Global Journal of Computer Sciences: Theory and Research

Volume 14, Issue 1, (2024) 01-14

www.gjcs.eu

Determination of reinforcement diameters of reinforced concrete deep beams with genetic algorithms

Mustafa Kaya[a *], Aksaray University Faculty of Engineering, Aksaray, Türkiye

Suggested Citation:

Kaya, M. (2024). Determination of reinforcement diameters of reinforced concrete deep beams with genetic algorithms. *Global Journal of Computer Sciences: Theory and Research. 14*(1), 01-14. https://doi.org/10.18844/gjcs.v14i1.9194

Received from; October 25, 2023, revised from; December 22, 2023 and accepted from February 05.

Selection and peer review under the responsibility of Assoc. Prof. Dr. Ezgi Pelin YILDIZ, Kafkas University, Turkey.

iThenticate Simiarity Rate: 15%

Abstract

Genetic algorithms, a stochastic research method, emerged by adapting the development process of biological systems to the computer environment. Operations carried out in genetic algorithms are performed on units stored in computer memory, similar to natural populations. Today, many linear or nonlinear methods have been developed for the solution of optimization problems. Since genetic algorithms are heuristic, they may not find the optimum result for a given problem. However, it gives very close to optimum values for problems that cannot be solved by known methods or whose solution time increases exponentially with the solution of the problem. Genetic algorithms are initially applied to nonlinear optimization problems. In this study, a genetic algorithm was applied to the single-span beam, a single-span beam with a gap in its body. While applying to the genetic algorithm the problems developed back-controlled selection, randomly mixed crossover, double-sensitivity mutation operators, and backward-controlled stopping criterion were used. As a result, developed genetic algorithm operators were applied to the too-big-sized beam problems. These beams' dimensions were too big but they weren't deep beams according to ACI 318-95 rules.

Keywords: Deep beams; genetic algorithms; reinforcement diameters; selection Operator.

* ADDRESS FOR CORRESPONDENCE: Kaya, Mustafa, Aksaray University Faculty of Engineering, Aksaray, Türkiye
E-mail address: kaya261174@hotmail.com / Tel.: +90 (0382) 288 31 21

1. Introduction

The basic principles of genetic algorithms were first introduced by [1]. The discovery of the crossover operator by Holland played a major role in the development of genetic algorithms. The first study on genetic algorithms in the literature is Holland's work on Machine Learning. Goldberg's [2] work on the control of gas pipelines, later influenced by this work, proved that genetic algorithms can have practical uses.

The problems for which genetic algorithms are most suitable are those that cannot be solved by traditional methods or whose solution time increases exponentially with the size of the problem. Until today, it has been tried to solve problems in different fields with genetic algorithms. Some of these fields of study are optimization, automatic programming, machine learning, economics, population genetics, evolution and learning, and social systems[3,4,5,6,7].

In the first step of the genetic algorithms, an initial population of subsets of all possible solutions is obtained [8]. Each member of the population is coded as an individual. Each individual is biologically equivalent to a chromosome. Every individual in the population has a fitness value. The fitness value determines which individual will move to the next population. The strength of an individual depends on their fitness value, and a good individual has a high fitness value if it is a maximization problem, and a low fitness value if a minimization problem [9].

Genetic algorithms used in solving any problem consist of the following components.

- Representation of individuals forming the population as a sequence (“chromosome”).
- Creation of the initial population.
- Determining the suitability of individuals and establishing the evaluation function
- Genetic operators for obtaining new populations.
- Control variables (Probabilities of the crossover and mutation operators)

1.1. Purpose of study

In this study, a genetic algorithm was applied to the single-span beam, a single-span beam with a gap in its body.

2. Methods and materials

2.1. Coding

The most important feature that distinguishes genetic algorithms from other methods is the use of codes instead of variables. The first step in applying the genetic algorithm to any problem is to choose the most appropriate coding type for the problem (table 1). In this study permutation coding was used.

