International Journal of Engineering & Applied Sciences (IJEAS)
Vol.15, Issue 1 (2023) 1-18
http://dx.doi.org/10.24107/ijeas.1232408
Int J Eng Appl Sci 15(1) (2023) 1-18

Comparison of Different Regulations and Metaheuristic Algorithms in Beam Design

Muhammed Çoşut, Gebrail Bekdaş, Sinan Melih Niğdeli[*]

Department of Civil Engineering, Istanbul University- Cerrahpaşa, İstanbul, 34320, Turkey
[*]E-mail address: muhammed.cosut@ogr.iuc.edu.tr, bekdas@iuc.edu.tr, melihnig@iuc.edu.tr [*]

ORCID numbers of authors:
0000-0002-9356-9030, 0000-0002-7327-9810, 0000-0002-9517-7313[*]

Received date: 11.01.2023
Accepted date:27.02.2023

Abstract

In this first study, the rectangular reinforced concrete beam's costs and cross-section sizes are found by using Harmony Search (HS), Differential Evolution Algorithm (DE), Jaya Algorithm, Teaching- Learning Based Algorithm (TLBO), Hybrid algorithm (Jaya-TLBO) and Flower Pollination Algorithm (FPA) separately by using ACI 318 building code. In addition, in order to better see how successful the algorithms are, the standard deviation of the algorithms used in the project in a certain number of iterations, price changes and in which iteration the minimum cost is compared. As a result of running different algorithms 5, 10, 15 and 20 times, separate values are recorded, and the average number of iterations of the algorithms for each is shown by finding the standard deviation values. Furthermore, Hybrid Algorithm reached the objective function in fewer iterations and their standard deviations reached 0 earlier. In the second study, the beam design is made according to the ACI 318, TS500 and Eurocode 2 regulations under certain loads by using a Hybrid Algorithm with different concrete classes. Optimization of this design is made using the Matlab program, and comparisons are made between regulations. Eurocode and TS500 design costs are roughly the same; however, ACI 318's design is the cheapest.

Keywords: Beam Design, Metaheuristic Algorithms, Building codes, Cost Optimization, Hybrid Algorithm.

1. Introduction

Over time, studies have been carried out in many areas around the world in order to apply sustainable and safe systems [1], and as a result, it has enabled some systems to be implemented, designed and made in a short time with different desired features according to restraints properties [2]. These have accelerated by spreading to many areas instead of being limited to only one area, and in the general sense of the recent studies, studies are carried out to use the world's resources more efficiently and by preventing their consumption, expense [3] and pollution [4]. While these studies are carried out under sustainability, designs have become the focal point of making this situation in the foreground. Increasingly complex problems are solved by metaheuristic algorithms easily and successfully [5] and enable to design of cost-effective structures [6].

Studies in the field of civil engineering have likewise gained momentum and in this process, many structures; design according to the type of use, the use of different materials and different systemic designs are provided. In different beam designs, section sizes and similar cases, the structure is selected and completed according to the purpose of use. However, in the design phase, the cross-section dimensions are assigned by the trial and error method, which is the traditional method, and analyses are made and the results are interpreted. However, such designs which have no optimal numerical solutions can prevent very effective results in terms of time and cost. By creating objective functions (cost, CO_2 emission which is prominent in structural design [7], displacement, etc.), reaching the best results in a short time is achieved with metaheuristic algorithms. Metaheuristic algorithms have been used more for various problems in recent years [8,9]. Although there are many studies conducted in this context, these studies differ according to the designs and purposes of the building elements. Chakrabarty [10] studied the optimization of the design cost and material consumption with the Nonlinear program model in his study. Bekdaş and Nigdeli [11] provided the optimum design by using the TLBO algorithm in their study. Zivari et al. [12] worked on optimum weight and material optimization. Guerra and Kiousis [13] performed the optimization of the beam using a sequential quadratic programming algorithm. In the study of Chutani and Singh [14], the Particle Swarm Algorithm performed the optimum study of the reinforced concrete beam design by using Indian regulation. Nigdeli and Bekdaş [15] carried out the optimum design according to the unfavorable live load in their study. Coello et. al. [16] utilized Genetic Algorithm to achieve optimum beam design. Ulusoy et al. [17] optimized the minimum cost for the reinforced concrete beam by using Bat Algorithm (BA), Harmony Search (HS), Teaching-Learning Based Optimization (TLBO). In addition, Ulusoy et al. [18] found the optimum design of multi-span frame structures which consist of reinforced concrete by using Harmony Search.

In this study, different metaheuristic algorithms are used for comparison to the effectiveness of beam design which is utilized in various areas of structural engineering. All algorithms have different features which can be about phase number, control parameters as well as combined different features. Therefore, they affect the needed iteration number that can be enough to reach the objective function. In the second study, 3 different regulations which are ACI 318 [19], Eurocode 2 [20] and TS500 [21] are used to design the beam according to various classes of concrete.

2. Materials and Methods

2.1. The Beam Design and Regulations

Beams are one of the generally preferred building elements which are a member of frame systems in building designs and this building element may differ according to their designs. These differences affect the operation of the beam and enable it to gain different properties. It will be observed that there is a difference in the calculations when it is designed as a rectangular cross-section of the beam and a T-section beam. Firstly, in order to endure the bending moment, the cross-sections of the beam are assigned according to regulations, and it has to use the needed reinforcement area [22, 23]. Secondly, the effect of corrosion and enough capacity should be considered, when engineers design structural elements [24]. Even though the tension zone of the cantilever beams (balcony) is to be in the upper part of the beam, the two columns inside the building will be in the lower part of the beam. Such differences have a very important effect on design and reinforcement placement. The beams are reinforced in the area where the tension zone will be formed according to the dead and live load applied to it. In this way, tensile forces will be met with reinforcements with much better tensile strength than concrete under loads, and situations such as breaking or cracking in the structure will be reduced.

Each regulation has some specific formulas for design problems. Some formula differences may be due to the results of the laboratory or depending on factors such as the situation in which measures are taken as a result of the structural errors experienced in the history of the country. In this way, when the design of the structure is carried out under certain loads, besides the fact that the system has different cross-section dimensions, there may also be differences in the use of the required reinforcement area. If the reinforcement area is used more or less, its effect on the system should be considered [25]. Furthermore, if the result of the necessary reinforced area

is smaller than the minimum reinforced area, it will be equal to the minimum reinforced area for 3 building codes.
Table 1 shows the beams' design constraints. The first of the constraint values given for TS500 gives the area where the depth of the stress block should be, while the second constraint is the comparison of reinforcement ratios. The reason why the 0.235* f_{cd}/f_{yd} equation is taken into account in the reinforcement ratio comparison is used to keep the deflection conditions under control [26].
For Eurocode 2, the g_1 is used to control whether compressive steel is necessary or not (k), and the g_2 restrains the stress block depth, as well as g_3 limits the maximum reinforced area.
For ACI 318, the g_1 compares to values of the stress block depth, while the g_2 compares the reinforced area. Also, all abbreviations have meanings and these are:

As	reinforcement area,
b	section width,
h	section height,
d	distance from the over-compression to the center of the longitudinal tensile reinforcement
z	internal force lever (moment lever)
fyk	characteristic yield strength of steel,
fyd	steel design yield strength,
fcd	concrete design compressive strength,
fck	characteristic compressive strength of concrete,
ρ_b	balanced reinforcement ratio

Table 1. The Beam's Constraints

Eurocode 2	$g_1 = k < 0.167$ $g_2 = z < 0.95 * d$ $g_3 = \frac{A_s}{b \times h} < 0.04$
TS500	$g_1 = 0 < d - \sqrt{a} < h$ $g_2 = \text{Reinforced Area} \leq \begin{cases} 0.85 \times \rho_b \\ 0.02 \\ 0.235 \times \frac{f_{cd}}{f_{yd}} \end{cases}$
ACI 318	$g_1 = 0 < d - \sqrt{a} < h$ $g_2 = \frac{A_s}{b \times h} < 0.75*0.85*k1*(f_{ck}/f_{yk})*600/(600+f_{yk})$

Eq. (1) calculate how much money should expend on concrete, while Eq. (2) calculates the money for needed steel; furthermore, Eq. (3) contributes to finding how much money is necessary for labor and formwork as well as Eq. (4) is used for the total cost for this design.

$$C_{concrete} = C_c \times L \times \frac{(b_w \times h - A_s)}{10^6} \tag{1}$$

$$C_{concrete} = C_s \times L \times \gamma_s \times \frac{A_s}{10^6} \tag{2}$$

$$C_{formwork-labor} = (C_k + C_{ki}) \times L \times \frac{(b_w \times h)}{10^6} \tag{3}$$

$$TotalCost = C_{concrete} + C_{steel} + C_{formwork-labor} \tag{4}$$

2.2. Metaheuristic Algorithms

Metaheuristic algorithms are algorithms that are inspired by events in nature and created by forming equations as a result of observations. Although it is frequently used in fields such as engineering, economy, logistics, finance and energy systems, it can also be used in different fields. When the optimization process and structural design process are combined, it can lead to finding the optimum objective function effectively and easily [27-30]. To give examples of these algorithms; Algorithms such as Simulation Annealing (SA), Flower Pollination Algorithm (FPA), Cuckoo Algorithm (CS) [31] and Artificial Bee Colony Algorithm (ABC) [32] can be given as examples. These algorithms have been created as examples from many areas and differences of life. Some of these are the Ant Colony Algorithm developed based on the movements of the ants, the Bat Algorithm developed by utilizing the features of the Bats, the Differential Evolution with the evolutionary developments based on the population, and the Harmony Search Method inspired by ensuring that the musical piece sounds the best to the listener [33]. Although each developed algorithm has different formulas, it differs according to whether it is single-stage or multi-stage. Because of these differences, some algorithms can give more efficient results in reaching the objective functions.

2.2.1 Teaching-Learning based optimization (TLBO)

This algorithm is developed by Rao et al. [34] in 2011, inspired by the learning interaction stages between teacher-student. It consists of two phases: the teacher phase and the student phase[35]. This feature makes it superior to other algorithms. The reason is that the objective function is compared 2 times in 1 iteration, and this allows the algorithm to complete the algorithm in a shorter time by reaching the objective function in fewer iterations.

```
Begin
% All needed constraints, variables and constants should be written
% The determination of population and iteration number
% Finding moment value
% Cross-Section lengths are generated randomly in terms of variable range.
% Finding reinforcement area
        -   Finding reinforcement ratio according to reinforcement area
        -   Comparing the maximum and minimum reinforcement area
        -   Generating the initial solution matrix
        -   Controlling the constraints and penalizing the objective function
The step of Teaching-Learning Phase
% Finding the mean and best value of initial solution matrix
% Finding the teaching factor (TF)
% Generating the variables
% Finding reinforcement area
        -   Finding reinforcement ratio according to reinforcement area
        -   Comparing the maximum and minimum reinforcement area
        -   Generating the new solution matrix
        -   Controlling the constraints and penalizing the objective function
% Comparing the initial and new matrix, and choosing best one.
End
```

Fig. 1. TLBO Pseudo Code

2.2.2 Differential evolution (DE)

It is an algorithm developed by Storn and Price [36], inspired by the natural evolutionary state of species. This algorithm has been successfully applied in a lot of areas [37,38,39,40] such as engineering [41], communication [42] and many different fields. It is a random search method that simulates mutation, re-arrangement and selection steps [37]. The pseudo-code is shown in Fig. 2.

```
Begin
% All needed constraints, variables and constants should be written
% The determination of population and iteration number
% Cross-Section lengths are generated randomly
% NECESSARY EQUATIONS SHOULD BE WRITTEN in HERE
% Generating the initial solution matrix
The step of Differential Evolution
% Generating p, q as well as r which change iteration number (Mutation process)
% Crossover operation and comparing variables
% If (rand () <= CR) || (kr == randkr)
        b= b_new;
        h= h_new;
% If not (rand () <= CR) || (kr == randkr)
        b=OPT (1, kr);
        h=OPT (2, kr);
% NECESSARY EQUATIONS SHOULD BE WRITTEN in HERE
% Generating the new solution matrix
% Comparing new and initial solution matrix, and choosing the best one.
End
```

Fig. 2. Differential Evolution Pseudo Code

2.2.3 Hybrid algorithm (TLBO-Jaya)

Hybrid algorithms are generally formed by combining various algorithms within themselves. In order to develop their structures, it can be combined 2 or more algorithms [43]. These algorithms can be made by changing 1 phase of 2-phase algorithms. For example, there is Teaching and Learning phase in the TLBO algorithm, and if a Hybrid Algorithm is desired, 1 phase from other algorithms is added instead of the Learning phase (it can be Jaya) and the algorithm is completed in this way. The efficiency of the algorithm is shown in comparison with the studies and it is observed that there is a more effective optimization process in general and it reaches the objective function more efficiently. Furthermore, Hybrid Algorithms of SA, HS and BBBC which have effective features when solving problems are developed [44].

.

.

Iteration Phase: # It include 2 phases and altering can be learning phase

Teaching Phase

$$X_{i,new} = X_{i,j} + rand\,()\left(X_{i,g_{best}} - |X_{i,j}|\right) - (TF)X_{i,ortalama}\ )$$

$$\mathrm{TF} = round\ (1 + rand\ ())$$

.....

Learning Phase $\Rightarrow$ Jaya Algorithm

$$\boxed{X_{i,new} = \begin{cases} AF_a < AF_b, & X_{i,j} + r()\left(X_{i,a} - X_{i,b}\right) \\ AF_a > AF_b, & X_{i,j} + r()\left(X_{i,b} - X_{i,a}\right) \end{cases}}$$

$\Downarrow$ (Changing between equations)

$$\boxed{X'_{i,new} = X_{i,j} + rand\,()(X_{i,g_{best}} - |X_{i,j}|) - r()(X_{i,g_{worst}} - |X_{i,j}|)}$$

Fig. 3. Hybrid Code equations' changing

Altering hybrid code can be like the Fig. 3 which includes TLBO and Jaya algorithm to combine.

