buildings

MDPI

Article

Critical Failure Mode Determination of Steel Moment Frames by Plastic Analysis Optimization Principles

Abdelmajeed Alkasassbeh *, Hatem H. Almasaeid and Bilal Yasin

Civil Engineering Department, Faculty of Engineering, Al Al-Bayt University, Mafraq 25113, Jordan; almasaeid.h@aabu.edu.jo (H.H.A.); yasinbilal@aabu.edu.jo (B.Y.)
* Correspondence: abdelmajeed-k@aabu.edu.jo

Abstract: Determining the failure or failure mode of structures has long been a challenge for civil engineers. Traditional methods for analyzing structures are costly and complex. Plastic analysis, which involves combining pre-defined mechanisms, offers a less complex approach. However, as the number of potential mechanism combinations, or the search space, increases with the growing complexity of structural members, the effectiveness of this method diminishes. To address this issue, optimizers have been applied in the field of structural engineering to efficiently solve problems with large search spaces. Population-based meta-heuristic algorithms are widely used for their reduced dependency on input parameters. This research focuses on implementing the plastic theory of steel frames using MATLAB software, employing virtual work concepts and pre-defined mechanism combinations. A novel binary dolphin echolocation algorithm is proposed based on the principles of the primary algorithm. This algorithm is then utilized to optimize the plastic analysis method and determine the failure load factor and critical failure mode for sample frames. Additionally, the grey wolf optimizer and whale optimization algorithm are applied to optimize the problem, and the performance of all three algorithms is compared. The results demonstrate that the proposed algorithm yields accurate results with a minor margin of error compared to the other two algorithms.

Keywords: plastic theory; steel moment frame; optimizer; dolphin echolocation; failure analysis

check for updates

Citation: Alkasassbeh, A.; Almasaeid, H.H.; Yasin, B. Critical Failure Mode Determination of Steel Moment Frames by Plastic Analysis Optimization Principles. *Buildings* **2023**, *13*, 2008. https://doi.org/10.3390/buildings13082008

Academic Editors: Krishanu Roy and Mohamed K. Ismail

Received: 27 April 2023
Revised: 30 June 2023
Accepted: 18 July 2023
Published: 7 August 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Structural members are of paramount importance in structural design due to their strength, durability, and versatility requirements [1–3]. Steel is an alloy of iron and carbon, and its mechanical properties make it an ideal material for building construction, specifically in seismic conditions [4–6]. They offer numerous benefits, including high fatigue resistance [7], buckling resistance [8], and fire resistance [9]. Structural steel members can be fabricated into a wide range of shapes and sizes, making them suitable for a variety of structural configurations, such as joints [10] and reinforcements [11]. Overall, structural steel members play a critical role in modern structural engineering and are essential to the construction of safe, efficient, and sustainable structures based on many studies performed in this regard.

Various researchers within the previous century have studied the problem of plastic theory by defining plastic hinges [12]. Over the years, many researchers have applied various methods to analyze steel structures. Neal and Symonds were the first to suggest the approach of combining pre-defined mechanisms [13,14], which was later expanded by others [15–18]. In this method, after the pre-defined mechanisms are combined, the mechanism corresponding to the minimum load factor is presented as the ultimate failure mechanism of the steel frame.

One problem with this procedure is the analysis of frames containing several spans and floors. Under these circumstances, combining the pre-defined mechanisms to detect the critical one is a tedious and, in some cases, impossible task. Hence, in these cases, a complementary process is required to determine the combination of mechanisms automatically

and quickly. To deal with this problem, artificial intelligence (AI) can be an appropriate choice. Generally, machine learning and deep learning approaches are two branches of AI, where numerous studies in different areas exist in the literature [19,20]. Artificial intelligence (AI) and machine learning (ML) are being increasingly used in structural engineering applications for tasks such as structural analysis, design optimization, and predictive maintenance [21–24]. One of the key advantages of AI and ML is their ability to process vast amounts of data and generate insights that can be used to improve structural performance and reduce maintenance costs. For example, neural networks as the most effective branch of ML algorithms can be used to analyze sensor data from structural health monitoring systems and identify patterns that indicate potential issues [25,26]. Additionally, AI can be used to optimize the design of complex structures by taking into account multiple factors such as load, materials, and geometry. As the field of AI and ML continues to evolve, it is expected that their use in structural engineering will become increasingly common and transformative.