TABLE I

Profile numbers and codes of these profiles

Profile Number	Profile Section
1	IPE200
2	IPE220
3	IPE240
4	IPE260

2.1.1. Permutation coding

In permutation coding, the chromosome length is equal to the number of design variables. Permutation coding is especially preferred in problems where design variables consist of more than one sub-variable. In this coding type, the codes of the design variables consist of randomly selected numbers between 1 and the number of design variables (Table 2).

TABLE II

Code of a random individual in permutation coding type

Number	1. profile	2. profile	3. profile	4. profile	5. profile
Code	3	1	4	2	3

2.2. Creating the initial population

Population Another important feature that distinguishes genetic algorithms from other methods is that they perform a search within a population of points, not point-to-point (Goldberg, 1989). Therefore, the first step of the genetic algorithm is the creation of the initial population.

2.2.1. Evaluation

In each generation, the fitness values of the individuals in the population are calculated with an evaluation function. The fitness value plays a role in determining which candidate solutions from the existing population will be used to obtain new candidate solutions that will form the next population. The evaluation function used in genetic algorithms is the objective function of the problem. However, in some problems, objective functions may be limited by design variables. In this case, objective functions constrained by design variables should be converted to unconstrained objective functions independent of design variables with two transformations. In the first transformation, the constrained objective function f(s) is transformed into the unconstrained objective function Ø(s) by applying the following transformation (Eq. 1). Error functions are used in this transformation.

$$Ø(s) = [\ f(s) + R.\sum \Phi(Z)] \quad (1)$$

If $Z > 0$ ise $\Phi(Z) = Z2$

If $Z \leq 0$ ise $\Phi(Z)=0$ (2)

f(s): Constrained objective function

R: Predetermined error coefficient (R=10,100…)

(Z): Error function

Ø(s): Unconstrained objective function

In the second transformation, the unconstrained objective function Ø(s) is transformed into the fitness function F(s) (Eq.3).

$$F(s)= Ø\ max-[\ f(s) + R\ .\sum \Phi(gj(x)\ ]\ F(s)= Ø\ max- Ø(s) \quad (3)$$

Ø max: The maximum value of the unconstrained objective function

F(s): Fitness function

The fitness values of individuals are calculated from Equation 3. Then, according to these fitness values, individuals to be used in the next stage are determined by any of the following methods. When any selection method is used in minimization problems, it is not possible to directly use the objective function to calculate the fitness values. Because these selection mechanisms are concerned with maximizing fitness. One method used to transform minimization problems into maximization problems is to multiply the objective function of the minimization problem by (-1). However, this method is not used in genetic algorithms due to the condition that the fitness values are positive. A minimization problem solved by genetic algorithms is transformed into a maximization problem by subtracting the objective function value of the problem from a large value determined at the beginning [2].

2.2.2. Selection methods

After the initial population is formed in each generation, the members of the new population are selected from among the members of the existing population through a selection process. Individuals with a high fitness value have a high probability of acquiring new individuals. This operator performs natural selection artificially. The fitness of natural communities is determined by the ability of individuals to withstand the barriers of growth and reproduction. The selection method used is summarized below.

2.3. Back controlled selection operator (BCSO)

In the current selection operators, the selection operation is performed among the individuals that make up the population. In this operator, the selection is carried out according to the fitness values of each individual. The BCSO differs from existing selection operators because the fitness value of the individual is compared with the fitness value in the previous generation If the fitness value of the individual is more than the one in the preceding generation, this individual would keep their position. Otherwise, if the fitness value of the same individual is less than or equal to the fitness value in the preceding generation, this individual would be discarded from the population [10].

2.4. Crossover operator

Crossover is a search operator that provides access to similar but unexplored regions of the search space by exchanging information between different solutions. With the crossover operator, two different individuals, which will give new points in the search space, are obtained by interchanging certain parts of two randomly selected individuals from the population. In this study, the randomly mixed crossover was used.

