ISSN: 2321-7782 (Online)

Volume 3, Issue 8, August 2015

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Specialized Genetic Operators for Classification of Management and Artistic Intelligence for Genetic Fuzzy Systems

Dr. Kunjal Mankad
Independent Researcher
Rajkot, India

***Abstract:** Genetic Algorithm (GA) is well known for providing two salient features such as the randomness and the evolution of the fittest genes using genetic operators. Crossover and mutation are two prime genetic operators which are capable to add genetic diversity in solution. These characteristics are useful in designing intelligent decision support system. The application is to provide classification of capability using theory of multiple intelligence. The paper further elaborates the need of genetic operators such as crossover, mutation and algebraic operator. The proposed research work provides design of specialized operator which combines different point crossover, mutation and arithmetic crossover. The paper presents conclusion which shows contribution of designed specialized operator to achieve optimum results for classification of suitable career field such as administrative and artistic. Results of this study clearly show differences between the different types of crossover, mutation and arithmetic crossover operators.*

***Keywords:** Crossover, Genetic Algorithm, Multiple Intelligence (MI), Mutation.*

I. Introduction

Intelligent decision support systems have gained popularity due to many advantages such as automatic decision support, classification, accurate and timely decisions, imprecision handling, and many more. The proposed research work focuses on design of genetic algorithm which provides classification of intelligence to successfully work in some specific fields such as entrepreneurship and arts. The paper proposes the design of specific genetic operators to achieve classification task. The theory of Multiple Intelligence (MI) is being utilized as application domain to achieve the classification of intelligence.

The driving force behind GAs is the unique cooperation between selection, crossover and mutation operator. A genetic operator is a process used in GAs to maintain genetic diversity. The most widely used genetic operators are recombination, crossover and mutation. Crossover produces offspring by recombining the information from two parents while mutation prevents convergence of the population by flipping a small number of randomly selected bits to continuously introduce variation. The main goal of the paper is to propose specialized crossover to design classification application using theory of multiple intelligence.

The introductory section of the paper provides the overview about the research work. The second section of the paper discusses application design of the genetic fuzzy system. The third section of the paper describes genetic algorithm along with discussion various genetic operators. The fourth section of the paper explains experiments of designing genetic algorithm in detail. This section provides extensive discussion of designing different point's crossover, different point's mutation and arithmetic crossover operator.

II. Application Design of Genetic Fuzzy System

In today's competitive environment, every individual must require few types of intelligence to achieve success in their respective area of working. The theory of Multiple Intelligence (MI) is being utilized internationally. The field of education and technology has contributed numerous research projects focusing on implementation of theory of MI for the last few decades. The Theory of Multiple Intelligence (MI) defines intelligence as potential ability to process a certain sort of information [1].

The proposed research work focuses on design of intelligent system which provides classification of intelligence to successfully work in some specific fields such as entrepreneurship and arts. Some people are very bright in field of management or entrepreneur while some people are naturally capable towards artistic fields such as musicians, singers, painters, writers, etc. Hence, it is very advantageous to have proper knowledge regarding skills of individuals. Here, the task is to decide appropriate career field using combination of intelligence from the set of intelligences. In order to decide appropriate career field for the student, five different types of intelligences from the set of MI are required to be measured.

