Computers and Concrete, *Vol. 22, No. 5 (2018) 493-500*
DOI: https://doi.org/10.12989/cac.2018.22.5.493

Developing a new mutation operator to solve the RC deep beam problems by aid of genetic algorithm

Mustafa Kaya*

Faculty of Engineering, Aksaray University, Aksaray, Turkey

(Received June 23, 2017, Revised November 14, 2018, Accepted November 29, 2018)

Abstract. Due to the fact that the ratio of their height to their openings is very large compared to normal beams, there are difficulties in the design and analysis of deep beams, which differ in behavior. In this study, the optimum horizontal and vertical reinforcement diameters of 5 different beams were determined by using genetic algorithms (GA) due to the openness/height ratio (*L/h*), loading condition and the presence of spaces in the body. In this study, the effect of different mutation operators and improved double times sensitive mutation (DTM) operator on GA's performance was investigated. In the study following random mutation (RM), boundary mutation (BM), non-uniform random mutation (NRM), Makinen, Periaux and Toivanen (MPT) mutation, power mutation (PM), polynomial mutation (PNM), and developed DTM mutation operators were applied to five deep beam problems were used to determine the minimum reinforcement diameter. The fitness values obtained using developed DTM mutation operator was higher than obtained from existing mutation operators. Moreover; obtained reinforcement weight of the deep beams using the developed DTM mutation operator lower than obtained from the existing mutation operators. As a result of the analyzes, the highest fitness value was obtained from the applied double times sensitive mutation (DTM) operator. In addition, it was found that this study, which was carried out using GAs, contributed to the solution of the problems experienced in the design of deep beams.

Keywords: evolutionary algorithms; artificial intelligence; genetic algorithms; mutation operator; deep beam

1. Introduction

Nilson (1997), which is different from classical beams due to the fact that the deep beams used in cutting walls, folded roof plaques, quay walls and silos have more than normal clearances compared to their openings. For these beams, the hypothesis that the sections with planes will remain in the plane after bending is not valid. Because of the high shear stress, the sections are significantly multiplied. Therefore, there are problems in the analysis of deep beams. According to ACI 318-99; ratio of the height of the beams to continuous beams 5/2, in simple beams; beams less than 5/4 are considered to be deep beams. In cutting account; The height of the beam opening is less than 5 and the beams loaded from the upper edge are considered as deep beams. Simple bearing beams loaded on the body height or on the lower edge are not considered to be deep beams. In addition, there is no explanation for ACI 318-99 for beams with cavities in the body. These elements should be designed as normal beams ACI 318-99 (1999).

Genetic algorithms (GAs) came into existence with the adaptation of biological processes developed for the computer environment. They use units stored in the computer memory in the same way as those used in natural populations. The initial population is used for the solution of optimization problems with a GA. It is preferred to start the application of a GA to a problem using an initial population because this lowers the risk of being caught in local optimum traps; it utilizes the limited objective functions; enables the exploration of new combinations with high the fitness values in each generation; it is applicable to problems in which the design variables are discontinuous and there is no need for the derivatives of the objective functions Kaya (2011). The first study about the principles of GA was by Holland (1975), under the title of Machine Learning. Influenced by Holland's work, Goldberg (1989) carried out research into the monitoring of gas pipelines and showed that there are practical uses of GA in the field. GA studies conducted in engineering dealt with topology, shape, and dimension optimizations. Castilho *et al* (2007) described the use of a modified GA as an optimization operator in structural engineering to minimize the production costs of slabs using precast pre-stressed concrete joists. Govindaraj and Ramasamy (2005), presented an application of GAs for the optimum detailed design of reinforced concrete continuous beams based on Indian Standard specifications. Sahab *et al*. (1999), presented a hybrid optimization algorithm based on a modified GA. Morover, Daoud and Kharma (2011), Gupta *et al*. (2010), Hwang *et al*. (2006), Kołodziej and Khan (2012), Lu *et al*. (2013), Omara and Arafa (2010), Rahmani and Vahedi (2008), Sathappan *et al*. (2011), Singh and Singh (2012), Zomaya *et al*. (1999) studied on genetic algorithm.

In their study Albayrak and Allahverdi (2011), presents a new greedy mutation operator, called Greedy Sub Tour Mutation (GSTM). To prove the quality of the proposed

*Corresponding author, Professor
E-mail: kaya261174@hotmail.com

ISSN: 1598-8198 (Print), 1598-818X (Online)

Table 1 Reinforcement diameters and codes

Diameter (mm)	Ø 10	Ø 12	Ø 14	Ø 16	Ø 18	Ø 20	Ø 22	Ø 24
Code	1	2	3	4	5	6	7	8

operator, the authors compared this developed mutation operator with benchmark mutation operators. The benchmark used for the tests is the TSPLIB, Reinelt (1991), and the mutation operators used are Exchange Mutation (EXC), Displacement Mutation (DIPS), Inversion Mutation (INV), Insertion Mutation (INS), Simple Inversion Mutation (SIM), Scramble Mutation (SCM) and Greedy Swap Mutation (GSM), all of them cited in Albayrak and Allahverdi (2011).