2.2.4 Jaya algorithm

Jaya algorithm which has developed by Rao in 2016 [45] is a method that has a similar approach to the TLBO algorithm [33]. This algorithm is frequently used in engineering problems because its variables are collected in a narrow area and scanned, and thus efficient results are obtained. Using this algorithm is fairly straightforward to apply [46]. Jaya aims to reach the objective function in fewer iterations and it is called "Victory". Jaya equation is shown in Eq. (1).

$$X'_{i,new} = X_{i,j} + rand(\,)\left(X_{i,g_{best}} - |X_{i,j}|\right) - rand(\,)\left(X_{i,g_{worst}} - |X_{i,j}|\right) \tag{4}$$

$X'_{i,new}$ The new value of variable
$X'_{i,best}$: The i. design variable value, which is the best value for the objective function in the initial matrix
$X'_{i,worst}$ The i. design variable value, which is the worst value for the objective function in the initial matrix
$X'_{i,j}$ The value of the candidate solution i. and j. in the initial matrix
rand () Randomly assigned state between 0 and 1

2.2.5 Harmony search (HS)

Harmony Search algorithm which was inspired by musical tones and best-sounding situations was developed by Geem et al. [47]. Harmony Search has been used in miscellaneous areas [48] such as engineering problems [49], hydraulic system design [50,51], steel frames [52,53] as well as retaining walls to reach objective function. It has some equations for formulas such as PAR is known as Pitch Adjustment Rate, as well as HCMR, is known as Harmony Memory Consideration Rate which takes a number between 0 and 1. The harmony search equation is shown Eq. (2).

$$\boldsymbol{X'_{i,new}} = \begin{cases} HCMR > rand(), \; X_{i,min} + rand() \times \left(X_{i,max} - X_{i,min}\right) \\ HCMR < rand(), \; X_{i,k} + rand\left(-\frac{1}{2},\frac{1}{2}\right) \times PAR \times \left(X_{i,max} - X_{i,min}\right) \end{cases} \tag{5}$$

$X_{i,max}$ Maximum value of the i. design variable
$X_{i,min}$ Minimum value of the i. design variable
$X_{i,k}$ The value of the candidate solution i. and j. in the initial matrix

2.2.6 Flower pollination algorithm (FPA)

It is an algorithm created by taking into account the changes in color and scent, inspired by the characteristics of flowers [54]. It enables the analysis to be completed by forming local pollination and global pollination situations within the algorithm.

3. Numerical Examples

3.1. Comparison of Different Metaheuristic Algorithms in Beam Design

Table 2 shows the design values which are restraints, variables, constants as well as the cost of the material. Fig. 4 shows beam distributed load and cross-section.

Table 2. Beam design values

Explanation	Symbol	Unit	Value
Minimum section width	b_{wmin}	mm	250
Maximum section width	b_{wmax}	mm	400
Minimum section height	h_{min}	mm	400
Maximum section height	h_{max}	mm	600
Distributed load	q	kN/m	32
Beam length	L	m	6
Compressive strength of concrete	f_{ck}	MPa	25
Yield strength of concrete	f_{yk}	MPa	420
Specific gravity of steel	γ_s	t/m³	7.86
Clear cover	d	mm	30
Cost of concrete per unit volume	C_c	TL/m³	1400
Cost of steel per unit weight	C_s	TL/ton	15050
Cost of formwork material-labour	C_k, C_{ki}	TL/m²	104-60

The objective function was generated to minimize design cost. Hence, it is important to find effective cross-sections and necessary reinforced areas for design. Moreover, these variables can change differences between their cost. For instance, when the cost of concrete increases, the ratio of concrete usage will decrease in the optimization process.

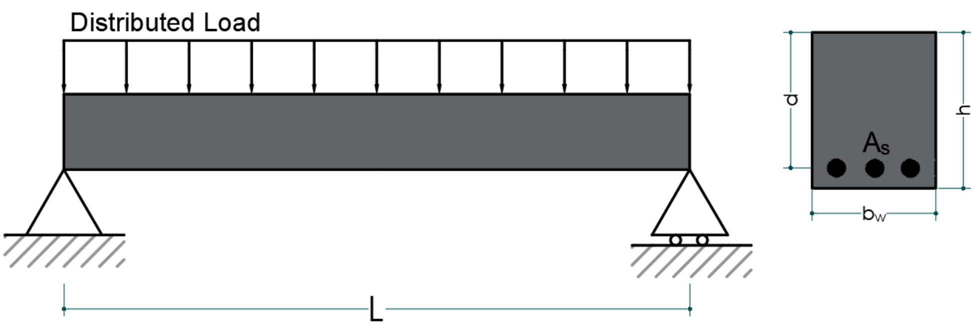

Fig. 4. Beam section and loading display

Table 3 demonstrates the result of optimizations with various algorithms which are Jaya, Teaching-Learning Based Optimization, Flower Pollination Algorithm, Hybrid Algorithm, Harmony Search as well as Differential Evolution. All of the algorithms are the approximately same cross-sections, reinforced area and cost.

Table 3. Optimization results for various algorithms

Explanation	b_w (mm)	h (mm)	Reinforced Area (mm²)	Cost (TL)
Jaya	250	410.5	1006.5	1668.8
TLBO	250	410.5	1006.5	1668.8
TLBO-Jaya	250	410.5	1006.5	1668.8
FPA	250	410.5	1006.5	1668.8
HS	250	410.5	1006.5	1668.8
DE	250	410.5	1006.5	1668.8

Table 4 illustrates the cases which are related to different running. Additionally, Case-1 has 5 runs, Case-2 has 10 runs, Case-3 has 15 runs as well as Case-4 has 20 runs. It can easily be seen that all cases show the average of runs to compare each other. Also, all cases have 2 different categories, namely Iter and S.D. Iter refers to how many average iterations the algorithm can reach the objective function. On the other hand, S.D. refers how many average (100) iterations the problem standard deviation will be 0. Also, the mean of standard deviation is taken for 100 iterations to compare the amount of changes. Standard deviation results are undeniable fact that when Hybrid and TLBO are used for problem, they generally take nearly the same value as the objective function.

Table 4. Comparisons of each algorithm

	Case-1		Case-2		Case-3		Case-4	
	Iter	S.D.	Iter	S.D.	Iter	S.D.	Iter	S.D.
JAYA	74.4	3.43	76.7	3.00	74.9	2.66	74.5	3.07
TLBO	61.4	0.016	59.6	0.02	58.8	0.016	59.9	0.02
HYBRID	35	0.0054	35.3	0.01	35.6	0.01	35.3	0.01
FPA	62.4	3.62	60	2.63	60.7	2.38	60.82	2.17
HS	10000+	4.5	10000+	3.7	10000+	3.58	10000+	4.04
DE	10000+	12.37	10000+	14.35	10000+	12.91	10000+	13.04

Although the HS approaches the objective function with less than 0.2% standard deviations (according to the average of 100 iterations) between 70-80 iterations, it is observed that it needs a lot of iterations to reach the objective function exactly. The reason for this may be that the maximum and minimum values of the variables are used in the formulas during the assignment of the cross-sections.

