

Research Article

A Genetic Algorithm with Fuzzy Crossover Operator and Probability

**Mohammad Jalali Varnamkhasti^{1,2} Lai Soon Lee^{1,3}
Mohd Rizam Abu Bakar^{1,3} and Wah June Leong^{1,3}**

¹ *Laboratory of Computational Statistics and Operations Research, Institute for Mathematical Research, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia*

² *Department of Basic Science, Islamic Azad University, Dolatabad Branch, Dolatabad, Esfahan, Iran*

³ *Department of Mathematics, Faculty of Science, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia*

Correspondence should be addressed to Lai Soon Lee, lee@math.upm.edu.my

Received 23 June 2011; Accepted 22 October 2011

Academic Editor: Lars Mönch

Copyright © 2012 Mohammad Jalali Varnamkhasti et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The performance of a genetic algorithm is dependent on the genetic operators, in general, and on the type of crossover operator, in particular. The population diversity is usually used as the performance measure for the premature convergence. In this paper, a fuzzy genetic algorithm is proposed for solving binary encoded combinatorial optimization problems. A new crossover operator and probability selection technique is proposed based on the population diversity using a fuzzy logic controller. The measurement of the population diversity is based on the genotype and phenotype properties. In this fuzzy inference system, the selection of the crossover operator and its probability are controlled by a set of fuzzy rules derived from the fuzzy logic controller. Extensive computational experiments are conducted on the proposed algorithm, and the results are compared with some crossover operators commonly used for solving multidimensional 0/1 knapsack problems published in the literature. The results indicate that the proposed algorithm is effective in finding better quality solutions.

1. Introduction

Premature convergence is a common problem in finding the optimal solution in Genetic Algorithm (GA) and it is strongly related to the loss of the population diversity. When population diversity is low, a GA will converge very quickly. On the other hand, if the diversity of the population is too high, it is very time consuming for a GA to converge and this may cause wastage in computational resources.

The performance of a GA is dependent on the genetic operators in general and on the type of crossover operator, in particular. During the evolution process by a GA, if the selected chromosomes are identical, some of the crossover operators have failed to create offspring that are different from their parents. Effective crossover in a GA is achieved through establishing the optimum relationship between the crossover and the search problem itself.

In this paper, a fuzzy genetic algorithm (FGA) is proposed for solving binary encoded combinatorial optimization problems like the Multidimensional 0/1 Knapsack Problem (MKP). The aim is to design a crossover operator and probability selection technique based on the population diversity using Fuzzy Logic Controller (FLC). The diversity of the population is measured on the basis of the genotype and phenotype characteristics of the chromosomes. In addition, a new technique based on Hamming Distance (HD), Fitness Value (FV), and Active Genes (AG) of the mate chromosomes is proposed during the sexual selection.

In recent years, the MKP has turned out to be the favourite playground for experimenting with metaheuristics. It serves as a suitably "difficult" test problem for developing advanced techniques in this field. The MKP is a general statement of any binary encoded problem with nonnegative coefficient. It is formulated as follows (Djannaty and Doostdar [1]):

$$\begin{aligned}
 \max \quad & f(x_1, x_2, \dots, x_n) = \sum_{j=1}^n p_j x_j \\
 \text{s.t.} \quad & \sum_{j=1}^n w_{ij} x_{ij} \leq c_i \quad i = 1, 2, \dots, m, \\
 & x_j \in \{0, 1\} \quad j = 1, 2, \dots, n, \\
 & \text{with } p_j > 0, w_{ij} \geq 0, c_i \geq 0,
 \end{aligned} \tag{1.1}$$

where n = number of objects; m = number of knapsacks; w_{ij} = consumption of resource i for object j ; c_i = capacity of the i th knapsack; p_j = profit associated with object j ; x_j = decision variable with object j .

The remainder of this paper is organized as follows. Literature review is given next in Section 2. Section 3 elaborates on the population diversity. In Section 4, the proposed FGA is described in detail. The computational results of the FGA for MKP based on the benchmark problem instances deflated from the literature are presented in Section 5. Finally, a brief conclusion is given in Section 6.

2. Literature

Evolutionary algorithms usually use the population diversity as the performance indicator for a number of reasons like avoiding premature convergence, controlling algorithm restarting or stopping when the population diversity drops below certain thresholds, the necessity for assessing a population of distinct Pareto-optimal solutions in an optimization problem, and the need for rapid adaptation of a population to dynamic problems (see Tomassini et al. [2], Morrison and De Jong [3], and Wineberg and Oppacher [4] for more details).

Some of the literature on the population diversity focuses on the genetic operators and their probabilities. Sywerda [5] and De Jong and Spears [6] compared the various crossover operators, particularly the numbers of crossover points. Their results demonstrated that sometimes the final output would be better if the number of crossover points was increased. Besides empirical analysis, substantial efforts have been invested in comparing, from theoretical perspectives, between mutation and crossover as well as between the various crossover operators [6]. However, these theories are not general enough to allow for predicting when to apply or what type of crossover operator to employ. For instance, those theories did not address the size of the study population even though it may affect the relative usefulness of the crossover operators [7]. Likewise, there is enough evidence suggesting that the relative usefulness of mutation may be influenced by the size of the population. Mutation seems to be more advantageous than crossover with small population sizes whereas crossover may be more advantageous than mutation with large population sizes [8]. Moreover, the relative usefulness of mutation and crossover are influenced by several other factors like, fitness function, representation, and selection scheme.

The selection of crossover probability p_c critically affects the behaviour and the performance of a GA. It controls the rate at which solutions are subordinate to crossover where for each problem the probability is chosen by the user. Goldberg and Sastry [9] generalized the schema theorem for p_c . They argued that the selection pressure corresponds to disruption of schema, and they showed that when the building blocks are compact (its genes are located close to each other in the chromosome string), the GA works well for a wide range of combinations of p_c and selection pressures.