Optimization problems in engineering (such as [27]) involve finding the most efficient and effective solutions to maximize performance, minimize costs, or achieve desired objectives within given constraints. Meta-heuristic optimizers are part of the algorithms in AI that are used to detect the optimum answers. There are different types of these algorithms, including the genetic algorithm, ant colony, bat, and so on. Meta-heuristic optimizers are becoming progressively admired in engineering processes because they have the following attributes:

1. Count on relatively simple concepts and are easy to apply.
2. Can get away from local optimum solutions.
3. Can be used in a variety of issues related to different categories.

Nature-based meta-heuristics, depicting biological or physical phenomena, solve optimization problems. A set of nature-inspired algorithms incorporates common methods that simulate the social attitude of animals. Some of these population-based algorithms are shown in Table 1 [28].

Kaveh et al. [20,29] and Kohama et al. [30] have presented studies on the determination of the failure load factor of plane frames employing the genetic algorithm. Kaveh and Jahanshahi have studied the limit plastic procedure of frames by means of the ant colony system and other similar algorithms [31–33]. Then, in 2016, Jahanshahi et al. calculated the failure load factor of plane frames using a neural network method and compared the results with the genetic and ant colony algorithms [34]. Kaveh et al. performed plastic analysis of plane frames using colliding bodies optimization and its extended version [35,36]. Further to the latest studies, Daryan et al. [37] modified dolphin echolocation algorithms to reach the minimum load factor and the corresponding failure mode of steel structures.

The main endeavor of researchers has been utilizing the genetic algorithm and ant colony system to optimize such problems in structural analysis. Recently, however, faster and more economical solutions have been required in structural analysis and design due to the development of larger and more complex structures. With this in mind, in the present study, a new algorithm with the principles of the dolphin echolocation algorithm (DE) is presented, which can be used for any binary problem. In order to evaluate the results of this algorithm, a comparison is made with two robust population-based algorithms presented in Table 1: the grey wolf optimizer (GWO) and the whale optimization algorithm (WOA). These algorithms define and optimize the same problem to detect the failure load factor and the critical failure mechanism of sample moment frames with rigid joints. The results are compared in terms of accuracy and convergence time.

Table 1. Population-based optimization algorithms.

Year	Concept	Algorithm
1995	Bird swarm	Particle swarm optimization (PSO)
2001	Honey bees	Marriage of honey bee optimization
2003	Fish swarm	Artificial fish-swarm algorithm (AFSA)
2005	Termite colony	Termite algorithm
2006	Ant colony	Ant colony optimization algorithm (ACO)
2006	Honey bee	Artificial bee colony (ABC)
2007	Parasitic wasp	Wasp swarm algorithm
2007	Monkey	Monkey search
2007	Wolf herd	Wolf pack search algorithm
2008	Bees	Bee collecting pollen algorithm (BCPA)
2009	Cuckoo	Cuckoo search (CS)
2009	Dolphin	Dolphin partner optimization (DPO)
2010	Bat herd	Bat-inspired algorithm
2010	Firefly	Firefly algorithm
2010	Animal group	Hunting search (HS)
2012	Bird mate	Bird mating optimizer (BMO)
2012	Krill herd	Krill herd (KH)
2012	Fruit	Fruit fly optimization algorithm (FOA)
2013	Dolphin	Dolphin echolocation algorithm (DE)
2014	Grey wolf	Grey wolf optimizer (GWO)
2016	Whale	Whale optimization algorithm (WOA)

2. Limit Analysis Principles Based on the Kinematic Theorem

The kinematic theorem of plastic analysis is a fundamental concept in structural engineering. It is used to analyze structures that undergo plastic deformation. According to this theorem, the internal forces, such as bending moments, shear forces, and axial forces, in a structure that undergoes plastic deformation can be determined based solely on its geometry and deformation. The material properties of the structure do not play a role in determining these internal forces.

In light of the complexities of kinematic theory, the contents of this section are briefly described in three sections. It is worth mentioning that this procedure has also been developed to solve other problems. For instance, Saedi Daryan and Palizi extended the same procedure for steel-braced frames [38] and developed the methodology with the aid of various nature-based optimization algorithms [39]. The classic plastic analysis theory was also extended to find the critical temperature of steel moment frames during a fire [40].