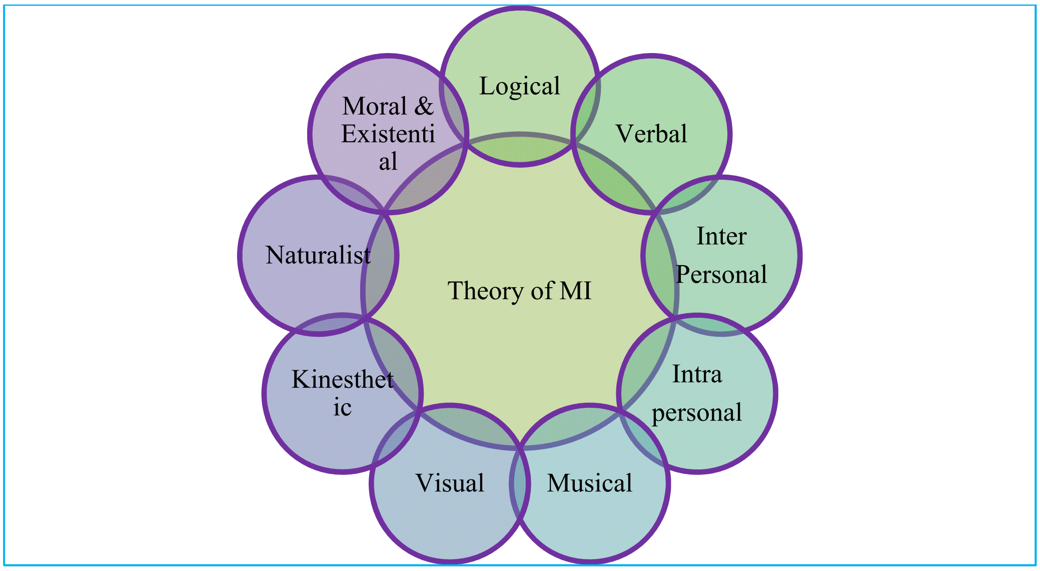


Figure 1: Theory of Multiple Intelligence Model

The core idea of designing an application using genetic fuzzy system using implementation of the theory of multiple intelligence has following objectives:

- To design the GA such as way that it becomes stronger to achieve faster evolution
- To hybrid with FL to handle imprecision
- To incorporate domain knowledge
- To make machine learn through supervised learning and to achieve inference mechanism
- To evolve the accurate solution for real life application
- The problem is to suggest the suitable career field to the students according to level of different types of intelligence he/she possess

In order to achieve the stated objectives, an evolutionary framework is designed using genetic-fuzzy hybridization. The paper focuses on design of genetic operators which further optimize the system and evolve off springs till it reaches convergence.

III. Designing Genetic Algorithm

Genetic Algorithm is an evolutionary-based search or optimization techniques that performs parallel, stochastic, but direct search method to evolve the best solution. Genetic Algorithms are based on principle of natural evolution which is popularly known as “Darwinian Evolution”. The general structure of genetic algorithm is as under:

Step A: Generate a population of random chromosomes

Step B: Repeat (each generation)

Calculate fitness of each chromosome

Repeat

Use selection to select pairs of parents

Generate offspring with crossover and mutation

Until a new population has been produced

Until best solution is good enough

GA starts to operate with more solutions compared to a starting population randomly created by taking determined fitness function variables into consideration. Then it tries to turn the solutions into an optimum solution by using genetic operators (elitism, selection, crossover, mutation) [2]. GA requires two main processes which are as under:

1. a string can represent a solution of the solution space, and
2. An objective function and hence a fitness function which measures the goodness of a solution can be constructed / defined.

A. *Encoding*

To apply GA for any optimization problem, it is necessary to represent chromosome using one of the encoding schemes. The techniques for encoding solutions vary from problems to problems. For the current research application, binary encoding scheme is designed.

B. *Fitness Function*

Fitness of the GA rule for each of the profession is calculated by finding the distance between the intelligence values of GA rule to that of training data set. Later penalty function is applied to identify the strength of the rule.

C. *Genetic Operators*

The prime focus of the proposed research work is on specialized genetic operators such as crossover, mutation and arithmetic crossover operators which are discussed as follows:

IV. Crossover

A. *Crossover*

Crossover is the main genetic operator. It operates on two parents (chromosomes) at a time and generates offspring by combining both chromosomes’ features. In weight selection problem, crossover plays the role of exchanging weights of the securities of two chosen parents in such a manner that the offspring produced by the crossover represents. Several crossover operators have been proposed for permutation representation, such as Partial-mapped crossover (PMX), Order crossover (OX), Position-based crossover (PX), heuristic crossover, and so on. There are different types of crossover operators available for different applications as per their requirements which are briefly discussed as under:

1. One Point Crossover

A random number is produced in one point crossover operator. When performing crossover, both parental chromosomes are split at a randomly determined crossover point. Afterward, a new child chromosome is created by appending the first part of the first parent with the second part of the second parent [3, 4].

2. Two Point Crossover

Two random numbers are produced in this crossover operator. However, an advantage of having more crossover points is that the problem space may be searched more thoroughly. In two-point crossover (TPC), two crossover points are chosen and the contents between these points are exchanged between two mated parents [5, 6].

3. Multipoint Crossover

In multipoint crossover operation, the crossover takes place at even and odd numbered sites. The crossover points are selected according to the length of the string. The bits lying between alternate pairs of sites are then interchanged [7].

4. Uniform Crossover

This operator is achieved as a result of exchanging bits from two individual parent chromosomes and maintaining a probability of 0.5 to produce off springs. In the given example, 2nd, 4th, 5th, 8th, 9th, 15th, 18th, and 20th bit positions are selected for swapping [8].