When the DE algorithm is used, it is similar to the HS algorithm in terms of the number of iterations to reach the objective function, and it is observed that this algorithm approaches the objective function with less than 0.75 standard deviations (according to the average of 100 iterations) in approximately 75-80 iterations. However, it is observed that there are large differences in the mean standard deviation values in 100 iterations. The most important factor affecting the formation of these differences is; It is expected to result from the analysis according to the randomly selected objective function value, instead of dealing with the best and worst values of the objective function in the iteration stage.

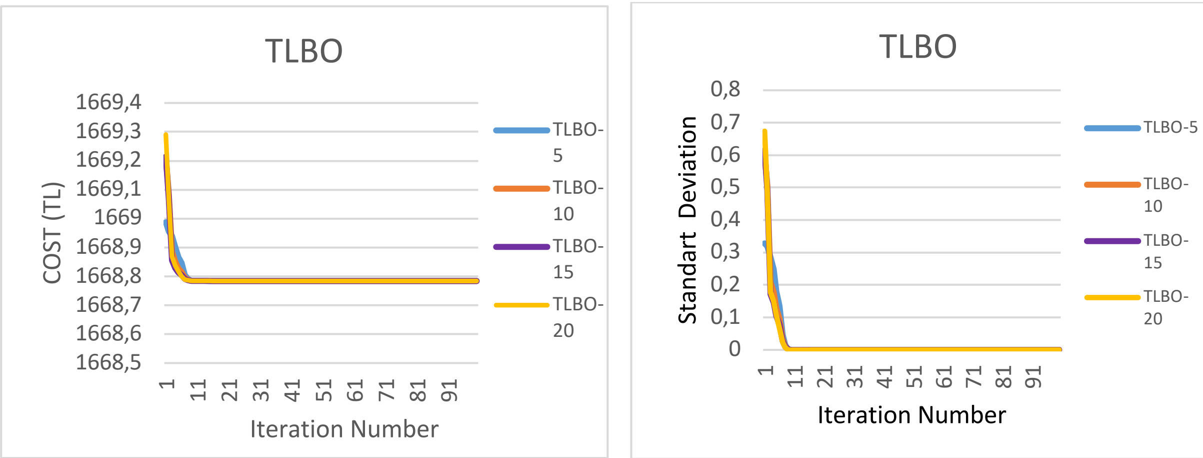

Fig. 5. TLBO results

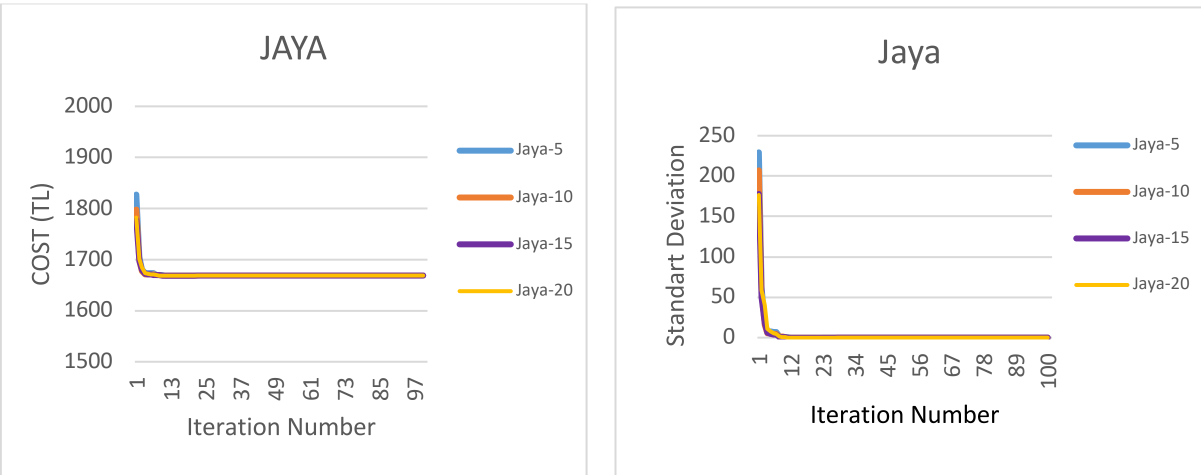

Fig. 6. Jaya Algorithms results

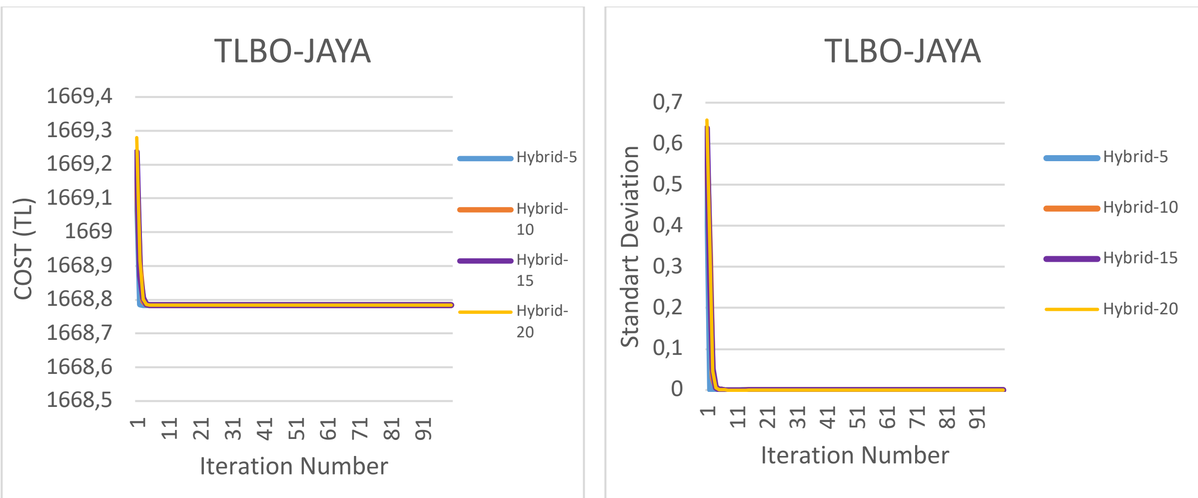

Fig. 7. Hybrid Algorithm results

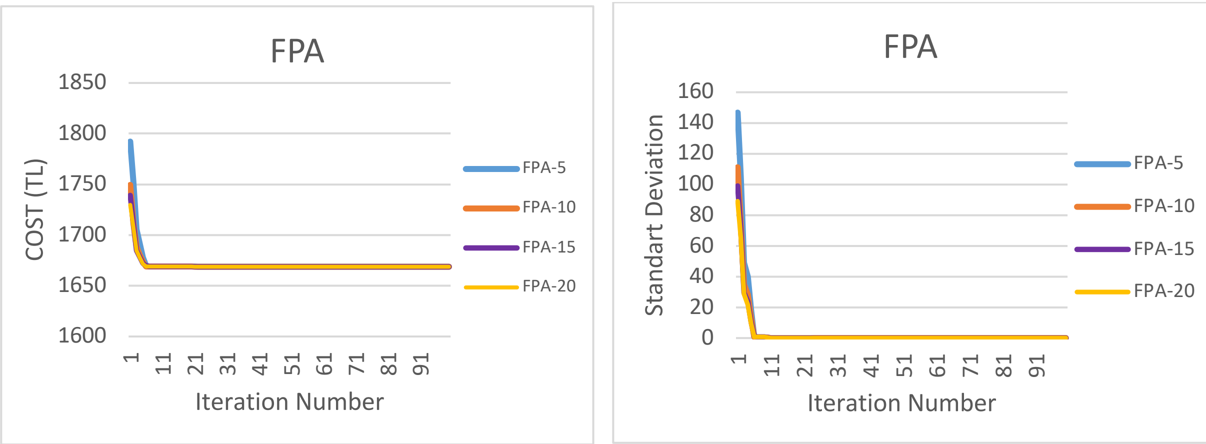

Fig. 8. Flower Pollination Algorithm results

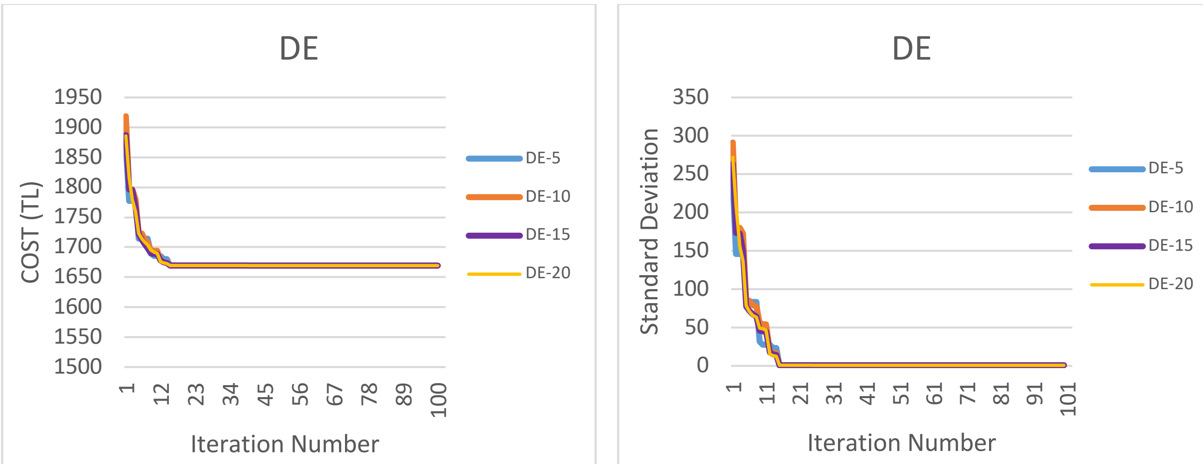

Fig. 9. Differential Evolution results

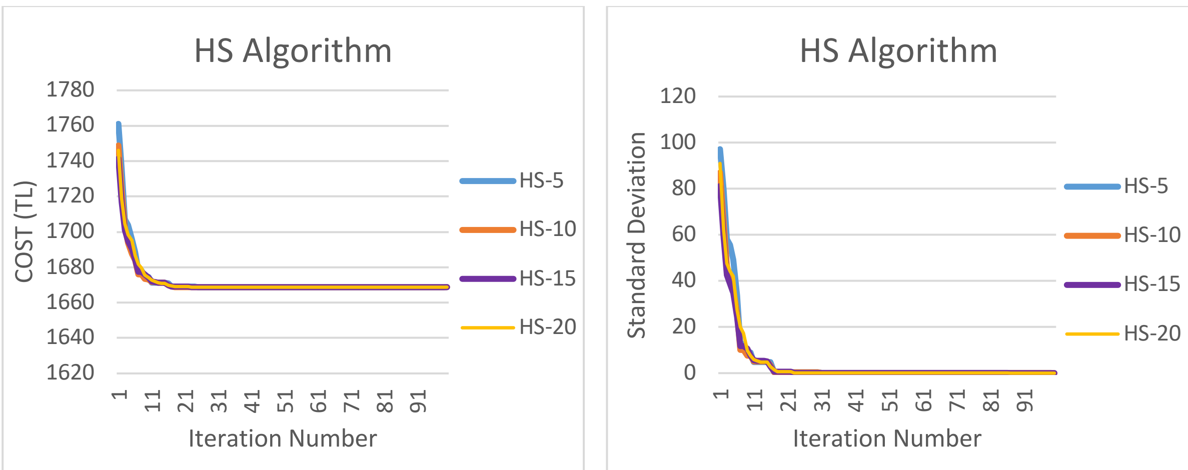

Fig. 10. Harmony Search results

Fig. (5-10) demonstrate the changing of cost and standard deviation by increasing iteration numbers. Looking at the line charts in more detail, at the beginning of the iterations, the beams' cost for Jaya algorithm is just over 1800 TL, FPA is just under 1800 TL as well as TLBO and Hybrid algorithms are approximately 