Adaptive techniques that alter the probability of applying an operator in proportion to the observed performance of the chromosomes in the course of a run were proposed by Davis [10]. Fernandes et al. [11] considered a GA that adapts the reproduction rate to the size of the population under investigation. Annunziato and Pizzuti [12] presented a technique for setting the genetic parameters during the implementation by adapting the population size and the operators' rate to the environmental constraint of maximum population size. On the other hand, Srinivas and Patnaik [13] adopted a messy approach towards the determination of p_c based on the various fitness values of the population. They presented an approach that eliminates the need for parameters in a crossover-based GA. Cao and Wu [14] proposed a systematic stochastic algorithm for optimization of the control parameters for GAs. In their research, a GA is modelled as a controlled Markov chain whose transition depended on the control probability of crossover and mutation. Hong et al. [15] introduced a dynamic GA that simultaneously used more than one crossover and mutation operators to generate the next generation. The operators' rate is varied along with the evaluation results of the respective offspring in the next generation.

Numerous researchers used different parameters as the input variables in FLC for dynamically control the crossover and mutation probabilities during the GA implementation. Song et al. [16] considered the alteration of the average fitness value between two consecutive generations as the input variables. Yun and Gen [17] later introduced a scaling factor for normalization of the input variables used in [16]. Li et al. [18] utilized the information of both the whole generation and the particular chromosomes as the input variables. Wang et al. [19] considered the differences in the fitness values between two consecutive generations, and Liu et al. [20] used the average fitness value and the best fitness value of chromosomes in each generation as the input variables.

Lee and Takagi [21] considered three inputs for the fuzzy system: $x_0 = f_{avg}/f_{best}$, $x_1 = f_{worst}/f_{avg}$, and x_2 is the change in best fitness since the last control action input

variables f_{avg} , f_{best} , and f_{worst} are the average, best, and worst fitness values, respectively. Corresponding to these inputs, three outputs are introduced for the current population size, crossover probability, and mutation probability. They tested the proposed technique on the inverted pendulum control task in [22]. However, this technique had two major defects:

- (i) only the phenotype characteristic is used as the input variable,
- (ii) using the metalevel GA to find the optimized fuzzy system for the dynamic parameterized GA is computationally expensive.

Im and Lee [23] presented adaptive crossover, mutation, and selection using fuzzy system for GAs. However, it is not clear how fuzzy system is based and which parameter of GA is used for the selection rules and the membership functions.

The main problem of FLC is how to gain the fuzzy rules the inference system is based on. Several approaches have been proposed by automatic rule base identification. A family of bacterial type evolutionary algorithms has been successfully applied for solving this task. The goal is to create more accurate fuzzy rule bases from input-output data sets as quickly as possible. A comprehensive review on these bacterial-type evolutionary algorithms is given in [24].

3. Population Diversity

The approach adopted in this study attempts to define a measure for the population diversity on the basis of the genotype and phenotype characteristics. The question may be raised as to why the phenotype and genotype characteristics are used together in this study. A simple example will be used to answer the question. Let $f(x)$ be a linear function for which a maximum value needs to be defined as follows:

$$\max f(x) = \sum_{i=1}^{15} ix_i \quad \text{where } x_i \in \{0, 1\}. \quad (3.1)$$

To solve (3.1) using GA, let us say five chromosomes with length 15 are considered as the initial population. In Table 1, the goal is to evaluate the population diversity based on the genotype property. To activate this, the HDs between the chromosomes with the highest fitness value and other chromosomes are calculated. Note that the HD can be defined as follows.

Definition 3.1 (Hamming Distance). Let $C_i = \{c_{i1}, c_{i2}, \dots, c_{im}\}$ and $C_j = \{c_{j1}, c_{j2}, \dots, c_{jm}\}$ be two chromosomes within a population with length m , the Hamming Distance (HD) between C_i and C_j is

$$\text{HD}(C_i, C_j) = \sum_{k=1}^m d(c_{ik}, c_{jk}), \quad (3.2)$$

where $d(c_{ik}, c_{jk}) = \begin{cases} 0, & \text{if } c_{ik} = c_{jk} \\ 1, & \text{otherwise.} \end{cases}$

As seen in Table 1, the HD is at most 2 while the length of chromosomes is 15. This means that the genetic diversity of the population is low. However, by considering the FVs of

Table 1: A population with low genetic diversity based on genotype property (HD) but with high population diversity based on phenotype property (FV).

| Chromosome | Hamming Distance (HD) | Fitness value (FV) |
|-----------------|-----------------------|--------------------|
| 010101111101111 | — | 100 |
| 010101111101001 | 2 | 73 |
| 010101111101110 | 1 | 85 |
| 010101111101100 | 2 | 71 |
| 110101111100111 | 2 | 89 |

the chromosomes, the range is somehow relatively high. Therefore, the genotype property of the chromosomes alone is not enough for explaining the diversity of the population.

In Table 2, a different initial population is used to evaluate the population diversity based on phenotype property. To achieve this, the FVs of the chromosomes are considered. As seen in Table 2, the range of differences between the chromosomes in terms of FVs is low, and hence it seems that the diversity of the population is low although the HD between the chromosomes is high. Therefore, the phenotype property of the chromosomes alone is not enough for justifying the diversity of the population.

In light of this, the genotype and phenotype properties are both considered in this paper for the measurement of the population diversity. In the case of the phenotypic measures, the methods presented by Srinivas and Patnaik [13] and Zhu and Liu [25] are used with slight modifications:

$$T_{1,t} = \frac{n_{f_i}}{N}, \quad (3.3)$$

$$T_{2,t} = \frac{f_{\max,t} - f_{\text{avg},t}}{f_{\max,t}},$$

where N = population size, n_{f_i} = number of unique fitness values in the population, $f_{\max,t}$ = maximum fitness value in generation t and $f_{\text{avg},t}$ = average fitness value in generation t .

