



New crossover operators using dominance and co-dominance principles for faster convergence of genetic algorithms

G. Pavai¹ · T. V. Geetha²

Published online: 22 January 2018
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

Abstract

Normally, genetic algorithm (GA) does not guarantee global optimum for all optimization problems. Crossover operators play a crucial part in the convergence of GAs to a solution. Hence, if crossover is designed to pass on genes that highly contribute to the fitness of individuals, to subsequent generations, the convergence can be obtained faster while obtaining the best possible solution for the given initial population. In this paper, we propose two new crossover operators called the dominance and co-dominance crossover operators, based on the dominance and co-dominance principles of human genetics, respectively, to achieve this in case of applications using integer and real encoding. The dominance crossover operator is designed such that the child obtains a gene (feature) from a parent whose value for a particular gene (feature) is dominant than its value in the other parent. On the other hand, the co-dominance crossover operator is designed such that a child obtains two values for the same gene from both the parents in case both alleles (gene values) are equally dominant. These crossover operators were designed to get the optimal solution in less number of generations without sacrificing the performance of GA. The experiments conducted on test functions and two different problems, namely clustering (Reuters-21578 dataset) and learning to rank (LETOR dataset), emphasize that global optimum in fewer number of generations is obtained using our proposed crossover operators.

Keywords Co-dominance · Crossover operator · Dominance · Faster convergence · Genetic algorithms

1 Introduction

Genetic algorithms are widely used in solving optimization problems for applications such as TSP, bioinformatics, computational biology and chemistry. They perform well but take a long time to converge to an optimal solution. This led to restricting the GAs to terminate in a certain number of generations. However, this was done at the cost of the quality of the solution since cutting short the number of generations could be directly mapped to reduction in the search space. Optimal solution may not be reached in case it is present in

the part of the solution space that is not explored due to the time constraint of GAs. Therefore, efforts in reducing the convergence time resulted in pathetic quality of the obtained solutions. This convergence of GAs to local minima is called premature convergence (Rocha and Neves 1999).

‘The premature convergence of a genetic algorithm arises when the genes of some high rated individuals quickly dominate the population, constraining it to converge to a local optimum’ according to Nicoara (2009). This gives a zero percent chance of the offspring being better than the parents, thereby making the search for an optimal solution futile. This is an important research problem for which many techniques have been proposed that either solves the problem completely or minimizes the problem to some extent. There are many known reasons for premature convergence of GAs of which the popular ones are small population size, lack of diversity in the population (Herrera et al. 1997; Nicoara 2009; Ramadan 2013; Rocha and Neves 1999) disproportionate relationship between exploration and exploitation 15. (Herrera et al. 1997), the genetic operators used (Ramadan 2013), less mutation rate (Mc Ginley et al. 2011; Rocha and Neves

Communicated by V. Loia.

✉ G. Pavai
pavai_gops@yahoo.co.in
T. V. Geetha
tv_g@hotmail.com

¹ Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India

² Department of Computer Science and Engineering, College of Engineering Guindy, Anna University, Chennai, India

1999), loss of efficacy of the crossover operator (Mc Ginley et al. 2011), incorrect application of selection pressure (Mc Ginley et al. 2011), fitness function (Jing and Lidong 2012), etc. Many techniques and new GA operators were proposed to handle the premature convergence problem which is discussed in the related work section. While some of these techniques aim at delaying the convergence with the aim of getting better results, some of the works aim at faster convergence without affecting the quality of the solution. We follow the second path wherein two new crossover operators are proposed in this work to achieve this.

Various studies have shown that the employed form of crossover can determine the performance of the GA (Spears et al. 1995). There are many existing crossover operators designed to handle this issue. However, according to Herrera et al. (2003) the performance of each crossover operator varies based on different parameters for the same problem (such as the number of dimensions, number of local optima, interdependency between genes, boundary constraints and noise). Moreover, the no free lunch theorem (Wolpert and Macready 1997) states that ‘there is no universal algorithm that performs better than all others on all possible problems’ which is also applicable to the crossover operators. Therefore, we have tried to improve the performance of the GA by designing new crossover operators that are specially designed to reduce time taken by GAs to converge to a solution without compromising on the quality of the solution obtained.

Crossover operation is the process of forming one or more children from two or more parents by choosing one of the parent’s alleles for a child’s gene. Crossover operators being the backbone of the GAs are applied to the mating pool with a hope of creating better offspring. It is said to be the most important operator in the case of GAs because of the building-block hypothesis that was proposed by Holland (1992) according to which certain important building blocks (genes) are responsible in producing fit individuals in the next generation. Crossover helps to blend the genetic information between chromosomes for exploring the search space. From the time GA was invented (De Jong and Spears 1992; Holland 1992), crossover operators are said to play a prominent role (De Jong and Spears 1992; Kita 2001) in making the GA find the required globally optimal solution by exploring the search space. Crossover rate determines the likelihood of the crossover operator being applied to a chromosome. The more effectively the crossover operation is performed, the higher is the quality of the solution. This relates to the importance of the crossover operator in improving the performance of GA (Liepins and Vose 1992). Furthermore, crossover operators are considered effective to solve the premature convergence problem (Herrera et al. 1997). This gives the hint that crossover can be used for faster convergence. The convergence to local optima should be dealt with by the design of the crossover operator when it is used for faster

convergence. This importance of the crossover operators in GA motivated us to carry out this work.

We have used Mendel’s law of dominance (Gardner et al. 2006; http://en.wikipedia.org/wiki/Mendelian_inheritance; Laird and Lange 2011) to pass on only the best value of the features (dominant values) to the next generation rather than randomly choosing a gene from either of the parents, which is the norm of most of the existing crossover operators. This in turn helps the GAs to converge to a solution comparatively in lesser number of generations than GAs using other crossover operators due to the passing of best gene values (dominant alleles). The concept of co-dominance (Deshpande and Kelkar 2008; http://en.wikipedia.org/wiki/Mendelian_inheritance) of genes that exist in the ABO blood groups has been used to design the new co-dominance crossover operator. In the case of the presence of more than one dominant allele for the same gene, one of them is randomly chosen for the offspring’s gene. We have come up with another crossover operator called the co-dominance crossover operator, which selects genes from both the parents when gene values of both the parents are equally dominant. This could bring out a better solution than the dominance crossover since one of the dominant genes (which could have been better than its selected counterpart when the chromosome as a whole is considered) is left out in dominance crossover operator. In our work, the best child is created from the two parents based on the best gene value for every gene that constitutes the overall chromosome.

Our major contribution in this work is making the GA to converge faster without affecting the quality of the solution. We achieve this by designing new crossover operators called the dominance crossover operator and the co-dominance crossover operator based on the dominance and co-dominance principles in human genetics. We apply the crossover operators to the genetic algorithm for optimization test functions (integer and real encoding), clustering problem on the Reuters-21578 dataset (integer encoding) and learning to rank for IR problem on the LETOR dataset (real encoding) to evaluate our newly proposed crossover operators. We have also come up with the multi-parent version of the new crossover operators. Furthermore, we have analyzed the effect of mutation on the newly proposed crossover operators.

The rest of the paper is organized as follows: Sect. 2 discusses the background in detail. Section 3 discusses in detail about our methodology. Section 4 discusses the evaluation of our work.

2 Background

This section brings out the work on existing techniques and various GA operators and their modified form to handle pre-

mature convergence. The existing techniques either delay the convergence or urge the convergence but avoid local optima. In addition, this section discusses crossover operators similar to the crossover operators proposed in this work. Furthermore, we describe the dominance and the co-dominance principles of human genetics in this section.

2.1 Methods to avoid premature convergence

In this section, we discuss the existing methods to avoid the premature convergence problem.

Linear scale transformation of a fitness function was used by Jing and Lidong (2012), since a good fitness function maintains the diversity in the population in the early stage so as to explore new areas of search space and should help in convergence in the later stage so that the optimal solution is found out.

The dynamic application of crossover and mutation operators based on selection pressure and average progress instead of using just an operator for crossover and mutation avoids premature convergence (Nicoara 2009). This process starts from generation one and goes along till the termination of GA.

Self-adaptive crossover and mutation probabilities proposed by Jing and Lidong (2012) declare that self-adaptive mutation and crossover probabilities help in avoiding the premature convergence problem by increasing the diversity at the cost of convergence time and vice versa so that the local minima problem is avoided.

Mutation introduces diversity to the population. Hence, a high mutation rate is expected to decrease the premature convergence problem. However, a very high value for mutation rate delays the convergence process, thereby increasing the time and resource consumption of GA. In adaptive crossover, an initial value is set for the mutation rate, and at proper time intervals, it is increased or decreased based on the standard deviation of the individuals in the population (Rocha and Neves 1999).

Biasing crossover and mutation technique uses violation score which is a weighted sum of rows and columns that cause constraint violations, to bias the genetic operators such as the crossover and mutation to preserve the basic building blocks and also improve the speed of convergence.

Avoiding close relative breeding proposed by Jing and Lidong (2012), discuss about the hamming distance-based close relative breeding avoidance method so that diversity in the population is maintained.

Herrera et al. (1997) proposed four crossover operators based on fuzzy connectives to maintain the exploration and exploitation levels which in turn helps delay the convergence only longer to get a good solution and not too longer.

The social disaster technique (SDT) was introduced by Kureichick et al. (1996) in order to avoid the premature con-

vergence of GA to local optima. It diagnoses the situations of loss of genetic diversity in the population and applies a catastrophic operator in order to give the required diversity to the population.

Standard population diversity (SPD) is a way of maintaining the diversity of the individuals in the population so as to avoid the premature convergence problem. However, the fitness of individuals is not considered in such cases (Mc Ginley et al. 2011). However, Mc Ginley et al. (2011) introduce healthy population diversity (HPD) that accounts to a healthy diversity among individuals of the population by quantifying diversity of individuals that show better fitness values

While the dynamic application of crossover and mutation operators is from the beginning to the end of the GA, population reinitialization is applied to a part of the population only at a point when the population is detected to be stagnant (Nicoara 2009). Earlier methods include reinitializing part of the population of every generation of the GA (Nicoara 2009).

Random offspring generation (ROG) tests an individual's genetic material, before crossover is performed, and in case they share common genetic material, one or both the offsprings are randomly generated since randomness increases diversity by introducing the GA to new and unexplored subsections of the solution space (Rocha and Neves 1999).

While the first five methods discussed here are based on adapting one of the components of GA, the next four methods aim at improving the diversity of the population at the same time trying not to delay the convergence too much and the last two methods introduce diversity with the help of introducing randomness to the population. We find that all these methods share one thing in common that is involving dynamic computation time and resources.

2.2 GA operators to avoid premature convergence

In this section, we discuss about the GA operators that were designed specially to handle the premature convergence problem.

The cleanup operator was designed specifically to handle premature convergence in the traveling salesman problem (TSP) (Mitchell et al. 2000). This operator was used to find and correct errors (such as unequal tour length, visiting a city more than once and not visiting certain cities in the tour) in the offspring formed as the result of crossover operation. This reduces the circulation of the invalid offspring, thereby promoting chances of obtaining a global optimum. However, the operator was designed for a specific domain which cannot be applied to other domains.

The convergence acceleration operator (Adra et al. 2009) gives better solutions in the objective space and uses neural network mapping schemes for predicting the required solution area in the decision variable space. Here, better solutions are obtained but not at the cost of the convergence time.

Duplication operator essentially creates the clone of the best-fit individual and the fabrication operator uses the statistical information from the elite chromosome base to fabricate a new individual (Chang et al. 1995) after which the regular GA operations are done. These new operators help in faster convergence to a better solution according to Chang et al. (1995). However, the statistical information of the area of search space which may hold the potential solutions that is not present in the population is not taken into consideration.

A probabilistic Boltzmann function is used for reduction process (Lozano et al. 1999), which does the job of a selection operator for selecting a subset of the population for the next generation. This reduces unfit individuals from participating in further evolution, thereby improving the chances of converging at global optimum rather than at local optimum. Unfit individuals may have good alleles for certain genes that may account to the globally optimum solution may be lost here.

The following mutation operators were designed to prevent the premature convergence problem.

Two groups of mutation were proposed by Ramadan (2013) where the Group I consists of five of the nine mutation operators, namely 1) Ends Exchange mutation (EsEm), 2) Group Insertion mutation (GIm), 3) Group Insertion mutation-2 (GIm2) 4) Reverse Ends mutation (REsm) and 5) Two Genes Exchange mutation (TGsEm), and Group II consists of the remaining four crossover operators, namely the 6) Reverse Ends Exchange mutation (REsEm), 7) Reverse Ends mutation (REm), 8) One Position Swap mutation (OPSm) and 9) Middle Reverse mutation (MRm). Group I mutations changes more than two links in the round trip (exploration purpose since major change is introduced) and Group II mutations changes only two links in the round trip (exploitation purpose since minor changes are introduced).

Gene-based adaptive mutation where the mutation rates are adjusted for the gene locations based on the feedback of the success or failure of the previously employed mutation rates (Uyar et al. 2004).

The improved mutation operator preserves the highest fit allele and uses the frequency of occurrences for calculating the fitness of the allele, thereby maintaining search in the promising areas of the solution space (Yusof et al. 2012).

