to better offsprings. Fitness value of chromosome as per proposed encoding is independent of position, so inversion does not affect the computation of fitness value.

Test runs for standard genetic algorithm have been implemented using MATLAB code. Comparison of four given cases is done for average fitness and maximum fitness and is shown in Fig. 2 and Fig. 3.

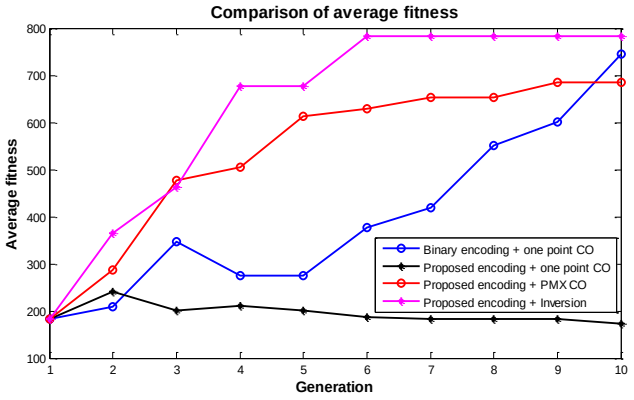

Fig. 2: Testing of Proposed Encoding by comparing Average fitness

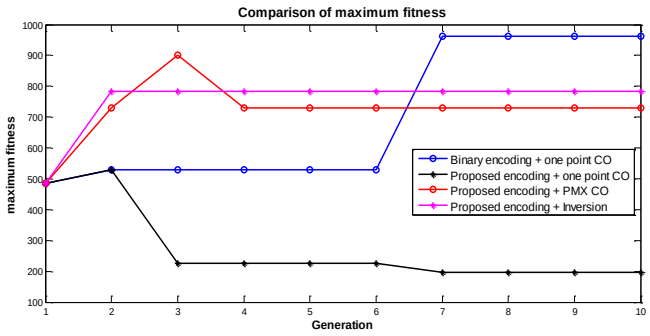

Fig. 3: Testing of Proposed Encoding by comparing Maximum fitness

It has been affirmed by the test runs that Proposed encoding is not confined to specific set of genetic operators. It has wide range of applicability and independence of genetic operator to use. The proposed encoding is independent of value and order, so it can be used with any type crossover or inversion operator. It can also be tested with different mutation operators. It helps in retaining good building blocks with higher fitness contribution as it supports inversion operator which was lacking in permutation encoding. Proposed encoding is independent of order of alleles in the chromosome, so it can be used with PMX crossover and can also be tested with order crossover etc.

VIII. CONCLUSION

Holland's Schema Theorem is based on the proposition that Genetic algorithm finds good solutions to a problem through the recombination of mutually useful features from different population members. The Schema Theorem places emphasis on the preferential survival of building blocks which are already beneficial to solutions within the population. Numerous encoding techniques exist with their respective pros and cons. Studying the existing encoding schemes, it has been established that encoding schemes are either function of value or order or value and order both. It has been generalised that crossover and mutation operators also vary with different encoding schemes. A naive encoding scheme is proposed in this paper which is independent of function of order or value. The proposed encoding scheme represents genes in the chromosome as its fitness contribution value.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The order of genes is immaterial. The proposed encoding supports one point crossover as in binary encoding and PMX crossover as in permutation encoding. The proposed encoding is also apt for inversion operation which is restricted to some encoding schemes. The naive encoding scheme based on fitness contribution is tested on simple genetic algorithm and test runs substantiate the independent nature of the proposed encoding that proves its superiority over other encoding schemes. This encoding scheme can be applied to all the applications where initially binary encoding is implemented and would prove as efficient method of encoding. The authors intend to carry forward the research on the proposed encoding to ascertain its benefits practically in real time applications.

REFERENCES

[1] D.E. Goldberg, *Genetic algorithms in search, optimisation, and machine learning*. Addison Wesley Longman, Inc., ISBN 0-201-15767-5, 1989.
[2] G. Mendel, Versuche ¨uber Pflanzen-Hybriden. In Verhandlungen des naturforschenden Vereins, Volume 4, Br¨unn, Naturforschender Verein zu Br¨unn, 1866, pp 3-47.
[3] J.Holland, Adaptation in natural and artificial systems, University of Michican Press, Ann Arbor, 1975.
[4] Mitsuo Gen, *Genetic algorithms and engineering design*, Wiley-IEEE, ISBN 0471127418, 1996.
[5] K. Deb, *Handbook of Evolutionary computation*, release 97/1. Oxford University Press, 1997.
[6] M.Mitchell, *An Introduction to genetic algorithms*, Prentice Hall of India, New Delhi, ISBN-81-203-1358-5,1996.
[7] A.E. Eiben and J.E.Smith., *Introduction to Evolutionary Computing*, Springer, First edition, ISBN 3-540-40184-9, 2003C.
[8] S. Zhonzhi, *Advanced Artificial Intelligence*, Volume 1 of Series on Intelligence Science, Scientific Publisher, ISBN981429134X, 9789814291347, 2011
[9] F. Rothlauf,, *Representation for genetic and evolutionary algorithms* (2.ed), Physica-Verlag, 2006
[10] John R. Koza, *Genetic programming: on the programming of computers by means of natural selection*,Volume 1, Complex adaptive systemsBradford BooksAuthorJohn R. KozaEdition4, illustrated, reprintPublisherMIT Press, ISBN0262111705, 9780262111706, 1992

Dr. Rakesh Kumar obtained his B.Sc. Degree, Master's degree – Gold Medalist (Master of Computer Applications) and PhD (Computer Science & Applications) from Kurukshetra University, Kurukshetra. Currently, he is Professor in the Department of Computer Science and Applications, Kurukshetra University, Kurukshetra, Haryana, India. His research interests are in Genetic Algorithm, Software Testing, Artificial Intelligence, Networking. He is a Senior member of International Association of Computer Science and Information Technology (IACSIT).

Mrs. Jyotishree is working as Assistant Professor in the Department of Computer Science and Applications, Guru Nanak Girls College, Santpura, Yamunanagar. She is M.Sc – Gold Medalist from Guru Jambheshwar University, Hisar and is pursuing PhD from Kurukshetra University. She is member of International Association of Computer Science and Information Technology (IACSIT). She has presented papers in 3 International and 5 National Conferences. Her research interests are Genetic Algorithms, Soft Computing methods and Networking.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)