Besides these generalized operators there are several other operators available for specific problem categories such as traveling salesperson problem, machine scheduling, resource allocation, vehicle routing, quadratic assignment problems, etc. These operators are enlisted as under [9,10,11]:

- Ordered Crossover
- Position Based Crossover
- Cycle Crossover
- Sub tour exchange Crossover
- Heuristic Crossover

5. Arithmetic Crossover

In this type of crossover operator, mathematical form is utilized. The weighted average of two chromosomal vectors X_1 and X_2 is calculated as $\alpha x1+\beta x2$, where $\alpha+\beta=1$, such that $\alpha>0$ and $\beta>0$. In arithmetic crossover (AC), arithmetic creates children that are the weighted arithmetic mean of two parents. Children are feasible with respect to linear constraints and bounds. Alpha is random value between [0,1]. If parent1 and parent2 are the parents, and parent1 has the better fitness value, the function returns the child [12].

Off-spring =alpha*parent1 + (1-alpha)*parent2 ----------- (1)

B. Mutation

Mutations are random alterations in genetic materials. It is designed such a way that genes are selected randomly and change the allele. Mutation is achieved by flipping the digits starting from a randomly chosen order. The bit wise mutation performed bit by bit by changing 0 to 1 and 1 to 0. The purpose of using mutation is to maintain diversity within the population and inhibit premature convergence [13].

V. AN EXPERIMENT

The score of the questionnaires based on the Theory of Multiple Intelligence (MI) has been collected as prerequisites and later the score of different intelligences of individual student is classified into four distinct levels such as Very High, High, and Low & Mid.

The theory of Fuzzy Logic is useful in order to store and process domain specific linguistic knowledge. The application designed here is a typical application of classification which has numerical information available while Fuzzy Logic based system deals with linguistic knowledge. A set of questioner is provided to the users which is a prerequisite process of system.

The score of questionnaires is basically numerical information. In order to convert, numerical information into linguistic knowledge, numerical information is classified into linguistic labels. Linguistic labels are associated to numerical score as per Table 1. For the current application, four linguistic labels are specified, such as:

1. Very High
2. High
3. Low
4. Mid

Rule architecture consists of the finite set of attributes with a finite number of possible values can be easily represented using binary encoding. To convert student's intelligence level as achieved by "Low", "Mid", "High", and "Very High" level into 2 bits binary system, the following approach has been used. Table 1 represents the associated integers to the level of intelligence in order to map various levels of intelligence [14].

TABLE 1
Mapping of Level of Intelligence

Level of Intelligence	Assigned Integer
Low	0
Mid	1
High	2
Very High	3

There are five different kinds of intelligences are considered for classification purpose. These are enlisted as under:

1. Interpersonal
2. Intrapersonal
3. Logical
4. Musical/Visual
5. Verbal

Here, the task is to decide appropriate career field using combination of intelligence from the set of intelligences. In order to have prediction of career fields for the students, following fields have been considered for the application:

1. Entrepreneurship and
2. Artist

Main architecture of GA rule considers five intelligences of a student for each of the future career fields. This can be typically shown using Figure 2 as under.

A_E	B_E	C_E	D_E	E_E	A_A	B_A	C_A	D_A	E_A

Figure 2: Structure of GA Rule

Where A_E represents the students' level in Logical Intelligence for the category, if the student's proposed career field is "Entrepreneurship". Similarly B_E, C_E, D_E, E_E represent the student's level in Interpersonal, Intrapersonal, Musical & Verbal intelligence, if the proposed career field is "Entrepreneurship". A_A represents the students' level in Logical Intelligence for the category, if the student's proposed career field is "Artist". Similarly B_E, C_E, D_E, E_E represent the student's level in Interpersonal, Intrapersonal, Musical & Verbal intelligence, if the proposed career field is "Artist".

a) Operator 1 – Mutation on 20 Bits

The first GA operator considered application of Mutation operator is applied on 20 bits of the GA rule.

The first GA operator considered application of Mutation operator is applied on 20 bits of the GA rule.

Parent 1

0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0

Figure 3(a): Binary Bit String for Parent 1

Mutation Over Parent 1 as shown in Figure 3 (b):

Child 1

1	1	0	1	1	0	1	1	1	1	0	1	1	0	1	0	1	1	1	1

Figure 3(b): Result of Mutation on Parent 1

Parent 2

0	1	1	0	0	0	0	1	0	1	1	1	1	1	1	1	1	0	0	1

Figure 3(c): Binary Bit String of Parent 2

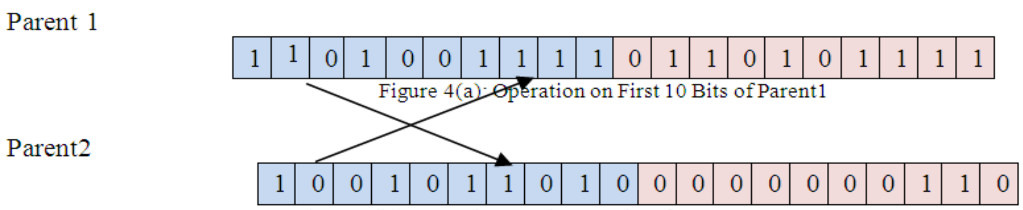
Child 2

1	0	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0	1	1	0

Figure 3(d): Result of Mutation on Parent 2

b) Operator 2 – Crossover on 10 Bits

A typical example of this operator is described in the Figure 4(a),(b),(c),(d).