Pavai and Geetha (2016) in their work on the survey of existing crossover operators have cited that crossover is effective in handling the premature convergence problem. They have classified some crossover operators as aiming faster convergence of GAs without affecting the quality of the obtained solution and some other crossover operators as aiming to avoid the premature convergence problem. They discussed the following crossover operators in their work for this purpose.

Rank and proximity-based crossover (RPBC) (Chakraborty and Hoshi 1999)—Chromosomes of similar ranks and from closer locations are crossed over so that the disruption of good chromosomes is avoided, and therefore, good solution is converged quicker.

MMX-BLXexploit and MMX-BLXexplore—These two operators were designed to have a check on the two most important functions of crossover, i.e., exploitation and exploration (Roy et al. 2015) so that convergence does not occur too early (checked by exploration) and not too late (checked by exploitation).

New crossover operator—The new crossover operator (Lunacek et al. 2006) preserves the valid bit combinations, thereby trying to utilize the sparse area containing useful solutions in the search space. This heuristic brought about faster convergence to effective solutions.

MPX, MLX—Multi-parent crossover operators and multi-parent crossover operators are inherently designed for faster convergence since an offspring is generated using features of more than two parents. MPX (multi-parent crossover with polynomial distribution) and MLX (multi-parent crossover with lognormal distribution) are multi-parent crossover operators (Patel and Raghuvanshi 2011) that target multi-objective optimization.

Frequency crossover—Frequency of genes is calculated between the parents and genes that are common to both are carried over to the offspring and the remaining genes are filled preserving the order of the genes in order to stabilize the order of the genes across generations (Ramadan 2013). This indeed reduces the diversity and increases the risk of premature convergence in the population. Hence, the nine mutation techniques proposed in the previous section were used along with this crossover operator to introduce diversity and reduce premature convergence.

Adapted crossover (Durand and Alliot 1998)—This crossover operator optimizes the global function by optimizing each variable separately considering the problem of completely separable functions. A single child is created from two parents, by choosing a gene that locally fits better from either parent, with an additional operator called the determinism operator.

Crossover operators to create diversity (and therefore reduce premature convergence)—Goldberg and Lingle's partially mapped crossover operator (PMX) maps a segment of one of the parent's chromosomes to a segment of the other parent's chromosome and exchanges the remaining genes (Goldberg et al. 1985). Oliver et al. (1987) create offspring from the parents by copying the gene value and its position from the parents to the offspring based on the feasibility of the chromosome in their cycle crossover operator (CX). Brady's sorted match crossover identifies similar subtours in the parents and generates offspring by replacing the least cost subtour with the most cost one

(Brady 1985). The chaotic crossover operator (Demirci et al. 2015) produces a single child from two parents by weighting the genes of both the parents considered using chaotic random number generator for getting near-global optimal solutions. This crossover operator improves the diversity of the population using the chaotic randomness, thereby avoiding the local minima problem. Gwak et al. (2016) proposed an ensemble of mean-centric and parent-centric crossover operators help in avoiding convergence to local optima.

Mutation is known for introducing diversity in the population that may help prevent the convergence to local optimum. However, on the other hand, mutation increases the number of generations to converge to a solution. Almost all of the methods discussed in Sect. 2.1 as well as the most GA operators discussed in Sect. 2.2 (other than crossover) aim at obtaining global optimum by increasing the diversity of the population. However, the more the population is diverse, the more number of generations are taken to converge to the solution. It is believed that crossover operator's basic nature is converging to a solution. Moreover, no diversity is introduced to the population by the crossover operator (Mc Ginley et al. 2011). Hence, crossover is the right candidate for faster convergence. Though the convergence is faster, there is no guarantee for a global optimum solution when existing crossover operators alone are used. Hence, the local minima problem alone needs to be concentrated on in the design of the crossover operator, leaving the convergence to be handled by the inherent nature of crossover operators to converge quickly.

Among the crossover operators designed to handle the premature convergence problem that was discussed in Sect. 2.1, the RPBC operator (Chakraborty and Hoshi 1999) is similar to our work in a way that it preserves the good genes just as we select and preserve the dominant genes. However, our crossover operators preserve the good (dominant) genes wherever they are not necessarily in close proximity as in RPBC. The adapted crossover (Durand and Alliot 1998) is also similar to our crossover operators in a way that a single best child for the parents considered is created. However, offspring formation is controlled by the determinism operator in adapted crossover, whereas no such operator influences our proposed crossover operators. Hence, we have used these two crossover operators as baselines to evaluate our newly designed crossover operators. Furthermore, the multi-parent crossover operators also aim at faster convergence which has motivated us to develop and test the multi-parent version of our newly proposed crossover operators. In addition to this, since mutation is also an important GA operator and it is widely used to handle the premature convergence problem, we have studied the effects of our crossover operators in the presence and absence of mutation.

2.3 Other dominance principle-based crossover operators

In this section, we discuss other crossover operators that are close to the design of our proposed crossover operators and how they differ from our work.

The dominance and diploidy-based crossover (Deshpande and Kelkar 2008) was designed for the binary strings. They have assumed the value of '1' to be dominant and a value of '0' to be recessive. We, however, have defined the notion of dominance for integer- and real-coded chromosomes for each gene based on each gene's contribution to the overall fitness of the entire chromosome which is far beyond the assumption described by Deshpande and Kelkar (2008).

Selective crossover (Vekaria and Clack 1998) also follows dominance principle, however, 'without diploidy' according to Vekaria and Clack (1998). This is basically an adaptive crossover where the dominance for every gene changes from generation to generation based on the increase or decrease in the overall fitness of the offspring formed by this crossover. Our method finds dominant gene values initially based on the individual's contribution to the fitness of the entire chromosome and dynamic adaptation of dominance does not happen in our method.

Based on the selective crossover, two new crossover operators were designed for genetic programming called simple selective crossover and dominance selective crossover by Yuen (2004). These two crossover operators are also adaptive in nature that changes the dominance vector in every generation in order to obtain the exact dominance vector for which the optimal solution can be obtained. However, our proposed crossover operators determine the dominance and the co-dominance vectors initially before starting the GA process.

In the next two sections, we discuss the basic principles of dominance and co-dominance from the biological perspective. The proposed crossover operators are based on these principles.

2.4 Dominance law of mendel

Mendel's law of dominance can be stated as, 'In a cross of parents that are pure for contrasting traits, only one form of the trait will appear in the next generation. Offspring that are hybrid for a trait will have only the dominant trait in the phenotype.' (http://en.wikipedia.org/wiki/Mendelian_inheritance).

While Mendel was crossing (reproducing) his pea plants (over & over & over again), he noticed something interesting. When he crossed pure tall plants with pure short plants, all the new pea plants (referred to as the F1 generation) were tall. Similarly, crossing pure yellow seeded pea plants and pure green seeded pea plants produced an F1 generation of

all yellow seeded pea plants. From these experiments, he concluded that if either of the parents have a dominant feature (e.g. Tall compared to short), the child gets the dominant gene for the height characteristic.

Though genetic algorithms advocate randomness in the population, in human genetics, we go in for a biased choice of the child's gene. The bias is that the child gets the dominant of the genes for each of its features. In our work, we find which parent has a dominant value for a feature and pass on only the dominant values from generation to generation. It is in effect selecting good genes from both the parents and forming the child. We agree that this does not exist in nature. However, there are works such as multi-parent crossover operators for GA (Kita et al. 1999), which is also against nature since a child is formed from one parent (asexual reproduction) or two parents (bisexual reproduction) only. The multi-parent crossover operators were developed from the view point of optimizing machine solvable problems. Our work also is based on the same roots.

2.5 Co-dominance

Co-dominance (Gardner et al. 2006; http://en.wikipedia.org/wiki/Mendelian_inheritance) is yet another trait that exists in nature, which we try to adapt to the genetic algorithms. Co-dominance is said to exist when two different traits of the same gene are both dominant and both the dominant genes are being expressed in the children formed.

ABO blood groups are a good example where co-dominance exists. Here, both 'A' and 'B' traits are dominant and the 'O' trait is recessive. As it goes with the dominance law, stated previously, When a parent with 'A' allele or 'B' allele and another parent with 'O' allele should form a child, the child has 'A' or 'B' blood group with one dominant allele ('A' or 'B') and a recessive allele 'O'. However, when both the parents contribute 'O' allele, the child gets a 'O' group. On the other hand, when a parent contributes 'A' trait and the other contributes 'B' trait, since both these traits are dominant, the child gets 'AB' blood group. This effect is called co-dominance.

We adapt co-dominance to the genetic algorithms and prove that GAs performance improves on using this concept to design the crossover operator, which is shown clearly in the subsequent sections.

3 Design of the crossover operators

We discuss the new crossover operators in this section. Our design consists of two main steps. In the first step, the dominance vector is determined which finds and stores the order of dominance of various values of genes, and in the second

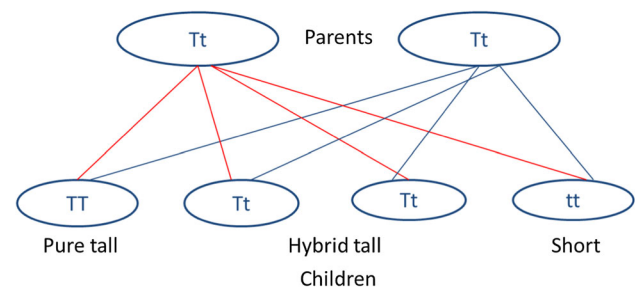


Fig. 1 Dominance-based crossover in human genetics (for the feature Height)

step, the actual GA is carried out with our new crossover operators.

3.1 Learning the dominance order of the alleles

We should first define what the dominant values are for each feature. For example, the dominant value for the feature height in human genetics is tall (Genotype-T), whereas the recessive value is short (Genotype-t). The dominance-based crossover in human genetics is explained in Fig. 1.

Definition 1 Given a set of features $f = \{f_1, f_2, \dots, f_n\}$, f_i denotes a feature height characteristic given by HC, $1 \leq i \leq n$. Denote by DBC^{HG} , a dominance-based crossover in human genetics. Let $V = \{T, t\}$ represent the alleles of a gene of parents $P_i, i \leq n$ of order 2. Let the genotype values for the feature height in HG be $HC = \{\langle TT \rangle, \langle Tt \rangle, \langle tt \rangle\}$, $|HC| = 2^n$ inclusive of repetitions in the hybrid genotype case, be a set of height features that, respectively, denote pure tall, hybrid tall and short. Therefore, it follows that height of a child (C) is $f_i(C(P_1, P_2)) \in HC$.

There is a one-fourth probability of getting a pure tall child and a half probability of getting a hybrid tall child and again a one-fourth probability of getting a short child when two hybrid tall parents mate (definition 1). Our dominance crossover operator works in a similar fashion. That is, when either parent has a dominant gene value, it is carried over to the child.

Definition 2 HG defines dominance as the gene value that expresses itself leaving behind the recessive one, whereas we define dominance as the gene value that contributes the most to the fitness of the chromosome.

Definition 2 is very important for our work since we determine which feature value is dominant and which is recessive based on this definition so that only dominant genes values are passed on to the next generation.

We assume that the dominant allele is the one that contributes most to the fitness value of the chromosome. The method of determining the dominance vector for each feature is described in Fig. 2. As described in Fig. 2, we consider

For every feature f_i ,

For every value V_j of f_i ,

Calculate fitness contribution V_{ij} to the chromosome

Dominance \leftarrow Sort(V_{ij}, j)

Fig. 2 Forming the dominance vector

each feature and every value of each feature one by one and the contribution of each of these feature values to the fitness of the entire chromosome is calculated. This is carried out using the following equation.

$$\bar{f} = \max_i f_i p_i \quad (1)$$

where \bar{f} is the fitness contribution of a gene T to the whole chromosome; f_i is the fitness contribution of an allele; T_i (it is '1' if the global optimum value is obtained using this allele); p_i is the frequency of the allele T_i in the population

Once the fitness values are calculated, the features are sorted based on their fitness values. When the sorting is done, we have the dominant alleles according to the order of dominance in the vector 'dominance' in case of integer encoding. When real encoding is used, the range of alleles is discretized by dividing the range into slots (ten slots for most problems discussed in this work—determined by trial and error method). In case there are more than one allele with the same fitness value (this is what we consider 'equally dominant'), they are randomly ordered in the 'dominance' vector (in the case of real encoding, all equally dominant alleles fall into the same slot). We will be using this vector in the dominance crossover operator which is discussed in the next section.

3.2 Dominance crossover operator

Once the dominance vector is obtained, we use it in the dominance crossover operator. We use the normal GA framework. However, in place of the basic crossover operator, we use our dominance crossover operator.

Definition 3 Let G be the set of genes in P_i and C_i where each $g_i \in \{d, r\}$ such that $d_{>_g} r$ (ie. d has dominance over r). However, among $g_i, d_k > d_j$ is chosen for crossover on g while $r_k < r_j$ is chosen on r for some k, j . *Dominance crossover* may be formally defined as follows:

$g(C) = \{g_i | P_j >_g P_k, \forall g_i \in G, 1 \leq i \leq n\}$ where n is the number of genes for each parent and child, $P_1 >_g P_2$ denotes dominance of a gene's allele d of P_j over that of P_k . Note that P_j and P_k may be used interchangeably to represent P_1 or P_2 , based on dominance at a position.