2. Stages of the research

2.1 Coding

The most important feature that distinguishes GAs from other operators is the use of codes that represent the variables, instead of the variables themselves. The first stage in applying a GA to any problem is to determine the most suitable type of coding for the respective problem. In this study, permutation coding was applied to the problems (Table 1).

2.2 Formation of the initial population

A GA realizes the search within a population formed by the points. While this population is being formed, attention should be paid to the dissimilarity of the individuals, which is why the individuals should be randomly formed. In this study, the initial population is formed of 100 dissimilar individuals.

2.3 Evaluation

The fitness values of the individuals within the population were calculated in each generation. These fitness values play a role in determining which individuals from the current population need to be used to obtain the new individuals to form the next population. The evaluation function that is used in the GA is the objective function of the problem. However, since the objective function was limited to the problem under obstacles, it was changed into an unlimited objective function using two transformations.

2.4 Selection

In each generation, the individuals of the new population are selected from the individuals of the current population. This operator artificially performs the natural selection. In the study, the serial selection was used, and the individuals were ordered using a linearly decreasing function. After the ordering process, the individuals with lower fitness values were discarded in a certain proportion, and individuals with higher fitness values were copied into the population to replace the discarded individuals.

2.5 Crossover

A crossover is an operator, which maintains the information exchange between different solutions, and thereby maintains the access of the research to undiscovered regions. Different individuals are obtained by interchanging certain parts of the two individuals that are randomly selected from the population using the crossover operator. In this study, a random mixed crossover operator was used.

2.6 Mutation

When examining a limited population, some of the genetic information is likely to be lost earlier over time. In the advancing generations, all the genes forming the chromosome may be the same, and it is not possible to change such a chromosome using a crossover operator. In such cases, for the individuals externally forming the population at a certain ratio, the code of these individuals can be changed. In this study, mutation operators, namely RM, BM, NURM, MPT, PM, PNM, and DTM operators were applied to the population for the analyses of all beams.

2.6.1 Random mutation (RM)

The RM used in Genetic Algorithms (GAs) is one of the classic schemes used, in which the value of a gene of the decision variable is replaced with a value generated randomly from its domain. The random consideration process in harmony (HS) search algorithm is defined by Geem *et al*. and it uses the same concept of RM. The decision variable to be replaced is randomly sampled from any solution stored in HM.

2.6.2 Boundary mutation (BM)

The BM is a scheme used in GA. In this type of mutation, the value of the decision variable in the BM is replaced randomly with its upper or lower bound (i.e., *xi* or *Uxi*).

2.6.3 Nonuniform random mutation (NURM)

The NURM increases the probability that the amount of the mutation will be close to 0 as the generation number increases. It helps the genetic algorithm in avoiding local optima in the early stages of the evolution then it helps the genetic algorithm to fine tune the solution in the final stages of evolution. This mutation can only be used for integer and float genes. This mutation scheme aimed at both improving single element tuning and reducing the disadvantage of RM.

If $stv=(v1, v2, v3\ldots\ldots vm)$ is a chromosome (t is the generation number) and the element vk was selected for this mutation, the result is a vector

$$s^t_v+1= (v_1, v_k^*\ldots.\ v_m) \quad \text{where} \tag{1}$$

$$v_k^* = v_k+ (x\ (t,\ U_{xi}\text{-}v_k)\ \text{if a ran} \tag{2}$$

$$v_k^* = v_k\text{-}\Delta\ x\ (t,\ v_k\text{-}L_{xi})\ \text{if a random} \tag{3}$$

L_{xi} and U_{xi} are lower and upper bounds for variable v_k. (4)

The function $\Delta(t, y)$ returns a value in the range $[0, y]$ such that the probability of $\Delta(t, y)$ being close to 0 increases as t increases. This property causes this operator to search the space uniformly in the initial generations (when t is small), and closer to the local values at later generations. In order to increase the probability of generating the new number closer to the previously generated values rather than a random choice.