1669 TL. For these graphs, differences between standard deviations also are observed and they have big differences compared to each other. At the first iteration, due to the fact that TLBO and Hybrid Algorithms have 2 phases, they approach objective functions easily compared to the other used algorithms. FPA is roughly 150 and Jaya is almost 250. DE has the biggest standard deviation. Harmony Search's standard deviations seem that it is an effective and good solution compared to DE, Jaya, and FPA. However, it cannot reach the objective function with fewer iterations. The standard deviation of TLBO and Hybrid algorithm.

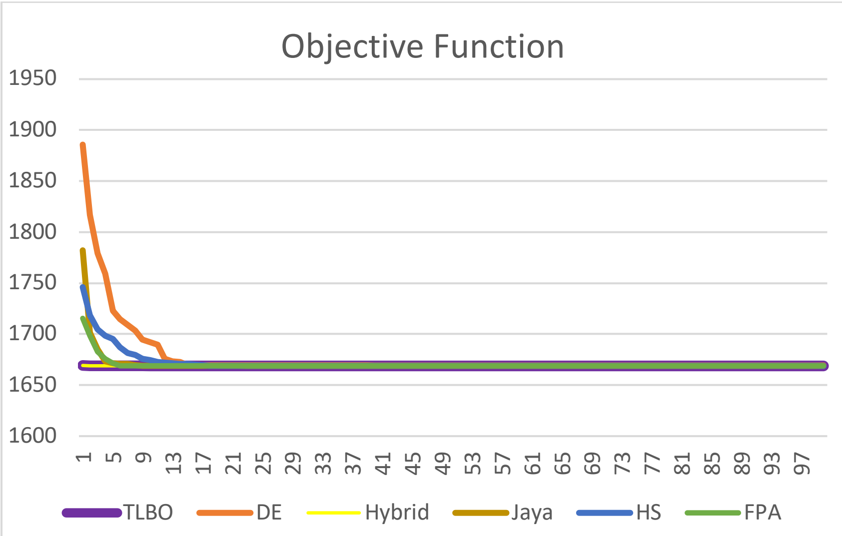

Fig. 11. Comparison of the algorithms in terms of cost

Fig. 11 compares by showing the variation of all used algorithms in this study according to the number of iterations. The X-axis shows the iteration number while Y-axis demonstrates the cost of the beam. It can easily be seen that Hybrid and TLBO algorithms are the most effective algorithms compared to the others. They generally approach the objective function in a few iterations. This feature allows complex problems to be solved easily and in a short time. Differential Evolution is the slowest one for approaching the objective function. Additionally, the initial cost is bigger than other algorithms and the second most expensive cost is from the Jaya algorithm, at approximately 1785 TL as well as the other 4 algorithms generally alter between 1668 TL and 1750 TL.