2.1. Pre-Defined Mechanisms

In the simulation process, concentrated horizontal and gravitational loads are applied on each story level and the middle of each opening, respectively. Figure 1 exemplifies the critical locations (plastic hinges) for a single-story frame with one opening. These points include the nodal points and the midpoints of the beams in the concentrated loading case.

According to the Neal and Symonds method [13,14], three pre-defined failure modes are considered: beam, floor, and node mechanisms, as depicted in Figure 2a–c, respectively, for a one-story frame with one opening.

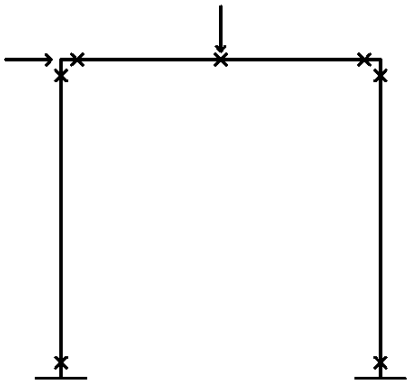
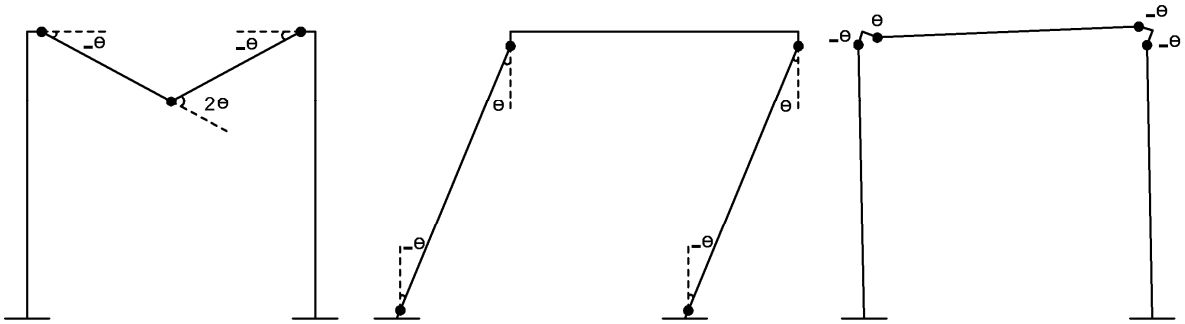


Figure 1. Plastic hinge possible locations.



(**a**) (**b**) (**c**)

Figure 2. Pre-defined mechanisms: (**a**) beam, (**b**) floor, and (**c**) node.

It is worth mentioning that for a frame with (a) openings and (b) floors, there are (ab) beam mechanisms and (b) floor mechanisms. Accordingly, the individual mechanisms are equal to [b (a + 1)]. The number of node mechanisms is the same value as the number of nodes in the structure minus the support nodes (at the base level). Although the node mechanisms do not create the work equation separately, they are the key to acquiring the minimum load factor in combined mechanisms.

The Kaveh and Jahanshahi method [29] has been used for coding simulation to create work equations related to plastic analysis theory.

2.2. Failure Loads

The failure load in a given mechanism is derived as Equation (1). The displacements and rotation of mechanisms are considered virtually, and the relationship between internal and external virtual work is written.

$$\lambda_c = \frac{\text{internal work expression}}{\text{external work expression}} \tag{1}$$

The external work is derived by the following equation, where P is the applied load and d is the vector of displacements at the location of the applied loads.

$$\text{external work expression} = P^T \cdot d \tag{2}$$

The internal work is also determined by the following equation, where M denotes the moment capacity of the r rotated members.

$$\text{internal virtual work} = M_p^T \cdot |r| \tag{3}$$

Considering that the plastic moment always resists joint rotations, the rotation values are always considered positive. Nodal mechanisms have been neglected during the combination of pre-defined mechanisms in the above equations. The reason that decreases the load factor is the node mechanisms; otherwise, the combination of two individual mechanisms results in a load factor value between the load factors of the two mechanisms. To implement node mechanisms automatically, the data of each joint and its containing node attached to each member are defined in a separate matrix.

2.3. Pre-Defined Mechanism Combinations

As the number of openings and floors grows, the number of mechanism combination cases increases. Thus, merging these pre-defined mechanisms to detect the correct failure mode will be tedious and time-taking. A frame with x individual mechanisms results in $(2^x - x - 1)$ possible combinations of independent mechanisms; however, not all are analytical failure mechanisms.