In the case when the genotypic measure is used, an equation similar to part of the technique proposed by Jassadapakorn and Chongstitvatana [26] is used:

$$T_{3,t} = \frac{\text{HD}(C_{f_{\max,t}}, C_{f_{\min,t}})}{L}, \quad (3.4)$$

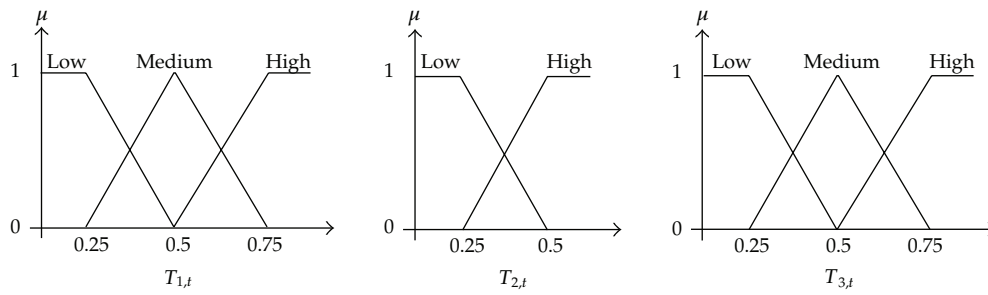
where $\text{HD}(C_{f_{\max,t}}, C_{f_{\min,t}})$ = HD between the chromosomes with the highest and the lowest FV, and L = length of the chromosome.

In this technique, the HD is calculated in each generation for only two chromosomes: the chromosome with the maximum fitness value ($C_{f_{\max}}$) and the one with the minimum fitness value ($C_{f_{\min}}$). It should be highlighted, however, that in the method of [26], the HD is calculated between the first selected chromosome and the candidate chromosome c_i , where $i = 1, 2, \dots$, population size. Therefore, our proposed method requires less time than that of [26].

$T_{1,t}$, $T_{2,t}$, and $T_{3,t}$ belong to the interval $[0, 1]$. As the $T_{1,t}$ and $T_{2,t}$ approach zero, the FVs of the chromosomes are almost identical and this shows that the convergence has taken place whereas if these are close to one, the population shows a high level of diversity. Similarly,

Table 2: A population with low population diversity based on phenotype property (FV) but with high genetic diversity based on genotype property (HD).

| Chromosome | Hamming Distance (HD) | Fitness value (FV) |
|-----------------|-----------------------|--------------------|
| 010101111101000 | — | 58 |
| 011010001010110 | 11 | 57 |
| 011010010010101 | 11 | 57 |
| 011000000010111 | 12 | 58 |
| 111100100010011 | 10 | 57 |

**Figure 1:** The set of linguistic labels associated with $T_{1,t}$, $T_{2,t}$, and $T_{3,t}$.

if $T_{3,t}$ is close to zero, then the genetic diversity is low and the population has the potential to converge very quickly. If it is near to one, the population shows a high level of genetic diversity.

When we speak of population diversity in terms of low, medium, or high diversity, we can see that the language becomes a fuzzy variable whose spatial denotation is imprecise. In this sense, fuzzy theory becomes easily understood because it can be made to resemble a high level language instead of a mathematical language. To describe this, fuzzy sets with names such as Low, Medium, or High are used to create a membership function. By determining the degree of membership of an input in the fuzzy sets of this membership function, one can see the role of membership functions play in decoding the linguistic terminology to the values a computer can use. In most cases, membership functions are designed by experts with knowledge of the system being analyzed. In this paper, with regards to $T_{1,t}$, $T_{2,t}$, and $T_{3,t}$, three membership functions are defined. The set of linguistic labels associated with $T_{1,t}$ and $T_{3,t}$ are Low, Medium, and High and those related to $T_{2,t}$ are Low and High. The semantic meaning of these labels is illustrated in Figure 1.

4. Fuzzy Genetic Algorithm

In this section, we concentrate on the discussion of the proposed FGA for solving binary encoded combinatorial optimisation problem specifically on the MKP defined in Section 1. In the remainder of this section, we explained the framework of the proposed FGA illustrated in Figure 2 in more detail.