Fig. 12 illustrates the altering between all used algorithms in terms of standard deviation. Standard deviation helps us to compare differences between current matrix elements and objective function. If the standard deviation is close to 0, it can exactly say that there are not too many differences between these values and they are nearly similar. DE and Jaya make up a large proportion, at 275 and just over 175 respectively. FPA and HS have nearly 70-90 for both algorithms. TLBO and Hybrid constitute of smallest difference. Therefore, their finding costs are close to the objective function.

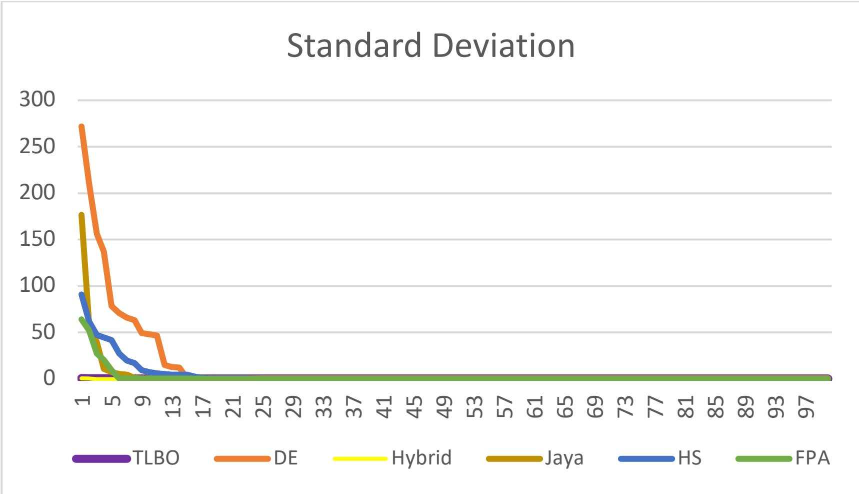

Fig. 12. Comparison algorithms in terms of standard deviations

3.2. Implementation of Beam Design according to Regulations Using Hybrid Algorithm

In the study in this section, the rectangular section reinforced concrete beam design is designed to be cost-optimal by using different regulations. ACI 318, TS500 and Eurocode 2 regulations have been added to the program for analysis of separate formulations. Hybrid Algorithm created by combining Metaheuristic Algorithms is used to achieve the objective function. The use of this type of algorithm is because it gives more efficient results than other algorithms. In section 4.1, it is observed that Hybrid Algorithm achieved much more efficient results than other algorithms, and in addition, it is observed that it reached the objective function in approximately 75% shorter iterations compared to the second-best algorithm (TLBO).

Maximum-minimum value of sections, distributed load value, the concrete class used, steel class yield strength, clear cover, steel specific gravity, and concrete-steel-formwork costs will be used as given in Table 5. In addition, cost changes that would occur if different concrete classes are used under the same specifications are also applied. These cost changes, the cost increases as the concrete compressive strength increases, the reinforcement class used will not change and there will be no difference in cost. However, with the change in the concrete class, the values to be used in the formulas will change and there will be differences in the cross-section dimensions and the reinforcement areas to be used. Due to these differences, there will be a difference in the cost value required for the beam design. Concrete classes of C25/30, C30/37 and C35/45 will be used for this study. With the change in concrete compressive strength, differences in objective functions can be observed. Table 5 shows the cost of concrete types which are increasing with the rise of the concrete strength.

Table 5. The cost of concrete

Concrete Classes	Cost of concrete (TL/m^3)
C25/30	1400
C30/37	1460
C35/45	1575

Tables 6, 7 and 8 illustrate the changing between building codes according to cross-section, reinforced area (A_s) as well as cost by using different types of concrete classes. It can easily be seen that when TS500 and Eurocode 2 are used for this problem, their results are found approximately the same. In the design process, if the using concrete class increases, the necessary amount of cost design goes up too. Moreover, b_w (width) sections are the same in both all 3 building codes and 3 various concrete classes although h (height) sections alter by changing concrete class. When the using concrete class is increased, the cost of design also will go up.

Table 6. Result of Hybrid Algorithm by using C25/30 Concrete

	b_w (mm)	h (mm)	A_s (mm²)	Cost (TL)
Eurocode	250	456.33	1060.3	1814.2
TS500	250	456.45	1060.6	1814.7
ACI 318	250	410.5	1006.5	1668.8

Table 7. Result of Hybrid Algorithm by using C30/37 Concrete

	b_w (mm)	h (mm)	A_s (mm²)	Cost (TL)
Eurocode	250	437.92	1091.6	1832
TS500	250	438.1	1091.9	1832.4
ACI 318	250	400	1019.1	1688.8

Table 8. Result of Hybrid Algorithm by using C35/45 Concrete

	b_w (mm)	h (mm)	A_s (mm²)	Cost (TL)
Eurocode	250	418.7	1136.2	1887.8
TS500	250	418.8	1136.4	1888.3
ACI 318	250	400	1003.5	1761.1

Fig. 13 shows the costs according to building regulations and the different classes of concrete. It is fact that ACI 318 is the least amount compared to others.

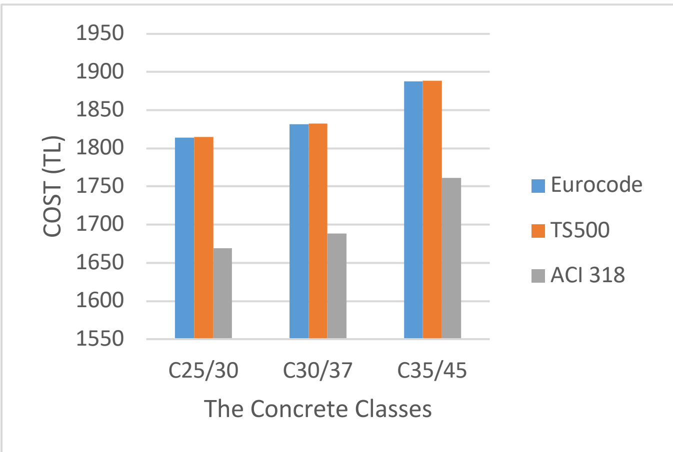

Fig. 13. The costs according to building codes

4. Discussion of Results

In this study, the rectangular reinforced concrete beam is designed ACI 318, TS500 and Eurocode 2. The first study is that ACI 318 regulation rules and boundary conditions, and algorithms are written in such a way that the system had the most optimum cross-section values