According to Definition 3, the dominance crossover operator creates one best child (C) for the given parents (P_1 and

Table 1 Co-dominance in ABO blood groups

Blood type	Genotype	
A	$i^A i^A$	AA
	$i^A i^O$	AO
B	$i^B i^B$	BB
	$i^B i^O$	BO
AB	$i^A i^B$	AB
O	$i^O i^O$	OO

For every feature f_i ,

For every value V_j of f_i ,

Calculate fitness contribution of V_{ij} to chromosome

Co-Dominance \leftarrow hash-table(V_{ij}, j)

Fig. 3 Determining co-dominance hash table

P_2) by selecting the most dominant value for each gene of the child from its parents. In case both the alleles are equally dominant, we choose one of them randomly and pass it on to the child.

3.3 Learning co-dominance of alleles

As we have described earlier, co-dominance is the presence of more than one dominant allele for the same gene. This is better understood with the help of Table 1. A and B blood groups are formed when the genes i^A or i^B are present regardless of whether the second gene is the same (i^A or i^B , respectively) or i^O . However, in the presence of each other (the genes i^A and i^B), both are equally dominant, thereby giving rise to the 'AB' blood group. In the absence of dominant genes, 'O' group is formed.

Determining co-dominance vector for each feature is described in Fig. 3. Calculating the fitness values is the same as we do for forming the 'dominance' vector. However, once the fitness values are calculated, when the features are sorted based on their fitness values, hash table is used instead of an array so that the 'equally dominant' alleles fall into the same bucket. We call this 'co-dominance' hash table. We will be using this hash table in the co-dominance crossover operator which is discussed in the next section.

3.4 Co-dominance crossover operator

Once the co-dominance hash table is formed, we can use this in the co-dominance crossover operator. The co-dominance-based crossover operator works in a similar fashion as that of the dominance crossover operator except when there is co-

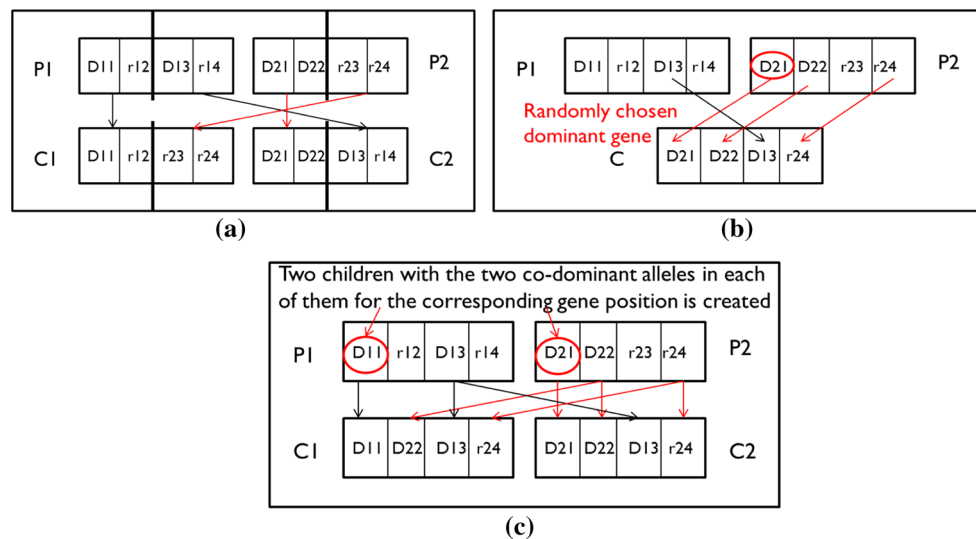


Fig. 4 Illustration of crossover operators. **a** Classical one-point crossover. **b** Dominance crossover. **c** Co-dominance crossover

dominance (multiple alleles of a gene show the same fitness value).

Definition 4 Co-dominance crossover extends dominance crossover. Co-dominance crossover may be formally defined as follows:

$g(C_s) = g_p^l U\{g_m \mid P_{j>g} P_k, \forall g_i \in G, 1 \leq m, p \leq n, m \neq p \text{ and } l=1,2 \text{ denoting the parents}\}$, for some i , where p denotes position of gene where both parents have an equally dominant allele and s is the number of children formed due to a co-dominance crossover on two parents, $1 < s < a$, $a = 2^b$, where $b = \text{number of equally dominant genes in both the parents}$.

As per Definition 4, we form two different children with the same set of alleles for all genes except in places where co-dominance is exhibited. In the gene positions where co-dominance is present, each child has one of the parents' alleles. We assume co-dominance helps in retaining the good genes in the population rather than losing them just because another dominant allele for the gene is present. This helps in widening the search space without losing the good genes.

3.5 Illustration of the crossover operators

The illustration of the newly designed crossover operators is given in Fig. 4. For a simpler understanding, we have considered the number of genes to be four. We have considered the illustration of the basic one-point crossover for the same reason.

Definition 5 Let \otimes denote one-point crossover operator. Then $P_1 \otimes P_2$ where $P_1 = \{g_1^i, g_2^i, \dots, g_n^i\}$ where $i=1,2$ and $C_i = \{g_1^i, g_2^i, \dots, g_n^i\}$ where $i=1,2$ and $g \in G$ where G is a set of all genes for the given application. Let crossover point

$CP=j$ where $1 < j < n$. $C_1 = \{g_1^1, g_2^1, \dots, g_j^1, g_{j+1}^2, \dots, g_n^2\}$ and $C_2 = \{g_1^2, g_2^2, \dots, g_j^2, g_{j+1}^1, \dots, g_n^1\}$.

As per Definition 5, the offspring for the one-point crossover is formed by swapping the segments before and after the cut-point. This is clearly shown in Fig. 4a (here, the cut-point is after the second gene). Consider child 'C1' in Fig. 4a to have a low fitness. This leads to subsequently dropping the child 'C1' from the next generation. The dominant allele 'D11' is lost in this case as it is removed from the solution space. Figure 4b shows the illustration of the dominance crossover operator, and Fig. 4c shows the illustration of the co-dominance crossover operator. When there is only one dominant allele in both the parents for a gene, it is chosen for the offspring by both the new crossover operators. When both the alleles are recessive, any one of them is randomly chosen for the gene of the offspring in both the new crossover operators. As we can see, 'D11' and 'D21' alleles are both dominant for the first gene. The dominance crossover operator selects 'D21' randomly for the first gene of the offspring leaving behind the allele 'D11' which could have been a potential point in the optimal solution. Therefore, it is evident that there is a possibility of losing an allele though it is dominant when the other parent involved in the dominance crossover operation also contains a dominant allele. On the other hand, the co-dominant crossover operator creates two children with both the dominant alleles in each child for the corresponding gene position. This is clearly depicted in Fig. 4c where 'D11' and 'D21' are both dominant alleles for the first gene position. Two children 'C1' and 'C2' are created with 'D11' as the allele and 'D21' as the allele in both the offspring, respectively. From this, it is clear that dominance crossover operator preserves the dominant alleles in all cases except when co-dominance is present. In the presence

Table 2 Hardy–Weinberg frequencies for two alleles

Genotype	TT	Tt	tt
Frequency	X^2	$2X(1 - X)$	$(1 - X)^2$

of co-dominance (i.e., multiple alleles being equally dominant for a gene), co-dominance crossover operator helps to preserve all good alleles.

The solution obtained by using the co-dominance crossover operator can be proved to be the optimal solution when the following assumption holds good: ‘When all the dominant alleles for all the gene positions are present in the initial population.’ The necessity for the presence of all alleles of all the genes is that no dominant allele should be missed out from the solution space. Since all the dominant alleles are already present in the initial population and due to the nature of the co-dominance crossover operator that all dominant alleles are preserved, each gene has the best possible allele, and hence, the resulting convergence is perfectly at the global optima. This is true for encodings such as binary and integer where there are a finite number of alleles for each gene and the initial population covers all of these alleles. From this, we conclude that our crossover operators guarantee the best solution for the given initial population, and if the initial population has all the alleles of all the genes, then the best solution obtained is also the optimal solution.

3.6 Convergence of the new crossover operators

In this section, we discuss the convergence of the crossover operators. We need to know about the famous Hardy–Weinberg Principle which states that ‘In the absence of evolutionary causes, like mutation, migration, natural selection, and random genetic drift, the gene frequencies of the population will remain unchanged from one generation to the next; the population will be in Hardy–Weinberg equilibrium’ (Felsenstein 2015). In Hardy–Weinberg equilibrium, the distribution of genotype frequencies will follow the binomial distribution. For example, if a gene has two alleles say ‘T-Dominant’ and ‘t-recessive,’ then the Hardy–Weinberg frequencies are given as shown in Table 2.

The Hardy–Weinberg frequencies all sum up to one as shown in equation 2. When this is true, the population is said to be in Hardy–Weinberg equilibrium.

$$X^2 + 2X(1 - X) + (1 - X)^2 = 1 \quad (2)$$

It is to be noted that no evolutionary causes are present according to the Hardy–Weinberg principle. However, in our case, ‘dominance’ is a force that influences crossover operation, which otherwise is done purely in a random fashion. In any generation, the frequencies all sum up to one as shown

in Eq. 2. As the result of ‘dominance’ acting on the population, the subsequent generations see a rise in the frequency of the Dominant alleles and reduction in the frequency of the recessive alleles as given in Eq. 3.

$$\text{As number of generations } \uparrow X^2 \approx 1 \text{ and } (1 - X)^2 \approx 0 \quad (3)$$

This can be argued as follows: If $X(n)$ is the number of ‘A’ alleles in the population, and $X(n + 1)$ is the number of ‘A’ alleles in the next generation, under Hardy–Weinberg equilibrium,

$$X(t) = X(t + 1) \quad (4)$$

According to our dominance principle,

$$X(t) < X(t + 1) \text{ if A is Dominant} \quad (5)$$

and

$$X(t) > X(t + 1) \text{ if A is recessive} \quad (6)$$

So, according to Eqs. 5 and 6, Eq. 3 happens. In the presence of dominance, i.e., substituting Eq. 3 in Eq. 2, the Hardy–Weinberg frequencies are not maintained in the presence of dominance since dominant genes grow in number and the recessive genes are reduced in number in subsequent generations. However, the Hardy–Weinberg equilibrium is still maintained. So, we can conclude that introducing dominance maintains the Hardy–Weinberg equilibrium, and hence, the convergence of the dominance crossover operator is unaffected. The same explanation also holds good for the co-dominance crossover operator since, it follows the same dominance principle.

4 Results and discussion

All the algorithms were implemented in Java, and the experiments were carried out on a PIV 2.8 GHz machine with 4 GB RAM under WINDOWS-7 platform.

We have evaluated our new crossover operators on test functions as well as applications. Section 4.1 shows the evaluation on test functions, and Sect. 4.2 shows the evaluation on applications.

4.1 Evaluation on test functions

The evaluation of the proposed crossover operators using test functions is carried out based on fitness and the number of generations to converge to a solution. Specifically this was done to check whether the proposed crossover operators were able to converge quickly to the global optimum. This section

discusses the performance of our crossover operators on the following test functions. We have considered different groups of functions such as the integer, mixed-integer and real-coded functions (since we claim that our crossover operators were designed for these two encodings) that have simple to very complex search spaces (in order to test the effectiveness of our crossover operators based on the complexity of the search space) and that are completely separable, partially separable and non-separable (in order to test the effectiveness of our crossover operators based on the separable nature of the test functions).

$$1. \text{ F1: Step function: } \min f(x) = \sum_{i=1}^{30} \|x_i\|$$

$$-100 \leq x_i \leq 100; \text{ integer}$$

$$\text{Global optimum: } (x_1, x_2, \dots, x_i : f) = (0, 0, \dots, 0; 0)$$

$$2. \text{ F2: Sphere function: } \min f(x) = \sum_{i=1}^{30} x_i^2$$

$$-5 \leq x_i \leq 5; \text{ integer}$$

$$\text{Global optimum: } (x_1, x_2, \dots, x_i : f) = (0, 0, \dots, 0; 0)$$

$$3. \text{ F3: Sphere function: } \min f(x) = \sum_{i=1}^{30} x_i^2$$

$$-100 \leq x_i \leq 100; \text{ integer}$$

$$\text{Global optimum: } (x_1, x_2, \dots, x_i : f) = (0, 0, \dots, 0; 0)$$

$$4. \text{ F4: Schwefel 2.4 function: } \min f(x) = \sum_{i=1}^D \left[100(x_i - 1)^2 + (x_1 - x_i^2)^2 \right]$$

$$0 \leq x_i \leq 10; \text{ real}$$

$$\text{Global optimum: } (x_1, x_2, \dots, x_i : f) = (1, 1, \dots, 1; 0)$$

$$5. \text{ F5: Rastrigin function: } \min f(x) = \sum_{i=1}^{30} (x_i^2 - 10 \cos(2\pi x_i)) + 10$$

$$-5.12 \leq x_i \leq 5.12; \text{ real}$$

$$\text{Global optimum: } (x_1, x_2, \dots, x_i : f) = (0, 0, \dots, 0; 0)$$

The functions (F1 to F5) are taken from Sahnehsaraei et al. (2012). The step function (F1) is the simplest test function which is discrete and unimodal. Next comes the sphere function (F2 and F3) which is discrete and multimodal. We have used the sphere function twice to test the crossover operators one for a search space of $[-5, 5]$ and another for a search space of $[-100, 100]$. This was done to show that our operators performed the best when the dominant alleles for all genes are present in the initial population. The Schwefel

2.4 (F4) function is continuous and has multiple optima. Convergence to the global optimum is difficult in this case which makes this problem important in assessing performance of GAs. Rastrigin function (F5) is a modified Sphere function with a cosine modular term. It has a large number of local minima apart from one global minimum and hence highly multimodal. However, this function is not deceptive since the minima are distributed at regular intervals.