According to Neal and Symonds [13,14], possible failure mechanisms begin with a pre-defined mechanism and then combine with others to decrease the load factor due to their combination. When no more mechanisms exist to reduce the load factor, the same loop is conducted for the next mechanism until all conditions have been investigated. The minimum load factor obtained so far is introduced as the critical load factor. However, the problem can be expressed as an optimization one to deal with it more efficiently.

3. Meta-Heuristic Optimization Algorithms

Optimizers can be used to merge the pre-defined mechanisms. This study aims to present a new algorithm based on the dolphin echolocation optimization algorithm called the binary DE algorithm and compare its results with popular meta-heuristic algorithms. These optimization algorithms are the GWO algorithm and the WOA and are briefly described in the upcoming sub-sections.

3.1. Binary Dolphin Echolocation Algorithm

The dolphin echolocation optimizer simulates the dolphin hunting strategy. Dolphins create a sound to detect their target's position. A dolphin can produce click-like sounds. The frequency of these sounds is higher than that of the sounds used for socialization and differs among dolphin species. When the sonar strikes a target, partial wave energy returns back to the dolphin. When the echo arrives, the animal creates another sound. The duration of sending and receiving the sound enables the dolphin to measure the target distance. By continuously emitting and receiving sounds, dolphins can trace targets [41]. Different modifications have also been made to the original DE algorithm to improve its performance in different engineering problems [24,37,42].

In the original DE, the user sets a convergence curve. Then, the algorithm follows its rules based on the defined diagram. This method performs by assigning a higher likelihood of the best answer occurring than others. The algorithm measures the probability of selecting the optimum solution in each loop based on previous knowledge.

The probability function is as follows [41]:

$$PP(Loop) = PP_1 + (1 - PP_1)\frac{Loop_i^{power} - 1}{(LoopNumber)^{power} - 1} \quad (4)$$

where PP is the pre-defined probability, PP_1 is the convergence factor of the first loop, which is randomly chosen, $Loop_i$ is the current loop number, and power is the curve degree. As can be seen, the curve of the above equation has the power degree. LoopNumber is the number of loops in which the algorithm must reach a convergence point. The user sets this value. The flowchart of this algorithm is shown in Figure 3 [41].