Parent 1

1	1	0	1	0	0	1	1	1	1	0	1	1	0	1	0	1	1	1	1

Figure 4(a): Operation on First 10 Bits of Parent1

Parent2

1	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	1	0

Figure 4(b): Operation on First 10 Bits of Parent 2

As a result following children are generated as shown in the Figure 3 (c) and Figure 3 (d).

Child 1

1	0	0	1	0	1	1	0	1	0	0	1	1	0	1	0	1	1	1	1

Figure 4c): Result of Operator Applied on Parent 1

Child 2

1	1	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0

Figure 4(d): Result of Operator Applied on Parent 2

c) Operator 3 – Crossover on 2 Bits

The third operator designed is Crossover applied on 2 bits of the GA rule. Each parent is shown in the Figure 5(a) and the Figure 5(b).

Parent 1

1	1	0	1	0	0	1	1	1	1	0	1	1	0	1	0	1	1	1	1

Figure 5(a): 2 Bit Crossover Operator on Parent 1

Parent 2

1	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	1	0

Figure 5(b): 2 Bit Crossover Operator on Parent 2

Application of the operator would lead to creation of following child as shown in the Figure 4(c) and Figure 4(d).

Child 1

1	0	0	1	0	0	1	1	1	1	0	1	1	0	1	0	1	1	1	1

Figure 5(c): Result of 2 Bit Crossover on Parent 1

Child 2

1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	1	0

Figure 5(d): Result of 2 Bit Crossover on Parent 2

d) GA Operator 4 – Mutation on 2 Bits

The fourth GA operator considered is Mutation applied on 2 bits of the GA rule. A typical example of this operator is described as shown in the Figure 6 (a), Figure 6 (b), Figure 6 (c) and Figure 6 (d).

Parent 1

1	1	0	1	0	0	1	1	1	1	0	1	1	0	1	0	1	1	1	1

Figure 6(a): First Parent String before Mutation Operation

Parent 2

1	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	1	0

Figure 6(b): Second Parent String before Mutation Operation

Application of the operator would lead to creation of following child as shown in the Figure 6(c) and figure 6(d).

Child 1

0	0	0	1	0	0	1	1	1	1	0	1	1	0	1	0	1	1	1	1

Figure 6(c): 2 Bits Mutation Operation on Parent 1

Child 2

0	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	1	0

Figure 6(d): 2 Bits Mutation Operation on Parent 2

e) GA Operator 5 – Arithmetic Crossover

The Fifth GA operator considered is Arithmetic Crossover. A typical example of this operator is described as shown in the Figure 7 (a), the Figure 7 (b), the Figure 7(c) and the Figure 7(d).

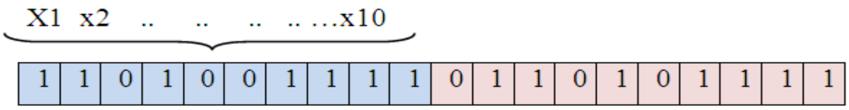
$$x' = \alpha x + \beta y$$

$$y' = \beta x + \alpha y$$

where $0 \geq \alpha, \beta \leq 1$ & $\alpha + \beta = 1$.

At first this rule is applied on 10 bits as following.

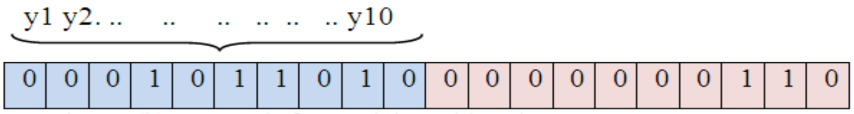
Parent 1



1 1 0 1 0 0 1 1 1 1 0 1 1 0 1 0 1 1 1 1

Figure 7 (a): Parent 1 before applying Arithmetic Crossover Operator

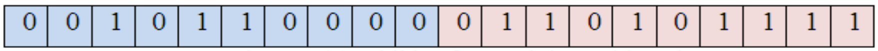
Parent 2



0 0 0 1 0 1 1 0 1 0 0 0 0 0 0 0 0 1 1 0

Figure 7(b): Parent 2 before applying Arithmetic Crossover Operator

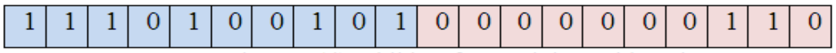
Child 1



0 0 1 0 1 1 0 0 0 0 0 1 1 0 1 0 1 1 1 1

Figure 7 (c): Child1 after applying Arithmetic Crossover

Child 2



1 1 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 1 1 0

Figure 7 (d): Child 2 after applying Arithmetic Crossover

The performance of comparison among applied operators is presented in Table 2 as under:

TABLE 2:

Genetic Operators' Performance Level in Convergence

Types of Operator	**Performance in Convergence**
20 bits Mutation	High
10 bits crossover	Very High
2 bits crossover	Average
2 bits mutation	Low
Arithmetic Crossover	Very Low

The above discussed operators are applied on chromosomal representation using stated methods. It has been observed that the most powerful operator is 10 bit crossover while arithmetic operator is found weak compared to different bits mutation and 2 bits crossover. The application is successfully designed using different types of operators shown in Table 2.

VI. Conclusion

In this paper a specialized crossover operator is proposed and experiments are conducted. The genetic algorithm design is presented to classify carrier fields. Encoding, fitness function design and genetic operations are prime tasks of GA design. The paper has extensively discussed the designs of specialized genetic operators for the problem of classifying two classes of applications. In order to test the operators, application of the theory of multiple intelligence is utilized. The specific implementations of genetic algorithm especially genetic operators are represented using examples. The design of operators is robust which greatly reduces the computational power of the system. The paper has presented novel design of a fitness function and application of genetic operators which enable the system to achieve desired level of convergence rate in evolutionary process. The most important advantage of the proposed method is that more variety is presented in crossover and mutation operators. The experiments and the results presented in the paper clearly expose the potential ability of the proposed method in optimization processing based on GA.

References

1. P. Carter, The complete book of intelligence tests. Chichester, West Sussex: John Wiley & Sons Ltd., 2005.p.5.
2. E.D.Goldberg, "Genetic Algorithms in Search Optimization and Machine Learning," Addison –Wesley Longman, 1989.
3. C. R. Reeves, J. E. Rome, Genetic Algorithms Principles and Perspectives, Kluwer Academic Publishers. Dordrecht, 2003.
4. T. Kellegoz, B. Toklu, J. Wilson, "Comparing efficiencies of genetic crossover operators for one machine total weighted tardiness problem," Applied Mathematics and Computation, 199, 590–598, 2008.
5. L. Booker, "Improving search in genetic algorithms," In Genetic Algorithms and Simulated Annealing, L. Davis (Ed.). Morgan Kaufmann Publishers, 1987.
6. M. Kaya, "The effects of two new crossover operators on genetic algorithm performance," Applied Soft Computing, 11, 881–890, 2011.
7. W.Spears, & K.Jong, An anlaysis of multipoint crossover. Proc of the foundation of Genetic Algorithms Workshop,1990,(pp. 301 315). Indiana.
8. G. Syswerda, Uniform crossover in genetic algorithms. Proc of 3rd International Conference on Genetic Algorithms ,1989, (pp. 2- Morgan Kaufmann Publishing.
9. N.P. Padhy, Artificial Intelligence and Intelligent Systems,2005, New Delhi: OXFORD University Press.
10. D.K.Pratihar, Soft Computing, 2008, New Delhi: Narosa.
11. S.Rajsekaran &V. Pai, Neural Networks, Fuzzy Logic and Genetic Algorithms Synthesis and Applications, 2010, New Delhi: PHI.
12. Y. Kaya ,M. Uyar, R. Tekdn, A Novel Crossover Operator for Genetic Algorithms: Ring Crossover
13. S.Sivanandam, & S.Deepa, Principles of Soft Computing, 2010, New Delhi, India: Wiley India.
14. K.B.Mankad, "A Genetic-Fuzzy Approach to Measure Multiple Intelligence", Ph.D. thesis,2013, Sardar patel university, vallabh vidyanagar, India.

AUTHOR(S) PROFILE

Dr. Kunjal Mankad, received the Ph.D degree in Computer Science in October 2013 from Sardar Patel University, Vallabh Vidyanagar, Gujarat. She received MCA degree in 2001 from Saurashtra University, Rajkot, Gujarat. During 2002 to 2013, She has worked as assistant professor at ISTAR, Vallabh Vidyanagar. She has presented many research papers in international conferences and international journals.