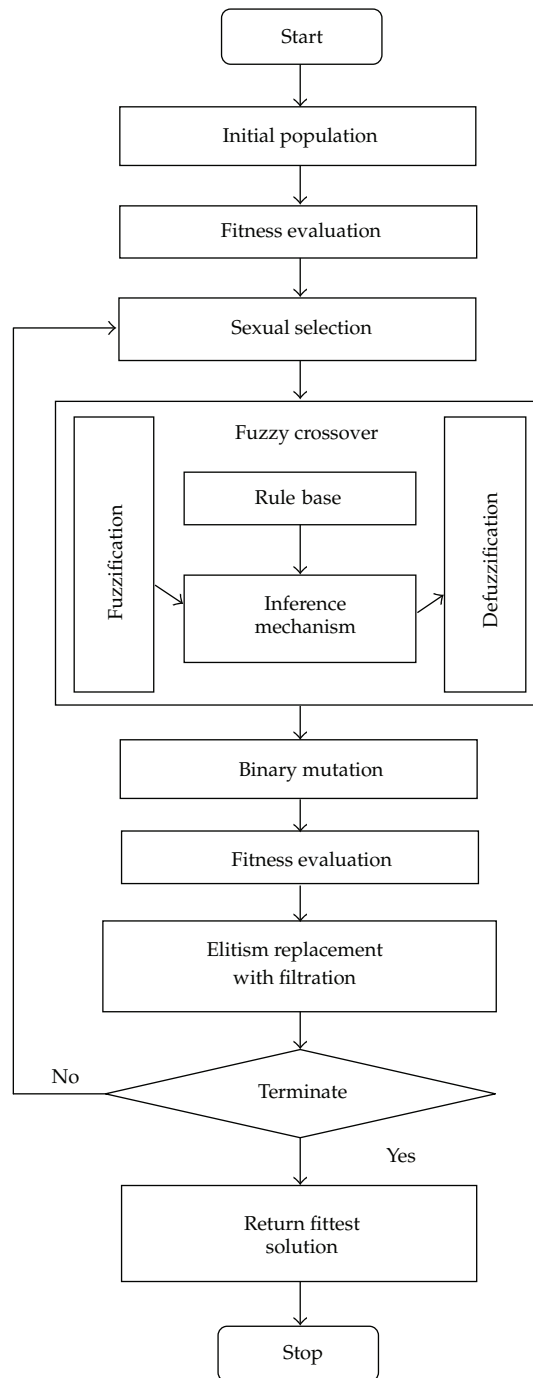


Figure 2: Framework of fuzzy genetic algorithm.

```

Let:  $S$  is a bit string belong to  $\{0,1\}^n$ .
Let:  $R_i$  = accumulated resources of constraint  $i$  in  $S$ .
for  $k = 1$  to  $N$  do
  set  $x_{kj} \leftarrow 0, \forall j \in J$ ;
  set  $T \leftarrow J$  ( $T$  is a dummy set);
  randomly select a  $j \in T$  and set  $T \leftarrow T - j$ ;
  while  $R_i + w_{ij} \leq c_i; \forall i \in I$  do
    set  $x_{kj} \leftarrow 1$ ;
    set  $R_i \leftarrow R_i + w_{ij}, \forall i \in I$ ;
    randomly select a  $j \in T$  and set  $T \leftarrow T - j$ ;
  end while
end for

```

Algorithm 1: Initial population (source: Chu and Beasley [27]).

4.1. Gene Representation and Population

For the MKP, a solution can be simply encoded by a string of 0 and 1s, where “1” encodes that an item is being selected and “0” means otherwise. The length of the chromosome corresponds to the number of items n , in the MKP.

A good initial population makes it easy for a GA to converge to good solutions while a poor initial population can prolong a GA convergence. There are different approaches in generating initial population for a GA. The most common method is by random generation. In this paper, we use the random generation method proposed by Chu and Beasley [27] to achieve better diversity in the population. The pseudo code for the initial population is given in Algorithm 1. Note that the population size is kept constant throughout the implementation.

After new offspring are created via the crossover and mutation operators, there may be some offspring that have infeasible solutions. There are several ways to deal with the infeasible solutions in a GA as stated below:

- (i) utilize an encoding method that produced only feasible solutions,
- (ii) apply a repair mechanism to transform any infeasible solution into a feasible solution,
- (iii) separate the evaluation of fitness between the feasible and infeasible solutions, or
- (iv) apply a penalty function to penalise the fitness of any infeasible solution without distorting the fitness landscape.

In this paper, we use the repair operator introduced by Chu and Beasley [27] that converts an infeasible solution into a feasible solution as presented in Algorithm 2.

4.2. Sexual Selection

In a standard GA, chromosomes reproduce asexually, that is, any two chromosomes may be the parents in crossover. Gender division and sexual selection inspired a model of gendered GA in which crossover takes place only between chromosomes of opposite sex. Since GA mimics natural evolution, then gender is one of the crucial elements in a GA.


```

Let:  $R_i$  = accumulated resources of constraint  $i$  in  $S$ .
initialize  $R_i = \sum_{j=0}^n w_{ij}x_j, \forall i \in I$ ;
for  $j = n$  to  $1$  do
    if  $(x_j = 1)$  and  $(R_i > c_i$  for any  $i \in I)$  then
        set  $x_j \leftarrow 0$ ;
        set  $R_i \leftarrow R_i - w_{ij}, \forall i \in I$ ;
    end if
end for
for  $j = 1$  to  $n$  do
    if  $(x_j = 0)$  and  $(R_i + w_{ij} \leq c_i$  for any  $i \in I)$  then
        set  $x_j \leftarrow 1$ ;
        set  $R_i \leftarrow R_i + w_{ij}, \forall i \in I$ ;
    end if
end for
    
```

Algorithm 2: Repair operator (source: Chu and Beasley [27]).

Table 3: The layout of male and female chromosomes.

| Chromosome | Gender | |
|------------|----------------|--------------------|
| | Generation t | Generation $t + 1$ |
| C_1 | Male | Female |
| C_2 | Female | Male |
| C_3 | Male | Female |
| C_4 | Female | Male |
| C_5 | Male | Female |

Inspired by the nongenetic sex determination system prevalent in some reptile species where sex is determined by the temperature at which the egg is incubated, the population are divided such that the male and female would be selected in an alternate way. The layout of the male and female chromosomes in each generation is different. In other words, when the number of generation is even, the chromosomes C_i and C_{i+1} are the male and female, respectively, and vice versa when the number of the generation is odd. For instance, suppose that a population size is five, the layout of males and females from this population is shown in Table 3.

During the proposed sexual selection, a female chromosome is selected by tournament selection size t from the female group. The selected female chromosome is denoted as female_chro. On the other hand, the selection of male chromosome (male_chro) for crossover is based on the HD from female_chro, FV, or AG of the male chromosome. Note that the AG can be defined as follows.

Definition 4.1 (Active Gene). The Active Gene denoted by $AG(C_i)$ is the number of nonzero gene in chromosome C_i . For instance, let $C_i = (1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 0)$, then $AG(C_i) = 6$.

