Computers and Concrete, *Vol. 11, No. 6 (2013) 541-569*
DOI: http://dx.doi.org/10.12989/cac.2013.11.6.541

Evaluation of the different genetic algorithm parameters and operators for the finite element model updating problem

Yildirim Serhat Erdogan and Pelin Gundes Bakir*

Istanbul Technical University, Department of Civil Engineering, Maslak, 34469, Istanbul, Turkey

(Received October 5, 2010, Revised December 22, 2011, Accepted December 2, 2012)

Abstract. There is a wide variety of existing Genetic Algorithms (GA) operators and parameters in the literature. However, there is no unique technique that shows the best performance for different classes of optimization problems. Hence, the evaluation of these operators and parameters, which influence the effectiveness of the search process, must be carried out on a problem basis. This paper presents a comparison for the influence of GA operators and parameters on the performance of the damage identification problem using the finite element model updating method (FEMU). The damage is defined as reduction in bending rigidity of the finite elements of a reinforced concrete beam. A certain damage scenario is adopted and identified using different GA operators by minimizing the differences between experimental and analytical modal parameters. In this study, different selection, crossover and mutation operators are compared with each other based on the reliability, accuracy and efficiency criteria. The exploration and exploitation capabilities of different operators are evaluated. Also a comparison is carried out for the parallel and sequential GAs with different population sizes and the effect of the multiple use of some crossover operators is investigated. The results show that the roulettewheel selection technique together with real valued encoding gives the best results. It is also apparent that the Non-uniform Mutation as well as Parent Centric Normal Crossover can be confidently used in the damage identification problem. Nevertheless the parallel GAs increases both computation speed and the efficiency of the method.

Keywords: reinforced concrete; finite element model updating; damage detection; genetic algorithms; statistical comparison

1. Introduction

There is a rising interest on vibration based structural damage identification methods within research circles over the last 20 years. These methods use only modal parameters such as natural frequencies, damping ratios and mode shapes which are sensitive to the damage in structural members. Vibration based methods allow the identification of both the severity, location and the extent of the damage. These methods can be classified as model based and non-model based. Non-model based methods that detect changes in the flexibility, damage index methods and continuous wavelet transform, do not require a numerical model. A comparative study on these methods was made by Humar *et al.* (2006). Cruz and Salgado (2009) use non-model based methods to detect damage in composite bridges under different crack depths and to identify the

*Corresponding author, Professor, E-mail: gundesbakir@yahoo.com

http://www.techno-press.org/?journal=cac&subpage=8 ISSN: 1598-8198(Print), 1598-818X (Online)

extent of the damage under different noise levels. Model based methods use a numerical model such as the finite element (FE) model of the structure. The main goal of those methods is to identify the structural changes (damage, deterioration) by comparing the numerical and experimental data such as modal parameters obtained from Structural Health Monitoring (SHM) applications and extracted from the measured time histories by means of system identification techniques (Bakir 2011, 2012, Bakir *et al.* 2011).

Finite Element Model Updating (FEMU) Technique is a model based method and has a wide range of applications especially in damage identification. Damage in structural components results with a reduction in stiffness. In FEMU applications, an objective function is established that includes the differences between the experimental and the numerical vibration data such as the frequencies and mode shapes. Usually this objective function is formulated as a nonlinear least square problem (Friswell and Mottershead 1995). Teughels *et al.* (2002), use a model based method to identify the stiffness reduction factors of a beam by minimizing the differences between the measured modal parameters and the corresponding analytical predictions. Bakir *et al.* (2007) has used the Trust Region Newton Technique to identify, locate and quantify damage in a portal frame. The results have shown that the technique acts like a regularization technique in frame type structures and the damage is successfully identified. An improved version of FEMU which is capable of incorporating the uncertainties arising from the modeling simplifications and the measurement errors are also proposed by Erdogan and Bakir (2013).

Much effort has been spent to minimize this objective function and obtain the stiffness reduction factors defined as reductions in the stiffness of the different finite elements. In most cases an ill-conditioned problem occurs when using gradient-based optimization techniques. Some investigations have been conducted to deal with the ill-conditioned problems in model updating (Li and Law 2010). However, these types of optimization methods do not guarantee to find the global optimum. Thus, further investigations are needed to establish a robust optimization routine in order to deal with problems that local optimization techniques cannot overcome. For this purpose, global optimization methods such as Genetic Algorithms (GAs), Simulating Annealing (SA) and Coupled Local Minimizers (CLM) have been used in the damage identification problems.

Levin and Lieven (1998) have conducted an experimental study and implemented the SA and GA methods to detect the damage at a plate wing structure. Bakir *et al.* (2008) have applied the CLM method to obtain the appropriate values of 24 stiffness reduction factors of a portal concrete frame and compared the method with some local techniques. Perera and Torres (2006) have used GAs to detect the simulated damage of a simply supported beam and verified the results using experimental data. They have used binary coded GAs (BCGA) and single point crossover which is the basic crossover operator of GAs. Roulettewheel selection has been chosen as the selection method. Hao and Xia (2002) have used a laboratory tested cantilever beam and a frame to assess the performance of a real-coded GA (RCGA). They have adopted the rouletwheel selection and uniform crossover as GA operators. Rao *et al.* (2004) have proposed a GA methodology to quantify the damage in structural members using the concept of residual forces. Two-point crossover and BCGA with tournament selection is adopted in minimizing the objective function. Chou and Ghaboussi (2001) have implemented GAs with different string representations called the implicit redundant representation to identify the reduction of the Young modulus and the cross-section areas of a plane truss structure.

In this study, various aspects of GAs have been investigated for the damage detection problem. There are various operators and parameters which affect the GA performance during the

optimization process. Most of the applications of GAs in the damage detection problem have not considered the effects of the GA operators and parameters on the results of FEMU. In these applications, the basic operators such as one point crossover and uniform mutation for RCGA are utilized and parameters are selected intuitively. Hence, a comparative study is necessary to assess the performance of the wide variety of GA parameters and operators. In this study, the performance of the essential GA operators and parameters is investigated and compared for certain criteria.

GAs are robust, accurate and effective search methods. Unfortunately, the convergence speed is too slow especially in the later generations. Thus, there is a need for the parallel implementation of the GA. Some different approaches have been proposed within this context (Goldberg 1989). The so-called multi-population parallel GAs (PGAs) is widely used in the GA literature (Meruane and Heylen 2010). In this present study, a comparison is also made between GAs and PGAs for various population sizes.

2. GA method

GAs was first proposed by Holland (1975) and is based on the theory of the survival of the fittest. It is a global optimization method that is very capable of finding the global optimum of various types of functions which are linear, nonlinear, continuous, discontinuous, convex, non-convex, unimodal and multimodal. Detailed information can be found in Boyer *et al.* (2008). GAs are first initiated with a population that contains a set of potential solutions. Then, GA operators evolve this population until an optimal or near optimal solution is obtained. There are various advantages of GAs in comparison with the gradient-based optimization techniques. GAs use a population and search for optimal solutions at different points in the search space. GAs do not need derivatives of the objective function and can find the global solutions with high probability. GAs can be simply coded and implemented for a variety of optimization problems.

Basic steps of GAs are encoding, selection, crossover and mutation. Design variables are coded as bit-strings by the use of an encoding schema. This type of GAs is referred to as BCGA. Other encoding implementations have also been used, including real-valued encodings (RCGA). After the initial population is randomly or heuristically generated, selection, crossover and mutation operators evolve the population through iterative applications. The selection method is implemented for the reproduction of the population. It chooses the relatively fit individuals according to their fitness values. Crossover can be deemed as a mixing operator. It combines the genes of parent chromosomes mated in the selection processes. In BCGA, this operator does not add new genes into the population; only juxtapose chromosome portions of the parents to create new individuals. The mutation operator is implemented with some small probabilities. In binary encoding, it is carried out by flipping bits at random between 0 and 1. The main purpose of this operator is to generate random diversity in the population by adding new genes.

A frequent problem associated with the GAs is the premature convergence in which highly fit individuals may rapidly dominate the population and cause a rapid convergence to the local optimum. To overcome this difficulty, the operators and the parameters of the GAs should be selected appropriately considering the population diversity-selection pressure balance. Hence each operator should be included in the GAs processes according to their exploration and exploitation capabilities. Another problem is the slow convergence of the GAs especially at further generations. To solve this problem a PGA can be used.

To apply a GA to a given problem, a decision must be taken on the way the parameters of the problem will be mapped into a finite string of symbols. The binary alphabet is often used for the coding purposes. The standard binary and the Gray Coding are the two popular instances. However, these coding methods have some disadvantages. The BCGA needs extra computation time because the variables have to be encoded and decoded. Furthermore, the BCGA has its precision limited by the binary representation of variables. In addition the binary coding schemas bring additional nonlinearities into the objective function. With respect to the BCGA, RCGA has become more popular since RCGA provides some advantages which makes it convenient for various types of optimization problems. Most physical problems have continuous variables in the search domain; hence it is feasible to use continuous real-valued variables. In addition, several GA operators can be constructed using the RCGA.

One of the major processes in GA is the selection. The selection pressure is a degree to which the best individuals are favored. High selection pressure enhances the selection probability of the fittest individuals. The convergence rate of the GAs is substantially determined by the selection pressure. Higher selection pressures imply higher convergence rates and vice versa. If the convergence rate is too high, the probability of converging to a local optimum is also too high. Considering this contradiction, appropriate selection operators must be used. In this study, two widely used selection operators are considered: namely, tournament and roulettewheel selections. Detailed information can be found in relevant books (Haupt and Haupt 2004).

The crossover operator, which mixes the genes of chromosomes and generates new individuals, has always been regarded as the main search operator. It takes place with a probability of P_c that is a value between 0.5-1. In BCGA, new variables cannot be generated by applying crossover operators. This is due to fact that the crossover operators only mix the existing genes and do not create new individuals. New genes can only be created by the mutation operator. But this is not valid for RCGA. Applying various crossover operators, it is possible to create variables in the vicinity of their parents. While limited number of operators is proposed for BCGA, several different operators were suggested for RCGA.

The main advantage of RCGA is that the real-valued genes allow the application of a wide variety of crossover operators. Herera *et al.* (2003) have classified some of these RCGA operators which have different levels of exploration and exploitation capabilities. The definition interval of variables may be divided into subintervals and classified as exploration and exploitation zones (Herrera *et al.* 2003). Using their capabilities, these operators generate some diversity in the population or use this diversity to create better individuals.

2.1 Crossover operators

Some basic expressions for the crossover operators will be given in the following paragraph in order to emphasize working principles of these operators. The basic operators of BCGA are the single point (SPOX), the two point (TPX) and the uniform crossovers (UX). These operators can also be used in RCGA. It is enough to replace the bit strings by real-valued genes in this case (Lee and El-Sharkawi 2008). In addition, the widely used crossover operators; **A**rithmetical **C**rossover (AX), BLX-α Crossover, **H**euristic **C**rossover (HX), **S**imulated **B**inary **C**rossover (SBX), **U**nimodal **N**ormally **D**istributed **C**rossover (UNDX), **L**inear **C**rossover (LX), **S**implex **C**rossover (SPX), **P**arent **C**entric **N**ormal **C**rossover (PNX), **F**uzzy **R**ecombination **C**rossover (FRX) are explained below. In RCGA, the parent and the offspring chromosomes are defined as $\boldsymbol{x^k} =$

$\{x_1^k, x_2^k, \ldots x_i^k, x_{i+1}^k, x_{i+2}^k, \ldots x_n^k\}$ and $\bar{\boldsymbol{x}}^{\boldsymbol{k}} = \{\bar{x}_1^k, \bar{x}_2^k, \ldots \bar{x}_i^k, \bar{x}_{i+1}^k, \bar{x}_{i+2}^k, \ldots \bar{x}_n^k\}$, where k and n are the number of parents and the design variables, respectively.