2.6.4 Makinen, and Toivonen mutation (MPT)

The MPT mutation operator given a chromosome x= (x_1; x_2;...; x_{ian}) the resulting chromosome y=(y_1; y_2;...; y_n) is generated as follows: for each gene xi generate a uniform random number RI [0,1] yi=(1-ti) x_i^{lower} + x_i^{upper}

Where

$$\mathrm{t_i} = \mathrm{t_i} - \mathrm{t_i}\left(\frac{\mathrm{t_i - r_i}}{\mathrm{t_i}}\right).^{\mathrm{b}} \qquad r_i<t_i \tag{5}$$

$$\mathrm{t_i} \qquad ri=ti \tag{6}$$

$$\mathrm{t_i} = \mathrm{t_i} + (1-\mathrm{t_i})\mathrm{t_i}\left(\frac{\mathrm{r_i - t_i}}{1-\mathrm{t_i}}\right).^{\mathrm{b}} \qquad r_i>t_i \tag{7}$$

And

$$\mathrm{t_i} = \frac{\mathrm{x_{i-}x_i^{lower}}}{\mathrm{x_i^{upper}-x_i}} \tag{8}$$

Where; x_i^{lower} and x_i^{upper} are the lower, and upper bounds of x_i.

The strength of the MPT mutation operator does not decrease as the generation number increases as it is the case in the non-uniform mutation operator (Deep and Thakur 2007).

2.6.5 Power mutation (PM)

The PM was proposed by Deep and Thankur (2007). The PM is based on power distribution, and its distribution function is given by

$$f(x)= px^{p-1} \qquad 0 \le x \le 1 \tag{9}$$

$$f(x)= x^{p} \qquad 0 \le x \le 1 \tag{10}$$

P is the index of the distribution.

Given an existing solution x, PM generates a new solution y as follows:

A uniform random number RI [0,1] was generated.

A random numbers were created according to the final distribution.

t was calculated as follows

$$\mathrm{t} = \frac{\mathrm{x_{-x}lower}}{\mathrm{x^{upper}-x}} \tag{11}$$

Where; x^{lower} and x^{upper} are the lower and upper bounds of x. The following formula is used to create the new solution y

$$y=x-s(x-x^{lower}t<r) \tag{12}$$

$$y=x+s(x^{upper-}x) \text{ otherwise} \tag{13}$$

2.6.6 Polynomial mutation (PNM)

In the PNM operator, n is the number of genes in the chromosome, Pm is the probability of mutation, and ηm is a non-negative value representing the distributed index. Each gene xi has an upper bound x_i^{upper} and a lower bound x_i^{lower} . The PNM works for each gene xi as follows

In the PNM operator, n is the number of genes in the chromosome, Pm is the probability of mutation, and ηm is a non-negative value representing the distributed index. Each gene xi has an upper bound x_i^{upper}, and a lower bound x_i^{lower}. The PNM works for each gene xi as follows:

1. A random value r is generated
2. If (r < Pm) then the gene will be mutated as follows

(a) Find the difference between the gene and both of its boundaries

(b) Divide the smallest difference by by x_i^{upper} x_i^{lower}

(c) Another random value r is generated

(d) The procedure sample to the left-hand side of the gene if $r \le 0.5$; otherwise its samples to the right-hand side

(e) The new value of xi is calculated where n is the number of genes in the chromosome, Pm is the probability of mutation and the gene is a non-negative value representing the distributed index. Each gene xi has a bound x_i^{upper} and a lower bound x_i^{lower}.

A major problem in this mutation appears when the value of the gene to be mutated is close to one of its boundaries. Namely, the algorithm may get trapped in a local optimum.

2.6.7 Double times sensitive mutation (DTM)

In the existing mutation operators, previously selected mutation operators were applied to the same sites of the chromosomes of the population.

The developed DTM operator differs from the existing mutation operators in that this operator was applied to the randomly selected members' randomly selected site of the population. In the developed mutation operator process applied in two steps every generation. In the first step, a member randomly selected between members in the population. In the second step, a site randomly selected between sites in the members. In this operator mutation process applied to the population double times sensitive than existing mutation operators, but between all the mutation operators developed double times sensitive mutation operator (DTM) used problems analysis completed longer time than existing mutation operators used problems analysis.

3. Applications

In this study, a new mutation operator in Genetic Algorithm is proposed; DTM operator. In the study, existing and developed the new mutation operator was applied to the five deep beam problems.

3.1 Deep beams

Deep beams used in the shear walls, folded roof plates, quay walls, and silos possess more than normal heights in proportion to their spaces; therefore, they exhibit different behaviors when compared with classical beam. The hypothesis suggesting that the sections which are level before bending, remain level after bending is not applicable to deep beams. Due to high shear stress, sections become significantly distorted, therefore problems arise in analyzing

Table 2 ACI 318-99 conditions of deep beams

	(L/h)	Load location		Analysis			
				Bending		Shear	
		Con.	Result	Con.	Result	Con.	Result
1	1.1	Upper	DB.	1.14<5/4	DB	1.1<5	DB
2	4	Upper	DB	4>5/4	Don't DB	4<5	DB
3	5.3	Upper	DB.	5.7>5/4	Don't DB	5.3>5	Don't DB
4	4	This beam loading from its below edge can't accept a deep beam					
5	4	This beam has a space on its body can't accept a deep beam					