At first, t male chromosomes are selected randomly from the male group. Then, the male_chro is selected, in order of preference based on

- (i) the maximum HD between the male chromosome and the female_chro,
- (ii) the highest FV of male chromosome (if more than one male chromosome is having the maximum HD),

Table 4: Abbreviations of crossover operators.

| Crossover operators | Abbreviation |
|-----------------------|--------------|
| 2-point crossover | 2PC |
| k -point crossover | KPC |
| Uniform crossover | UC |
| Segregation crossover | SC |
| Inversion crossover | IC |

Table 5: Levels of crossover operator based on CA.

| Low | Medium | High |
|---------|------------|-------------|
| 2-point | k -point | Segregation |
| | Uniform | Inversion |

- (iii) the highest AG of the male chromosome (if more than one male chromosome having the highest FV is found), or
- (iv) random selection.

The computational experiments of the proposed sexual selection compared to other selection mechanisms from the literature can be found in Varnamkhasti and Lee [28]. Computational results on MKPs showed that the proposed sexual selection mechanism is comparable in providing high-quality solutions in reasonable execution time.

4.3. Fuzzy Crossover Operator and Probability

In this subsection, we propose a crossover operator and probability selection technique based on the population diversity using FLC. For selecting the most appropriate crossover operator, the structure of the crossover and the population diversity must be considered. In this paper, some commonly used crossover operators for binary encoding are considered as the default crossover operators. These operators and their abbreviations are given in Table 4. It is worth mentioning that, if crossover is not applied based on the given probability, then the two offspring are simply duplicates of their parents.

When the genetic diversity of the population is high ($T_{3,t} \geq 0.50$), the 2PC operator performs relatively well compared to other crossover operators especially in terms of the computational time. The genetic diversity of the population is medium ($0.25 \leq T_{3,t} \leq 0.50$) when some loci in the chromosomes are the same. In such cases, the 2PC operator is not an appropriate choice, but the KPC, and UC operators would be more suitable. Both operators are capable of keeping the genetic diversity of the population while improving the fitness values of the chromosomes. On the other hand, the genetic diversity of the population is low ($T_{3,t} \leq 0.25$) when some chromosomes are identical. The 2PC, KPC, and UC operators failed to produce offspring that are different from their parents. In this case, the SC and IC operators will be more appropriate. Based on the above explanation of the relative crossover operators and genetic diversity, three levels for Crossover Ability (CA) called Low, Medium, and High are introduced and presented in Table 5.

As discussed earlier, genetic diversity alone is not enough to justify the population diversity as a whole. Hence, we propose a fuzzy system which takes three input variables

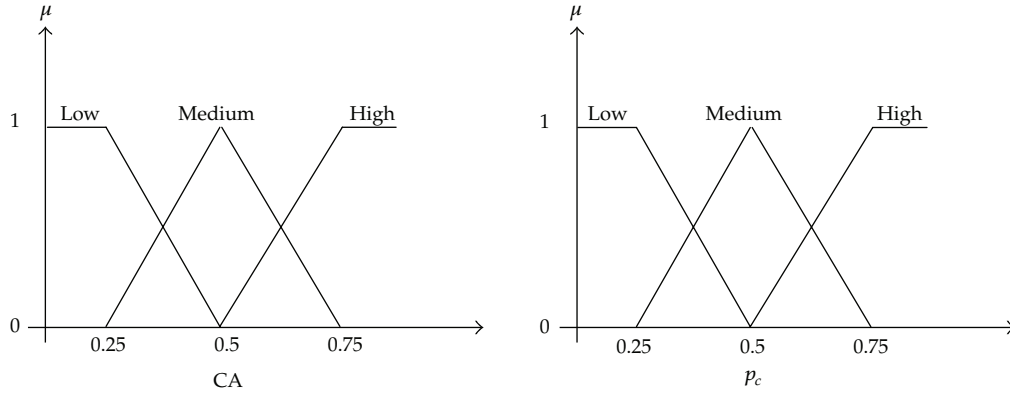


Figure 3: The set of linguistic labels associated with CA and p_c .

and produces two output variables: $T_{1,t}$, $T_{2,t}$, and $T_{3,t}$ are the input variables while CA and p_c are the output variables. The set of linguistic labels associated with the output variables comprises the descriptions of Low, Medium, and High. For each linguistic term, there is a triangular fuzzy set that defines its semantic meaning as shown in Figure 3.

The linguistic rules describing the control system consist of two parts: an antecedent block (between the **IF** and **THEN**) and a consequent block (following **THEN**). By making this type of evaluation, fewer rules can be evaluated, thus simplifying the processing logic and perhaps even improving the fuzzy logic system performance. Each rule has the possibility of generating a single rule for each output variable. In this paper, the inputs are combined logically using the AND/OR operator to produce output $(x_i, \mu_i(CA))$ and $(y_i, \mu_i(p_c))$ response values for all the expected inputs:

$$\begin{aligned}\mu_i(CA) &= \max\{\mu_i(T_{1,t}), \mu_i(T_{2,t}), \mu_i(T_{3,t})\}, \\ \mu_i(p_c) &= \min\{\mu_i(T_{1,t}), \mu_i(T_{2,t}), \mu_i(T_{3,t})\},\end{aligned}\quad (4.1)$$

where $i = 1, 2, \dots$, number of rules.