The arithmetic crossover is proposed by Michalewics (1996). In this technique, two new individuals are obtained according to Eqs. (1) and (2)

$$x_i^{-1} = \lambda x_1^2 + (1-\lambda) x_i^2 \tag{1}$$

$$x_i^{-2} = \lambda x_1^2 + (1-\lambda) x_i^1 \tag{2}$$

Where λ is a uniform random number in the interval [0, 1]. As presented in Herera *et al.* (2003), the AX produces new genes $\bar{x}_i$ in the exploitation interval of parent genes x_i^1 and x_i^2. Consequently, AX has no exploration capability.

The BLX-α is suggested by Eshelman and Schaffer (1993). Two offsprings are obtained in the interval $[x_{\min} - I.\alpha, x_{\max} + I.\alpha]$ where $\boldsymbol{x_{min}} = \min\left[x_i^1, x_i^2\right]$, $\boldsymbol{x_{max}} = \max\left[x_i^1, x_i^2\right]$ and $\boldsymbol{I} = \boldsymbol{x_{max}} - \boldsymbol{x_{min}}$. The parameter α defines the interval that the new offsprings are generated. This parameter determines the exploration degree of the operator. For $\alpha = 0.5$, the probability of being inner and outer of exploitation interval is equal for the individuals. In this study, BLX-α operator is included in the comparison for the values of 0.1, 0.3 0.5, 0.7, 1 and 1.4 for the α parameter. The action interval of BLX-α is given in Herera *et al.* (2003).

The heuristic crossover proposed by Wright (1991), returns a child that lies on the line containing the two parents, a small distance away from the parent with the better fitness value in the direction away from the parent with the worse fitness value. The fitness values of the parents are used to determine the direction of the search. Using Eq. (3), new individuals are guaranteed to be in the exploration interval as shown in Fig. 1.

$$\bar{\boldsymbol{x}} = \boldsymbol{x^2} + r(\boldsymbol{x^1} - \boldsymbol{x^2}) \tag{3}$$

where the parameter r is the distance measured from the best individuals and takes values greater than 1. This parameter is selected as 1.2, 1.6 and 2 in this study.

The simulated binary crossover introduced in Dep and Agraval (1995) creates two new offsprings using Eqs. (4) and (5).

$$\bar{x}_i^1 = \frac{1}{2}\left[(1-\beta_k)x_i^1 + (1+\beta_k)x_i^2\right] \tag{4}$$

$$\bar{x}_i^2 = \frac{1}{2}\left[(1+\beta_k)x_i^1 + (1-\beta_k)x_i^2\right] \tag{5}$$

$$\beta(u) = \begin{cases} 2u^{1/(\eta+1)} & if\ u \le \frac{1}{2} \\ [2(1-u)]^{1/(1+\eta)} & if\ u > \frac{1}{2} \end{cases} \tag{6}$$

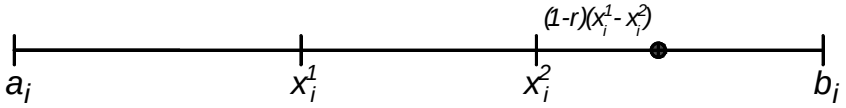


Fig. 1 Action interval of HX operator

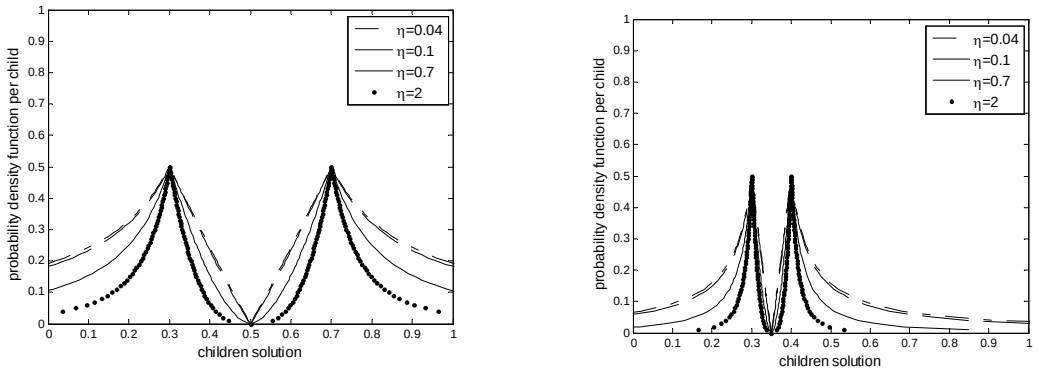


Fig. 2 The probability density functions for The SBX operator: (a) Parents are 0.3 and 0.7 and (b) Parents are 0.3 and 0.4

where u is a random number that lies in the interval [0, 1]. In Eq. (6), η is the distribution index that determines the distribution of the new individuals. For bigger values of η, new offsprings are created near parents and a small value of η allows distant solutions to be selected as children solutions. In this study, the values of 0.04, 0.1, 0.7, 1 and 2 are selected for η. If the difference between the parent genes is small, the probability of creating children near parents is high and vice versa. In Fig. 2, the probability distribution of the offsprings is given. The offsprings are widely distributed when parents are relatively distant as shown in Fig. 2(a). The opposite stuation can be observed in Fig. 2(b). This property is an important property of SBX to exhibit self-adaptation. In the initial population, if the parents are far from each other, it is expected that the children solutions will be distant and in later generations the parents will be closer and the children solution will be closer too. This is a dynamic property of SBX operator.

The unimodal normal distributed crossover was suggested by Ono *et al.* (1997). The UNDX uses three parents in order to create two offsprings. Children solutions are created by an ellipsoidal probability distribution. This probability distribution has one axis which is formed along the line joining two of the three parent solutions and another axis, for which the distance in the orthogonal direction is determined by the perpendicular distance of the third parent from the axis. The UNDX preserves the population statistics such as the covariance and the mean vector if certain parameters are chosen appropriately. Unlike the BLX-α operator, this operator assigns more probability for creating solutions in the exploitation interval. The method is outlined below:

(1) The parents x^1, x^2, x^3 are selected randomly.

(2) The mean of x^1 and x^2 is calculated, $x_p = \dfrac{x^1 + x^2}{2}$.

(3) The difference vector $\boldsymbol{d}$ of parent vectors x^1 and x^2 is computed as $d=x^1-x^2$.

(4) The line connecting the x^1 and x^2 is called the primary search line, the distance from the third parent x^3 to the primary search line is the distance D.

(5) A new individual is created using Eq. (7).

$$\bar{x} = x_p + \xi d + D\sum_{i=1}^{n-1} \eta_i e_i \tag{7}$$

where $\xi \sim N(0, \sigma_\xi^2)$ and $\eta \sim N(0, \sigma_\eta^2)$. $N(0, \sigma^2)$ is the normal distribution with variance σ^2. Variance of the distribution, σ_ξ^2 and σ_η^2 are 1/4 and $0.35^2/n$, respectively, where n is the number of design variables. Vectors $\boldsymbol{e_i}$ are the orthonormal bases that span the subspace perpendicular to $\boldsymbol{d}$.

The simplex crossover was introduced by Tsutsiu *et al.* (1999). More than two, say m, individuals can be used to create m offsprings. Certain number of chromosomes constitutes a simplex in the solution space. The simplex given in Tsutsiu *et al.* (1999) is generated by using three chromosomes. This simplex is expanded with a parameter ε and the new individuals are randomly chosen from this expanded simplex. The procedure is described as follows:

(1) If the population size is N, $m+1$ parent vectors $x^0, x^1 \ldots . x^m$ are chosen randomly within this population and the mean $O = \frac{1}{m+1}\sum_{k=0}^{m} x^k$ is calculated.

(2) A random number using Eq. (8) is generated where u is a random number in the interval [0,1]

$$r_k = u^{1/(1+k)}, k = 0,1,2,...m \tag{8}$$

(3) Y_i and C_i are obtained from Eqs. (9) and (10), respectively.

$$Y_k = 0 + \varepsilon(x_k - 0), k = 1,2,...m \tag{9}$$

$$C_k = \begin{Bmatrix} 0 & : k = 0 \\ r^{(k-1)}(Y_{k-1} - Y_k + C_{k-1}) & : k = 1,2,...m \end{Bmatrix} \tag{10}$$

(4) The new individual is obtained as $\bar{x} = Y_m + C_m$

If the parameter ε is taken $\sqrt{m+2}$, and if m is chosen as m=n, where n is number of design variables, population statistics such as the mean vector (mean of the parents) and the covariance matrix are preserved (Higuchi *et al.*). If the parameter ε is chosen as twice of the parameter of BLX-α and the parent number m is taken as 2, SPX corresponds to BLX-α operator.

The linear crossover which is proposed by Wright (1991) can produce three new individuals using two parents according to the Eqs. (11)-(13). Two fittest individuals are included in the next population. Each individual lies in the different portions of the search space for both exploration and exploitation (Herera *et al.* 2003).

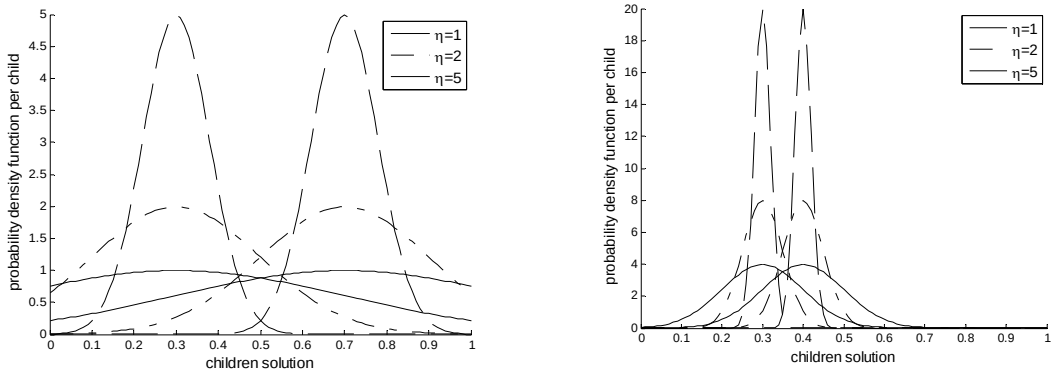
$$\bar{x}_i^1 = \frac{1}{2}x_i^1 + \frac{1}{2}x_i^2 \tag{11}$$

$$\bar{x}_i^2 = \frac{3}{2}x_i^1 - \frac{1}{2}x_i^2 \tag{12}$$

$$\bar{x}_i^3 = -\frac{1}{2}x_i^1 + \frac{3}{2}x_i^2 \tag{13}$$

The parent centric normal crossover proposed by Ballester and Carter (2004) generates new individuals which are assumed to be distributed according to the Gaussian probability density function. More than two offsprings can be created by PNX. The distribution function provides new individuals to be distributed in a wide portion of the search space. The probability density function of the offsprings is given in Fig. 3. Eq. (14) is used to create new offsprings.

$$\bar{x}_i = \begin{cases} N(x_i^1, |x_i^1 - x_i^2|/\eta) \\ N(x_i^2, |x_i^1 - x_i^2|/\eta) \end{cases} \tag{14}$$



(a) (b)

Fig. 3 The probability density functions for The PNX operator: (a) Parents are 0.3 and 0.7 and (b) Parents are 0.3 and 0.4

$N(\mu, \sigma)$, corresponds to the normal distribution with mean μ and a standard deviation of σ. In Eq. (14), η is a control parameter. For the large values of η, the offsprings are created closer to the parents. The values of 1, 2 and 5 are selected for this parameter.