Each problem was run 50 times (as in Sahnehsaraei et al. (2012) and Vekaria and Clack (1998)) and each time with a different initial population. However, the same initial population was used for a run of all the four crossover operators considered for evaluation. The initial population size was taken as 100 in the light of the existing literature such as Sahnehsaraei et al. (2012) for all the runs. We have restricted the maximum number of generations for each run as 100. The boundary constraints, additional constraints if any, encoding of the genes and the global optimum values along with the number of dimensions for all these functions are given in the above description of the functions. We have considered two baseline crossover operators, namely the RPBC (Chakraborty and Hoshi 1999) and the adapted (Durand and Alliot 1998) crossover operators as baselines since they are similar to our work, i.e., crossover operators aiming for faster convergence with better fitness values. Neither of the crossover operators described in Sect. 2.3 although close to our dominance method are not taken as baselines for the below reasons. i) The dominance and diploidy-based crossover (Deshpande and Kelkar 2008) was basically designed for binary strings and we evaluate for real- and integer-encoded functions. ii) the selective (Vekaria and Clack 1998), simple selective (Yuen 2004) and the dominance selective (Yuen 2004) crossover operators are adaptive in nature, whereas our crossover operators are not adaptive. Moreover, other hybrid (Herrera et al. 2003) and ensemble-based (Gwak et al. 2016) crossover operators were not used as baselines since we have designed two standalone crossover operators which are heuristic in nature and hence cannot be compared with synergetic effect producing methods. Mutation was not used here so as to study the complete effect of the crossover operators alone on the performance of GA (the performance of crossover in the presence of mutation is discussed later). Roulette wheel selection was used for selection of individuals for mating (except RPBC where rank and proximity-based selection is used).

The best (B), average (A) and worst (W) fitness (F) and number of generations (G) of convergence for the five test functions in 50 runs is summarized in Table 3. From the table, it is clearly evident that the dominance and co-dominance crossover operators perform better than the baselines. The RPBC and adapted crossover operators perform similar to each other. However, the adapted crossover operator seems to be better in terms of both fitness as well as number

of generations as given in the table. Moreover, the co-dominance crossover operator produced even better results than the dominance crossover operator, however, with a few more generations. The delay in convergence is due to the co-dominance crossover's principle of retaining both the dominant alleles of a gene in case two dominant alleles considered for crossover are present in a gene of the parent.

As given in Table 3, both the dominance and the co-dominance crossover operators have obtained the global minimum values for the functions F1 and F2 in all the runs, thereby giving a best-, average- and worst-case fitness to be a global minimum for the respective functions. F3 is 30-dimensional. From the results, it is clear that even higher dimensional functions can be solved to obtain global optimum using the newly proposed crossover operators. However, it is very clear that function F3 converges a few generations later when compared to the function F2. The results of the rest of the functions are not solved by the proposed crossover operators to get the global minimum fitness of '0' in every run of the GA. However, we can see that the functions F4 and F5 have a best-case fitness value which is equal to the global optimum even though the worst and the average fitness is not the global minimum value using both our proposed crossover operators (except for F4 and F5 which does not have a global optimum for dominance crossover). On a closer examination of the initial population in such cases, we found that the initial population had alleles for all the genes that produced the global minimum as given in the description of the respective functions. From this, it is very clear that all the dominant alleles if present in the initial population surely make the function converge to the global minimum. We also found that in the rest of the cases where global optimum is not found the initial population did not contain the required alleles for the genes that are required for global optimum convergence. Thus, our statement 'The solution obtained by using the co-dominance crossover operator can be proved to be the optimal solution when the following assumption holds good: When all the dominant alleles for all the gene positions are present in the initial population' is experimentally proved here.

While the integer-encoded functions (F1, F2 and F3) give near-global optimum solutions, the real-encoded functions (F4 and F5) give solutions that are far from the global minimum. This is due to the fact that real encoding has a wider range of values for the genes, thereby reducing the possibility of the presence of the dominant genes in the initial population. This prevents the functions from converging at the global optimum. However, Table 3 shows that the best-case results of co-dominance crossover on both these functions are global optimum.

On examination of the initial population of these cases, it was found that the necessary dominant genes were present which caused the convergence to global optimum. In con-

Table 3 Best (B), average (A) and worst (W) fitness (F) and number of generations (G) of convergence for the test functions in 50 runs

		RPBC	Adapted	Dominance	Co-dominance
F1	F-B	0.41	0.022	0	0
	A	0.44	0.054	0	0
	W	0.46	0.122	0	0
	G-B	25	21	12	13
	A	22	19	12	13
	W	21	17	12	13
F2	F-B	0.45	0.044	0	0
	A	0.51	0.15	0	0
	W	0.53	0.172	0	0
	G-B	27	23	13	14
	A	24	20	13	14
	W	23	18	13	14
F3	F-B	4.314	4.149	0	0
	A	5.147	4.447	0.13	0.0252
	W	5.27	4.682	0.17	0.071
	G-B	39	32	23	27
	A	36	30	22	26
	W	34	29	22	26
F4	F-B	9.239	6.843	0.00024	0
	A	12.747	7.781	0.882	0.741
	W	14.430	9.235	2.336	2.762
	G-B	39	34	18	23
	A	34	31	17	22
	W	29	28	17	22
F5	F-B	10.432	7.518	0.000651	0
	A	10.892	7.983	0.887	0.69
	W	11.552	8.769	3.543	2.997
	G-B	45	40	21	28
	A	44	38	20	26
	W	42	35	19	23
F6	F-B	0.33	0	0	0
	A	0.38	0.012	0	0
	W	0.41	0.15	0	0
	G-B	37	21	9	13
	A	34	20	9	13
	W	31	18	8	13
F7	F-B	12.227	11.052	8	8
	A	14.948	11.93	8	8
	W	15.025	14.021	8	8
	G-B	25	20	8	8
	A	24	18	8	8
	W	23	17	9	10
F8	F-B	4.1	1.78	0	0
	A	5.24	1.839	0.00015	0.00011
	W	9.53	1.895	0.00025	0.000016
	G-B	23	22	18	22

Table 3 continued

		RPBC	Adapted	Dominance	Co-dominance
F9	A	21	20	16	22
	W	19	19	16	21
	F-B	6.561	0	0	0
	A	8.284	4.257	1.325	0.87
	W	11.225	7.462	3.824	3.371
	G-B	51	45	25	39
F10	A	49	42	23	36
	W	45	39	22	32
	F-B	8.479	7.663	0.0005	0
	A	10.521	7.218	0.975	0.813
	W	13.482	8.341	3.81	3.67
	G-B	38	32	19	25
	A	35	31	19	23
	W	31	30	19	23

clusion, the proposed crossover operators perform less than expected when real-valued chromosomes are used to represent the solution since the range of real values is vast, and hence, it is difficult to find the exact global optimum gene values, when the initial population does not contain all possible gene values or the optimum gene values. The same reason is applicable for the difficulty in finding the global optimum in the case of using real chromosomes. When compared to the results obtained using the baseline crossover operators, the proposed crossover operators give better solutions.

The highlight of the proposed crossover operators is the number of generation at which the solution was obtained. As given in Table 3, both the proposed crossover operators showed a remarkable decrease in the number of generations taken to obtain the solution. The importance of the proposed crossover operator is that the optimal solution (which is global optimum in case all dominant genes are present in the initial population) is obtained in a remarkably less number of generations. As we said earlier, the bias introduced by the dominance and co-dominance principles in the selection of alleles while forming the offspring is the reason for the faster convergence. Furthermore, the dominance crossover operator converges to a solution in generations less than or equal to that of the co-dominance crossover operator even when they work with the same initial population which contains all the dominant alleles (best-case performance of dominance and co-dominance crossover operators for functions F4 and F5 in Table 3). This is due to the co-dominance principle of retaining the co-dominant genes so that no dominant allele is unnecessarily lost. If there exists more than one dominant value for a gene, then the gene may get lost due to the randomness while choosing an allele for a gene in the

dominance crossover operator, whereas this does not happen in the co-dominance case as both the alleles are retained.

Apart from the test functions given here, we also evaluated the same set of crossover operators on two other test functions, namely the Hansen function and the Shubert function (Jamil and Yang 2013) both having more than one global optima. The test results are not produced here due to a similarity they show with the evaluations in Table 3. However, some important points from the evaluation are given here. Neither of the crossover operators considered for evaluation converged (all members of the population become the same). However, some or all of the global optima were found out using the crossover operators in the evaluation. To be more particular, the dominance and co-dominance crossover operators found the global optima in lesser number of generations, whereas the other crossover operators found them a little later. In fact the co-dominance crossover operator was able to find all the global optima (in less than 22 generations) in cases where the initial population had all the dominant genes.

However, all functions discussed so far are separable. We have also used partially separable functions to test the effect of dominance and co-dominance on these kinds of functions. In a partially separable function, there is dependence between the variables unlike the functions that we saw earlier which do not have any dependence between them. Solving the partially separable functions is highly dependent on two important variables, namely 'n'—the number of genes and 'k'—the number of neighbors per gene. Consider the following function F6 Durand and Alliot (1998).

$$6. \text{ F6: } F(x_1, x_2, \dots, x_n) = \sum_{0 < i \neq j \leq n} \delta(x_i, x_j)$$

$$\delta(x_i, x_j) = 1 \text{ if } x_i \neq x_j \text{ and } 0 \text{ if } x_i = x_j$$

Every 'x' takes a value of either '0' or '1'

$$\text{Global optima: } (x_1, x_2, \dots, x_n) = (0, 0, \dots, 0) \text{ and } (1, 1, \dots, 1)$$

The function F6 is a partially separable problem with say $n=10$. Every 'x' takes a value of either '0' or '1'. The dominant alleles for all genes are found as described earlier. However, this function being partially separable, we treat a group of genes which are dependent and find the dominant alleles for the genes in that group rather than considering the genes independently as described earlier in Eq. 1. The interdependence between groups is handled by carrying the chosen dominant allele for the next group. In case of F6, a group is formed by any two genes. A minimum fitness of '0' is obtained when both the genes have the same alleles (either '0' or '1'). Consider this group to be genes one and two and the next group to be genes two and three. The value of either '0' or '1' for the gene two which we obtained in

the previous group is carried over for the next group. Overall fitness is calculated and the set of alleles contributing to the global optimum are stored as the dominant alleles. We were able to find at least one global minimum using the dominance crossover since based on the frequency of either '0' or '1' in the initial population that optimum is found out. The other optimum is found during some runs. However, our co-dominance crossover operator was able to find both the global minima during all the runs as both '0' and '1' are found to be co-dominant. This way, numerous unfit individuals are created, and hence, the time taken to find the solutions is greater than that of the dominance crossover operator. We arrived at the global minimum in an average of nine generations for F6 when ' $n = 10$ ' using dominance crossover operator and thirteen generations using co-dominance crossover operator as shown in Table 3.

The proposed crossover operators performed better than the baselines for F6. Notably, the co-dominance crossover operator found both the global minima. This is a simple case. Consider that ' x ' takes a value between 0–4 rather than '0' and '1' as described earlier. We will have five global optima in this case. The dominance crossover was able to identify at least one of the global optima in an average of nine generations, whereas the co-dominance crossover operator was able to identify all these global optima in an average of eighteen generations. This shows that the number of alleles for a gene affects the number of generations at which all the global optima are found in the case of the co-dominance crossover. However, this was possible only when the initial population contained all the alleles for all the genes. The RPBC and the adapted crossover did not find all the global optima. The adapted crossover that was designed to specifically handle the partially separable problems performed better than the RPBC both in terms of the fitness and in terms of the number of generations. The adapted crossover in few runs found more global optima when compared to dominant crossover and those were runs found to have missing dominant genes in the initial population. This is due to the dependence of dominant gene calculation on the frequency of the alleles which is not the case in the adapted crossover operator. The function F6 is a simple case of the partially separable functions where the numbers of genes that are dependent are only one at a time. Hence, we tried to assess the performance of our crossover operators on more complex functions.