2.4.1. *Randomly mixed crossover*

In this crossover operator, beginning the process the existing crossover operators were numbered. Then, the existing crossover operators were applied randomly to the chromosome couples in the population of the same generation [11].

2.5. Mutation operator

When working on a limited population, there is a possibility that some genetic information in the population will be lost prematurely after a while. All the genes that make up a chromosome can be the same. It is not possible to replace such a chromosome with the crossover operator. In this case, the gene in the randomly selected site on the chromosome is modified by interfering with the chromosomes that make up the population at a certain rate [2]. It is recommended to take the mutation rate between 0.0001 and 0.05 [12].

2.5.1. Double times sensitive mutation (DTSM)

The developed DTSM operator differs from the existing mutation operators in that this operator was applied to the randomly selected members' randomly selected sites of the population. In the developed mutation operator process is applied in two steps every generation. In the first step, a member is randomly selected between members in the population. In the second step, a site is randomly selected between sites in the members. In this operator mutation process applied to the population is double times sensitivity mutation operators [13].

3. Results

3.1. Stopping criteria

In GA problem analysis, the analysis terminates at a previously determined time, generation numbers, or fitness values. The stopping of the analysis according to the determined fitness values provides more accomplished results than the previously determined generation numbers and time. Moreover, previous stopping criteria do not provide enough investigation into the design area.

3.1.1. The backward controlled stopping criterion (BCSC)

The difference between the newly developed stopping criterion and the available stopping criteria is that a very high generation number is not entered when the newly developed stopping criterion is used. When the analysis starts; if the fitness value obtained from the following generation is; bigger than the value obtained from the previous generation, the maximum generation number increases by one hundred; if equal, the maximum generation number increases by fifty; and if smaller than the maximum generation number does not change. If a higher fitness value cannot be found in the last step of the analysis, the analysis stops automatically, and the maximum fitness value obtained from the first step is accepted as the maximum fitness value for this analysis. Also, every problem at the completion time was determined for all stopping criteria and was given at the tables.

3.2. Reinforced concrete deep beams

Some reinforced concrete beams show different behavior from the classical beam behavior because they have a very large height compared to their spans. The thickness of these beams in the direction perpendicular to the plane is much smaller than both their openings and their depths. Although there is no full harmony between the regulations of the countries in the definition of these elements, some of the definitions made are as follows.

According to TS 500; The ratio of beam span to height for continuous beams is 2.5, and for simple beams, beams with a ratio of beam span to beam height less than 2 are considered high beams (TS 500, 1984).

According to ACI 318-95; In the bending calculation; in continuous beams, the ratio of beam clear span to beam height is 5/2, and for simple span beams, beams with a ratio of beam clear span to beam height less than 5/4 are considered high beams (ACI, 1999).

According to ACI 318-95: In the shear calculation; If the ratio of the beam span to the beam height is less than 5 and the beam is loaded from the top face, the beam is considered a high beam (ACI 318-95, 1999).

Another definition can be made as follows. The beams, which are carried by the compressive belt formed between the load and the reactions, are considered high beams. This is achieved if the single load is closer to the beam's useful height at the support for beams loaded with a single load, or if the ratio of beam span to beam height is less than 5 for beams loaded with evenly distributed load (318-95, 1999). Such elements are used in walls and tanks of rectangular tanks, floor diaphragms, shear walls, corrugated roof sheets, bunkers, quay walls, and silos [14].

3.2.1. Determination of reinforcement diameters in a single-span deep beam

The most suitable reinforcement diameter was determined using genetic algorithms by keeping the reinforcement spacing in the deep beam constant. The deep beam was considered an element consisting of 3x3 plates. Then, the deep beam, which consists of these nine plates, is analyzed with the finite element method under the two loads acting from the upper edge of the beam, and the forces acting on these plates are determined. The required horizontal and vertical reinforcement areas were determined for each plate. By selecting the reinforcement spacings on the horizontal and vertical axis, the process of determining the most suitable reinforcement diameter to be used in the horizontal and vertical axis for each plate was started. Since the determination of the most suitable reinforcement diameter is an optimization problem, genetic algorithms were used to solve this problem.