The fuzzy recombination crossover is suggested by Voight *et al.* (1995). New individuals are created using a probability density function that resembles the membership functions of the fuzzy logic theory. New individuals are selected from the interval given by Eqs. (15) and (16). Parameter *d* determines the distribution of the offsprings. For small values, new individuals are created near parents and vice versa. The values of 0.3, 0.5 and 0.9 are used for the parameter *d*. The distribution of offsprings is given in Fig. 4.

$$x_i^1 - d|x_i^1 - x_i^2| \leq r \leq x_i^1 + d|x_i^1 - x_i^2| \tag{15}$$

$$x_i^2 - d|x_i^1 - x_i^2| \leq r \leq x_i^2 + d|x_i^1 - x_i^2| \tag{16}$$

In order to increase the performance of GAs using these operators, exploitation and exploration intervals of those crossover operators should be well understood. For the further generations,

exploitation interval becomes narrow. Hence, new individuals resemble their parents more. In these stages, most of the crossover operators search the solution locally. If the solution in these stages is closer to the optimum solution, this local search increases the efficiency of the method. Otherwise some other operators are needed that have high exploration capabilities. In Figs. 2-4, the distribution of the offsprings obtained from SBX, PNX and FRX is given for various distribution indexes. When the parents are distant, the probability of generating offsprings far from parents is high and vice versa. To achieve diversity in the population and find the global solution efficiently, distribution indexes should be appropriately determined.

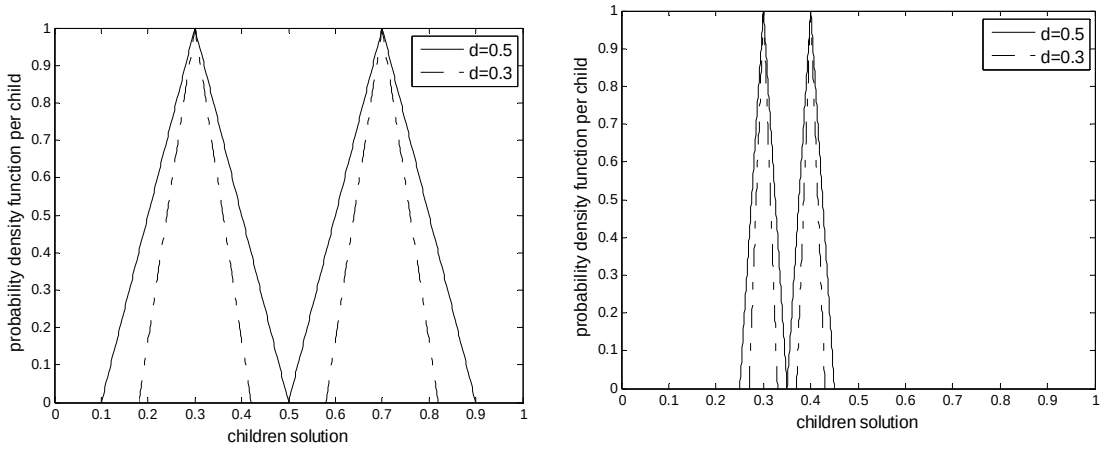


Fig. 4 Normalized probability density functions for The FRX operator: (a) Parents are 0.3 and 0.7 and (b) Parents are 0.3 and 0.4

2.2 *Mutation operators*

Mutation takes place with small probabilities P_m=0.01-0.10. It produces random diversity in the population. The main purpose of this operator is to prevent the search process to get stuck at local minima. For large probabilities of mutation, randomness increases and the convergence rate decreases. In this study, different mutation operators are investigated for the damage identification problem. In BCGA, the mutation simply changes the bits from 0 to 1 or from 1 to 0 with certain probabilities. Nevertheless, several mutation operators are suggested for RCGA. Some of them are outlined below. For basic equations for these mutation operators corresponding references can be appealed.

In uniform Mutation (UM), gene values change to any values that lie between the lower and upper limits of the gene (design variable). The change is purely random in RCGA. The Non-uniform mutation **(**NUM) operator was proposed and developed by Michalewics (1996) to reduce the effects of randomness of the uniform mutation. When the generation increases, the change in the variables becomes smaller. In this operator, a control parameter is used to determine the degree of the dependence on the generation number. This technique can be viewed as a dynamic mutation technique as the changes in the parameters depend on the generation number.

Makinen, Periaux and Toivanen Mutation (MPTM) was proposed by Makinen *et al.* (1999) and implemented on the shape optimization problems in electromagnetics. In this method, a distribution parameter is proposed which controls the amount of change in the mutated gene. The Power Mutation (PM) proposed by Deep and Thakur (2007) is based on power distribution. The distribution index which is similar to one used in MPTM determines the performance of the PM. For small values of this parameter, the amount of change in the genes become small and for the large values of this parameter, the amount of change in the genes becomes higher. For this distribution index, the values of 0.1, 0.5 and 0.9 are used in this study as proposed by the authors.

2.3 Parallel Genetic Algorithms (PGAs)

The application of GAs into the complicated physical problems requires high computational resources. Time consuming objective functions are needed for the formulation of these problems. Nevertheless, the GAs do not have effective local search capabilities. Hence, the convergence speed decreases especially in the later generations. The PGAs solve these problems and increase the numerical performance. Lim *et al.* (2007) gives a brief review on different types of PGAs. In this study, multi-population PGAs is used.

Multi-Population PGAs: The multi population PGAs increases the solution quality as well as the computation speed. In this technique, PGAs consist of several subpopulations and GA operators are implemented on each subpopulation separately. Exchange of the individuals between subpopulations called *migration,* occurs occasionally. There are some parameters that affect the performance of PGAs. These are the migration rate, the migration interval and the migration direction. The migration rate defines the number of individuals that are exchanged. The migration interval is the number of generations required to exchange the chromosomes and the migration direction is the migration topology for transferring the individuals. Meruane and Heylen (2010) give the appropriate values of these parameters for the damage detection problem with FEMU. The population model of the multi-population PGAs can be found in Lim *et al.* (2007).

In this paper, the multi-population PGAs are performed on a parallel cluster with distributed memory architecture and Intel Xeon 2.33 GHz processors. This system is founded in the National Center for High Performance Computing at ITU together with some other computing server systems. The system has 192 computing nodes and 1004 computing cores. The PGA which is programmed in Matlab Computing Language version R2009b and MATLAB Distributed Computing Server toolbox is used to execute the independent GA operations on the cluster.

3. Model definition and comparison set up

There are several numerical studies that compare the GA operators' performance for a test bed which includes functions such as Sphere, Schwefel, RosenBrock, Rastrigin and Ackley (Boyer and Martinez 2008). These functions are difficult to optimize as some of them are nonlinear, multimodal, discontinuous or have high epistasis. In this study, the performance of the several GA operators and parameters is evaluated for the damage identification problem with FEMU. The problem is formulated as a nonlinear least square problem which is given in Eq. (23(a)).

$$f(\boldsymbol{a}) = \frac{1}{2}\|r(\boldsymbol{a})\|^2 = \frac{1}{2}\left\|\begin{matrix} r_f(\boldsymbol{a}) \\ r_s(\boldsymbol{a}) \end{matrix}\right\|^2 \tag{17a}$$

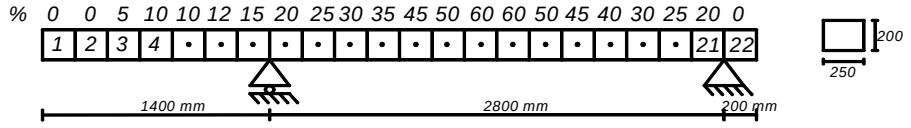


Fig. 5 Finite element model of the beam and the damage distribution

$$r_f(\boldsymbol{a}) = \frac{\lambda_j(\boldsymbol{a}) - \tilde{\lambda}_j}{\tilde{\lambda}_j} \tag{17b}$$

$$r_s(\boldsymbol{a}) = \frac{\phi_j^l(a)}{\phi_r^l(a)} - \frac{\tilde{\phi}_j^l}{\tilde{\phi}_r^l} \tag{17c}$$

$$a_i^e = \frac{x_i^e - x_0^e}{x_0^e} \Rightarrow x_i^e = x_0^e(1 - a_i^e) \tag{17d}$$

In Eq. (17(a)), $f(\boldsymbol{a})$ is a nonlinear function of the damage parameters which are the reduction of the bending stiffness *EI* of the beam given in Fig. 5. This function includes the differences between the experimental and the numerical eigenfrequencies as well as the mode shapes. The $\lambda_j(\boldsymbol{a})$ and $\tilde{\lambda}_j$ are the updated and experimental eigenvalues, respectively. The $\phi_j^l(\boldsymbol{a})$ and $\tilde{\phi}_j^l$ are the updated and experimental mode shapes. The first ten frequencies and eight mode shapes are included in the objective function. The damage scenario is given in Fig. 5 as percentage reduction of the bending stiffness of the beam. The elastic modulus is chosen as 30000 MPa.

Many comparisons are presented including encoding, selection, crossover and mutation operators in the next section. A comparison between PGA and GA is also given. Comparisons were made based on some defined criteria in the GA literature. These criteria are the reliability, accuracy and efficiency (Deep and Thakur 2007). Reliability can be defined as the number of independent successful runs that satisfy certain criteria. Accuracy is the measure of exactness in finding the global optimum and the efficiency is the measure of the convergence rate. GAs are repeated 20 times for each operator and parameter. The mean of the best fitnesses, standard deviations and the number of function evaluations is recorded considering these criteria.

The results are also compared using some statistical tests in order to check whether there are significant differences between classifiers or not. The Friedman test with corresponding post hoc tests is carried out in order to statistically validate the results. A comprehensive pairwise comparison is presented by using Holm's procedure. The detailed information about relevant statistical procedures can be found in Demsar, 2006, Garcia and Herrera, 2008.