7. F7: $\min f(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2$
Subject to :

$$x_1 + 2x_2 + x_4 \geq 4.0$$

$$x_2 + 2x_3 \geq 3.0$$

$$x_1 + 2x_5 \geq 5.0$$

$$x_1 + 2x_2 + 2x_3 \leq 6.0$$

$$2x_1 + x_3 \leq 4.0$$

$$x_1 + 4x_5 \leq 13.0$$

$$0 \leq x_i \leq 3 \quad i = 1, 2, \dots, 5; \text{ integer}$$

Global optimum: $(x_1, x_2, x_3, x_4, x_5 : f) = (1, 1, 1, 2, 8)$

8. F8: Tripod function: $f(x) = p(x_2)[1 + p(x_1)] + |x_1 + 50p(x_2)[1 - 2p(x_1)]| + |x_2 + 50[1 - 2p(x_2)]|$

$$-100 \leq x_i \leq 100; \text{ integer}$$

$$\text{where } p(x) = 1 \text{ for } x \geq 0$$

Global optimum: $(x_1, x_2 : f) = (0, -50; 0)$

9. F9: Griewank function:

$$F(x_1, x_2, \dots, x_n) = 1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right)$$

$$-600 \leq x_i \leq 600; \text{ real}$$

Global optimum: $(x_1, x_2, \dots, x_i : f) = (0, 0, \dots, 0; 0)$

10. F10: Rosenbrock's valley function: $\min f(x) = \sum_{i=1}^{29} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$

$$-2.048 \leq x_i \leq 2.048; \text{ real}$$

Global optimum: $(x_1, x_2, \dots, x_i : f) = (1, 1, \dots, 1; 0)$

The function F7 is taken from Deep et al. (2009) and it is a non-scalable function. This function is used to test the integer-coded problem and it takes integer values for all genes. This function has additional constraints to be satisfied as given in the above description of the function F7. Hence, this can be considered as a partially separable function. Our crossover operators work as described for function F6 with a group being a constraint given in the description of function F7. The results from Table 3 show that the global optimum is achieved during every run. This is possible since the solution space is integer and is finite, and hence, the solution space contains all the dominant alleles for all the genes.

Next comes the tripod function (F8) which is taken from Schoene (2011). Although this function is two dimensional, it is highly deceptive in that the GA might have the problem of getting 'trapped in the large neighborhoods of the two local optima' according to Schoene (2011). The function F8 is a deceptive function as mentioned earlier. The table shows that the dominance and the co-dominance crossover operators have obtained the global optimum of '0' in the best case though the function is deceptive. We can also see that the RPBC is worst affected crossover for this function as shown in Table 3. This is mainly due to the fact that RPBC concentrates on similar ranked individuals which eventually lead to

getting trapped in the local minima. The table also shows that the adapted crossover performs better than the RPBC in this case. This is due to the local optimization principle it follows. We do not concentrate on the specific areas such as in the RPBC. At the same time, the dominance principle is better than the local optimization principle followed by the adapted crossover operator, and hence, the results obtained for the dominance and the co-dominance crossover operator are better. This function in spite of being a non-separable problem converges to global optimum in lesser number of generations when compared to the baseline operators. One of the main reasons for this is the problem being two dimensional and having discrete solution space.

The Griewank function (F9) is a partially separable function where the dependence is higher. This function is taken from Durand and Alliot (1998). This is a very complex function with many local minima and one global minimum. We considered $n = 10$ for this function. The number of slots for this function alone is sixty instead of ten as in the other cases owing to the extremely large range of the allele values (determined by trial and error method). Owing to the fact that the search space is continuous and due to this all dominant alleles may not be present in the initial population, both the dominance and the co-dominance crossover operators did not find the global minimum during all the runs. However, both these proposed crossover operators found the global minimum in fewer number of generations (average of 23 for dominance crossover and 36 for co-dominance crossover). Furthermore, the adapted crossover operator being specifically designed for partially separable problems did find the global minimum in some of the runs (with an average of 42 generations). The RPBC suffered the most with none of the runs we considered finding the global minimum. As we can see, F8 being a separable function converges better in lesser number of generations when compared to F9 which is a partially separable function. This is because of the very high dependence between the genes and the solution space being vast.

The Rosenbrock's valley function (F10) taken from Sahnehsaraei et al. (2012) is continuous, is non-separable and has a single global optimum inside a long, narrow, parabola-shaped flat valley. Convergence to the global optimum is difficult in this case which makes this problem important in assessing performance of GAs. Table 3 clearly shows that it does not converge to the global optimum using dominance and co-dominance operators as well due to the high dependence between the genes and large continuous search space.

The comparison between the crossover operators shows that dominance and co-dominance crossover operators perform better than the baselines in the absence of mutation and with Roulette wheel selection. However, the performance of crossover is highly dependent on these two genetic operators. Hence, we have evaluated the same test functions for the two proposed crossover operators when different selection and

mutation mechanisms were employed. For this, we considered the gene-based adaptive mutation (Uyar et al. 2004) and improved mutation (Yusof et al. 2012) owing to the fact that unlike most crossover operators, these were designed for faster convergence of genetic algorithms though generally mutation introduces diversity in the population, thereby leading to an increase in the number of generations to converge to a solution. Furthermore, some other mutation operators were proposed for encodings other than integer and real. Since we deal with only these two types of encoding, we have used these two mutation operators for evaluation. A probabilistic Boltzmann function (Lozano et al. 1999) and the rank and proximity-based selection (RPBS) (Chakraborty and Hoshi 1999) used for the RPBC operator are used as selection mechanisms. Again, these two selection mechanisms were designed for faster convergence of genetic algorithms with the RPBC showing better performance when compared to the other baseline. We considered these four operators since our main motive behind designing the proposed crossover operators is reducing the number of generations taken to converge to a solution without affecting the quality of the solution obtained and these operators were proved to do this.

Table 4 clearly shows that whatever the mutation or selection operator is, there is no or less significant difference in the fitness obtained or the number of generations required to obtain the solution in the case of dominance crossover for functions F1 and F2. This is due to the fact that the search spaces are small and the alleles are integers, i.e., neither mutation nor selection mechanisms affect the performance of the proposed crossover operators on these type of functions since the initial population has all possible alleles to the genes in the initial population itself. The same is also true for the functions F6 and F7. In the case of the rest of the functions, variations are observed. Both dominance and co-dominance crossover operators take more number of generations to obtain the optimal solution in the presence of mutation than in the absence of mutation. This is due to the small amount of diversity introduced in the population by the mutation operators. Comparatively, the co-dominance crossover with mutation takes more number of generations to obtain the optimal solution than the dominance crossover in the presence of mutation. This is due to the co-dominance principle of retaining the co-dominant alleles in order to make sure that no dominant allele is lost as described earlier. Furthermore, the dominance crossover did not find the global optimum in the case of functions F4 and F5 in the absence of mutation as shown in Table 3. However, in the presence of improved mutation, the dominance crossover operator also was able to find the global optimum and that too in less number of generations when compared to the co-dominance crossover operator as shown in Table 4. However, the dominance crossover operator took more number of generations to obtain the global optima in the presence of mutation than what the co-dominance crossover

Table 4 Best (B), average (A) and worst (W) fitness (F) and number of generations (G) of convergence for the test functions in 50 runs for dominance and co-dominance crossover operators using different selection and mutation operators

Selection →	Mutation →	Crossover →	Probabilistic Boltzmann function		RPBS		Probabilistic Boltzmann function		RPBS	
			Gene-based adaptive mutation		Gene-based adaptive mutation		Improved mutation		Improved mutation	
			Dominance	Co-dominance	Dominance	Co-dominance	Dominance	Co-dominance	Dominance	Co-dominance
F1	F-B		0	0	0	0	0	0	0	0
	A		0	0	0	0	0	0	0	0
	W		0	0	0	0	0	0	0	0
	G-B		12	13	12	13	12	13	12	13
	A		12	13	12	13	12	13	12	13
	W		12	13	12	13	12	13	12	13
F2	F-B		0	0	0	0	0	0	0	0
	A		0	0	0	0	0	0	0	0
	W		0	0	0	0	0	0	0	0
	G-B		13	14	13	14	13	14	13	14
	A		13	14	13	14	13	14	13	14
	W		13	14	13	14	13	14	13	14
F3	F-B		0	0	0	0	0	0	0	0
	A		0.13	0.11	0.11	0.10	0.09	0.08	0.09	0.05
	W		0.16	0.13	0.16	0.13	0.15	0.13	0.14	0.12
	G-B		29	35	29	36	28	37	28	37
	A		28	33	27	35	27	36	26	35
	W		28	32	26	33	27	33	25	33
F4	F-B		0.00009	0	0.00004	0	0	0	0	0
	A		0.759	0.757	0.752	0.749	0.743	0.737	0.731	0.725
	W		1.73	1.68	1.56	1.51	1.48	1.4	1.42	1.39
	G-B		34	41	35	44	37	45	39	45
	A		33	40	33	41	36	43	37	44
	W		31	40	32	40	35	41	34	42
F5	F-B		0.000489	0	0.000353	0	0.000294	0	0	0
	A		0.791	0.58	0.762	0.51	0.737	0.45	0.712	0.37
	W		3.134	2.391	3.072	2.226	2.848	2.128	2.779	2.004
	G-B		31	38	33	39	34	41	35	43
	A		30	35	32	37	33	38	33	42
	W		28	32	30	34	32	36	32	40

Table 4 continued

Selection → Mutation → Crossover →	Probabilistic Boltzmann function		RPBS		Probabilistic Boltzmann function		RPBS	
	Gene-based adaptive mutation		Gene-based adaptive mutation		Improved mutation		Improved mutation	
	Dominance	Co-dominance	Dominance	Co-dominance	Dominance	Co-dominance	Dominance	Co-dominance
F6	F-B	0	0	0	0	0	0	0
	A	0	0	0	0	0	0	0
	W	0	0	0	0	0	0	0
	G-B	10	13	14	11	15	11	16
	A	10	14	14	11	15	11	16
	W	10	15	14	11	16	11	17
	F-B	8	8	8	8	8	8	8
	A	8	8	8	8	8	8	8
	W	8	8	8	8	8	8	8
	G-B	10	13	13	11	14	12	14
F7	A	10	13	13	11	14	12	14
	W	10	13	13	12	14	13	15
	F-B	0	0	0	0	0	0	0
	A	0.00011	0.00009	0.00008	0.00008	0.00006	0.00007	0.00005
	W	0.00023	0.00019	0.000178	0.00018	0.000146	0.00015	0.000129
	G-B	24	32	35	30	36	31	38
	A	21	31	34	28	33	29	37
	W	19	29	31	27	31	28	35
	F-B	0	0	0	0	0	0	0
	A	1.241	1.115	1.003	1.127	0.079	1.074	0.051
F8	W	3.147	1.984	1.539	2.116	1.517	1.674	1.381
	G-B	36	42	44	42	45	43	46
	A	35	41	42	40	44	42	45
	W	33	40	40	38	42	40	42
	F-B	0.00035	0	0	0.0086	0	0.0075	0
	A	0.879	0.431	0.373	0.685	0.301	0.653	0.293
	W	3.62	2.76	2.59	3.11	2.36	2.97	2.159
	G-B	22	30	32	27	35	28	37
	A	21	28	31	24	31	27	35
	W	20	25	28	23	28	25	34
F9	F-B	0	0	0	0	0	0	0
	A	1.241	1.115	1.003	1.127	0.079	1.074	0.051
	W	3.147	1.984	1.539	2.116	1.517	1.674	1.381
	G-B	36	42	44	42	45	43	46
	A	35	41	42	40	44	42	45
	W	33	40	40	38	42	40	42
	F-B	0.00035	0	0	0.0086	0	0.0075	0
	A	0.879	0.431	0.373	0.685	0.301	0.653	0.293
	W	3.62	2.76	2.59	3.11	2.36	2.97	2.159
	G-B	22	30	32	27	35	28	37
F10	A	21	28	31	24	31	27	35
	W	20	25	28	23	28	25	34
	F-B	0	0	0	0	0	0	0
	A	1.241	1.115	1.003	1.127	0.079	1.074	0.051
	W	3.147	1.984	1.539	2.116	1.517	1.674	1.381
	G-B	36	42	44	42	45	43	46
	A	35	41	42	40	44	42	45
	W	33	40	40	38	42	40	42
	F-B	0.00035	0	0	0.0086	0	0.0075	0
	A	0.879	0.431	0.373	0.685	0.301	0.653	0.293

Table 5 Analysis of variance for test functions

Source	DF	Sum of square	Mean square	F	Pr > F
Analysis of variance (Y1)					
Model	11	2615.365	237.760	2.090	0.030
Error	84	9555.875	113.760		
Corrected	95	12171.240			
Computed against model $Y = \text{Mean}(Y)$					

took to obtain the global optima in the absence of mutation. Moreover, on analyzing different runs of different functions, it was clear that though the initial population did not contain all possible alleles for all the genes, global optimum was obtained due to the effect of mutation. On comparing the performance of selection operators, it is found that the RPBS selection gives better results when compared to the probabilistic Boltzmann function used as a selection operator. This is due to the fact that RPBS selects parents that are closer in proximity, thereby helping the offspring to get dominant alleles of different genes in an area of the search space quickly than when the probabilistic Boltzmann function was used as a selection operator.

Statistical analysis of the newly proposed crossover operators on the above test functions was carried out using ANOVA (<http://www.uwsp.edu/psych/stat/12/anova-1w.ht>). Ten runs of the GA using each of the twelve forms of crossover operators (RPBC, adapted, dominance without mutation (DOM without mutation), co-dominance without mutation (CODOM without mutation), probabilistic Boltzmann function (PBF) + gene-based adaptive mutation (GBA) + DOM, PBF + GBA + CODOM, rank and proximity-based selection (RPBS) + GBA + DOM, RPBS + GBA + CODOM, PBF + improved mutation (IM) + DOM, PBF + IM + CODOM, RPBS + IM + DOM and RPBS + IM + CODOM) are executed. The best-case fitness and number of generations to converge to a solution are obtained.