and reinforcement area in terms of cost. These algorithms are performed using a Matlab program with different metaheuristic algorithms such as TLBO, Hybrid, Jaya, FPA, HS, DE. Each algorithm differs from the other due to its features such as having different formulas and being of 1 or 2-phase algorithms. Using different algorithms in this way; When the algorithms reach the objective function, the section sizes, reinforcement areas and cost values are compared, as well as the comparison of how many iterations the algorithms reached to the optimum value and how close they got to the objective function. As a result of the optimization, it is observed that the dimensions of the beam sections, the reinforcement area to be used in the beam and the total cost value for the system are the same for each algorithm. In the case of applying a distributed load of 32 kN/m to a 6-meter-long beam, the beam width is calculated as 250 mm, the height is calculated as 410.5 mm, and the reinforcement area is calculated as 1006.5 mm^2. In the case of cost calculation, after these values are found, an expense of approximately 1668.8 TL will be expected. While TLBO and Hybrid algorithms, which have 2 phases, reach this cost value in very few iterations, it takes a little longer for other algorithms to reach the objective function. Also, FPA reaches the objective function about the same iteration number. Looking at the other algorithms, Jaya generally reaches objective function roughly in 75 iterations. In the case of using HS and DE algorithms, it has been observed that reaching the objective function is more than 10,000 iterations. However, despite being like this, it is seen that the standard deviation values of HS are very close to the objective function on average. However, it is seen that the standard deviation value of DE for 100 iterations is higher than the others, and this may be because the objective function values chosen randomly in the formulas will affect the efficient finding of the sections. Jaya, on the other hand, appears to have reached the objective function in approximately 45 iterations, even though it seems expensive at first due to the randomly assigned values.
As the second study, a rectangular reinforced concrete beam design is applied by using different regulations depending on the same loadings and material properties. In these designs, the changes between them are controlled by using different concrete classes for each regulation. As the strength of the concrete used increases, the cost value also increases. In general, section dimensions, reinforcement area and cost results are approximately the same for Eurocode 2 and TS500, while ACI 318 takes different values according to these regulations and the cost value is calculated less. When controls are made between costs, it has been observed that ACI 318 has approximately 6.5%-8.2% less cost compared to other regulations. For C25/30 concrete class, Eurocode 2, TS500 and ACI 318 design costs take different amounts, at 1814.2 TL, 1814.7 and 1668.8 respectively. C30/37 concrete class, Eurocode 2, TS500 and ACI 318 design costs take different amounts, at 1832 TL, 1832.4 and 1688.8 respectively. C35/45 concrete class, Eurocode 2, TS500 and ACI 318 design costs take different amounts, at 1887.8 TL, 1832.4 and 1761.1 respectively.

5. Conclusion

As a result of this study, these findings were obtained.

- There are various metaheuristic algorithms that are inspired by nature. They can reach the objective function in the different iterations because of differences between their formulization and the number of stages and phases.

- Hybrid algorithms that combined with 2 or more algorithms generally reach the objective function the with least iterations compared to other used metaheuristic algorithms.

- Building codes can influence the design of structures because of their design properties. Therefore, when a system is designed, differences in cross-section dimensions, reinforcement area and cost values can be observed.

References

[1] Bekdaş, G., Niğdeli, S. M., Yang, X.S., A novel bat algorithm based optimum tuning of mass dampers for improving the seismic safety of structures. *Engineering Structures*, 159, 89-98, 2018.

[2] Wang, S., Peng J., Kang, S., Evaluation of Compressive Arch Action of Reinforced Concrete Beams and Development of Design Method. Engineering Structures, 191, 479-492, 2019.

[3] Yücel, M., Bekdaş, G., Nigdeli S. M., Minimizing the Weight of Cantilever Beam via Metaheuristic Methods by Using Different Population-Iteration Combinations. *WSEAS Transactions on Computers,* 19, 69-77, 2020.

[4] Nesheim S., Mela K., Malo K. A., Labonneta N., Optimization framework for cost and carbon emission of timber floor elements. *Engineering Structures*, 252, 113485, 2022.

[5] Seyyedabbasi, A., WOASCALF: A new hybrid whale optimization algorithm based on sine cosine algorithm and levy flight to solve global optimization problems. *Advances in Engineering Software,* 179-103272, 2022.

[6] Çoşut, M., Bekdaş, G., Niğdeli, S.M. Cost Optimization and Comparison of Rectangular Cross-section Reinforced Concrete Beams Using TS500, Eurocode 2, and ACI 318 Code. *Proceedings of 7th International Conference on Harmony Search, Soft Computing and Applications, Lecture Notes on Data Engineering and Communications Technologies,* 140, 83-91, 2022.

[7] Cakiroglu, C., Islam, K., Bekdaş, G., Billah, M., CO_2 Emission and Cost Optimization of Concrete-Filled Steel Tubular (CFST) Columns Using Metaheuristic Algorithms. *Sustainability*, 13 (14): 8092, 2021.

[8] Cakiroglu, C., Bekdaş, G., Kim, S., Geem, Z. W., Optimisation of Shear and Lateral-Torsional Buckling of Steel Plate Girders Using Meta-Heuristic Algorithms. *Applied Sciences*, 10 (10): 3639, 2020.

[9] Cakiroglu, C., Islam, K., Bekdaş, G., Kim, S., Geem, Z. W., Metaheuristic Optimization of Laminated Composite Plates with Cut-Outs, *Coatings*, 11 (10): 1235, 2021.

[10] Chakrabarty, B. K., Model for Optimal Design of Reinforced Concrete Beam. *Journal of Structural Engineering,*118 (11), 1992.

[11] Bekdaş, G., Niğdeli, S. M., Optimum design of reinforced concrete beams using teaching-learning based optimization. *International Conference on Optimization Techniques in Engineering,* 3, 7-9, 2015.

[12] Zivari, A., Habibi, A., Khaledy, N., Development of an analytical method for optimum design of reinforced concrete beams considering both flexural and shear effects. *Computers and Concrete an International Journal,* 117-123, 2019.

[13] Guerra, A., Kiousis, P. D., Design optimization of reinforced concrete structures. *Computer and Concrete*, 313-334, 2006.

[14] Chutani, S., Singh, J., Design Optimization of Reinforced Concrete Beams. *Journal of The Institution of Engineers (India): Series A*, 98, 429-435, 2017.

[15] Nigdeli, S. M., Bekdaş, G., Optimum Design of RC Continuous Beams Considering Unfavourable Live-Load Distributions, *KSCE Journal of Civil Engineering,* 21(4), 1410-1416, 2017.

[16] Coello, C. C., Hernandez, F. S., Ferrera, F. A., Optimal design of reinforced concrete beams using genetic algorithms. *Expert Syst. Appl.,* 12(1), 101-108, 1997.

[17] Ulusoy, S., Kayabekir, A. E., Bekdaş, G., Nigdeli, S. M., Metaheuristic algorithms in optimum design of reinforced concrete beam by investigating strength of concrete. *Challenge Journal of Concrete Research Letters*, 11 (2), 26-30, (2020).

[18] Ulusoy, S., Kayabekir, A. E., Bekdaş, G., Nigdeli, S. M., Optimum Design of Reinforced Concrete Multi-Story Multi-Span Frame Structures under Static Loads. *International Journal of Engineering and Technology*, 10 (5), 26-30, (2018).