4. Numerical comparisons

In this section, all numerical comparisons are presented for all basic GA operators. The performance of multiple crossovers is also evaluated at the end of this section. In first step, BCGA and RCGA are compared with each other. BCGA is divided into two parts, the first one is the standard bit binary GA (SBBGA) and the second is Gray coding binary GA (GCBGA). For a fair comparison, the basic GA operators: namely, single-point crossover and uniform mutation with the roulettewheel selection method is selected as GA operators. Maximum generation number is taken

Table 1 Comparisons of encoding methods with variable mutation rates

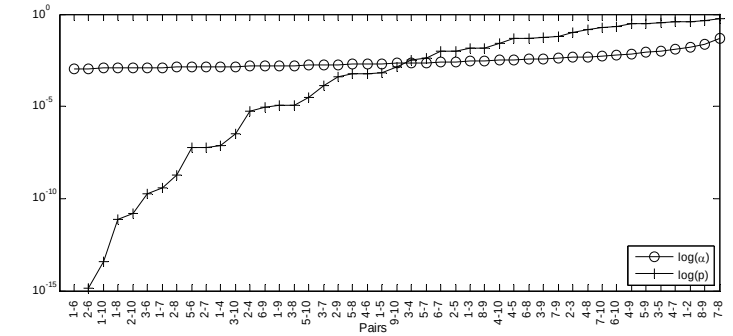
Classifier no	1	2	3	4	5	6	7	8	9	10
Coding/mut.rate	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
SBBGA	0.4175	0.4006	0.2813	0.2914	0.2337	0.2823	0.1698	0.2174	**0.1575**	0.2512
	0.1670	0.2765	0.1826	0.1888	0.2368	0.2257	0.1869	0.2468	**0.1371**	0.2394
	%5	%5	-	-	%10	-	%50	-	%30	-
GCBGA	0.0070	0.0043	**0.0038**	0.0044	0.0058	0.0099	0.0097	0.0087	0.0101	0.0160
	0.0064	0.0050	**0.0035**	0.0043	0.0050	0.0087	0.0085	0.0077	0.0046	0.0129
	-	%20	%35	%45	-	-	-	-	-	-
RCGA	1.0499	0.6939	0.0868	0.0290	0.0355	**0.0169**	0.0275	0.0232	0.0559	0.0174
	1.1872	1.5923	0.1588	0.0274	0.0344	0.0153	0.0342	0.0213	0.1315	**0.0099**
	-	-	-	-	-	%70	-	%15	-	%15

as 500. For the purpose of determining an optimum mutation rate, the methods were tested for each mutation rate in the interval 0.01-0.10 with an increment of 0.01. The population number is selected as N=100 and P_c is chosen as 0.80. Results are given in Table 1. The first and the second rows are the mean values and the standard deviations of the objective functions for 20 runs, respectively. The third row is the ratio of the best objective function values obtained among different P_m values for 20 runs.

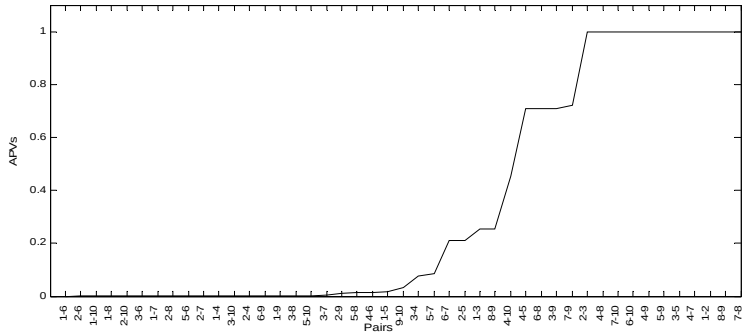
The results obtained for SBBGA is not efficient. The best mean fitness value is obtained for $P_m = 0.09$ and 50% of the total best fitnesses is obtained for $P_m = 0.07$. Actually, these mutation rates are relatively high values for BCGA. When the standard deviations for low values of the probability of mutation are relatively small, the mean of the best fitness values is higher for SBBGA. The results obtained by the GCBGA are significantly different from the other two methods since Gray coding is successful to overcome the Hamming distance problem. The results are also more stable. Best results are found for the mutation rates of 0.02-0.04 which are the expected values for BCGA. The results of RCGA are better than SBBGA but worse than the GCBGA. The reason is that the mutation operator causes more random changes in the genes of RCGA than in BCGA. When we apply the operators explained in Section 2, the performance of RCGA will increase and be superior to BCGA.

The Friedman test with Holm's procedure for pairwise comparisons is applied to mutation probabilities for RCGA. In order to make the pairwise comparison more apparent, two different types of figure are given for all comparisons. In the first type, the p-values and the corresponding confidence levels (α values) are plotted. The logarithmic scale is used in order to make the differences between p-values and α more obvious. In the second type, adjusted p-values (APVs) are plotted. APVs take into account that multiple tests are conducted. An APV can be compared directly with any chosen significance level α (Garcia and Herrera 2008). The tables are also given for corresponding statistical comparisons in Appendix A.

For the mutation probabilities, Iman-Davenport's statistic is calculated as 108 which is quite higher than critical value 1.88 at a=0.05. Due to the fact that the critical value is lower than the calculated statistic, Holm's procedure can be carried out for pairwise comparisons. Aforementioned graphics are given in Fig. 6. In Figs. 6(a) and 6(b), x axis shows compared pairs. Each mutation probability is numbered as classifiers in Table 1. As seen in Fig. 6, there are significant differences for the half of the classifiers. Particularly, performance for mutation



(a) *p*-values versus α



(b) Adjusted *p*-values

Fig. 6 Statistical pairwise comparison of mutation probabilities of RCGA

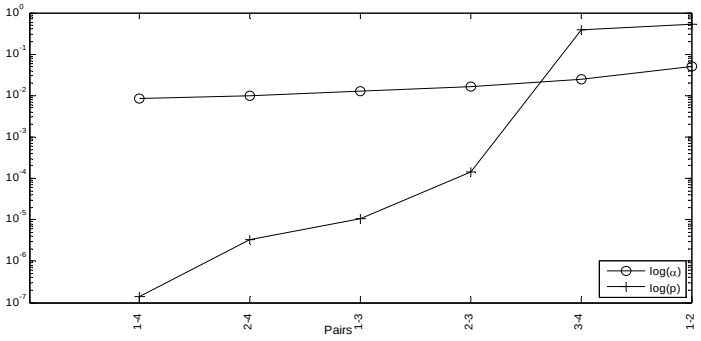
probability 0.06 (classifier 6) has significantly better than the other mutation probabilities. As apparent in Fig. 6(b), null hypothesis can be rejected for all pairs that include classifier 6 at a certain level of significance except pair 6-10.

In the second step, the roulettewheel (RW) and the tournament (TR) selection together with different tournament size are compared. For the roulettewheel selection method, linear scaling is used to prevent the domination of the fittest individuals. Tournament size is selected as 2, 4 and 8 for the tournament selection method to investigate the effects of the selection pressure. The other operators and parameters are as follows: N=100, P_c=0.80, P_m=0.06, crossover is heuristic crossover with r=1.6 and mutation is the non-uniform mutation. Maximum generation number is taken as 1000 and $f - f_{opt} \leq 10^{-5}$ is determined as the stopping criteria where f_{opt}=0. Success rate is defined as follows: if any run satisfies the inequality before the maximum generation number is reached, then it is considered as a successful run.

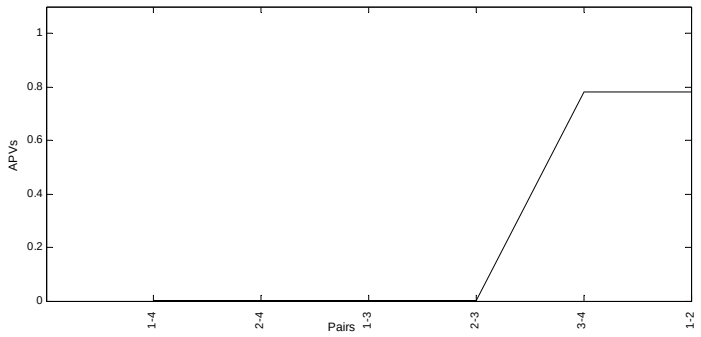
As shown in Table 2, results obtained using roulettewheel are more accurate and reliable. Increasing the tournament size effects the results unfavorably. As the tournament size is enlarged, the selection pressure increases which results in a decrease in the population diversity. This undesirable phenomenon can be prevented by the roulettewheel selection operator together with a

Table 2 Comparison of the selection methods

Classifier No	Selection method	Mean of the best objective function values	Standard deviation	Success rate (%)	Average number of function evaluations of successful runs
1	RW	7.56E-06	1.06E-05	70	66300
2	TR2	1.36E-05	1.33E-05	50	69700
3	TR4	9.10E-05	9.74E-05	0	-
4	TR8	1.50E-04	1.18E-04	0	-



(a) *p*-values versus α



(b) Adjusted *p*-values

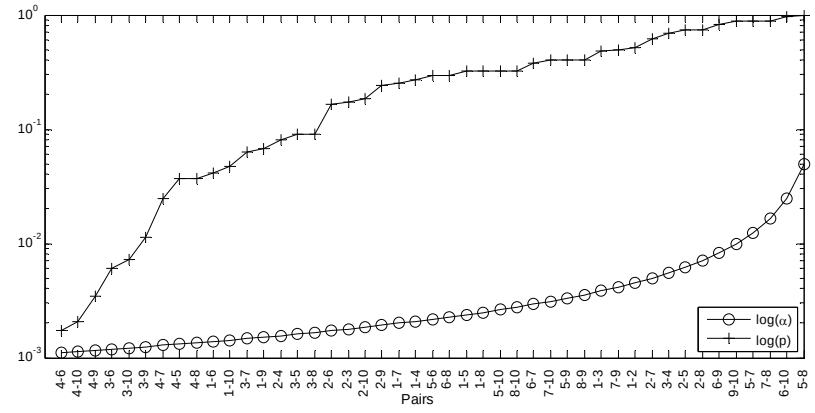
Fig. 7 Statistical pairwise comparison of selection methods

scaling technique called linear scaling.

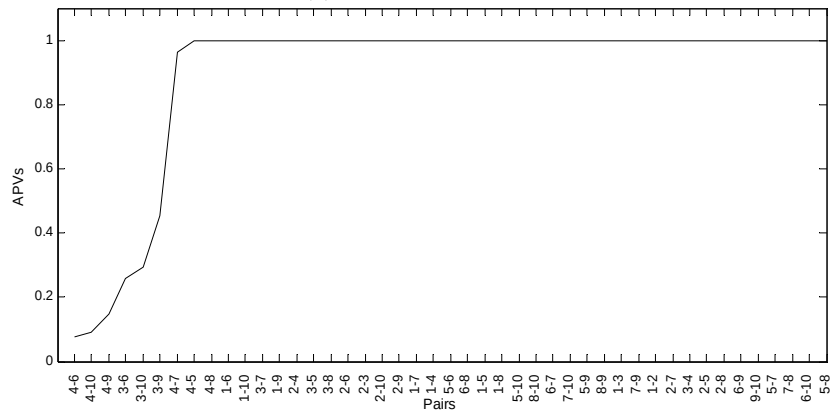
The statistical tests carried out for the selection methods also support the results presented in Table. 2. For this case, the Iman-Davenport's statistic is 44.97 and the critical value is 2.76. Hence, subsequent post hoc test can be applied for pairwise comparison. The results for the statistical tests are presented in Fig. 7. As apparent in Fig. 7(a), the null hypothesis for the pairs RW-TR4 (1-3), RW-TR8 (1-4), TR2-TR4(2-3) and TR2-TR8 (2-4) can be rejected at a=0.05. . However, the null hypothesis for the pairs RW-TR2 (1-2) which give the best results can be rejected at a=0.80 (see Fig. 7(b)).