The purpose of this test is to find out the similarity and differences if any in the crossover operators considered in terms of the number of generations taken to obtain the optimal solution. We consider the number of generations for the statistical tests instead of the fitness values since only best-case values are considered for the statistical tests and most crossover operators considered here produced an optimal solution in the best case. Hence, it is only the matter of the number of generations required to obtain the optimal solution as this is the most important aspect of the crossover operators considered in this work since obtaining an optimal solution in reduced number of generations is the main motive behind this work. So our null hypothesis is 'There is no significant difference in the number of generations taken by the twelve classes of crossover operators mentioned earlier.' The probability corresponding to the F value ($\text{Pr} > F$) is 0.030 as given in Table 5 shows that the null hypothesis is rejected. There-

Table 6 Summary of all pairwise comparisons (Tukey's (HSD))

Category	LS means (Y1)	Groups	
RPBC	37.125	A	
RPBS + IM + CODOM	34.500	A	B
PBF + IM + CODOM	33.500	A	B
RPBS + GBA + CODOM	32.125	A	B
Adapted	30.750	A	B
PBF + GBA + CODOM	30.500	A	B
RPBS + IM + DOM	28.375	A	B
PBF + IM + DOM	27.500	A	B
RPBS + GBA + DOM	26.000	A	B
PBF + GBA + DOM	24.500	A	B
CODOM without mutation	23.125	A	B
DOM without mutation	17.625		B

fore, there is significance in the results. The Tukey's HSD table obtained for this test (shown in Table 6) also confirms the presence of significance between the performances of the different crossover classes considered. From the table, it is clear that RPBC (which took the highest number of generations to obtain the optimal solution) and dominance without mutation (DOM without mutation—which took the least number of generations to obtain the optimal solution) fall in distinct groups, whereas the other groups of crossover fall between these two classes of crossover.

Furthermore, the Dunnett analysis (analysis of the differences between the control category DOM without mutation) of the differences between the different classes of crossover and DOM without mutation crossover shown in Table 7 also confirms that there is a very high significance between the number of generations required by DOM without mutation crossover and RPBC. This table also shows that RPBS+IM+CODOM and PBF+IM+CODOM groups of crossover also join the RPBC in showing significant difference with DOM without mutation crossover operator. The prolonged time taken by these two crossover operators can be attributed to the combined effect of the co-dominance and mutation (each separately increases the time to find the optimal solution).

Important points that we observed from our evaluation on test functions are as follows:

Table 7 Q1/Dunnett (two-sided)/analysis of the differences between the control category Q1-DOM without mutation and the other categories with a confidence interval of 95%:

Contrast	Difference	Standardized difference	Critical value	Critical difference	Pr > Diff	Significant
DOM without mutation versus RPBC	−19.500	−3.657	2.805	14.958	0.004	Yes
DOM without mutation versus RPBS + IM + CODOM	−16.875	−3.164	2.805	14.958	0.019	Yes
DOM without mutation versus PBF + IM + CODOM	−15.875	−2.977	2.805	14.958	0.032	Yes
DOM without mutation versus RPBS + GBA + CODOM	−14.500	−2.719	2.805	14.958	0.062	No
DOM without mutation versus adapted	−13.125	−2.461	2.805	14.958	0.115	No
DOM without mutation versus PBF + GBA + CODOM	−12.875	−2.414	2.805	14.958	0.127	No
DOM without mutation versus RPBS + IM + DOM	−10.750	−2.016	2.805	14.958	0.284	No
DOM without mutation versus PBF + IM + DOM	−9.875	−1.852	2.805	14.958	0.378	No
DOM without mutation versus RPBS + GBA + DOM	−8.375	−1.570	2.805	14.958	0.574	No
DOM without mutation versus PBF + GBA + DOM	−6.875	−1.289	2.805	14.958	0.781	No
DOM without mutation versus CODOM without mutation	−5.500	−1.031	2.805	14.958	0.925	No

Bold represent that the difference in the number of generations taken to converge to the solution is very high in the three highlighted cases

- Dominance crossover operator takes the least number of generations to converge of all crossover operators evaluated.
- However, co-dominance crossover operator shows better performance when compared to the dominance crossover operator in some cases especially in the presence of multiple global optima.
- The performances of all the crossover operators show a similar pattern for all the functions considered in the evaluation.
- The minimum, maximum and the average fitness values and the number of generations to converge to the solution using our crossovers are almost the same on different runs.
- However, the proposed crossover operators perform less than expected when
 - a. Real-valued chromosomes are used to represent the solution since the range of real values is vast, and hence, it is difficult to find the exact global optimum gene values.
 - b. When the initial population does not contain all possible gene values—in particular the optimum gene values in which case there is the possibility of getting only the sub-optimal solution. The absence of mutation makes it impossible to reach the optimal solution when the optimal gene value is not present in the initial population. The same reason is applicable for the difficulty in finding the global optimum in the case of using real chromosomes.
- The proposed crossover operators are also suitable for partially separable problems with dependent genes and non-separable problems.
- Our crossover operators converge when there is only one global optimum.
- Mutation increases the number of generations to find the global optima.
- Dominance crossover could find the global optimum in the presence of the improved mutation operator but in more number of generations than the co-dominance crossover operator without mutation.
- Convergence does not occur when there is more than one global optimum using our proposed crossover operators which is also the same case with the other crossover operators considered for evaluation.
- However, co-dominance crossover operator finds all the global optima in less than 22 generations when all the dominant alleles required are present in the initial population and in the absence of mutation.

We proceed with the discussion of evaluation of our proposed crossover operators on test applications on different aspects such as the effect of crossover in the presence and absence of mutation and in the multi-parent scenario in the following section.

4.2 Evaluation on test applications

This section shows the evaluation of the proposed crossover operators on test applications. We have evaluated the performance of the dominance and the co-dominance crossover operators in terms of the quality of the converged solution and the number of generations required to converge, highlighting that faster convergence is achieved without compromising the quality of the solution. We have used the Reuters-21578 (<http://www.daviddlewis.com/resources/testcollections/reuters21578>) dataset for the clustering prob-

lem and LETOR bench marked dataset (Liu et al. 2007) for the learning to rank problem for IR to evaluate our new crossover operators. The two applications showcase the performance of our new crossover operators for entirely different problems, namely clustering and learning to rank. The Reuters-21578 dataset is integer-encoded and has finite number of alleles for the genes. So it converges to the global optimum. The LETOR dataset is real-encoded and the range of values that a gene can hold is large, and hence, the performance is less when compared to that of the Reuters-21578. However, the solution obtained is the best possible solution for the given initial population which is evident from the fact that the co-dominance and the dominance operators perform better than the baselines considered. Hence, we chose to use these applications to evaluate the proposed crossover operators. The applications also show that the operators are effective for different encoding schemes, namely integer encoding for clustering and real-valued encoding for learning to rank. The baselines and other parameters for both the problems are given in Table 8 and are further discussed in the forthcoming sections. We have carried out statistical significance tests with ANOVA and proved the performance of our crossover operators. Furthermore, we have discussed the performance of our crossover operators in the presence and absence of mutation in order to study the effect of diversity of the population on the quality and speed of convergence. Multi-parent crossover operators are generally been designed to achieve faster convergence. Hence, we show the cumulative effect of using multi-parents with our newly designed crossover operators on the quality of performance and speed of convergence.

Reuters-21578 dataset

The Reuters-21578 text categorization test collection is a resource for research in information retrieval, machine learning and other corpus-based research methods. Each document is represented as a vector where a feature is a word in the document and its value is the frequency count of the word. We have carried out clustering on this dataset using our new crossover operators and evaluated them against the baseline crossover operators, namely the one-point, two-point and the uniform crossover operators, respectively. The reason for choosing the basic crossover operators as baselines here is because there are no popular integer-encoded crossover operators (permutation—a special case of integer encoding has many crossover operators designed specifically for it). However, we have also considered two other baselines, namely the RPBC (Chakraborty and Hoshi 1999) and the adapted (Durand and Alliot 1998) crossover operators as baselines since they are similar to our work, i.e., crossover operators aiming for faster convergence with better fitness values. The fitness function we have used for this method is average silhouette coefficient (Al-Zoubi and Rawi

2008). The description of the other parameters is given in Table 8.

Evaluation parameters such as cluster purity, F -measure, entropy, Precision and Recall, are often used for evaluating the performance of clusters. However, we use the silhouette coefficient since the other evaluation parameters require the dataset to provide details on the clusters that each data point belongs to. The silhouette coefficient works well even without these details.

The silhouette coefficient of a data point i is calculated as follows:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (7)$$

where $s(i)$ is the silhouette coefficient; $a(i)$ is the average dissimilarity of the data point i with all other data points in the same cluster; $b(i)$ is the average dissimilarity of the data point i with the closely associated cluster.

The silhouette coefficient of a cluster can be calculated by finding the mean of all the data points in the cluster. The silhouette coefficient can take the following values:

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)}, & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1, & \text{if } a(i) > b(i) \end{cases} \quad (8)$$

This definition suggests that the range of values that the silhouette coefficient can take is between -1 and 1 . A value closer to one is considered better for the clusters formed.

LETOR dataset

Learning to rank for information retrieval deals with the construction of a ranking model, e.g., a ranking function, that achieves the best result on test data in the sense of optimization of a performance measure (e.g., $P@10$, NDCG, error rate, classification accuracy and mean average precision) taking a collection of queries, corresponding retrieved documents, and their relevance judgments provided by human annotators as input. We have used a benchmarked dataset called the LETOR dataset, specifically designed for the Learning to rank problem in IR which is discussed in the next section. The LETOR dataset (Liu et al. 2007), contains a 134-length vector of feature representations for the documents corresponding to the queries for which they were retrieved, their relevance judgments for the respective queries that fetched them. The dataset has three partitions one for training, another for validation and the last for test. It has been released at the website of Microsoft Research. The important parameters for this evaluation are given in Table 8. Since the representation is real encoding (continuous), we discretize the alleles by dividing the weights into 10 slots.

We have evaluated the new crossover operators against baselines such as the fusion operator (Beasley and Chu 1996)

Table 8 Description of GA parameters

Parameter	Description	
Initial population sizes	25, 50, 75, 100	
Maximum number of generations	50	
Termination condition	Maximum number of generations reached or convergence reached	
Applications	Clustering	Learning to rank
Dataset used	Reuters-21578—text categorization test collection (http://www.daviddlewis.com/resources/testcollections/reuters21578)	LETOR dataset (Liu et al. 2007) contains a 134-length vector of feature representations for the documents corresponding to the queries for which they were retrieved, their relevance judgments for the respective queries that fetched them. The dataset has three partitions one for training, another for validation and the last for test
Baselines used	One-point, two-point, uniform, RPBC, adapted crossover operators	Fusion, heuristic, dissociated, RPBC, adapted crossover operators
Number of features	Different for different documents	134
Encoding	Integer count of the word frequency	Real-random weighted vector
Fitness function	Average silhouette coefficient (Al-Zoubi and Rawi 2008)—This coefficient has been used for the evaluation of clustering because other evaluation parameters require the dataset to provide details on the clusters that each data point belongs to while this is not required for average silhouette coefficient	P@5

and the heuristic operator (Wright 1991) since similar to our crossover operator they also produce only one offspring as a result of the crossover operation which is also heuristic in nature. The other baseline considered is the dissociated crossover operator (Vrajitoru 1998) which finds its way to our list of baselines since the application it was designed for is learning to rank for information retrieval (LETOR) (Liu et al. 2007), which is a real-coded application which we have used to evaluate our new crossover operators.

4.2.1 Performance of the dominance and co-dominance crossover operators

Tables 9 and 10 show the evaluation of the effect of crossover operators for two different applications, clustering (Reuters-21578 dataset) and learning to rank (LETOR dataset). We use average silhouette coefficient for the evaluation of clustering while we use the P@5 for the learning to rank problem. The baselines used for both cases can be classified as general crossover operators and crossover operators (RPBC and adapted crossover) designed for faster convergence. In the case of clustering the general crossovers used are the classic

operators such as one-point, two-point and uniform crossover operators since dataset used for clustering is integer-encoded and these operators have been adapted for integer encoding. In the case of learning to rank we have used fusion (Beasley and Chu 1996) and heuristic crossover (Wright 1991) operators which are based on heuristics (similar to our crossover operators) while dissociated crossover operator (Vrajitoru 1998) is an operator specifically designed for the learning to rank problem. We have carried out the evaluations in both cases with four different sets of different sizes of initial population of 25, 50, 75, 100.

Table 9 shows the average silhouette coefficient value for the baselines used and the newly designed crossover operators while Table 10 shows the P@5 values of the baselines used and the same newly designed crossover operators. We find that the effect of population sizes is similar for both applications. In both applications for all crossover operators, the performance is lower for an initial population size of 25, but the effect of initial population size on performance for sizes of 50 or higher is not substantial.