Therefore, the proposed fuzzy system with three input variables will have 18 ($3 \times 2 \times 3$) rules for each output variable (CA and p_c). This fuzzy rule base is collectively presented in Table 6. The fuzzy outputs for all rules are finally aggregated to one fuzzy set. To obtain a crisp decision from this fuzzy set, we use the Centre of Gravity approach for defuzzification:

$$\begin{aligned}x_{CA} &= \frac{\sum_{i=1}^n \mu_i(CA) x_i}{\sum_{i=1}^n \mu_i(CA)}, \\ y_{p_c} &= \frac{\sum_{i=1}^n \mu_i(p_c) y_i}{\sum_{i=1}^n \mu_i(p_c)}.\end{aligned}\quad (4.2)$$

4.4. Elitism Replacement with Filtration

After the fuzzy crossover and the standard binary mutation operators are applied, elitism replacement technique is used as the replacement strategy. The offspring have to compete with their parents in order to allow transition into the new population. In other words,

Table 6: Fuzzy rule base for CA and p_c .

| Rule | $T_{1,t}$ | $T_{2,t}$ | $T_{3,t}$ | CA | p_c |
|------|-----------|-----------|-----------|--------|--------|
| 1 | Low | Low | Low | High | High |
| 2 | Low | Low | Medium | High | High |
| 3 | Low | Low | High | Medium | Medium |
| 4 | Low | High | Low | High | Medium |
| 5 | Low | High | Medium | Medium | Medium |
| 6 | Low | High | High | Medium | Low |
| 7 | Medium | Low | Low | High | High |
| 8 | Medium | Low | Medium | Medium | Medium |
| 9 | Medium | Low | High | Medium | Medium |
| 10 | Medium | High | Low | Medium | Medium |
| 11 | Medium | High | Medium | Medium | Medium |
| 12 | Medium | High | High | Medium | Low |
| 13 | High | Low | Low | High | High |
| 14 | High | Low | Medium | Medium | Medium |
| 15 | High | Low | High | Low | Medium |
| 16 | High | High | Low | Low | Medium |
| 17 | High | High | Medium | Low | Low |
| 18 | High | High | High | Low | Low |

fitter chromosomes will survive for the next generation and they are never lost unless better solutions are found. In the elitism replacement technique, both parent and offspring populations are considered together as a single population. Then this population is sorted in a nonincreasing order of their associated fitness value and the first half of the chromosomes from this combined population are selected as the chromosomes of the new population for the next generation.

What is meant, within this context, by saying that two chromosomes are identical is that the loci of their genes are equal. Existence of the identical chromosomes is one of the critical problems for premature convergence. When a large proportion of the chromosomes in the population are identical, the diversity of the population will be lost and premature convergence occurs. In order to overcome this problem, the filtration technique is used to add diversity to the new population. In this technique, one of the identical chromosomes is kept while the others are removed and replaced by new feasible chromosomes that are generated randomly. As the filtration procedure involves the process of “identifying”, “regenerating” and “reevaluating”, of the new chromosomes, which requires a certain amount of computation time, it is sensible to just invoke the procedure every R generation (where R is a parameter, e.g., 100) or when there is at least 10 percent of the population that are identical.

5. Computational Results

The benchmark data set tested in this paper inclusive of 270 MKPs is proposed in [27]. This data set was extensively utilized in the literature for the testing of MKP algorithms. These problems include $n = 100, 250, 500$ variables and $m = 5, 10, 30$ constraints. For each category of n variables, three tightness ratios $\alpha = 0.25, 0.50, 0.75$ are considered. This set of

problems contains 27 different problem sets, each having 10 randomly generated instances, thus a total of 270 problems.

In these problem sets, w_{ij} are drawn from discrete uniform generator $U(0, 1000)$ and the right hand side coefficients c_i , $i \in \{1, \dots, m\}$, are set using $c_i = \alpha \sum_{j=1}^n w_{ij}$ where α is the tightness ratio. The objective function coefficients p_j , $j \in \{1, \dots, m\}$ are correlated to w_{ij} and are generated as $p_j = \sum_{i=1}^m (w_{ij}/m) + 500e_j$, where e_j is a real number drawn from the continuous uniform generator $U(0, 1)$.

The performance of these approaches is measured by the Percentage Deviation (PD) between the benchmark problems taken from the ORLibrary [29] and our results, that is:

$$PD = \frac{|B_i - R_i|}{B_i} \times 100\%, \quad (5.1)$$

where B_i is the benchmark result and R_i is the computational result obtained by the proposed algorithms.

In order to scrutinize the performance and the ability of the proposed FGA compared to standard genetic algorithms (SGAs), the crossover operators mentioned in Table 4 are used in the computational experiments. The difference between the proposed FGA and a SGA is with regards to the use of the crossover operator and its probability. In other words, a SGA will employ one crossover operator with fixed probability, $p_c = 0.70$ during implementation. Thus, there will be five SGAs as compared to one FGA in the experiment. Each algorithm is implemented with a population size of 100. Sexual selection described in Section 4.2, the replacement strategy explained in Section 4.4, and standard binary mutation with probability $p_m = 1/L$ (L is the length of chromosome) are used as the default genetic operators. All algorithms were coded in C++ and ran on a Pentium IV with 2.0 GHz CPU and 2.0 GB of RAM. For each problem instance, 30 runs were performed. In order to have a fair comparison between the algorithms in this experiment, we employed a duration of 20 CPU seconds per run. Results are listed in Table 7. The benchmark column represents the average computational results of 10 problem instances from each problem set of the MKPs that are compiled in [29]. For each algorithm, the entries report the average PD computed over 10 problem instances with 30 runs each, that is, 300 runs. For each category of m , the final line gives the average PD over all values of n . The final line of Table 7 gives the overall average PD over all categories of m .

We first observed that the FGA performs significantly better than the SGAs. This shows that the FGA is able to produce better quality solutions compared to SGAs in a fixed CPU time. There is clear evidence from Table 7 that, on average, the FGA is the best algorithm followed by the SGAs with SC, IC, UX, KPC, and finally the SGA with 2PC. The algorithms find problems of $m = 30$ to be the most challenging. With other things (i.e., n and α) being equal, when we increase the value of m , then PD will increase. Note that for all the algorithms, fewer generations are executed within the time limit as the number of knapsacks or constraints become larger.