Table 3 Results obtained from mutation operators

Classifier No	Mutation operator	Mean of the best objective function values	Standard deviation	Success rate (%)	Average number of function evaluations of successful runs
1	UM	1.535E-05	1.751E-05	45	73600
2	PM-0.1	7.06E-05	0.000162	55	74900
3	PM-0.5	7.99E-05	0.000253	45	75500
4	PM-0.9	1.87E-05	1.543E-05	35	72600
5	NUM-3	8.42E-06	6.474E-06	65	69200
6	NUM-5	7.56E-06	1.07E-05	75	66300
7	NUM-8	1.01E-05	1.208E-05	60	**56100**
8	MPTM-4	9.28E-06	**7.567E-06**	55	69300
9	MPTM-10	6.43E-06	6.35E-06	75	71200
10	MPTM-16	**5.53E-06**	5.86E-06	**80**	70500



(a) *p*-values versus α



(b) Adjusted *p*-values

Fig. 8 Statistical pairwise comparison of mutation operators for RCGA

Table 4 Results from comparison of crossover operators

Classifier No	Crossover Operators	Mean of the best objective function values	Standard deviation	Success Rate (%)	Average number of function evaluations of successful runs
-	SPOX	0.000223	0.000151	-	-
-	TPX	0.000287	0.000279	-	-
-	AX	0.000461	0.000222	-	-
-	UX	0.001053	0.002048	-	-
1	UNDX	8.28E-05	3.21E-05	-	-
2	SPX	3.22E-05	2.28E-05	5	54300
-	SBX-0.04	1.12E-05	7.64E-06	45	78967
-	SBX-0.1	4.30E-06	4.75E-06	85	78805
-	SBX-0.7	6.04E-06	5.38E-06	75	67266
3	SBX-1	4.22E-06	4.85E-06	85	66482
-	SBX-2	8.58E-06	6.95E-06	60	70666
4	LX	5.46E-06	5.64E-06	80	85119
-	HX-1.2	3.53E-05	2.89E-05	25	74520
5	HX-1.6	6.29E-06	5.33E-06	90	72894
-	HX-2	9.26E-06	7.51E-06	60	75008
-	BLX-0.1	0.000145	0.000109	-	-
-	BLX-0.3	5.30E-05	9.91E-05	20	66125
-	BLX-0.5	9.53E-06	6.23E-06	50	69310
6	BLX-0.7	3.73E-06	5.70E-06	80	**52731**
-	BLX-1	4.01E-06	3.73E-06	**95**	68415
-	BLX-1.4	5.38E-06	**3.67E-06**	90	76628
-	PNX-0.5	1.63E-05	1.04E-05	25	89050
7	PNX-1	**3.06E-06**	5.28E-06	90	69188
-	PNX-2	4.11E-06	4.68E-06	85	58070
-	PNX-5	3.85E-05	3.23E-05	15	74866
-	FRX-0.3	4.39E-05	3.14E-05	15	74867
-	FRX-0.5	2.78E-05	2.24E-05	25	71980
8	FRX-0.9	8.19E-06	8.19E-06	70	70685

However, the null hypothesis for the pairs RW-TR2 (1-2) which give the best results can be rejected at (see Fig. 7(b)).

In Table 3, the comparison results are given for mutation operators. The GA parameters and operators are N=100, P_m and P_c and are 0.06 and 0.80, respectively. The crossover operator is the heuristic crossover with r=1.6. The stopping criteria are the same as given in the previous section.

The NUM operator which is a dynamic mutation operator and the MPTM gives the best results. The amount of mutation can be adjusted by the selection of the appropriate parameter values for the NUM, MPTM and PM. However, only the NUM operator enables tuning of the mutation amount dynamically for successive generations. There are no significant differences in the mean of the best objective function values for different mutation operators. But PM and UM results are less reliable than the other two methods. This fact is due to more random changes in the genes caused

by PM and UM. It is clear that the use of NUM and MPTM as the primary mutation operator is feasible.

The Iman-Davenport's statistic and critical value are 2.5664 and 1.88, respectively. In this case, the calculated statistic value is just above the critical value. The results of the Holm procedure are given in Fig. 8. However, the Holm procedure cannot detect a significant difference between the performances of the mutation operators. In this situation, some more complicated statistical tests such as Hochberg and Hommel procedures can be applied. Since the mutation operator is the secondary operator in GA, additional statistical test will not be carried out to see if there is a significant difference between mutation operators.

In the next paragraphs, the performance of the crossover operators is compared and the results are presented. The GA operators and parameters used in this comparison are, N=100, P_c and P_m are 0.80 and 0.08, respectively. The stopping criteria are the same as explained in the first comparison. The results are given in Table 4 and Fig. 9.

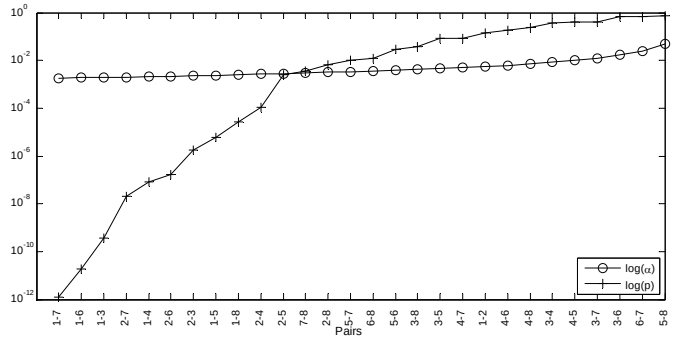
As shown in Table 4, the PNX operator with the distribution parameters 1 and 2 and BLX-α, with the distribution parameters 0.7, 1 and 1.4, outperform all methods in terms of all comparison criteria. For small values of the distribution parameter η, PNX performs better. For large values of η, the results worsen. If the results obtained by PNX, are investigated, it can be observed that the number of the mean function evaluations of successful runs for PNX-2 is smaller than that obtained for the PNX-1. However, the mean values and the standard deviations are better for PNX-1. Hence, it can be deduced that there is a trade-off between the reliability and the efficiency. When the probability of creating near parent solutions is high, the efficiency of the operator increases, the reliability decreases. The same is true for the SBX-0.7 - SBX-0.1 and the BLX-0.7 – BLX-1. The results obtained for the BLX-α operator is quite good. For the increasing values of the parameter α which determines the width of the search space, solutions are getting better. However, for the values of α greater than 1, the results slightly worsen because of the fact that the new individuals are generated far away from their parents which increases the randomness. This is also valid for the HX, SBX, PNX operators. If the results in Table 4 are investigated for these operators, it is observed that increasing or decreasing values for these parameters after a certain parameter value causes the solutions to worsen. Hence, determining the optimal parameter values for these operators is crucial in order to obtain a suitable distribution for the offsprings. A suitable distribution of the new individuals is the necessary condition to find the optimal solution. The LX operator which creates new individuals in the vicinity of both the exploration and the exploitation intervals gives excellent results. However, the required number of function evaluations to reach the optimum is higher than the other methods. The results obtained by HX are quite good. This method with the appropriate parameter (e.g. 1.6), is as reliable as SBX and LX and the mean of the best fitness values and the standard deviations obtained from HX are slightly less than SBX, PNX, BLX-α and LX. The operators SPX and UNDX have difficulties exploring the search space. The mean fitness values for these operators are not too bad, but the success rates are almost zero. These operators with the proposed parameter values are inadequate. It is apparent that the population statistics such as the mean vector and the covariance matrix are not preserved in the FEMU problem.

Although there are no significant differences between UNDX, SPX, SBX, HX, BLX-α, PNX and FRX in terms of the mean fitness values, the difference is significant for SPOX, TPX, AX and UX. The AX operator which creates new offsprings in the exploitation interval is incapable to find good solutions in terms of all the comparison criteria. The SPOX, TPOX and UX do not create new genes via crossover. They only use the parent genes and mix them to create new individuals.

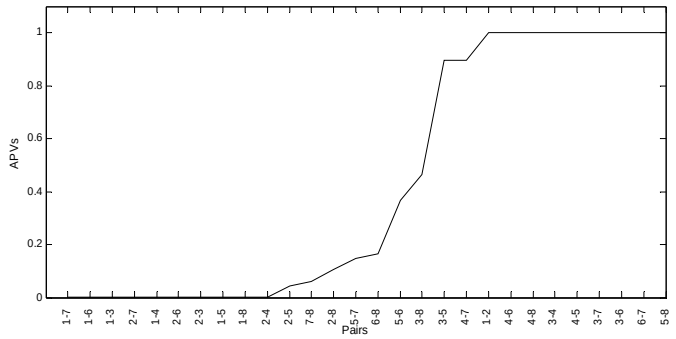
Hence, they cannot consider the exploration-exploitation degree. As shown in Table 4, the results are insufficient for these operators. As a result, the PNX-1, PNX-2, the BLX-1, BLX-0.7, the SBX-0.1, SBX-0.7, SBX-1, the HX-1.6 perform better than the other operators.

The conclusion drawn from Table 4 is also supported by the statistical analysis results. The statistical analysis is carried out in two stages. In the first stage, same crossover operators with different parameter values are compared. In second stage, the crossover operators with the best parameter values which is determined in the first stage are compared with each other. These operators are numbered in Table 4 as classifiers of the second stage. The tables of the results belong to first stage is given in Appendix A and the results of the second stage is given in Fig. 9. The Iman-Davenport's statistic and critical value are 18.92 and 2.01, respectively. As apparent, in Fig. 9a, the *p*-values of the half of the pairs are below the significance level of $\alpha = 0.05$. Especially the pairs including the UNDX and SPX operators are below that level. It is due to fact that the performances of these operators are obviously worse than the other operators. Nevertheless, the performances of the operators HX-1.6, LX, SBX-1, BLX-0.7 and the PNX-1 cannot be separated at a certain significance level as seen in Fig. 9b. Hence, it can be said that these operators performs better than other operators at the specified significance level. However, an obvious performance difference between these operators is not apparent from statistical stand point.

In Table 5, the multi-population PGAs mentioned in Sec. 2.3 is compared with the sequential GAs and the results are presented for four differet population size. The subpopulation number is selected as 10 for PGAs.. The migration interval and the migration rates are taken as 100 and 0.30,



(a) *p*-values versus α



(b) Adjusted *p*-values

Fig. 9 Statistical pairwise comparison of crossover operators

respectively. The roulettewheel selection, the NUM mutation operator and the operator PNX are adopted for the selection, mutation and the crossover, respectively. The mutation rate P_m=0.06 and a crossover rate P_c=0.80 are selected.

As shown in Table 5, although the success rates remain unchanged, the convergence speeds increase 3 to 5 times. If the mean generation numbers are compared for PGAs and GAs, the convergence is higher for the PGAs except for the population size of 150. This means that the PGAs become more effective through the increasing size of the subpopulation. For the population sizes of 200 and 300, the mean generation number for PGAs is less than the GAs. This is due to fact that PGAs provide more diversity to the population. In Fig. 10, the variation of the best and the mean values of the objective function of the population versus the generation number is given for the best of the 20 independent runs. The population size and the maximum generation number are selected as 200 and 1000, respectively.