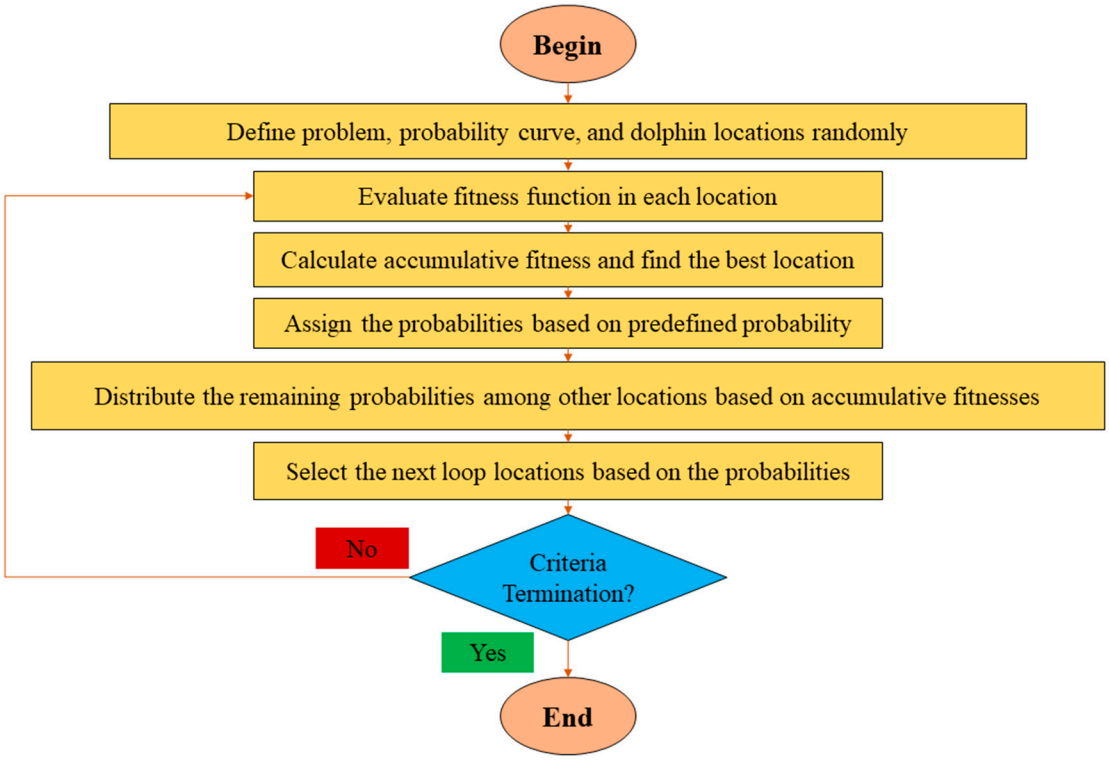


Figure 3. The dolphin echolocation procedure.

In the following, the parameters of the DE are denoted:

(A) Number of iterations (loops): The computational cost grows with the loop number.
(B) Definition of convergence criterion: The first try suggests a linear convergence, and then other values are tried for better answers.
(C) Effective radius (Re): This parameter is dependent on the size of the search area. Choosing a value of less than 1/4 of the search space is usually recommended.
(D) Number of locations (NL): A logical value should be taken based on trial and error.
(E) ε: This parameter is used to apply random variables, and it is better to use a value less than the minimum possible fit. These random variables are added to avoid local and fast convergence of the algorithm.

The superiority of meta-heuristic algorithms is that these algorithms are less dependent on input parameters and are easily operated with less possibility of being trapped in local optimum solutions.

In addition to having this feature, the DE algorithm has another feature that distinguishes it from other algorithms of the same category; this algorithm assigns a pre-defined probability to the best answer by determining the P(Loop) in each stage, where this value varies in each loop. The presence of this property in the DE algorithm controls the convergence process and causes the answer to be less likely to be local. The larger the probability of the first loop, the smaller the search space and the closer the algorithm is to the final answer. On the other hand, if more than a reasonable limit is selected, it can possibly become stuck in the local optimum solution (due to its faster convergence) and neglect some parts of the search space. The pre-assigned probability is suggested to be 0.1 or 10% in most cases.

Despite the superior capabilities of the DE algorithm and its high strength in solving various complex problems, especially in finding the optimal size and layout of lateral bearing systems, tackling binary problems with the aid of this algorithm has received less attention from previous researchers. Hence, in this research, by making modifications to DE, this algorithm is first developed for binary problems and then applied to the problem

of plastic theory of steel frame structures. The code is implemented in MATLAB software and can be used for any other binary problems.

First, the probability triangle (Figure 4) suggested in the dolphin algorithm is removed. Since there are only two options, 0 and 1, in the search space for each variable, using a neighbor triangle that distributes the fitness in the neighborhood of each alternative is practically meaningless.

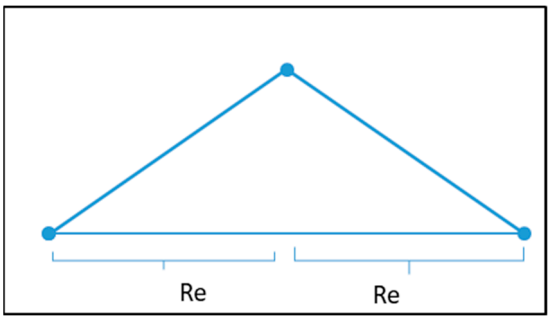


Figure 4. Triangular probability distribution of DE.

In the next step, changes have been made to the definition of (PP_1). As mentioned, the probability of the answer in the first loop is assumed to be 10%. In the binary case, this assignment causes errors during the convergence process. In the DE algorithm, for each variable, (PP_1) is the probability assigned to the best solution in the first step, and $(1 - PP_1)$ is the probability assigned to other options. Since there are only two options in binary space, (PP_1) is the probability of the superior solution, and $(1 - PP_1)$ is the probability of the other one. Considering that the probability of the superior solution should be more significant, (PP_1) must be higher than 50%. Hence, the algorithm holds the appropriate response with a probability above 50%.

In order to prevent premature convergence, the parameter (ε), which was responsible for adding random responses to the algorithm, is modified as Equation (5):

$$\varepsilon = (1 - PP_1) \cdot \text{Random} \tag{5}$$

where (Random) is the ratio that the number of random responses changes accordingly and (ε) is the percentage of random responses in each step that gradually reaches zero.