3.2.1.1. Coding

For this problem, it was decided that the most appropriate coding type was permutation coding. In our problem, we have 18 design variables and horizontal and vertical reinforcement diameters for each plate. Since the number of design variables will be equal to the chromosome length, our chromosome length has been determined as 16. There are 8 rebar diameters for each of our design variables (Table 3).

TABLE III

The design variables used in the problem and the codes representing these variables

Diameter	Ø10	Ø12	Ø14	Ø16	Ø18	Ø20	Ø22	Ø24
Code	1	2	3	4	5	6	7	8

The chromosome structure of any individual used in the problem is given in Table 4. While obtaining chromosomes the numbers are randomly chosen from 1 to 8 (Table 4).

TABLE IV

The chromosome structure of any individual used in the problem

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
5	3	4	7	6	2	8	1	null	null	5	7	4	3	6	3	4	5

3.3. Creating the initial population

In Table 5., the initial population consisting of 10 individuals was given.

TABLE V

The initial population of 10 individuals

Individual number	Individual Code																	
1	7	2	8	1	5	3	6	4	null	Nilsson	3	4	7	2	1	5	4	2
2	3	5	8	1	2	4	7	6	null	null	3	8	5	1	7	4	8	1

3	3	1	7	6	2	4	8	5	null	null	6	4	7	5	1	8	3	7
4	5	6	3	4	1	7	2	8	null	null	8	2	1	5	3	6	1	2
5	8	7	3	2	4	5	1	6	null	null	4	1	8	2	6	3	4	5
6	5	4	6	3	7	2	1	8	null	null	5	4	8	2	6	3	4	5
7	4	3	5	1	2	6	7	8	null	null	7	3	3	1	2	7	2	8
8	5	1	2	6	8	7	3	4	null	null	2	6	8	4	1	6	1	4
9	4	7	5	6	8	1	3	2	null	null	2	8	1	3	8	7	3	6
10	4	6	8	2	5	1	7	3	null	null	3	5	4	6	7	1	6	7

3.4. Evaluation

In the evaluation phase, first of all, an objective function suitable for the problem was determined (Eq. 4).

$$f(s) = \min [\Sigma \rho . l . A] \tag{4}$$

Since this objective function is constrained by design variables, the function given in Eq. 4 is transformed into an unconstrained objective function (Eq.7). This function on independent of the design variables by using the error functions given in Eq. 5 and Eq. 6.

$$fh = As.cs / Fe \tag{5}$$

$$\text{If } fh=1 \text{ ise } \Phi(g)=fh \tag{6}$$

$$Ø(s) = [R.\sum \Phi(g)] \tag{7}$$

The problem-specific error coefficient is taken as R=1. The following conversion is made to the unconstrained objective function Ø(s) to obtain the fitness function (Eq. 8).

$$F(s)=\max [1- Ø(s /Øort] \tag{8}$$

Then the fitness values of the individuals in the initial population are calculated. The fitness values are used to identify the individuals who will form the next population.

3.5. Applying the copy operator to the population

The individuals forming the initial population are ranked according to their fitness values. After this ranking, individuals whose fitness values were lower than the predetermined success limit were considered unsuccessful and expelled from the population. Successful individuals are copied instead of unsuccessful individuals are expelled from the population (Table 6). In this study, the success limit was accepted as 50%. Individuals numbered 6, 2, 5, 4, and 3 in the initial population were expelled from the population because their fitness values were below the success limit, and individuals numbered 1, 8, 7, 10, and 9 were copied instead (Table 7).