As apparent in Fig. 10, the best and mean fitness values coincide in 850.th and 950.th generation numbers for GAs and PGAs, respectively. This means that PGAs increases diversity

Table 5 Comparison for GAs and PGAs

Method	Populationsize	Meangeneration	Successrate (%)	Mean executiontime (sn)
PGA	100	610	85	77
	150	589	100	97
	200	325	100	65
	300	340	100	90
GA	100	634	85	316
	150	435	100	355
	200	403	100	385
	300	388	100	549

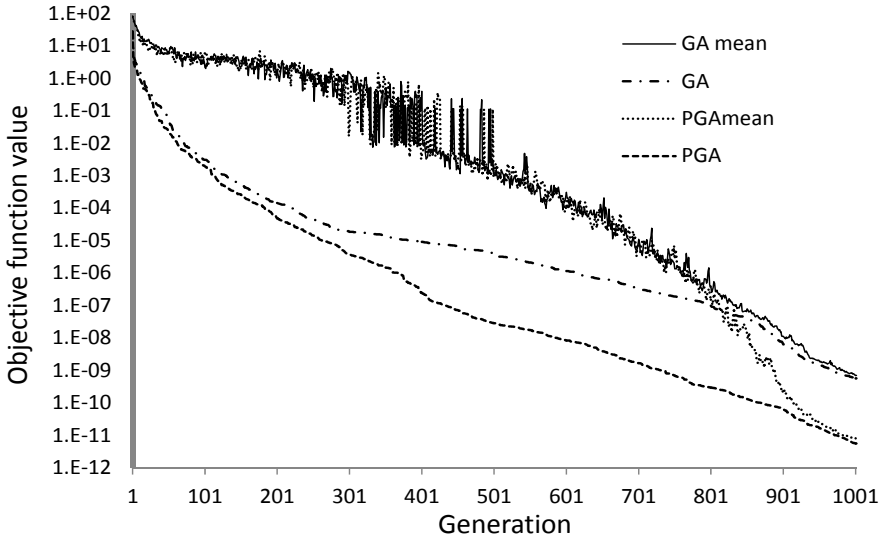
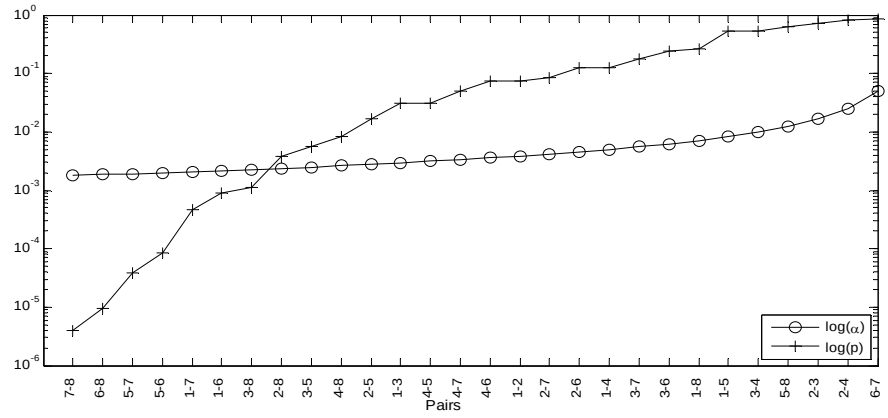


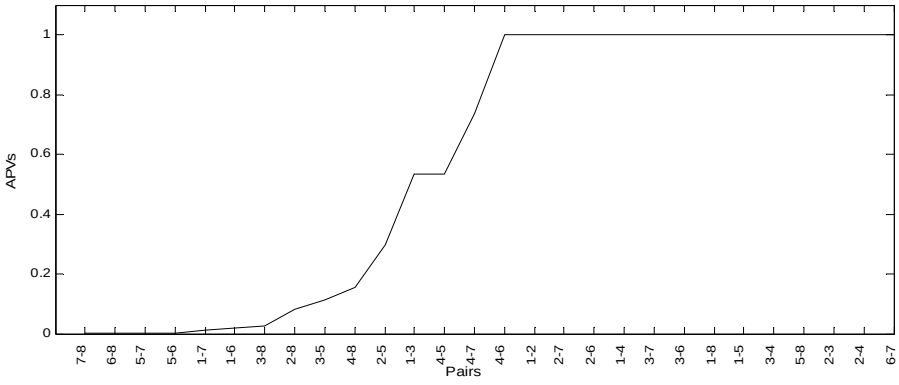
Fig. 10 Variation of the best and the mean objective function values with generation number

Table 6 Comparison of multiple crossover operators

Classifier No	Crossovermethods	Successrate (%)	Mean of the best fitness values	Average number of function evaluations of successful runs
1	PNX1	30	4.993E-07	306520
2	PNX2	60	1.831E-07	253400
3	PNX2-BLX0.3-SBX0.04	65	1.055E-07	264464
4	PNX2-SX1.6-1.2-SBX0.04	55	9.933E-08	256980
5	FRX0.5-SX1.6-1.2-UNDX	55	2.924E-06	314380
6	BLX0.7-SX1.6-1.2- PNX1	**90**	2.782E-08	269300
7	PNX1- PNX2- PNX4	**100**	1.833E-08	300140
8	BLX0.2-BLX0.5-BLX0.8	30	1.504E-06	196510



(a) *p*-values versus α



(b) Adjusted *p*-values

Fig. 11 Statistical pairwise comparison of multiple crossover operators

and prevent the premature convergence more effectively than the GAs. It should also be stated that objective function values of both PGA and GA are expected to stabilize at the further generations due to loss of population diversity.

In addition to the above comparisons, the results from the multiple use of crossover operators are presented. Some forms of crossover operators might be more suitable than others at the different stages of the genetic process. Multiple use of different crossover operators provides an effective search of the solution space. Nevertheless exploration-exploitation degree can be balanced by the use of crossover operators together.

In Table 6, results obtained by different combinations of crossover operators and parameters are presented. In each generation, six offsprings were created by the multiple implementations of three crossover operators. The two most promising offsprings substitute their parents in the next generation. The combinations are compared with the PNX operator which gives the best results in previous comparisons. For a fair comparison, the maximum generation number is determined as 2000 and 5000 for multiple and single crossover. Stopping criteria is taken $f - f_{opt} \leq 10^{-7}$.

As can be observed, there are significant differences between some of the multiple and single crossover operators. The combination at the last row of Table 6 shows a poor performance. But if we look into the last column of Table 6, for this operator, the mean number of function evaluation of successful runs is less than the others. This means that the method is effective but not reliable. In other words, the method finds the optimum point relatively fast, however it can also convergence to non-optimum points for different runs. The reason could be the premature convergence of the algorithm. The GA generally convergences to a non-optimum point if the diversity of the population lost due to fact that one of the fittest individual dominate the population in early iterations. In order to avoid this phenomenon, selection pressure could be reduced by adjusting some parameters belong to GA operators or appropriate combinations of the operators are sought as done in this study. Examples of these combinations can be increased by considering different parameters and operators as well. The statistical analysis results are well compatible with the above results. The Iman-Davenport's statistic and critical value are 6.20 and 2.01, respectively. The pairwise comparisons which are given in Fig. 11 show that there are no significant differences between the performances of best classifiers 2, 3, 6 and 7. However, the success rate of the classifier 7 is better than the others.

5. Conclusions

In this paper, the performance of the several GA operators and the performance of the parameters were compared for the damage identification problem with finite element model updating (FEMU). The genetic algorithms (GAs) is affected by the application of different operators and parameters. It should not be expected that the performance of these operators and parameters is the same for all class of problems. Hence, choosing the appropriate ones considering the problem at hand is crucial. Thus, the results obtained in this study will be valid for the damage detection problem.

For the mutation operator, whose primary goal is to provide random diversity in the population, the dynamic mutation operator, non-uniform mutation (NUM) is found to be the best mutation operator. This is due to the fact that the NUM operator tends to search the space uniformly in initial generations and more locally in the later. Among the GA crossover operators, the parent

centric normal crossover (PNX) achieves the best overall performance. The simulated binary crossover (SBX) which uses a different probability distribution to create offsprings, the heuristic crossover (HX) and the BLX-α also give close results to those obtained from the PNX operator. The selection of the appropriate corresponding values of the parameters is vitally important. These parameters should be determined considering the exploration-exploitation capabilities of these operators. Unfortunately, giving an explicit formula is not possible. Thus, it is necessary to make a parametric study for each problem separately.

A trade-off between the reliability and the efficiency comes up with the use of the crossover operators which create offsprings via some distribution functions. This trade-off should be considered in the selection of the feasible parameter values. The use of the mutation operators that have local search capabilities is adequate. However, the crossover operators which create offsprings too close to their parents don't give sufficient results. For the selection operator, the roulettewheel selection with linear scaling outperforms the tournament selection with different tournament sizes. The parallel genetic algorithms (PGAs) enhance the performance of GAs in terms of the convergence speed and the efficiency. It should also be stated that the real coded genetic algorithms (RCGA) with special operators outperforms binary coded genetic algorithms (BCGA).

Acknowledgments

In the present study, the parallel computations are carried out in the framework of the project supported by ITU National Center for High Performance Computing with Grant Number: 20682009.This support is acknowledged. The second author is the promoter of the TUBITAK Project No: 107M573. This support is also gratefully acknowledged here.

References

Bakir, P.G., Reynders, E. and De Roeck, G. (2008), "An improved finite element model updating method by the global optimization technique Coupled Local Minimizers", *Comput. Struct.*, **86**(11-12), 1339-1352.

Bakir, P.G., Reynders, E. and De Roeck, G. (2007), "Sensitivity-based finite element model updating using constrained optimization with a trust region algorithm", *J. Sound Vib.*, **305**(1-2), 211-225.

Bakir, P.G. (2011), "The combined deterministic stochastic subspace based system identification in buildings", *Struct. Eng. Mech.*, **38**(3), 315-332.

Bakir, P.G. (2012), "Instrumentation and system identification of a typical school building in Istanbul", *Struct. Eng. Mech.*, **43**(2), 179-197.

Bakir, P.G., Alkan, S. and Eksioglu, E.M. (2011), "A comparative study on the subspace based system identification techniques applied on civil engineering structures", *Smart Struct. Syst.*, **7**(2), 153-167.

Ballester, P.J. and Carter, J.N. (2004), *An effective real-parameter genetic algorithm with parent centric normal crossover for multimodal optimisation*, Genetic and Evolutionary Computation – GECCO, 901-913.

Boyer, D.O. and Martinez, C.H. and Pedrajas, N.G. (2008), "Robust confidence intervals applied to crossover operator for real-coded genetic algorithms", *Soft Comput.*, **12**(8), 809-833.

Chou, J.H. and Ghaboussi, J. (2001), "Genetic algorithm in structural damage detection", *Comput. Struct.*, **79**(14), 1335-1353.

Cruz, P.J.S and Salgado, R. (2009), "Performance of vibration-based damage detection methods in bridges", *Comput.-Aided Civil Infrastruct. Eng.*, **24**(1), 62-79.