Note: All length's and forces units are mm, and KN respectively

Note: 5/4 value is ACI 318-99 criteria for simple span deep beam

DB: Deep beam, Con: Beam conditions

Note: All length's and forces units are mm, and KN respectively

Note: 5/4 value is ACI 318-99 criteria for simple span deep beam

the deep beams. In the bending computations according to ACI 318-99; of the continuous beams, those with a beam space/beam height ratio of less than 5/2, and of the simple beams, those with beam space/beam height ratio less than 5/4 are considered to be deep beams. In the shear computation; beams with a beam space/beam height ratio less than 5, and those loaded from their upper edges, are considered to be deep beams. A simple supported beams which are loaded along their body height or from their lower edges are not considered to be deep beams. Since there is no explanation in ACI 318-99 with regard to the beams with space in their bodies Nilson ADD (1997), these elements should be considered normal beams. The deep beams dimensions were given in Table 2.

DB: Deep beam, Con: Beam conditions

As usual, the design basis is that

$$V_u = \phi.(V_n) \tag{14}$$

Where ϕ=0.85 for shear, and

$$V_n = V_c + V_s \tag{15}$$

Regardless of the amount of reinforcement provided, the nominal strength is not to be taken greater than the following

$$l_n / d < 2 \tag{16}$$

$$2 \le l_n / d < 5\,2 \quad V_n = \frac{2}{3}.(10 + l_n / d).b_w.d.\left(f_c^{'}\right)^{0.5} \tag{17}$$

$$V_c = \left(3.5 - 2.5\frac{M_u}{(V_u.d)}\right).\left(1.9.f_c^{'}\right)^{0.5} + 2500.\rho_w.\left(\frac{V_u.d}{M_u}\right).b_w.d \tag{18}$$

When the shear force *Vu* at factored loads exceeds the design shear strength of the concrete, shear reinforcement has to carry the excess shear. The contribution of the web steel *Vs* is to be calculated from Eq. (19).

$$V_s = \left[\frac{A_v}{s}\left(\frac{1}{12} + \frac{l_n}{12.d}\right) + \frac{A_{vh}}{s_2}\left(\frac{11}{12} - \frac{l_n}{12.d}\right)\right] f_y.d \tag{19}$$

In the first stage, the finite element models of all beams were created. The beams were divided into a total of 9 shell elements and then analysed according to their load conditions, and the effect of the forces in the vertical, and horizontal directions. Among these forces, compressive forces were supposed to be carried by the concrete, tensile forces by the reinforcements. For all beams, the horizontal and vertical reinforcement spaces were taken to be 20 cm, and the beam thickness was taken as 50 cm. In the models, *fck*=30 MPa concrete and reinforcement *fyk*=420 Mpa reinforcement were used.

After the linear analysis via a finite-element based program on the first stage, the horizontal and vertical reinforcement diameters were determined via GA in the second stage, and the steps in this process are given below.

1. Constructing the initial population
2. Decoding the permutation coding for the design variables of each member and finding their sequence numbers in the variable list.
3. Unconstrained function values for each member calculated.
4. Calculating the fitness value for each member.
5. Sequential selection operator applied to the population. Copying the members into the mating pool according to their fitness, and coupling them randomly. Copying the members into the mating pool according to their fitness, and coupling them randomly.
6. Crossover operator was used while obtaining new generation from population.
7. Applying the developed DTM operator to the population.
8. Replacing the initial population by the new population and repeating steps 3 to 8 until the termination criteria were fulfilled. The chromosome length in this study was 18. While the codes in the first 9 sites within the chromosomes represent the diameter of the horizontal reinforcements, the codes in the other 9 sites represent the diameter of the vertical reinforcements.

The chromosome length in this study was 18. While the codes in the first 9 sites within the chromosomes represent the diameter of the horizontal reinforcements, the codes in the other 9 sites account for the diameter of the vertical reinforcements.

The reinforcement diameters used in this study range between Ø10~Ø24 and the corresponding codes range between 1 and 8. These codes are used in the formation of the individuals within the initial population.

In the study, 25 individuals with the lowest fitness values were discarded and replaced by 25 individuals with the highest fitness values that were copied into the population.

In the analyses, different mutation operators were applied to five different beams, completion periods of the analyses, obtained fitness values, and reinforcement weights at the beams were compared. The reinforcement diameters determined for the beams are given in Fig. 1-Fig. 5.

For the deep beams; the objective function, the penalty function, and the constrained objective function are given in Eqs. (21), (22), and (23), respectively.

$$\min W(x) = \sum_{s=1}^{C_s} A_s.L_s.\rho_s \tag{20}$$

Where *As* is the reinforcement area, the specific gravity of reinforcement, and is the reinforcement length.