TABLE VI

The fitness values of the individuals that make up the unit

Fitness value		Sorted fitness value		Fitness values after copying	
Individual Number	Fitness value	Individual Number	Fitness value	Individual Number	Fitness value
1	0.4172	6	0.3923	1	0.4172
2	0.3989	2	0.3989	8	0.4189
3	0.4152	5	0.4127	10	0.4233
4	0.4134	4	0.4134	7	0.4203
5	0.4127	3	0.4152	9	0.4562
6	0.3923	1	0.4172	1	0.4172
7	0.4203	8	0.4189	8	0.4189
8	0.4189	10	0.4233	10	0.4233
9	0.4562	7	0.4203	7	0.4203
10	0.4233	9	0.4562	9	0.4562

TABLE VII

Initial population after copying P'(t)

Individual Number	The New Population Formed at the End of the Copy Process																	
1	7	2	8	1	5	3	6	4	8	6	3	4	7	2	1	5	4	2
8	5	1	2	6	8	7	3	4	5	4	2	6	8	4	1	6	1	4
10	4	6	8	2	5	1	7	3	2	8	3	5	4	6	7	1	6	7
7	4	3	5	1	2	6	7	8	5	2	7	3	3	1	2	7	2	8
9	4	7	5	6	8	1	3	2	1	7	2	8	1	3	8	7	3	6
1	7	2	8	1	5	3	6	4	8	6	3	4	7	2	1	5	4	2
8	5	1	2	6	8	7	3	4	5	4	2	6	8	4	1	6	1	4
10	4	6	8	2	5	1	7	3	2	8	3	5	4	6	7	1	6	7
7	4	3	5	1	2	6	7	8	5	2	7	3	3	1	2	7	2	8
9	4	7	5	6	8	1	3	2	1	7	2	8	1	3	8	7	3	6

3.6. Applying the crossover operator to the initial population

The initial population was crossed over to obtain individuals with higher fitness values. In this application, a point crossover method is used. The individuals forming the population are divided into two groups. These two groups called Parent 1 (Table 8) and Parent 2 (Table 9) were crossed with each other. After the crossover process, two new communities called Child 1 (Table 10) and Child 2 (Table 11) were formed. While child 1 from these two new populations was kept in the population, child 2 was expelled from the population. A new group was produced to replace child 2 who was expelled from the population. Later, child1 and translator1 are combined (Table 12).

TABLE VIII

The first parent to be a crossover

Individual Number	Individual Code																	
1	7	2	8	1	5	3	6	4	8	6	3	4	7	2	1	5	4	2
10	4	6	8	2	5	1	7	3	2	8	3	5	4	6	7	1	6	7
9	4	7	5	6	8	1	3	2	1	7	2	8	1	3	8	7	3	6
8	5	1	2	6	8	7	3	4	5	4	2	6	8	4	1	6	1	4
7	4	3	5	1	2	6	7	8	5	2	7	3	3	1	2	7	2	8

TABLE IX

The second parent to be a crossover

Individual Number	Individual Code																	
8	5	1	2	6	8	7	3	4	5	4	2	6	8	4	1	6	1	4
7	4	3	5	1	2	6	7	8	5	2	7	3	3	1	2	7	2	8
1	7	2	8	1	5	3	6	4	8	6	3	4	7	2	1	5	4	2
10	4	6	8	2	5	1	7	3	2	8	3	5	4	6	7	1	6	7
9	4	7	5	6	8	1	3	2	1	7	2	8	1	3	8	7	3	6

TABLE X

The first child formed as a result of the crossover operation

Individual Number	Individual Code																	
1	7	2	8	1	5	3	6	4	8	4	2	6	8	4	1	6	1	4
2	4	6	8	2	5	1	7	3	2	2	7	3	3	1	2	7	2	8
3	4	7	5	6	8	1	3	2	1	6	3	4	7	2	1	5	4	2
4	5	1	2	6	8	7	3	4	5	8	3	5	4	6	7	1	6	7
5	4	3	5	1	2	6	7	8	5	7	2	8	1	3	8	7	3	6

TABLE XI

The population was brought from outside instead of the second population formed as a result of the crossover process.