Deep, K. and Thakur, M. (2007), "A new mutation operator for real coded genetic algorithms", *Appl. Math. Comput*., **193**(1), 211-230.
Demsar, J. (2006), "Statistical comparisons of classifiers over multiple data sets", *J. Mach. Learn. Res*., **7**(1), 71-30.
Dep, K. and Agraval, R.B. (1995), "Simulated binary crossover for continuous search space", *Complex Syst*., **9**, 115-148.
Erdogan, Y.S. and Bakir, P.G. (2013), "Inverse propagation of uncertainties in finite element model updating through use of fuzzy numbers", *Eng. Appl. Artif. Intel*., **26**(1), 357-367.
Eshelman, L.J. and Schaffer, J.D. (1993), *Real-coded genetic algorithms and interval schemata*, D.L. Whitley (Ed.), Foundation of Genetic Algorithms II, Morgan Kaufmann, San Mateo, CA, 187-202.
Friswell, M.I. and Mottershead, J.E. (1995), *Finite element model updating in structural dynamics*, Kluwer Academic Publishers, Dordrecht, The Netherlands.
Garcían, S. and Herrera, F. (2008), "An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons", *J. Mach. Learn. Res*., **9**(12), 2677-2694.
Goldberg, D.E. (1989), *Genetic algorithms in search, optimization, and machine learning, massachusetts*, Addison-Wesley.
Hao, H. and Xia, Y. (2002), "Vibration-based damage detection of structures by genetic algorithms", *J. Comput. Civil Eng.-ASCE*, **16**(3), 222-229.
Holland, J.H. (1975), *Adaptation in natural and artificial systems*, Ann Arbor: The University of Michigan press.
Herrera, F., Lozana, M. and Sanchez, A.M. (2003), "A taxonomy for the crossover operator for real-coded genetic algorithms: an experimental study", *Int. J. Intell. Syst*., **18**(3), 309-333.
Higuchi, T., Tsitsui, S. and Yamamura, M. (2000), "Theoretical analysis of simplex crossover for real-coded genetic algorithms", Lecture Notes In Computer Science, **1917**, *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, 365-374.
Humar, J., Bagchi, A. and Xu, H. (2006), "Performance of vibration-based techniques for the identification of structural damage", *Struct. Health Monit*., **5**(3), 215-227.
Lee, K.Y. and El-Sharkawi, M.A. (2008), *Modern heuristic techniques*, John Wiley & Sons, Inc. Hoboken, New Jersey.
Levin, L.I. and Lieven, N.A.J. (1998), "Dynamic finite element model updating using simulated annealing and genetic algorithms", *Mech. Syst. Signal Pr*., **12**(1), 91-120.
Li, X.Y. and Law, S.S. (2010), "Consistent regularization for damage detection with noise and model errors", *AIAA J*., **48**(4), 777-787.
Lim, D., Ong, Y.S., Jin, Y., Sendhoff, B. and Lee, B.S. (2007), "Efficient hierarchical parallel genetic algorithms using grid computing", *Future Gener. Comp. Sy*., **23**(4), 658-670.
Makinen, R.A.E., Periaux, J. and Toivanen, J. (1999), "Multidisciplinary shape optimization ın aerodynamics and electromagnetics using genetic algorıthms", *Int. J. Numer. Meth. Fl*., **30**,149-159.
Meruane, V and Heylen, W. (2010), "Damage detection with parallel genetic algorithms and operational modes", *Struct. Health Monit*., **9**(6), 481-496.
Michalewicz, Z. (1996), *Genetic algorithms + data structure =evolution programs*, Springer, New York.
Ono, I. and Kobayashi, S. (1997), "A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover", *Proceedings of 7th ICGA*, 246-253.
Perera, R. and Torres, R. (2005), "Structural damage detection via modal data with genetic algorithms", *J. Struct. Eng*., **132**(9), 1491-1501.
Rao, M.A., Srinivas, J. and Murthy, B.S.N. (2004), "Damage detection in vibrating bodies using genetic algorithm", *Comput. Struct*., **82**(11-12), 963-968.
Teughels, A., Maeck, J. and De Roeck, G. (2002), "Damage assesment by FE model updating using damage functions", *Comput. Struct*., **80**(25), 1869-1879.
Tsutsui, S., Yamamura, M. and Higuchi, T. (1999), "Multi-parent recombination with simplex crossover in real coded genetic algorithms", *Proceedings of the Genetic and Evolutionary Computation Conference (GECC0-1999)*, 657-664, Morgan Kaufmann, San Francisco.

Wright, A. (1991), *Genetic algorithms for real parameter optimization*, G.J.E. Rawlins (Ed.), Foundations of Genetic Algorithms, 205-220.

Voight, H.M., Mühlenbein, H. and Cvetkovic, D. (1995), "Fuzzy recombination for the breeder genetic algorithm", *Proceedings of the 6th Int Conf Genetic Algorithms*, San Mateo, CA, Morgan Kaufmann, 104-111.

CC

Appendix A

Table A1 Pairwise comparisons of the mutation probabilities

Hypothesis	$z = (R_0 - R_i/SE)$	p	α	APV
0.01-0.06	8.8257	0.0000e+000	0.0011	0.0000e+000
0.02-0.06	7.9902	1.3323e-015	0.0011	5.8620e-014
0.01-0.10	7.5724	3.6637e-014	0.0012	1.5754e-012
0.01-0.08	6.8413	7.8504e-012	0.0012	3.2972e-010
0.02-0.10	6.7368	1.6191e-011	0.0012	6.6381e-010
0.03-0.06	6.3712	1.8750e-010	0.0013	7.5001e-009
0.01-0.07	6.2668	3.6855e-010	0.0013	1.4374e-008
0.02-0.08	6.0057	1.9053e-009	0.0013	7.2402e-008
0.05-0.06	5.4312	5.5969e-008	0.0014	2.0709e-006
0.02-0.07	5.4312	5.5969e-008	0.0014	2.0709e-006
0.01-0.04	5.3790	7.4901e-008	0.0014	2.6215e-006
0.03-0.10	5.1179	3.0898e-007	0.0015	1.0505e-005
0.02-0.04	4.5434	5.5347e-006	0.0015	1.8265e-004
0.06-0.09	4.4390	9.0386e-006	0.0016	2.8924e-004
0.01-0.09	4.3868	1.1505e-005	0.0016	3.5667e-004
0.03-0.08	4.3868	1.1505e-005	0.0017	3.5667e-004
0.05-0.10	4.1779	2.9426e-005	0.0017	8.5335e-004
0.03-0.07	3.8123	1.3768e-004	0.0018	3.8550e-003
0.02-0.09	3.5512	3.8350e-004	0.0019	1.0355e-002
0.05-0.08	3.4467	5.6740e-004	0.0019	1.4752e-002
0.04-0.06	3.4467	5.6740e-004	0.0020	1.4752e-002
0.01-0.05	3.3945	6.8750e-004	0.0021	1.6500e-002
0.09-0.10	3.1856	1.4444e-003	0.0022	3.3222e-002
0.03-0.04	2.9245	3.4500e-003	0.0023	7.5901e-002
0.05-0.07	2.8723	4.0752e-003	0.0024	8.5579e-002
0.06-0.07	2.5589	1.0499e-002	0.0025	2.0998e-001
0.02-0.05	2.5589	1.0499e-002	0.0026	2.0998e-001
0.01-0.03	2.4545	1.4108e-002	0.0028	2.5395e-001
0.08-0.09	2.4545	1.4108e-002	0.0029	2.5395e-001
0.04-0.10	2.1934	2.8280e-002	0.0031	4.5248e-001
0.04-0.05	1.9845	4.7202e-002	0.0033	7.0803e-001
0.06-0.08	1.9845	4.7202e-002	0.0036	7.0803e-001
0.03-0.09	1.9323	5.3327e-002	0.0038	7.0803e-001
0.07-0.09	1.8800	6.0103e-002	0.0042	7.2123e-001
0.02-0.03	1.6189	1.0546e-001	0.0045	1.0000e+000
0.04-0.08	1.4623	1.4367e-001	0.0050	1.0000e+000
0.07-0.10	1.3056	1.9169e-001	0.0056	1.0000e+000
0.06-0.10	1.2534	2.1008e-001	0.0063	1.0000e+000
0.04-0.09	0.9922	3.2108e-001	0.0071	1.0000e+000
0.05-0.09	0.9922	3.2108e-001	0.0083	1.0000e+000
0.03-0.05	0.9400	3.4721e-001	0.0100	1.0000e+000
0.04-0.07	0.8878	3.7465e-001	0.0125	1.0000e+000
0.01-0.02	0.8356	4.0340e-001	0.0167	1.0000e+000
0.08-0.10	0.7311	4.6470e-001	0.0250	1.0000e+000
0.07-0.08	0.5745	5.6566e-001	0.0500	1.0000e+000

Table A2 Pairwise comparisons of selection operators

Hypothesis	$z = (R_0 - R_i/SE)$	p	α	APV
RW-TR8	5.2664	1.3912e-007	0.0083	8.3474e-007
RW-TR4	4.6540	3.2551e-006	0.0100	1.6275e-005
RW-TR4	4.4091	1.0381e-005	0.0125	4.1524e-005
TR2-TR4	3.7967	1.4663e-004	0.0167	4.3989e-004
TR4-TR8	0.8573	3.9127e-001	0.0250	7.8253e-001
RW-TR2	0.6124	5.4029e-001	0.0500	7.8253e-001

Table A3 Pairwise comparison of mutation operators

Hypothesis	$z = (R_0 - R_i/SE)$	p	α	APV
PM-0.9-NUM-5	3.1334	1.7280e-003	0.0011	7.7758e-002
PM-0.9-MPTM-16	3.0812	2.0619e-003	0.0011	9.0722e-002
PM-0.9-MPTM-10	2.9245	3.4500e-003	0.0012	1.4835e-001
PM-0.5-NUM-5	2.7417	6.1118e-003	0.0012	2.5669e-001
PM-0.5-MPTM-16	2.6895	7.1559e-003	0.0012	2.9339e-001
PM-0.5-MPTM-10	2.5328	1.1315e-002	0.0013	4.5258e-001
PM-0.9-NUM-8	2.2456	2.4730e-002	0.0013	9.6445e-001
PM-0.9-NUM-3	2.0889	3.6714e-002	0.0013	1.0000e+000
PM-0.9-MPTM-4	2.0889	3.6714e-002	0.0014	1.0000e+000
UM-NUM-5	2.0367	4.1679e-002	0.0014	1.0000e+000
UM-MPTM-16	1.9845	4.7202e-002	0.0014	1.0000e+000
PM-0.5-NUM-8	1.8539	6.3750e-002	0.0015	1.0000e+000
UM-MPTM-10	1.8278	6.7577e-002	0.0015	1.0000e+000
PM-0.1-PM-0.9	1.7495	8.0208e-002	0.0016	1.0000e+000
PM-0.5-NUM-3	1.6973	8.9648e-002	0.0016	1.0000e+000
PM-0.5-MPTM-4	1.6973	8.9648e-002	0.0017	1.0000e+000
PM-0.1-NUM-5	1.3839	1.6638e-001	0.0017	1.0000e+000
PM-0.1-PM-0.5	1.3578	1.7453e-001	0.0018	1.0000e+000
PM-0.1-MPTM-16	1.3317	1.8296e-001	0.0019	1.0000e+000
PM-0.1-MPTM-10	1.1750	2.3999e-001	0.0019	1.0000e+000
UM-NUM-8	1.1489	2.5059e-001	0.0020	1.0000e+000
UM-PM-0.9	1.0967	2.7278e-001	0.0021	1.0000e+000
NUM-3-NUM-5	1.0445	2.9627e-001	0.0022	1.0000e+000
NUM-5-MPTM-4	1.0445	2.9627e-001	0.0023	1.0000e+000
UM-NUM-3	0.9922	3.2108e-001	0.0024	1.0000e+000
UM-MPTM-4	0.9922	3.2108e-001	0.0025	1.0000e+000
NUM-3-MPTM-16	0.9922	3.2108e-001	0.0026	1.0000e+000
MPTM-1-MPTM-10	0.9922	3.2108e-001	0.0028	1.0000e+000
NUM-5-NUM-8	0.8878	3.7465e-001	0.0029	1.0000e+000
NUM-8-MPTM-16	0.8356	4.0340e-001	0.0031	1.0000e+000
NUM-3-MPTM-10	0.8356	4.0340e-001	0.0033	1.0000e+000
MPTM-1-MPTM-4	0.8356	4.0340e-001	0.0036	1.0000e+000
UM-PM-0.5	0.7050	4.8080e-001	0.0038	1.0000e+000
NUM-8-MPTM-10	0.6789	4.9720e-001	0.0042	1.0000e+000
UM-PM-0.1	0.6528	5.1389e-001	0.0045	1.0000e+000
PM-0.1-NUM-8	0.4961	6.1981e-001	0.0050	1.0000e+000
PM-0.5-PM-0.9	0.3917	6.9530e-001	0.0056	1.0000e+000
PM-0.1-NUM-3	0.3395	7.3427e-001	0.0063	1.0000e+000