The constrained objective function was transformed into an unconstrained objective function as shown in Eq. (7). Total fault "*C*" expressed in Eq. (24). Also, objective function expressed in Eq. (33).

$$g_i(x)=\left(\frac{\rho_s.L_s.A_s.n_s.f_{yd}}{Fe_i}\right)-1 \tag{21}$$

$$\text{If} \qquad g_i(x)\geq 0 \qquad c_i=g_i(x) \tag{22}$$

$$\text{If} \qquad g_i(x)<0 \qquad c_i=g_i(x).100 \tag{23}$$

$$C=\sum c_i \tag{24}$$

$$\phi(s)=W(x)(1+KC) \tag{25}$$

K: a coefficient selected for the problem taken to be 10 in this study.

: the negligence coefficient.

C: The total fault

In the first transformation, the constrained objective function was transformed into an unconstrained objective function as expressed in Eq. (26).

$$\phi(x)=\sum \phi(s)/\phi(s)_{\max} \tag{26}$$

Where Fei is the required force met by reinforcements in the x or y-direction in the ith zone,

In the second transformation, the unconstrained objective function was transformed to a fitness function *F*(*s*). The fitness values of the members were calculated according to Eq. (27).

$$F(s)=\phi(x)_{\max}-\phi(x) \tag{27}$$

3.1.1 Deep beam 1

Deep beam 1 was loaded from its upper edge, and it is accepted as deep beam since the bending and shear computations for the *L/h* ratio (*L/h*=1.14) is less than 5/4 and 5 (Fig. 1).

The results of the analyses conducted for this beam showed that the DTM operator was completed in the longest period, but the analysis using RM operator was completed in the shortest period (Table 3). The analysis completed in the longest period was 16% longer than the analysis completed in the shortest period. Regarding the fitness values of the same problem, the highest fitness value was obtained from the DTM operator, and the lowest fitness value was obtained from the RM operator (Table 4). The maximum fitness value was found 20% greater than the minimum value. According to the total reinforcement diameter of this beam, the lowest reinforcement weight was obtained from the PNM operator, and the heaviest reinforcement weight was obtained from the RM operator (Table 5). The lowest reinforcement weight 11% lighter than the heaviest reinforcement weight.

3.1.2 Deep beam 2

Deep beam 2 was loaded from its upper edge. Even though this beam is not accepted as a deep beam since the bending computations for the *L/h* ratio (*L/h*=4) are more than 5/4, but less than 5, it is considered a deep beam in the shear computation (Fig. 2).

The analysis results showed that the PNM operator completed in the longest period, but the analysis using MPT operator was completed in the shortest period (Table 3). Also, the analysis completed in the longest period was 13% longer than the analysis completed in the shortest period. The highest fitness value was obtained from the DTM operator, and the lowest fitness value was obtained from the PM operator (Table 4). The maximum fitness value was found 17% greater than the minimum fitness value. According to the total reinforcement diameter of the deep beam 2, the lowest reinforcement weight was obtained from