Individual Number	Individual Code																	
1*	1	5	1	2	5	2	6	3	6	1	5	1	1	5	1	1	8	1
2*	2	6	2	3	3	3	3	4	3	4	2	4	3	3	3	2	6	2
3*	3	5	3	3	3	3	3	4	3	4	2	4	3	3	3	2	6	2
4*	3	1	5	6	4	7	3	7	3	8	8	5	4	6	7	3	6	3
5*	5	7	5	7	8	6	7	8	6	7	7	4	3	3	3	2	6	2

Table XII

The new population formed as a result of the crossover operation is P''(t)

Individual Number	Individual Code																	
1	7	2	8	1	5	3	6	4	8	4	2	6	8	4	1	6	1	4
2	4	6	8	2	5	1	7	3	2	2	7	3	3	1	2	7	2	8
3	4	7	5	6	8	1	3	2	1	6	3	4	7	2	1	5	4	2
4	5	1	2	6	8	7	3	4	5	8	3	5	4	6	7	1	6	7
5	4	3	5	1	2	6	7	8	5	7	2	8	1	3	8	7	3	6
1*	1	5	1	2	5	2	6	3	6	1	5	1	1	5	1	1	8	1
2*	2	6	2	3	3	3	3	4	3	4	2	4	3	3	3	2	6	2
3*	3	5	3	3	3	3	3	4	3	4	2	4	3	3	3	2	6	2
4*	3	1	5	6	4	7	3	7	3	8	8	5	4	6	7	3	6	3

3.7. Application of the mutation operator to the initial population

While the program is running, after a certain period, all of the genes that make up the chromosome may be the same. It is not possible to replace such a chromosome with the crossover operator. The population is mutated according to a predetermined mutation rate (Table 13). In this study, the mutation rate was accepted as 0.01.

Table XIII

The new community formed as a result of the mutation process is P'''(t)

Individual Number	Individual Code																	
1	7	7	8	1	5	3	6	4	8	4	2	6	8	4	1	6	1	4
2	4	6	8	5	5	1	7	3	2	2	7	3	3	1	2	7	2	8
3	4	7	5	6	8	1	3	2	1	6	6	4	7	2	1	5	4	2

4	5	6	2	6	8	7	3	4	5	8	3	5	4	6	7	1	6	7
5	4	3	5	1	2	6	7	8	5	7	2	8	1	1	8	7	3	6
6	1	5	8	2	5	2	6	3	6	1	5	1	1	5	1	1	8	1
7	2	6	2	3	2	3	3	4	3	4	2	4	3	3	3	2	6	2
8	3	5	3	3	3	3	3	4	3	4	5	4	3	3	3	2	6	2
9	3	1	5	6	4	7	3	7	3	8	8	5	4	6	7	3	6	1
10	5	7	5	7	2	6	7	8	6	7	7	4	3	3	3	2	6	2

The new population obtained by applying the transcription crossover and mutation operators to the initial population was accepted as the initial population for the next generation. In this application, the processes are repeated until the number of elements with the features we have specified is 10 (Table 14). The fitness value the highest person fitness value and code are given in Table 15, and the reinforcement diameters are given in Table 16.

TABLE XIV

The fitness values, and codes of the 10 most suitable individuals in the problem of determining the most suitable reinforcement diameter in high beams using genetic algorithms.

No	Fitness Value	Codes of Eligible Individuals																	
1	0.5260	4	6	4	6	3	6	5	5	5	1	5	1	1	6	1	3	5	3
2	0.4565	4	6	4	6	3	6	5	5	5	1	5	1	1	6	1	3	5	3
3	0.4947	4	6	4	6	3	6	5	5	5	1	5	1	1	6	1	3	5	3
4	0.4451	3	6	3	3	4	3	3	5	3	3	6	3	4	3	4	3	5	3
5	0.4660	4	6	4	6	3	6	5	5	5	1	5	1	1	6	1	3	5	3
6	0.5252	3	5	3	4	6	4	7	4	7	1	2	1	1	7	1	1	8	1
7	0.5028	3	5	3	1	3	1	7	8	7	1	4	1	1	7	1	1	2	1
8	0.5417	3	5	3	1	3	1	7	4	7	1	4	1	1	7	1	1	2	1
9	0.4966	3	5	3	1	3	1	7	4	7	1	4	1	1	7	1	1	2	1
10	0.5341	3	5	3	1	3	1	7	4	7	1	4	1	1	7	1	1	2	1