[19] ACI 318-95., Building code requirement for structural concrete and commentary, 1995.

[20] Eurocode 2., Design of Concrete Structures – Part 1-1: General Rules and rules for buildings, EN-1-2, 1991.

[21] Turkish Standardization Institute., "Design and Construction of Concrete Structures", Ankara, Turkey, TS500, 2000.

[22] Pierott, R., Hammad, A. W. A., Haddad, A., Garcia, S., Falcon, G., A Mathematical Optimization Model for the Design and Detailing of Reinforced Concrete Beams. *English Structure,* 245, 112861, 2021.

[23] Kulkarni, A. R., Bhusare, V., Structural optimization of reinforced concrete structures. *Int. J. Eng. Res.,* 5(07), 123-127, 2016.

[24] Zhao, S., Guo, J., Investigation on Electrochemical Repair of Reinforced Concrete Structure Cracks and Their Bonding Performance. *Alexandria Engineering Journal*, 2022.

[25] Tabsh, S. W., Safety of reinforced concrete members designed following ACI 318 building code. *Engineering Structures*, 19(10), 843-850, 1997.

[26] Doğangün A., *Calculation and Design of Reinforced Concrete Structures*, Birsen Publishing and Distribution, 2019.

[27] Xu, A., Li, S., Fu, J., Misra, A., Zhao, R., A hybrid method for optimization of frame structures with good constructability. *Engineering Structures*, 276,115336, 2023.

[28] Ma, Y., Chen, R., Bai, J, Zuo, W., Shape optimization of thin-walled cross section for automobile body considering stamping cost, manufacturability and structural stiffness. *Int. J. Automot. Technology*, 21 (503), 12, 2020.

[29] Zuo, W. J., Bai, J. T., Cross-sectional shape design and optimization of automotive body with stamping constraints. *Int. J. Automot. Technol.,* 17:1003–11, 2016.

[30] Jootoo, A., Lattanzi, D., Hybridizing topology optimization and evolutionary computation to support computer-aided engineering design. *Presented at the ASCE International Workshop on Computing in Civil Engineering,* 18–25, 2017.

[31] Zhang, Z, Ding, S., Jia, W., A hybrid optimization algorithm based on cuckoo search and differential evolution for solving constrained engineering problems. *Eng. Appl. Artif. Intell.,* 85, 254–68, 2019.

[32] Karaboga, D., An idea based on Honey Bee Swarm for Numerical Optimization. *Technical Report*-TR06, 1-10, 2005.

[33] Bekdaş, G., Nigdeli, M. N., Yücel, M., & Kayabekir, A. E., *Yapay Zeka Optimizasyon Algoritmaları ve Mühendislik Uygulamaları*, Seçkin Yayıncılık, Ankara, 2021.

[34] Rao, R.V., Teaching-learning based optimization: A novel method for constrained mechanical design optimization problems. *Computer Aided Design*, 43, 303-315, 2011.

[35] Öztürk, H.T., Dede, T., Türker, E., Optimum design of reinforced concrete counterfort retaining walls using TLBO, Jaya algorithm. *Structures*, 25, 285-296, 2020.

[36] Qin, A. K., Huang, V. L., Suganthan, P. N., Differential Evolution Algorithm with Strategy Adaptation for Global Numerical Optimization. *IEEE Transactions on Evolutionary Computation,* 13(2), 398-417, 2009.

[37] Storn, R., On the usage of differential evolution for function optimization. *In Biennial Conference of the North American Fuzzy Information Processing Society* – NAFIPS. IEEE, 519–523, 1996.

[38] Wu, C. Y., Tseng, K. Y., Truss structure optimization using adaptive multi-population differential evolution. *Struct. Multidiscip. Optim*., 42(4), 575–90, 2010.

[39] Donate, J. P., Li X., Sanchez. G. G., de Miguel, A. S., Time series forecasting by evolving artificial neural networks with genetic algorithms, differential evolution and estimation of distribution algorithm. *Neural Comput. Appl*., 22(1), 11–20, 2013.

[40] Lee, A. L., Nguyen, T. T., Ho, H. V., Dang, T. H, Bui, X. T., Static and frequency optimization of folded laminated composite plates using an adjusted Differential Evolution algorithm and a smoothed triangular plate element. *Compos Struct*., 127:382–94, 2015.

[41] Rogalsky, T., Derksen, R. W., Kocabiyik, S., Differential evolution in aerodynamic optimization. *in Proc. 46th Annu. Conf. of Can. Aeronaut. Space Inst.*, Montreal, QC, Canada, May, 29–36, 1999.

[42] Omran, M. G. H., Mahdavi, M., Global-Best Harmony Search. *Applied Mathematics and Computation*, 198, 543-656, 2008.

[43] Tang, H., Huynh, T.N., Lee, J., A novel adaptive 3-stage hybrid teaching-based differential evolution algorithm for frequency-constrained truss designs. *Structures*, 38, 934-348, 2022.

[44] Ficarelli, E., Lamberti, L., Degertekin S. O., Comparison of three novel hybrid metaheuristic algorithms for structural optimization problems. *Computers and Structures*, 244-106395, 2021.

[45] Rao, R. V., Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations,* 7, 19-34, 2016.

[46] Warid, W., Hizam, H., Mariun, N., Abdul-Wahab, N. I., Optimal Power Flow Using the Jaya Algorithm. *Energies*, 9,678, 2016.

[47] Yang, X. S., Harmony Search as a Metaheuristic Algorithm. *Studies in Computational Intelligence*, 191, 1-14, 2009.

[48] Bekdaş, G., Cakiroglu, C., Kim, S., Geem, Z. W., Optimization and Predictive Modelling of Reinforced Concrete Circular Columns. *Materials,* 15 (19): 6624, 2022.

[49] Bekdaş, G., Cakiroglu, C., Islam, K., Kim, S., Geem, Z. W., Optimum Design of Cylindrical Walls Using Ensemble Learning Methods. *Applied Science,* 12 (4): 2165, 2022.

[50] Bekdaş, G., Cakiroglu, C., Kim, S., Geem, Z. W., Optimal Dimensioning of Retaining Walls Using Explainable Ensemble Learning Algorithms, *Materials,* 15 (14): 4993, 2022.

[51] Geem, Z.W., Cho, Y.H. Optimal design of water distribution networks using parameter-setting-free harmony search for two major parameters. *J. Water Resour. Plan. Manag*., 137, 377–380, 2011.

[52] Cakiroglu, C., Bekdaş, G., Geem, Z. W., Harmony Search Optimization of Dispersed Laminated Composite Plates, *Materials,* 13 (12): 2862, 2020.

[53] Degertekin S.O. Optimum design of steel frames using harmony search algorithm. *Struct. Multidiscip. Optim*. 36:393–401, 2007.

[54] Yang, X. S. (2012). Flower pollination algorithm for global optimization. In Unconventional Computation and Natural Computation: 11th International Conference, UCNC 2012, Orléan, France, September 3-7, 2012. Proceedings 11 (pp. 240-249). Springer Berlin Heidelberg.