Tables 9 and 10 show that though RPBC and the adapted crossover operators perform less than our two new proposed

Table 9 Fitness and number of generations to converge (in 10 runs) for Reuters-21578 dataset

	IP = 25		IP = 50		IP = 75		IP = 100	
	Fitness	Generations	Fitness	Generations	Fitness	Generations	Fitness	Generations
One-point	0.41	21	0.63	38	0.55	36	0.59	34
Two-point	0.43	26	0.68	41	0.59	38	0.62	42
Uniform	0.55	30	0.81	45	0.7	42	0.73	48
RPBC	0.62	18	0.82	22	0.84	21	0.87	25
Adapted	0.65	15	0.85	19	0.86	16	0.89	20
Dominance	0.73	8	0.92	9	0.91	8	0.93	10
Co-dominance	0.77	9	0.93	10	0.94	9	0.97	10

Table 10 Fitness and number of generations to converge (in 10 runs) for LETOR dataset

	IP = 25		IP = 50		IP = 75		IP = 100	
	Fitness	Generations	Fitness	Generations	Fitness	Generations	Fitness	Generations
Fusion	0.13	32	0.21	30	0.22	33	0.25	37
Heuristic	0.19	34	0.28	42	0.29	44	0.28	47
Dissociated	0.25	22	0.35	26	0.34	27	0.36	40
RPBC	0.22	18	0.34	22	0.37	22	0.39	24
Adapted	0.27	15	0.38	19	0.42	20	0.45	18
Dominance	0.35	10	0.51	11	0.53	13	0.55	13
Co-dominance	0.36	10	0.55	12	0.55	15	0.59	14

crossover operators, they do perform reasonably better than the other baselines considered since these two baselines were designed for obtaining better solution despite faster convergence. However, these graphs show that the dominance and co-dominance crossover operators which we have designed for faster convergence perform even better than the RPBC and the adapted crossover operators irrespective of the application. Further it is observed that the co-dominance crossover operator performs marginally better than dominance crossover operator in most cases. This happens when the co-dominant alleles are predominant in the initial population. As we can see, the best solution obtained using the co-dominance crossover operator has a fitness value of 0.97 for the Reuters-21578 dataset, whereas it is only 0.59 for the LETOR dataset. The reason for this is that since Reuters-21578 dataset is integer-encoded and almost all alleles are present in the initial population. However, LETOR is real-coded and since we have divided the allele values into slots, the exact allele value of the optimal solution is may not be available. However, for the given initial population, the new crossover operators milk the best solutions. In the next section we show that the better performance with faster convergence is achieved using our new crossover operators.

4.2.2 Number of generations to converge to the solution

Tables 9 and 10 also show the number of generations taken to converge for all the baselines and the new crossover operators on the Reuters-21578 dataset and the LETOR datasets, respectively. The graphs bring out the fact that in both applications when the population size is 25, all the crossover operators invariably converge quicker to the solution when compared to population sizes greater than or equal to 50. However, the faster convergence for initial populations of 50 or higher does not affect the performance.

It can be seen from the tables that RPBC and the adapted crossover are better in convergence when compared to other baselines since basically they have been designed for faster convergence. RPBC considers two better fitness individuals that are closer in proximity as parents for the crossover operation rather than randomly choosing the parents, thereby aiming at faster convergence. However, there is a good chance of the closer proximity individuals to have a repetition of a common set of genes, and hence, the variations in genes are dealt with in the later part of the GA, thereby increasing the number of generations taken to converge to the solution. On the other hand, the adapted crossover operator shows a performance that is similar to the RPBC operator and follows a similar pattern in the number of generations to converge to a solu-

Table 11 Minimum, maximum and average values for fitness and number of generations to converge (in 10 runs)

Crossover	Dataset	Parameter	Minimum	Maximum	Average
Dominance	Reuters-21578	Fitness	0.91	0.96	0.945
		Number of generations	8	10	9
	LETOR	Fitness	0.51	0.56	0.53.5
		Number of generations	11	13	12
Co-dominance	Reuters-21578	Fitness	0.93	0.98	0.955
		Number of generations	9	11	10
	LETOR	Fitness	0.55	0.59	0.57
		Number of generations	12	15	13.5

Table 12 Analysis of variance for Reuters-21578

Source	DF	Sum of square	Mean square	F	Pr > F
Analysis of variance					
Model	2	0.130	0.065	1.636	0.214
Error	26	1.035	0.040		
Corrected	28	1.165			

tion. The variation operator biases the allele selection in the offspring formation process to achieve faster convergence.

The graphs show that dominance and co-dominance crossover operators perform significantly much better than all the baselines considered including RPBC and adaptive crossover operators. The improvement is so significant that this can be claimed as the strength of our work, i.e., an improved performance with an even more improved speed of convergence. The reason for such a remarkable improvement in speed of convergence is due to the presence of the dominance effect in the crossover part which is responsible for increasing the dominant genes and decreasing the number of recessive genes in the subsequent generations as given in Eq. 3. The dominance and the co-dominance crossover operators bias the selection of alleles based on its contribution to the fitness of the entire chromosome without considering proximity of the chromosomes (as in RPBC) or the variation in the parents of the crossover (as in adaptive crossover).

The dominance crossover operator converges comparatively quicker or at the same generation as that of the co-dominance crossover operator. The reason for the smaller increase in the number of generations with the co-dominance crossover operator is because of the narrow extension of the search space by creating two chromosomes when co-dominance exists. This happens because we do not eliminate the good allele just because the other allele considered is good as well. We widen the search space without losing the good genes in the case of co-dominance crossover operator. As we had already found out in our evaluation, since an initial population size ≥ 50 shows very little difference in the results obtained, for further evaluation we consider an initial population size of 50.

4.2.3 Statistical tests

Statistical analysis of the newly proposed crossover operators was carried out. Ten runs of the GA using each of the new crossover operators are executed. The minimum, maximum and the average fitness and number of generations to converge to a solution are obtained as given in Table 11. As we can see from the table, there is no significant variation in the fitness values and the number of generations to converge. This proves that the newly proposed crossover operators behave almost the same way for different runs of the GA. To further add value to this, we have done the statistical significance test using ANOVA (<http://www.uwsp.edu/psych/stat/12/anova-1w.ht>) on both the datasets and the results are discussed in the forthcoming subsections.

Statistical significance test using ANOVA—Reuters-21578

Our next evaluation is the statistical significance test using one-way ANOVA (<http://www.uwsp.edu/psych/stat/12/anova-1w.ht>). We compared the performance of three samples of the Reuters-21578 dataset for statistical evaluation purpose. The purpose of this test is to find out if the co-dominance crossover operator performs the same way irrespective of the sample dataset used. So our Null hypothesis is ‘There is no significant difference in the average silhouette coefficient values of all the three samples.’ Here we directly test the ANOVA significance for co-dominance crossover and not the dominance crossover since the performance of the two operators in spite of being the same in most cases is better for the former in some cases.

The expected result is that there is no difference in the performance of all the three samples. The probability corresponding to the F value ($\text{Pr} > F$) is 0.214 as given in Table 12

Table 13 Sample3/Tukey's (HSD)/analysis of the differences between the categories with a confidence interval of 95%

Contrast	Difference	Standardized difference	Critical value	Pr > Diff	Significant
Sample3 versus Sample2	0.151	1.646	2.485	0.245	No
Sample3 versus Sample1	0.138	1.504	2.485	0.305	No
Sample1 versus Sample2	0.013	0.146	2.485	0.988	No
Tukey's d critical value:			3.515		

Table 14 Analysis of variance for LETOR

Analysis of variance:					
Source	DF	Sum of square	Mean square	F	Pr > F
Model	2	0.050	0.025	0.485	0.621
Error	26	1.336	0.051		
Corrected	28	1.386			

shows that the null hypothesis is accepted. Therefore, there is no significance in the results. The Tukey's HSD table given in Table 13 also confirms the absence of significance between the performances on the different samples considered.

Statistical significance test using ANOVA-LETOR

We compared the performance of three samples of the LETOR dataset for statistical evaluation purpose. The purpose of this test is to find out if the co-dominance crossover operator performs the same way irrespective of the sample dataset used. So our null hypothesis here also is 'There is no significant difference in the P@5 values of all the three samples.' Here also we directly test the ANOVA significance for co-dominance crossover and not the dominance crossover for the same reason as given in Sect. 4.2.1.

The expected result is that there is no difference in the performance of all the three samples. The probability corresponding to the F value ($\text{Pr} > F$) is 0.621 as given in Table 14 says that the null hypothesis is accepted. Therefore, there is no significance in the results. The Tukey's HSD table given in Table 15 also confirms the absence of significance between the performances on the different samples considered.

4.2.4 Multi-parent version of crossover operators and convergence

As we have already mentioned in Sect. 2, the multi-parent crossover operators are inherently designed to converge faster to a solution. Hence, we have implemented the multi-parent version of both the dominance and co-dominance crossover operators. We have carried out the evaluation of the two-parent, three-parent, four-parent and five-parent versions of the dominance crossover operator on the Reuters-21578 and the LETOR datasets. The process of dominance crossover is the same as the normal two-parent version dis-

cussed earlier. The other versions have the respective number of parents of which the dominant gene is carried over to the offspring.

As shown in Figs. 5 and 6, in general as the number of parents increases there is uniform increase in speed of convergence. In the case of the two-parent and three-parent versions of the dominance crossover operator, the convergence is comparatively quite slow during the initial phase and faster during the later phase of evolution. However, for the four-parent and five-parent versions convergence is faster during the initial phase and slow during the later phase. This behavior is due to the fact that the dominant genes from the two- and three-parent dominance crossover operators are less in number in the offspring that was created during crossover when compared to the four-parent and the five-parent dominance crossover. In the case of four-parent and the five-parent dominance crossover, most of the dominance alleles have already been converged, the choice of alleles is less during the later stages, and the convergence is almost near completion at the later stages.

4.2.5 Dominance and co-dominance Operators in the presence of mutation

The very argument of the inability of the crossover operators to explore into new areas of the search space without mutation motivated us to evaluate the new crossover operators in the absence and presence of mutation. The mutation operator that we have considered is the improved mutation operator that is normally used for faster convergence in spite of introducing diversity (Yusof et al. 2012).

As shown in Fig. 7, the presence or absence of mutation with the new crossover operators brings about no change in the fitness value of the converged solution for the Reuters-21578 dataset. The presence of mutation only results in delaying the convergence process to a small extent. The presence of mutation allows for the exploration of search areas that are not covered by the initial population while this is not necessary in cases where all the dominant alleles are accounted for in the initial population itself and mutation only results in unnecessary exploration. However, this effect is not propagated to subsequent generations due to dominance principle, and hence, there is no delay in convergence.

Table 15 Sample3/Tukey's (HSD)/analysis of the differences between the categories with a confidence interval of 95%

Contrast	Difference	Standardized difference	Critical value	Pr > Diff	Significant
Sample3 versus sample2	0.101	0.969	2.485	0.603	No
Sample3 versus sample1	0.069	0.661	2.485	0.788	No
Sample1 versus sample2	0.032	0.316	2.485	0.947	No
Tukey's d critical value:			3.515		

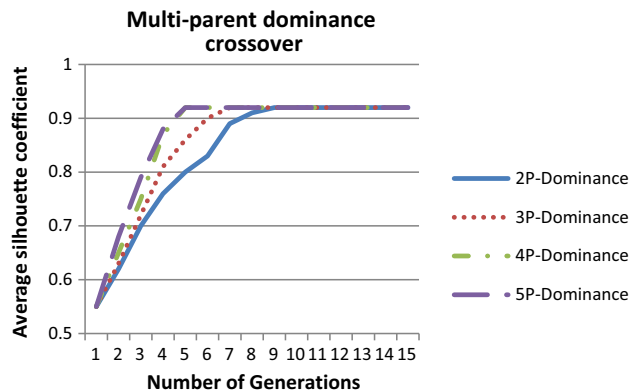
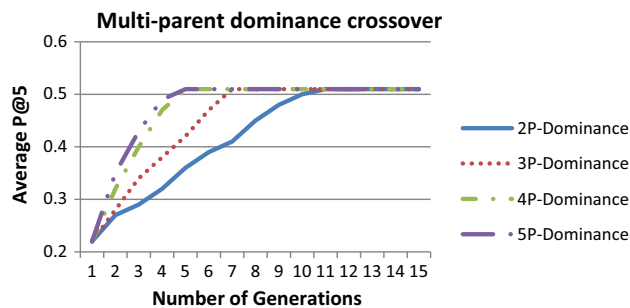
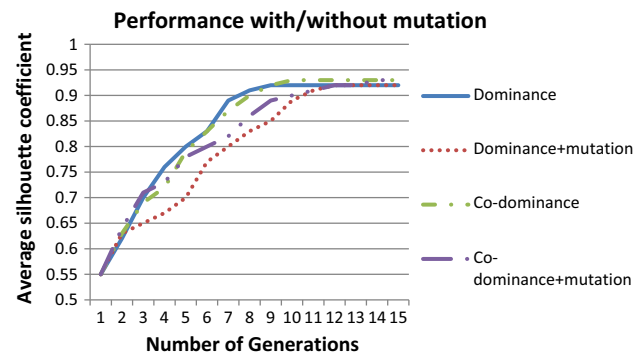
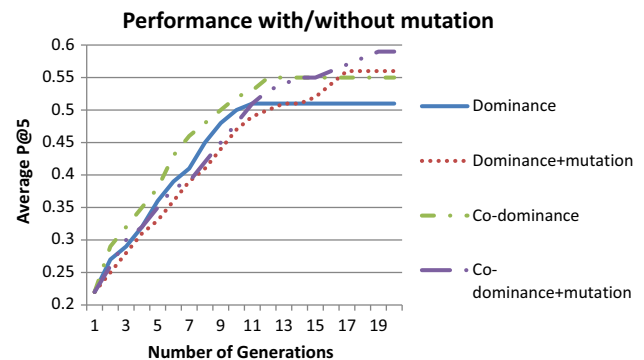
**Fig. 5** Performance of the multi-parent version of dominance crossover operator—Reuters-21578 (initial population = 50)**Fig. 6** Performance of the multi-parent version of dominance crossover operator—LETOR (initial population = 50)

Figure 8 shows the effect of the new crossover operators with and without mutation on the LETOR dataset. In this case, although mutation delays the convergence it results in improving the fitness of the solution obtained. The presence of mutation gives better solutions in LETOR dataset due to the fact that the initial population does not contain all possible alleles for all the genes as we had discussed earlier and mutation is necessary for exploring the search space to find missed dominant alleles.