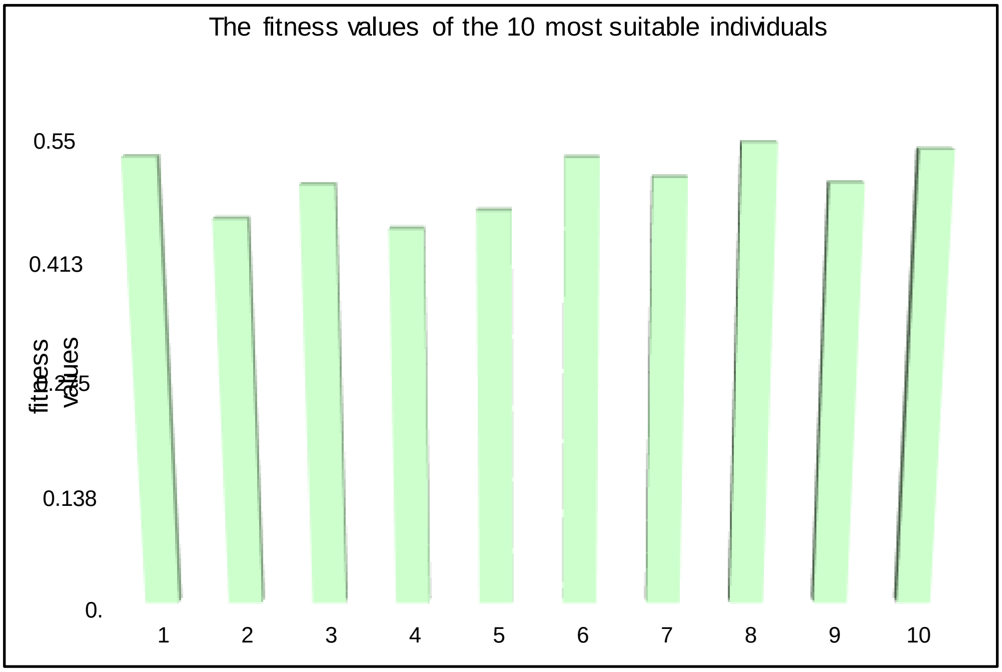

Fig. 1. The fitness values of the 10 most suitable individuals

The fitness value and code of the most suitable individual in the solution of the high beam are given in Fig. 1 with genetic algorithms. The code of the most suitable individual in the solution of the high beam is given in Fig. 1. with genetic algorithms and the reinforcement diameters corresponding to this code

TABLE XV

The fitness value and code of the most suitable individual

Fitness value	Code of the Most Suitable Individual
0.4825	2 5 2 3 3 3 3 4 3 4 2 4 3 3 3 2 6 2

TABLE XVI

Genetic algorithms and the reinforcement diameters corresponding to this code

Region Number	1	2	3	4	5	6	7	8	9
Horizontal Reinforcement code	2	5	2	3	3	3	4	3	4
Horizontal Reinforcement Diameter	Ø12	Ø18	Ø12	Ø14	Ø14	Ø14	Ø16	Ø14	Ø16
Vertical Reinforcement code	4	2	4	3	3	3	2	6	2
Vertical Reinforcement Diameter	Ø16	Ø12	Ø16	Ø14	Ø14	Ø14	Ø12	Ø20	Ø12

4. Conclusion

Genetic algorithms, a stochastic research method, emerged by adapting the development process of biological systems to the computer environment. Operations carried out in genetic algorithms are performed on units stored in computer memory, similar to natural populations. Today, many linear or nonlinear methods have been developed for the solution of optimization problems. It is accepted that the design variables are continuous with these methods of optimization problems. Due to a large number of design variables and constraints in some problems, the use of traditional optimization methods in solving such problems sometimes gives incorrect results or the solution time becomes too long.