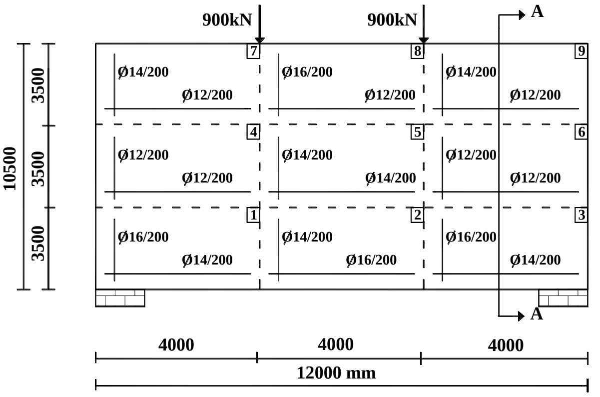

Fig. 1 The fittest reinforcement diameters of deep beam 1

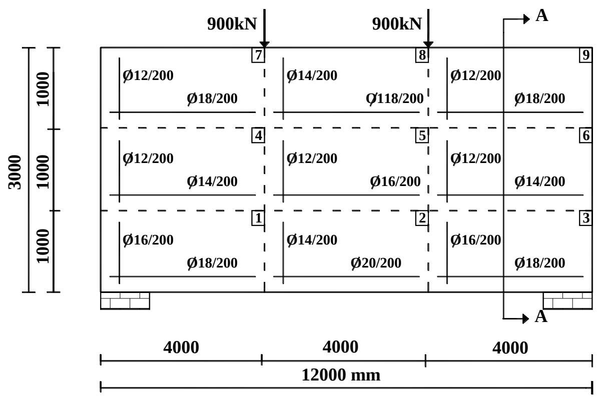

Fig. 2 The fittest reinforcement diameters of deep beam 2

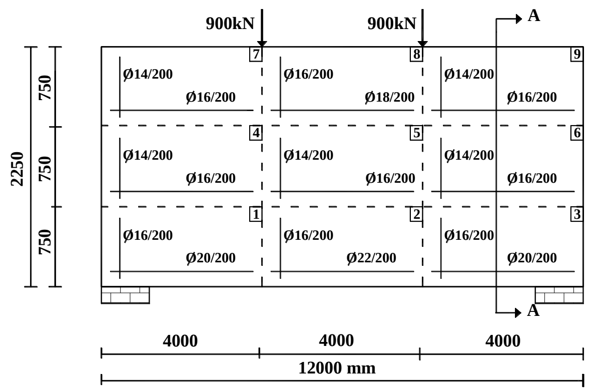

Fig. 3 The fittest reinforcement diameters of deep beam 3

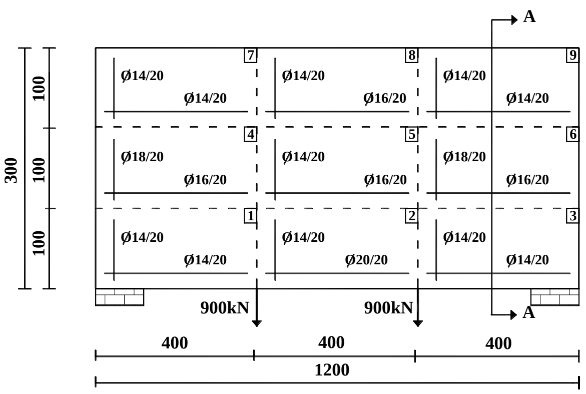

Fig. 4 The fittest reinforcement diameters of deep beam 4

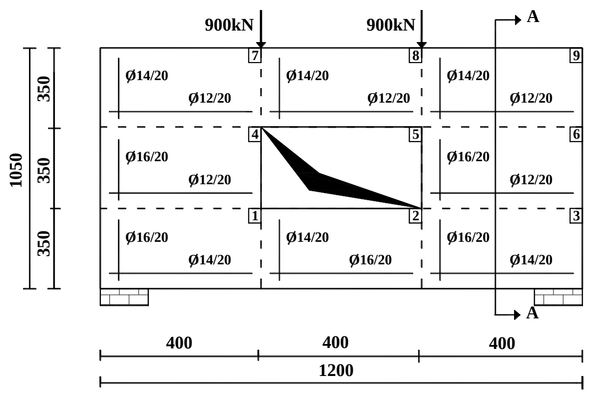

Fig. 5 The fittest reinforcement diameters of deep beam 5

Table 3 The computation times of analysis

No	RM	BM	NRM	MPT	PM	PNM	DTM
1	213	217	221	226	237	243	249
2	227	242	231	218	239	247	243
3	187	207	191	194	219	217	224
4	191	190	215	204	199	187	222
5	193	225	221	201	241	207	237

Table 4 The maximum fitness values of analysis

No	RM	BM	NRM	MPT	PM	PNM	DTM
1	0,69	0,82	0,78	0,76	0,63	0,71	0,83
2	0,68	0,70	0,69	0,71	0,64	0,72	0,74
3	0,75	0,77	0,78	0,76	0,75	0,82	0,81
4	0,84	0,93	0,87	0,95	0,89	0,83	0,85
5	0,71	0,77	0,67	0,69	0,72	0,79	0,73

Table 5 Deep beams reinforcement weight (kN)

No	RM	BM	NRM	MPT	PM	PNM	DTM
1	14215	13780	14025	14210	13472	13763	12615
2	4957	5135	4981	4973	4987	5173	4931
3	4578	4530	4503	4494	4613	4032	4553
4	4896	4840	4781	4910	4932	4765	4712
5	9180	9410	9168	9352	9393	9178	9463

the DTM operator, and the heaviest reinforcement weight was obtained from the BM operator (Table 5). The lowest reinforcement weight 4% lighter than the heaviest reinforcement weight.

3.1.3 Deep beam 3

Deep beam 3 did not accept as a deep beam in the bending, and shear computations because of its the *L/h* ratio (*L/h*=5.33) (Fig. 3).

The results of the analyses conducted for deep beam 3 showed that the DTM operator completed in the longest period, yet the analysis using RM operator was completed in the shortest period (Table 3). Besides, the analysis completed in the longest period was 19% longer than the analysis completed in the shortest period. For deep beam 3, the highest fitness value was achieved by the application of the PNM operator, and the lowest fitness value was achieved by the application of the RM operator (Table 4). There was a 9% difference between the maximum and minimum fitness values. According to the total reinforcement diameter of this deep beam, the lowest reinforcement weight was obtained from the PNM operator, and the heaviest reinforcement weight was obtained from the RM operator (Table 5). The lowest reinforcement weight 11% lighter than the heaviest reinforcement weight.