Important points that we observed from our evaluation on test applications are as follows:

- When the population size is small, the crossover operators help converge to a solution quickly due to the narrow search space.

**Fig. 7** Performance of the proposed crossover operators with and without mutation—Reuters-21578 (initial population = 50)**Fig. 8** Performance of the proposed crossover operators with and without mutation—LETOR (initial population = 50)

- This adversely affects the performance, i.e., the performance in this case is comparatively less.
- The performance of all the crossover operators considered here is more or less the same when the initial population size ≥ 50 .
- Dominance crossover operator takes the least number of generations to converge time of all crossover operators evaluated.
- However, co-dominance crossover operator shows better performance when compared to the dominance crossover operator in some cases.
- The performances of all the crossover operators are similar for both the datasets considered in the evaluation.
- The minimum, maximum and the average fitness values and the number of generations to converge to the solution

using our crossovers are almost the same on different runs.

- Statistical significance tests using one-way ANOVA prove this.
- Multiple-parent version helps in even faster convergence than its two-parent counterpart.
- The higher the number of parents, the faster is the convergence process.
- There is no drastic change in convergence or performance beyond four-parent crossover.
- Mutation delays the convergence process.
- The solution obtained when mutation is used along with crossover is better than when mutation is not used in cases where the initial population does not hold all possible alleles for all the genes.
- Mutation does not bring about any change in the fitness of the converged solution when the initial population holds all possible alleles for all the genes.

The next section presents the concluding remarks of our work.

5 Conclusions, implications and limitations of the study

This paper describes two new crossover operators based on the dominance and co-dominance principle existing in nature, and the results show that the new crossover operators perform well against the classical baseline crossover operators and the baselines designed for faster convergence. The dominance crossover operator converges faster. However, the co-dominance crossover operator gives the best solution.

Our new crossover operators work well for the problems where each gene's fitness contribution is independent of the others, i.e., separable problems. They also work well when there is dependency between the genes especially partially separable problems when compared to the baselines. Our crossover operators also work well on non-separable problems where the number of genes is less and the solution space is comparatively smaller.

The multi-parent version of the proposed crossover operators seems to be promising with respect to the number of generations to converge to a solution. This provides ample scope for future work as we would like to further explore on this area by designing new multi-parent crossover operators for dependent as well as independent genes in the chromosomes.

We would also like to bring out the effect of mutation in the crossover operator itself so that the extra time consumed for mutation is avoided without any compromise on the quality of the results that mutation provides. Moreover, we would like to explore the chances of mutation operator in finding the dominant genes that are not present in the initial population which

is the reason behind some of the runs not converging to global optimum using our newly proposed crossover operators.

The highlight of our method is the reduced run-time due to the reduction in the number of generations. However, this includes only the run-time of the GA after the dominance vector and the co-dominance hash table are determined. We have not included the time to create the dominance vector and the co-dominance hash table in the run-time evaluation. This accounts for a considerable amount of time especially in case of partially separable and non-separable problems which we want to decrease by proposing other bio-inspired crossover techniques that could combine both the tasks of finding dominance and the crossover as a single task as future work.

Acknowledgements This study was funded by Anna University, Chennai, Tamil Nadu, India (Anna Centenary Research Fellowship).

Compliance with ethical standards

Conflict of interest G. Pavai has received research grants from Anna University, Chennai, Tamil Nadu, India and works with SRM University, Chennai, Tamil Nadu, India. T. V. Geetha works with Anna University, Chennai, Tamil Nadu, India.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

- Adra SF, Griffin IA, Fleming PJ (2009) A convergence acceleration operator for multiobjective optimization. *IEEE Trans Evol Comput* 13:825–847
- Al-Zoubi M, Rawi M (2008) An efficient approach for computing silhouette coefficients. *J Comput Sci* 4:252–255
- Beasley JE, Chu PC (1996) A genetic algorithm for the set covering problem. *Eur J Oper Res* 94:392–404
- Brady RM (1985) Optimization strategies gleaned from biological evolution. *Nature* 317:804–806. <https://doi.org/10.1038/317804a0>
- Chakraborty G, Hoshi K (1999) Rank based crossover—a new crossover technique to improve convergence in genetic algorithms. In: *Proceedings of the congress on evolutionary computation (CEC 1999)*, vol 2. IEEE, Washington DC, US, pp 1595–1602
- Chang PC, Wang YW, Liu CH (1995) New operators for faster convergence and better solution quality in modified genetic algorithm. *Advances in Natural Computation, Lecture Notes in Computer Science (LNCS 1995)*, vol 3611. Springer, Changsha, China, pp 983–991
- Deep K, Singh KP, Kansal ML, Mohan C (2009) A real coded genetic algorithm for solving integer and mixed integer optimization problems. *Appl Math Comput* 21:505–518
- De Jong KA, Spears WM (1992) A formal analysis of the role of multi-point crossover in genetic algorithms. *Ann Math Artif Intell* 5:1–26
- Demirci H, Ozcerit AT, Ekiz H, Kutlu A (2015) Chaotic crossover operator on genetic algorithm. In: *Proceedings of second international conference on information technology*, Singapore, 2015
- Deshpande AS, Kelkar RB (2008) Advanced genetic operators and techniques—an analysis of dominance and diploidy, reordering

- operator in genetic search. In: Proceedings of the ninth WSEAS international conference on evolutionary computing, Sofia, Bulgaria, pp 27–33
- Durand N, Alliot JM (1998) Genetic crossover operator for partially separable functions. In: Proceedings of 3rd annual conference on genetic programming, Morgan Kaufmann, Madison, United States, pp 487–494
- Felsenstein J (2015) Theoretical evolutionary genetics. Department of Genome Sciences and Department of Biology, University of Washington Box 355065, Seattle, Washington, pp 98195–5065
- Gardner EJ, Simmons MJ, Snustad DP (2006) Principles of genetics, 8th edn, Wiley. ISBN: 8126510439, 9788126510436
- Goldberg DE, Lingle R (1985) Alleles, Loci and the TSP. In: Grefenstette JJ (ed) Proceedings of the first international conference on genetic algorithms and their applications. Lawrence Erlbaum, Hillsdale, New Jersey, pp. 154–159
- Gwak J, Jeon M, Pedrycz W (2016) Bolstering efficient SSGAs based on an ensemble of probabilistic variable-wise crossover strategies. *Soft Comput* 20:2149–2176
- Herrera F, Lozano M, Verdegay JL (1997) Fuzzy connectives based crossover operators to model genetic algorithms population diversity. *Fuzzy Sets Syst* 92:21–30
- Herrera F, Lozano M, Sanchez AM (2003) A taxonomy for the crossover operator for real-coded genetic algorithms: an experimental study. *Int J Intell Syst* 18:309–338
- Holland JH (1992) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. The MIT Press, Cambridge
- http://en.wikipedia.org/wiki/Mendelian_inheritance
- Jamil M, Yang XS (2013) A literature survey of benchmark functions for global optimization problems. *Int J Math Model Numer Opt* 4:150–194
- Jing J, Lidong M (2012) The strategy of improving convergence of genetic algorithm. *Telkomnika* 10:2063–2068
- Kita H (2001) A comparison study of self-adaptation in evolution strategies and real-coded genetic algorithms. *Evol Comp* 9:223–241
- Kita H, Ono I, Kobayashi S (1999) The multi-parent unimodal normal distribution crossover for real-coded genetic algorithms. In: Proceedings of the 1999 congress on evolutionary computation, IEEE Washington DC US, pp 1588–1595
- Kureichick V, Melikhov AN, Miaghick VV, Savelev OV, Topchy AP (1996) some new features in the genetic solution of the traveling salesman problem. In: Proceedings of 2nd international conference of the integration of genetic algorithms and neural network computing and related adaptive computing with current engineering practice, Plymouth, UK, pp 231–247
- Laird NM, Lange C (2011) The fundamentals of modern statistical genetics, statistics for biology and health. Springer, Berlin. https://doi.org/10.1007/978-1-4419-7338-2_2
- Liepins GE, Vose MD (1992) Characterizing crossover in genetic algorithms. *Ann Math Artif Intell* 5:27–34
- Liu TY, Xu J, Qin T, Xiong W, Li H (2007) LETOR: Benchmarking learning to rank for information retrieval. In: Proceedings of SIGIR 2007 workshop on learning to rank for information retrieval, ACM, Amsterdam, Netherlands, pp 3–10
- Lozano JA, Larranaga P, Grana M, Albizuri FX (1999) Genetic algorithms: bridging the convergence gap. *Theor Comput Sci* 229:11–22
- Lunacek M, Whitley D, Ray I (2006) A crossover operator for the k-anonymity problem. In: Proceedings of genetic and evolutionary computation conference, (GECCO 2006), ACM, Seattle, Washington, USA, pp 1713–1720
- Mc Ginley B, Maher J, O’Riordan C, Morgan F (2011) Maintaining healthy population diversity using adaptive crossover, mutation and selection (ACROMUSE). *IEEE Trans Evol Comput* 15:692–714
- Mitchell GG, O’donoghue D, Trenaman A (2000) A new operator for efficient evolutionary solutions to the travelling salesman problem. In: Proceedings of applied informatics, Innsbruck, Austria, pp. 771–774
- Nicoara ES (2009) Mechanisms to avoid the premature convergence of genetic algorithms. *Pet Gas Univ Ploiesti Bull Math I* 61:87–96
- Oliver IM, Smith DJ, Holland JRC (1987) A study of permutation crossover operators on the TSP. In: Grefenstette JJ (ed) Proceedings of the first international conference on genetic algorithms and their applications, Lawrence Erlbaum, Hillsdale, New Jersey, pp 224–230
- One Way ANOVA—University of Wisconsin—Stevens Point [online]. <http://www.uwsp.edu/psych/stat/12/anova-1w.ht>
- Patel R, Raghuvanshi MM (2011) Multi-objective optimization using multi parent crossover operators. *J Emerg Trends Comput Inf Sci* 2:99–105
- Pawai G, Geetha TV (2016) A survey on crossover operators. *ACM Comput Surv* (accepted for publication—awaiting proof)
- Ramadan SZ (2013) Reducing premature convergence problem in genetic algorithm: application on travel salesman problem. *Comput Inf Sci* 6:410–418
- Rocha M, Neves J (1999) Preventing premature convergence to local optima in genetic algorithms via random offspring generation. In: Proceedings of 12th international conference on industrial and engineering applications of artificial intelligence and expert systems (IEA/AIE 1999), multiple approaches to intelligent systems, Lecture Notes in Computer Science (LNCS 1999), vol 1611, Springer, Cairo, Egypt, pp 127–136
- Roy A, Schaffer JD, Laramie CB (2015) New crossover operators for multiple subset selection tasks. *Comput Commun Collab* 3:42–62
- Sahnehsaraei MA, Mahmoodabadi MJ, Sahnehsaraei SE (2012) Solving of single-objective problems based on a modified multiple-crossover genetic algorithm: test function study. In: Proceedings of international conference on systems, signal processing and electronics engineering, Dubai, UAE, pp 193–198
- Schoene T (2011) Step-optimized particle swarm optimization, M.Sc. thesis, Department of Computer Science, University of Saskatchewan, Saskatoon
- Spears WM, McDonnell JR, Reynolds RG, Fogel DB (1995) Adapting crossover in evolutionary algorithms. In: Proceedings of the 4th annual conference on evolutionary programming, MIT Press, Cambridge, MA, pp 367–384
- The Reuters-21578. <http://www.daviddlewis.com/resources/testcollections/reuters21578>
- Uyar SE, Eryigit GC, Sariel S (2004) An adaptive mutation scheme in genetic algorithms for fastening the convergence to the optimum. In: Proceedings of the 3rd Asia Pacific international symposium on information technologies, pp 461–465
- Vekaria K, Clack C (1998) Selective crossover in genetic algorithms: an empirical study. *Parallel Problem Solving from Nature—PPSN V*, Lecture Notes in Computer Science, vol 1498. Springer, Berlin Heidelberg, pp 438–447
- Vrajitoru D (1998) Crossover improvement for the genetic algorithm in information retrieval. *Inf Process Manag* 34:405–415
- Wolpert DH, Macready WG (1997) No free lunch theorem for optimization. *IEEE Trans Evol Comput* 1:67–82
- Wright AH (1991) Genetic algorithms for real parameter optimization. In: Rawlins GJE (ed) Foundations of genetic algorithms I. Morgan Kaufmann, San Mateo, pp 205–218
- Yuen CC (2004) Selective crossover using gene dominance as an adaptive strategy for genetic programming. M.Sc. thesis. Department of Computer Science, University College London, London WC1E 6BT, UK
- Yusof R, Khairuddin U, Khalid M (2012) A new mutation operator for faster convergence in genetic algorithm feature selection. *Int J Innov Comput Inf Control* 8:7363–7379