PM-0.1-MPTM-4	0.3395	7.3427e-001	0.0071	1.0000e+000
NUM-5-MPTM-10	0.2089	8.3453e-001	0.0083	1.0000e+000
MPTM-4-MPTM-10	0.1567	8.7551e-001	0.0100	1.0000e+000
NUM-3-NUM-8	0.1567	8.7551e-001	0.0125	1.0000e+000
NUM-8-MPTM-4	0.1567	8.7551e-001	0.0167	1.0000e+000
NUM-5-MPTM-16	0.0522	9.5835e-001	0.0250	1.0000e+000
NUM-3-MPTM-4	0	1.0000e+000	0.0500	1.0000e+000

Table A4 Pairwise comparison of BLX-a operator

hypothesis	$z = (R_0 - R_i/SE)$	p	α	APV
BLX-0.1- BLX-0.7	6,8458	7.6077e-012	0.0033	1.1412e-010
BLX-0.1- BLX-1	6,0851	1.1641e-009	0.0036	1.6298e-008
BLX-0.1- BLX-1.4	4,9864	6.1512e-007	0.0038	7.9965e-006
BLX-0.3- BLX-0.7	4,8173	1.4546e-006	0.0042	1.7455e-005
BLX-0.3- BLX-1	4,0567	4.9762e-005	0.0045	5.4739e-004
BLX-0.1- BLX-0.5	3,8877	1.0119e-004	0.0050	1.0119e-003
BLX-0.3- BLX-1.4	2,9580	3.0960e-003	0.0056	2.7864e-002
BLX-0.5- BLX-0.7	2,9580	**3.0960e-003**	0.0063	2.7864e-002
BLX-0.5- BLX-1	2,1974	2.7992e-002	0.0071	1.9594e-001
BLX-0.1- BLX-0.3	2,0283	4.2522e-002	0.0083	2.5513e-001
BLX-0.3- BLX-0.5	1,8593	6.2979e-002	0.0100	3.1490e-001
BLX-0.7- BLX-1.4	1,8593	6.2979e-002	0.0125	3.1490e-001
BLX-0.5- BLX-1.4	1,0987	2.7190e-001	0.0167	8.1570e-001
BLX-1- BLX-1.4	1,0987	2.7190e-001	0.0250	8.1570e-001
BLX-0.7- BLX-1	0,7606	4.4687e-001	0.0500	8.1570e-001

Tablo A5 Pairwise comparison of PNX operator

Hypothesis	$z = (R_0 - R_i/SE)$	p	α	APV
PNX-1-PNX-5	4.6540	3.2551e-006	0.0083	1.9531e-005
PNX-2-PNX-5	3.9192	8.8849e-005	0.0100	4.4425e-004
PNX-1-PNX-0.5	3.9192	8.8849e-005	0.0125	4.4425e-004
PNX-2-PNX-0.5	3.1843	1.4509e-003	0.0167	4.3526e-003
PNX-5-PNX-0.5	0.7348	4.6243e-001	0.0250	9.2487e-001
PNX-1-PNX-2	0.7348	4.6243e-001	0.0500	9.2487e-001

Table A6 Pairwise comparison of SBX operator

Hypothesis	$z = (R_0 - R_i/SE)$	p	α	APV
SBX-0.04-SBX-1	3.2000	1.3743e-003	0.0050	1.3743e-002
SBX-0.04-SBX-0.1	2.7000	6.9339e-003	0.0056	6.2406e-002
SBX-1-SBX-2	2.5500	1.0772e-002	0.0063	8.6178e-002
SBX-0.1-SBX-2	2.0500	4.0364e-002	0.0071	2.8255e-001
SBX-0.04-SBX-0.7	1.9500	5.1176e-002	0.0083	3.0706e-001
SBX-0.7-SBX-2	1.3000	1.9360e-001	0.0100	9.6800e-001
SBX-0.7-SBX-1	1.2500	2.1130e-001	0.0125	9.6800e-001
SBX-0.1-SBX-0.7	0.7500	4.5325e-001	0.0167	1.0000e+000
SBX-0.04-SBX-2	0.6500	5.1569e-001	0.0250	1.0000e+000
SBX-0.1-SBX-1	0.5000	6.1708e-001	0.0500	1.0000e+000

Tablo A6 Pairwise comparison of FRX operator

Hypothesis	$z = (R_0 - R_i/SE)$	p	α	APV
FRX-0.3-FRX-0.9	3.0042	2.6631e-003	0.0167	7.9894e-003
FRX-0.5-FRX-0.9	2.6879	7.1895e-003	0.0250	1.4379e-002
FRX-0.3-FRX-0.5	0.3162	7.5183e-001	0.0500	7.5183e-001

Tablo A7 Pairwise comparison of HX operator

Hypothesis	$z = (R_0 - R_i/SE)$	p	α	APV
HX-1.2-HX-1.6	2.5298	1.1412e-002	0.0167	3.4236e-002
HX-1.2-HX-2	2.2136	2.6857e-002	0.0250	5.3713e-002
HX-2-HX-1.6	0.3162	7.5183e-001	0.0500	7.5183e-001

Table A8 Pairwise comparison of best crossover operators

Hypothesis	$z = (R_0 - R_i/SE)$	p	α	APV
UNDX-PNX-1	7.1005	1.2432e-012	0.0018	3.4810e-011
UNDX-BLX-0.7	6.7132	1.9044e-011	0.0019	5.1418e-010
UNDX-SBX-1	6.2613	3.8172e-010	0.0019	9.9248e-009
SPX-PNX-1	5.6158	1.9563e-008	0.0020	4.8906e-007
UNDX-LX	5.3576	8.4322e-008	0.0021	2.0237e-006
SPX-BLX-0.7	5.2285	1.7087e-007	0.0022	3.9299e-006
SPX-SBX-1	4.7767	1.7821e-006	0.0023	3.9207e-005
UNDX-SX-1.6	4.5185	6.2285e-006	0.0024	1.3080e-004
UNDX-FRX-0.9	4.1957	2.7199e-005	0.0025	5.4398e-004
SPX-LX	3.8730	1.0751e-004	0.0026	2.0427e-003
SPX-SX-1.6	3.0338	2.4146e-003	0.0028	4.3464e-002
PNX-1-FRX-0.9	2.9047	3.6756e-003	0.0029	6.2485e-002
SPX-FRX-0.9	2.7111	6.7063e-003	0.0031	1.0730e-001
SX-1.6-PNX-1	2.5820	9.8233e-003	0.0033	1.4735e-001
BLX-0.7-FRX-0.9	2.5174	1.1821e-002	0.0036	1.6550e-001
SX-1.6-BLX-0.7	2.1947	2.8186e-002	0.0038	3.6642e-001
SBX-1-FRX-0.9	2.0656	3.8867e-002	0.0042	4.6641e-001
SBX-1-SX-1.6	1.7428	8.1361e-002	0.0045	8.9497e-001
LX-PNX-1	1.7428	8.1361e-002	0.0050	8.9497e-001
UNDX-SPX	1.4846	1.3764e-001	0.0056	1.0000e+000
LX-BLX-0.7	1.3555	1.7524e-001	0.0063	1.0000e+000
LX-FRX-0.9	1.1619	2.4528e-001	0.0071	1.0000e+000
SBX-1-LX	0.9037	3.6616e-001	0.0083	1.0000e+000
LX-SX-1.6	0.8391	4.0139e-001	0.0100	1.0000e+000
SBX-1-PNX-1	0.8391	4.0139e-001	0.0125	1.0000e+000
SBX-1-BLX-0.7	0.4518	6.5138e-001	0.0167	1.0000e+000
BLX-0.7-PNX-1	0.3873	6.9854e-001	0.0250	1.0000e+000
SX1.6-FRX-0.9	0.3227	7.4689e-001	0.0500	1.0000e+000

Table A9 Pairwise comparison of multiple crossover operator

Hypothesis	$z = (R_0 - R_i/SE)$	p	α	APV
7-8	4.6108	4.0115e-006	0.0018	1.1232e-004
6-8	4.4263	9.5862e-006	0.0019	2.5883e-004
5-7	4.1189	3.8066e-005	0.0019	9.8971e-004
5-6	3.9344	8.3393e-005	0.0020	2.0848e-003
1-7	3.5041	4.5807e-004	0.0021	1.0994e-002
1-6	3.3197	9.0126e-004	0.0022	2.0729e-002
3-8	3.2583	1.1207e-003	0.0023	2.4654e-002
2-8	2.8894	3.8601e-003	0.0024	8.1062e-002
3-5	2.7665	5.6666e-003	0.0025	1.1333e-001
4-8	2.6436	8.2037e-003	0.0026	1.5587e-001
2-5	2.3975	1.6507e-002	0.0028	2.9713e-001
1-3	2.1517	3.1421e-002	0.0029	5.3416e-001
4-5	2.1517	3.1421e-002	0.0031	5.3416e-001
4-7	1.9672	4.9158e-002	0.0033	7.3737e-001
4-6	1.7827	7.4630e-002	0.0036	1.0000e+000
1-2	1.7827	7.4630e-002	0.0038	1.0000e+000
2-7	1.7214	8.5176e-002	0.0042	1.0000e+000
2-6	1.5369	1.2431e-001	0.0045	1.0000e+000
1-4	1.5369	1.2431e-001	0.0050	1.0000e+000
3-7	1.3524	1.7623e-001	0.0056	1.0000e+000
3-6	1.1680	2.4282e-001	0.0063	1.0000e+000
1-8	1.1066	2.6845e-001	0.0071	1.0000e+000
1-5	0.6148	5.3871e-001	0.0083	1.0000e+000
3-4	0.6148	5.3871e-001	0.0100	1.0000e+000
5-8	0.4919	6.2281e-001	0.0125	1.0000e+000
2-3	0.3690	7.1215e-001	0.0167	1.0000e+000
2-4	0.2458	8.0583e-001	0.0250	1.0000e+000
6-7	0.1845	8.5363e-001	0.0500	1.0000e+000