3.1.4 Deep beam 4

Deep beam 4 was loaded from its lower edge and is not accepted as a deep beam in both the bending and shear computations, disregarding the beam dimensions (Fig. 4).

The results of the analyses conducted for deep beam 4 showed that the DTM operator continued the longest period, but the analysis using BM operator was continued the shortest period (Table 3). The analysis continued the longest period was 17% longer than the analysis completed in the shortest period. Using the same beam, the highest fitness value was obtained from the analysis obtained from the PM operator, whereas the lowest fitness value was obtained from the analysis of the RM mutation operator (Table 4). The maximum fitness value obtained from these analyses was 5% greater than the minimum fitness value. According to the total reinforcement diameter of the deep beam 4, the lowest reinforcement weight was obtained from the DTM operator, and the heaviest reinforcement weight was obtained from the MPT operator (Table 5). The lowest reinforcement weight 4% lighter than the heaviest reinforcement weight.

3.1.5 Deep beam 5

There is no explanation in ACI 318-99 concerning beams with space in their bodies; therefore, this beam is accepted as normal in both the bending and shear computations (Fig. 5).

In was seen that the RM operator applied application was completed in the shortest period, but the PM operator applied analysis completed in the longest period (Table 3). Besides, PM operator applied application analysis took 24 % longer than the RM operator applied application analysis. Regarding the fitness values of the same beam, the highest fitness value was obtained from the PNM operator, and the lowest value was obtained from the NURM operator (Table 4). The maximum fitness value obtained from these analyses was 18% greater than the minimum fitness value.

According to the total reinforcement diameter of this beam, the lowest reinforcement weight was obtained from the PNM operator, and the heaviest reinforcement weight was obtained from the RM operator (Table 5). The lowest reinforcement weight 16% lighter than the heaviest reinforcement weight.

4. Conclusions

In this study performance of developing double times mutation operator compared with existing mutation operators. While comparing the performance of developed and existing mutation operators five deep beams, which differed from each other regarding their *L/h* ratio, load conditions, and due to the existence of space in the body of the beam, were used. In the analysis completion period, the fitness value, and beams total reinforcement weights were compared.

In the analysis of deep beam 1, the longest completion period was obtained from the DTM mutation operator. In the same analysis, and the highest fitness value was obtained from this mutation operator. The highest fitness value obtained DTM mutation operator used analysis was completed in the longest period. According to the total reinforcement weight of this beam, the lowest reinforcement weight was obtained from the same operator.

In the analysis of deep beam 2, the longest completion period was obtained from the PNM mutation operator. In the same analysis, the highest fitness value was obtained from DTM mutation operator. DSSM operator used analysis was completed in the longest period, but the highest fitness value could not obtain from this operator. The maximum the fitness value obtained from the PNM mutation operator. According to the total reinforcement weight of this beam, the lowest reinforcement weight was obtained from the DTM operator.

In the analysis of deep beam 3, the longest completion period was obtained from the DTM mutation operator. In the same analysis, the highest fitness value was obtained from PNM mutation operator. The highest fitness value obtained PNM mutation operator used analysis, and DTM operator used analysis was completed in the longest period. According to the total reinforcement weight of this beam, the lowest reinforcement weight was obtained from the PNM operator.

In the analysis of deep beam 4, the longest completion period was obtained from the DTM mutation operator used analysis. In the same analysis, the highest fitness value was obtained from the MPT mutation operator. The DTM mutation operator used analysis was completed in the longest period, but the maximum fitness value could not obtain from this operator. The highest the fitness value obtained from MPT mutation operator used analysis. According to the total reinforcement weight of this beam, the lowest reinforcement weight was obtained from the DTM operator.

In the analysis of deep beam 5, the longest completion period was obtained from the DTM mutation operator. In the same analysis, the highest fitness value was obtained from PNM mutation operator used analysis. DTM operator used analysis was completed in the longest period, but the highest fitness value could not obtain from this operator. The maximum fitness value obtained from PNM mutation operator used analysis. According to the total reinforcement weight of this beam, the lowest reinforcement weight was obtained from the PNM operator.

The comparison of the results showed that when the DTM operator was applied, the GA was able to undertake searches in the widest dimensions within the design space, and found new individuals, which were likely to be optimal solutions. In parallel with the increased the fitness values, lower reinforcement weights were obtained from the deep beams, to which the DTM operator was applied.

It can be said that the DTM facilitated a more extensive search of the design space than existing mutation operators. Moreover, the DTM gave better results in solving problems than existing mutation operators that were used in this research.

References

ACI 318-99 (1999), American Concrete Institute.