6. Conclusion

Genetic algorithms usually use the population diversity as the performance measure for a number of reasons such as avoiding premature convergence. This paper proposed techniques

Table 7: Comparative computational results (20 CPU seconds per run).

| m | n | α | Benchmark | SGA | | | | | |
|----------------|-----|----------|-----------|--------------|--------------|--------------|--------------|--------------|--------------|
| | | | | 2PC | KPC | UX | SC | IC | FGA |
| 5 | 100 | 0.25 | 24197.20 | 1.380 | 0.849 | 0.822 | 0.862 | 0.905 | 0.537 |
| | | 0.50 | 43252.90 | 1.147 | 0.371 | 0.344 | 0.388 | 0.392 | 0.187 |
| | | 0.75 | 60471.00 | 1.005 | 0.307 | 0.285 | 0.299 | 0.300 | 0.196 |
| | 250 | 0.25 | 60409.70 | 1.309 | 0.200 | 0.196 | 0.205 | 0.201 | 0.110 |
| | | 0.50 | 109284.60 | 1.165 | 0.100 | 0.091 | 0.089 | 0.096 | 0.053 |
| | | 0.75 | 151555.90 | 0.994 | 0.057 | 0.66 | 0.044 | 0.057 | 0.022 |
| | 500 | 0.25 | 120615.50 | 0.711 | 0.701 | 0.052 | 0.055 | 0.055 | 0.034 |
| | | 0.50 | 219503.10 | 0.620 | 0.629 | 0.029 | 0.030 | 0.033 | 0.005 |
| | | 0.75 | 302354.90 | 0.499 | 0.440 | 0.019 | 0.019 | 0.022 | 0.004 |
| Average | | | | 0.981 | 0.406 | 0.278 | 0.221 | 0.229 | 0.128 |
| 10 | 100 | 0.25 | 22601.90 | 2.503 | 1.308 | 1.288 | 1.280 | 1.283 | 0.735 |
| | | 0.50 | 45659.10 | 1.282 | 0.583 | 0.574 | 0.570 | 0.582 | 0.480 |
| | | 0.75 | 59555.60 | 1.097 | 0.309 | 0.305 | 0.316 | 0.310 | 0.118 |
| | 250 | 0.25 | 58993.90 | 1.630 | 0.407 | 0.399 | 0.395 | 0.396 | 0.218 |
| | | 0.50 | 108706.40 | 1.523 | 0.169 | 0.163 | 0.158 | 0.160 | 0.107 |
| | | 0.75 | 151330.40 | 0.974 | 0.128 | 0.125 | 0.120 | 0.127 | 0.073 |
| | 500 | 0.25 | 118565.50 | 1.208 | 0.180 | 0.175 | 0.168 | 0.171 | 0.105 |
| | | 0.50 | 217274.60 | 0.988 | 0.085 | 0.085 | 0.080 | 0.085 | 0.043 |
| | | 0.75 | 302556.00 | 0.811 | 0.059 | 0.057 | 0.054 | 0.052 | 0.008 |
| Average | | | | 1.335 | 0.359 | 0.352 | 0.349 | 0.352 | 0.210 |
| 30 | 100 | 0.25 | 21654.60 | 3.755 | 2.739 | 2.730 | 2.718 | 2.723 | 1.217 |
| | | 0.50 | 41431.30 | 2.702 | 1.200 | 1.197 | 1.190 | 1.192 | 0.894 |
| | | 0.75 | 59199.10 | 1.693 | 0.725 | 0.725 | 0.720 | 0.724 | 0.479 |
| | 250 | 0.25 | 56875.90 | 2.660 | 1.020 | 1.017 | 1.014 | 1.019 | 0.936 |
| | | 0.50 | 106673.70 | 1.602 | 0.427 | 0.426 | 0.415 | 0.416 | 0.240 |
| | | 0.75 | 150443.50 | 1.109 | 0.160 | 0.160 | 0.152 | 0.157 | 0.030 |
| | 500 | 0.25 | 115473.50 | 1.581 | 0.405 | 0.405 | 0.398 | 0.400 | 0.328 |
| | | 0.50 | 216156.90 | 1.290 | 0.117 | 0.114 | 0.109 | 0.110 | 0.007 |
| | | 0.75 | 302353.40 | 1.041 | 0.103 | 0.103 | 0.099 | 0.101 | 0.075 |
| Average | | | | 1.937 | 0.766 | 0.764 | 0.757 | 0.760 | 0.467 |
| Average | | | | 1.418 | 0.511 | 0.465 | 0.442 | 0.447 | 0.268 |

for the measurement of the population diversity based on the phenotype and genotype properties. The main contribution of this study is the crossover operator and probability selection technique based on the population diversity using fuzzy logic controllers. The performance of the proposed fuzzy genetic algorithm compared to other standard genetic algorithms for solving the multidimensional 0/1 knapsack problems showed that the proposed algorithm is effective in finding better and comparable solutions.

Future research efforts will investigate extending the proposed fuzzy genetic algorithm to include the fuzzy mutation operator and probability selection technique for other binary encoded combinatorial optimisation problems. In addition, we may consider to extend the research for integer encoded fuzzy genetic algorithm.