Since genetic algorithms are heuristic, they may not find the optimum result for a given problem. However, it gives very close to optimum values for problems that cannot be solved by known methods or whose solution time increases exponentially with the solution of the problem. Genetic algorithms are initially applied to nonlinear optimization problems. In this study, a genetic algorithm was applied to the single-span beam, a single-span beam with a gap in its body. While applying to the genetic algorithm the problems developed back-controlled selection, randomly mixed crossover, double-sensitivity mutation operators, and backward-controlled stopping criterion were used. As a result, developed genetic algorithm operators were applied to the too-big-sized beam problems. These beams' dimensions were too big but they weren't deep beams according to ACI 318-95 rules.

References

[1] J. H. Holland, "Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. MIT Press, 1992.

[2] Goldberg DE., "Genetic Algorithms in Search, Optimization, and Machine Learning." Accessed: May 19, 2024. [Online]. Available: https://www.nature.com/articles/s41588-023-01650-1

[3] R. Hatanaka, K. Tamagawa, N. Haruta, and A. Sugimoto, "The impact of differential transposition activities of autonomous and nonautonomous hAT transposable elements on genome architecture and gene expression in Caenorhabditis inopinata," GENETICS, p. iyae052, Apr. 2024, doi: 10.1093/genetics/iyae052.

[4] A. Okazaki and J. Ott, "Machine learning approaches to explore digenic inheritance," Trends in Genetics, vol. 38, no. 10, pp. 1013–1018, Oct. 2022, doi: 10.1016/j.tig.2022.04.009.

[5] J. Pigeon and B. A. Hassan, "Timing neurogenesis: a clock or an algorithm?" Current Opinion in Genetics & Development, vol. 85, p. 102156, Apr. 2024, doi: 10.1016/j.gde.2024.102156.

[6] S. Lu, H. Jiang, C. Li, B. Hong, P. Zhang, and W. Liu, "Genetic Algorithm for TMS Coil Position Optimization in Stroke Treatment," Front. Public Health, vol. 9, p. 794167, Mar. 2022, doi: 10.3389/fpubh.2021.794167.

[7] P. B. Munroe, N. Aung, and J. Ramírez, "Genetic analysis of cardiac dynamic flow volumes identifies loci mapping aortic root size," Nat Genet, vol. 56, no. 2, pp. 197–199, Feb. 2024, doi: 10.1038/s41588-023-01650-1.

[8] Pelayo Quirós and Fernando Sánchez Lasheras, "Methodology for the projection of population pyramids based on Monte Carlo simulation and genetic algorithms | Applied Intelligence." Accessed: May 19, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s10489-023-04492-w

[9] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs. in Artificial Intelligence. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992. doi: 10.1007/978-3-662-02830-8.

[10] M. Kaya, "The effects of a new selection operator on the performance of a genetic algorithm," Applied Mathematics and Computation, vol. 217, no. 19, pp. 7669–7678, Jun. 2011, doi: 10.1016/j.amc.2011.02.070.

[11] M. Kaya, "The effects of two new crossover operators on genetic algorithm performance," Applied Soft Computing, vol. 11, no. 1, pp. 881–890, Jan. 2011, doi: 10.1016/j.asoc.2010.01.008.

[12] O. Hasançebi and F. Erbatur, "Layout optimization of trusses using improved GA methodologies," Acta Mechanica, vol. 146, no. 1, pp. 87–107, Mar. 2001, doi: 10.1007/BF01178797.

[13] M. Kaya, "Developing a new mutation operator to solve the RC deep beam problems by aid of genetic algorithm," 2018, doi: 10.12989/cac.2018.22.5.493.

[14] A. Nilson, DESIGN OF CONCRETE STRUCTURES. 1997. Accessed: May 19, 2024. [Online]. Available: https://trid.trb.org/View/680675