Albayrak, M. and Allahverdi, N. (2011), “Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms”, *Exp. Syst. Appl.*, **38**, 1313-1320.

Berry, A. and Vamplew, P. (2004), “PhD can mutate: a simple dynamic directed mutation approach for genetic algorithms”, *Proceedings of International Conference on Artificial Intelligence in Science and Technology*, 200-205.

Castilho, V.C., Mounir, M.K.D. and Nicoletti, M.C (2007), “Using a modified genetic algorithm to minimize the production costs for slabs of precast pre-stressed concrete joints”, *Constr. Build. Mater.*, **85**, 78-90.

Daoud, M.I. and Kharma, N. (2011), “A hybrid heuristic-genetic algorithm for task scheduling in heterogeneous processor networks”, *J. Parallel Distrib. Comput.*, **71**, 1518-1531.

Deep, K. and Thakur, M. (2007), “A new crossover operator for real coded genetic algorithms”, *Appl. Math. Comput.*, **188(**1), 895-911.

Goldberg, D.E. (1989), *Genetic Algorithms in Search. Optimization And Machine Learning. Reading*, Addison-Wesley, MA.

Govindaraj, V. and Ramasamy, J.V. (2005), “Optimum detailed design of reinforced concrete continuous beams using genetic algorithms”, *Comput. Struct.*, **84**(1-2), 34-48.

Gupta, S., Agarwal, G. and Kumar, V. (2010), “Task scheduling in multiprocessor system using genetic algorithm, Machine Learning and Computing (ICMLC)”, *2010 Second International Conference on IEEE*, 267-271.

Hansen, N. and Ostermeier, A. (2001), “Completely derandomized self adaptation in evolutionary strategies”, *IEEE Tran. Evol. Comput.*, **9**(2), 159-195.

Heitzunger, C. and Selberherr, S. (2002), “An extensible TCAD optimization framework combing gradient based and genetic optimizers”, *Microelectron. J.*, **33**, 61-68.

Holland, J.H. (1975), *Adaptation in Natural And Artificial Systems*, The University of Michigan Press, Ann Arbor.

Hwang, R., Gen, M. and Katayama, H. (2006), “A performance evaluation of multiprocessor scheduling with genetic algorithm”, *Asia Pacif. Manage. Rev.*, **11**(2), 67-72..

Kaya, M. (2011), “The effects of a new selection operator on the performance of genetic algorithm”, *Appl. Math. Comput.*, **217**, 7669-7678.

Kaya, M. (2011), “The effects of two new crossover operators on genetic algorithm performance", *Appl. Sof. Comput.*, **11**, 881-890.

Kołodziej, J. and Khan, S.U. (2012), "Multi-level hierarchic genetic-based scheduling of independent jobs in dynamic heterogeneous grid environment", *Inf. Sci*., **214**, 1-19.

Korejo, I., Yang, S. and Li, C. (2010), "A directed mutation operator for real coded genetic algorithms, Evo applications, Part I", LNCS 6024, 491-500.

Lobo, F., Lima, C. and Michalewicz, Z. (2007), "Parameter setting in evolutionary algorithms", *Stud. Comput. Intel.*, **54**, 47-76.

Lu, H., Niu, R., Liu, J. and Zhu, Z. (2013), "A chaotic non-dominated sorting genetic algorithm for the multi-objective automatic test task scheduling problem", *Appl. Soft Comput*., **13**, 2790-2802.

Nilson, A.D.D. (1997), *Design of Concrete Structures*, The McGraw Hill CoMpanies, İnc.

Omara, F.A. and Arafa, M.M. (2010), "Genetic algorithms for task scheduling problem", *J. Parallel Distrib. Comput*., **70**, 13-22.

Pratihar, D.K. (2008), *Soft Computing*, Alpha Science International Ltd., Oxford, UK, 2008.

Rahmani, A.M. and Vahedi, M.A. (2008), "A novel task scheduling in multiprocessor systems with genetic algorithm by using Elitism stepping method", *J. Comput. Theor. Eng.*, **1**(1), 1.

Sahab, M.G., Ashour, A.F. and Toropov, V.V. (1999), "A hybrid genetic algorithm for reinforced concrete flat slab buildings", *Comput Struct.*, **83**(8-9), 551-559.

Sathappan, O., Chitra, P., Venkatesh, P. and Prabhu, M. (2011), "Modified genetic algorithm for multiobjective task scheduling on heterogeneous computing system", *Int. J. Inf. Technol. Commun. Converg*., **1**, 146-158.

Singh, J. and Singh, G. (2012), "Improved task scheduling on parallel system using genetic algorithm", *Int. J. Comput. Appl*., **39**, 17-22.

Zhou, Q. and Li, Y. (2003), "Directed variation in evolutionary strategies", *IEEE Tran. Evol. Comput.*, **7**(4), 356-366.

HK