References

- [1] F. Djannaty and S. Doostdar, "A hybrid genetic algorithm for the multidimensional knapsack problem," *International Journal of Contemporary Mathematical Sciences*, vol. 3, no. 9, pp. 443–456, 2008.
- [2] M. Tomassini, L. Vanneschi, F. Fernández, and G. Galeano, "A Study of Diversity in Multipopulation Genetic Programming," in *Proceedings of the Artificial Evolution*, vol. 2936 of *Lecture Notes in Computer Science*, pp. 243–255, 2004.
- [3] R. W. Morrison and K. A. De Jong, "Measurement of population diversity," in *Proceedings of the Artificial Evolution*, vol. 2310 of *Lecture Notes in Computer Science*, pp. 1047–1074, 2002.
- [4] M. Wineberg and F. Oppacher, "Distance between populations," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2003)*, vol. 2724 of *Lecture Notes in Computer Science*, pp. 1481–1492, 2003.
- [5] G. Sywerda, "Uniform crossover in genetic algorithms," in *Proceedings of the 3rd International Conference on Genetic Algorithms*, pp. 2–9, Morgan Kaufmann, 1989.
- [6] K. A. De Jong and W. M. Spears, "A formal analysis of the role of multi-point crossover in genetic algorithms," *Annals of Mathematics and Artificial Intelligence*, vol. 5, no. 1, pp. 1–26, 1992.
- [7] K. A. De Jong and W. M. Spears, "An analysis of the interacting roles of population size and crossover in genetic algorithms," in *Proceedings of the Parallel Problem Solving from Nature*, vol. 496 of *Lecture Notes in Computer Science*, pp. 38–47, 1991.
- [8] L. J. Eshelman, R. A. Caruana, and J. D. Schaffer, "Biases in the crossover landscape," in *Proceedings of the 3rd International Conference on Genetic Algorithms*, pp. 10–19, Morgan Kaufmann, 1989.
- [9] D. E. Goldberg and K. Sastry, "A practical schema theorem for genetic algorithm design and tuning," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 328–335, 2001.
- [10] L. Davis, "Adapting operator probabilities in genetic algorithms," in *Proceedings of the 3rd International Conference on Genetic Algorithms*, pp. 61–69, Morgan Kaufmann, 1989.
- [11] C. Fernandes, R. Tavares, and A. C. Rosa, "niGAVaPS—outbreeding in genetic algorithms," in *Proceedings of the Symposium on Applied Computing, (ACM '00)*, pp. 477–482, 2000.
- [12] M. Annunziato and S. Pizzuti, "Adaptive parameterization of evolutionary algorithms driven by reproduction and competition," in *Proceedings of the European Symposium on Intelligent Techniques, (ESIT '00)*, pp. 246–256, 2000.
- [13] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 4, pp. 656–667, 1994.
- [14] Y. J. Cao and Q. H. Wu, "Optimization of control parameters in genetic algorithms: a stochastic approach," *International Journal of Systems Science*, vol. 30, no. 5, pp. 551–559, 1999.
- [15] T.-P. Hong, H.-S. Wang, W.-Y. Lin, and W.-Y. Lee, "Evolution of appropriate crossover and mutation operators in a genetic process," *Applied Intelligence*, vol. 16, no. 1, pp. 7–17, 2002.
- [16] Y. H. Song, G. S. Wang, P. Y. Wang, and A. T. Johns, "Environmental/economic dispatch using fuzzy logic controlled genetic algorithms," in *Proceedings of the Generation, Transmission and Distribution*, vol. 144, no. 4, pp. 377–382, 1997.
- [17] Y. Yun and M. Gen, "Performance analysis of adaptive genetic algorithms with fuzzy logic and heuristics," *Fuzzy Optimization and Decision Making*, vol. 2, no. 2, pp. 161–175, 2003.
- [18] Q. Li, D. Zheng, Y. Tang, and Z. Chen, "A new kind of fuzzy genetic algorithm," *Journal of University of Science and Technology Beijing*, vol. 23, no. 1, pp. 85–89, 2001.
- [19] C.-H. Wang, T.-P. Hong, and S.-S. Tseng, "Integrating membership functions and fuzzy rule sets from multiple knowledge sources," *Fuzzy Sets and Systems*, vol. 112, no. 1, pp. 141–154, 2000.
- [20] H. Liu, Z. Xu, and A. Abraham, "Hybrid fuzzy-genetic algorithm approach for crew grouping," in *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications, (ISDA '05)*, pp. 332–337, September 2005.
- [21] M. A. Lee and H. Takagi, "Dynamic control of genetic algorithms using fuzzy logic techniques," in *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 76–83, 1993.
- [22] M. A. Lee and H. Takagi, "Integrating design stages of fuzzy systems using genetic algorithms," in *Proceedings of the 2nd IEEE International Conference on Fuzzy System*, pp. 612–617, April 1993.
- [23] S.-M. Im and J.-J. Lee, "Adaptive crossover, mutation and selection using fuzzy system for genetic algorithms," *Artificial Life and Robotics*, vol. 13, no. 1, pp. 129–133, 2008.
- [24] J. Botzheim, L. Gál, and L.T. Kóczy, "Fuzzy rule base model identification by bacterial memetic algorithms," *Recent Advances in Decision Making*, vol. 222, pp. 21–43, 2009.

- [25] K. Q. Zhu and Z. Liu, "Population diversity in permutation-based genetic algorithm," in *Proceedings of the 15th European Conference on Machine Learning, (ECML '04)*, vol. 3201 of *Lecture Notes in Computer Science*, pp. 537–547, September 2004.
- [26] C. Jassadapakorn and P. Chongstitvatana, "Diversity control to improve convergence rate in genetic algorithms," in *Proceedings of the 4th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2003)*, vol. 2690 of *Lecture Notes in Computer Science*, pp. 421–425, 2003.
- [27] P. C. Chu and J. E. Beasley, "A genetic algorithm for the multidimensional knapsack problem," *Journal of Heuristics*, vol. 4, no. 1, pp. 63–86, 1998.
- [28] M. J. Varnamkhasti and L. S. Lee, "A genetic algorithm based on sexual selection for the multidimensional 0/1 knapsack problems," 10 pages, *International Journal of Modern Physics: Conference Series*. In press.
- [29] "OR Library," <http://people.brunel.ac.uk/~mastjib/jeb/info.html>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