According to the definitions, for each variable, at each step, the probability of the superior option is calculated by Equation (6):

$$PP = (PP(Loop_i)) \cdot (1 - \varepsilon) + 0.5\ \varepsilon \tag{6}$$

As a result, the probability of the other response is determined by Equation (7):

$$PP' = (1 - PP(Loop_i)) \cdot (1 - \varepsilon) + 0.5\ \varepsilon \tag{7}$$

3.2. Grey Wolf Optimizer

Grey wolves hunt in a pack. The groups contain a population of 5–12 members. The members of the group are ranked according to Figure 5.

Considering this figure, Alphas are the leaders of the groups and are in charge of decision making. Betas follow the Alphas' commands and help Alphas in decisions. Both male and female Betas are the best alternative candidates for leadership when an Alpha dies or becomes too old. The lowest-ranked wolves are Omegas, who have to follow others. Delta wolves are called under-command and are ranked before Omegas. Deltas must follow Alphas and Betas, but they have a higher rank than Omegas. The group follows some steps for hunting prey by first tracing and approaching the prey. Then, they follow and harass the prey to make it stop. Finally, they attack the target.

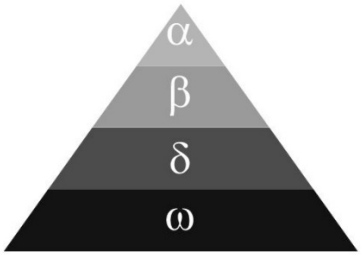


Figure 5. Ranking pyramid of wolves [43].

The hunting method of wolves is similar to an optimization process and can be modeled mathematically as the grey wolf optimizer. In this algorithm, the best answers are ranked according to the wolves' group ranking: the best responses are ranked in order as Alpha, Beta, and Delta, and the rest of the responses are considered as Omegas. To simulate mathematically, it is assumed that Alpha, Beta, and Delta contain better information about the position of the target. Hence, the first three responses are saved. Others update their position based on the three best responses.

Overall, the exploration phase starts with a random population of animals. The relations correspondent to this algorithm can be stated as follows:

$$\vec{Q} = \left| \vec{T} \cdot \vec{Y}_p(t) - \vec{Y}(t) \right| \tag{8}$$

$$\vec{Y}(t+1) = \vec{Y}_p(t) - \vec{R} \cdot \vec{Q} \tag{9}$$

$$\vec{R} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \tag{10}$$

$$\vec{T} = 2 \cdot \vec{r}_2 \tag{11}$$

$$\vec{Q}_\alpha = \left| \vec{T}_1 \cdot \vec{Y}_\alpha - \vec{Y} \right|, \vec{Q}_\beta = \left| \vec{T}_3 \cdot \vec{Y}_\beta - \vec{Y} \right|, \vec{Q}_\delta = \left| \vec{T}_3 \cdot \vec{Y}_\delta - \vec{Y} \right| \tag{12}$$

$$\vec{Y}_1 = \vec{Y}_\alpha - \vec{R}_1 \cdot \left(\vec{Q}_\alpha \right), \vec{Y}_2 = \vec{Y}_\beta - \vec{R}_2 \cdot \left(\vec{Q}_\beta \right), \vec{Y}_3 = \vec{Y}_\delta - \vec{R}_3 \cdot \left(\vec{Q}_\delta \right) \tag{13}$$

$$\vec{Y}(t+1) = \frac{\vec{Y}_1 + \vec{Y}_2 + \vec{Y}_3}{3} \tag{14}$$

where t is the current repetition, $(\vec{\mathbf{R}})$ and $(\vec{\mathbf{T}})$ are the coefficient vectors, $(\vec{\mathbf{Y}}_\mathbf{p})$ and $(\vec{\mathbf{Y}})$ are the position vector of the target and the animal, respectively. The vectors R and T are determined as Equations (10) and (11) and a in these equations decreases linearly from 2 to 0. $(\vec{\mathbf{r}}_\mathbf{1})$ and $(\vec{\mathbf{r}}_\mathbf{2})$ are random vectors in the range [0, 1]. In Equation (14), the position vector of wolves is updated, Alpha, Beta, and Delta determine the target position, and other animals update their position. Hence, a and T are two input parameters that need to be tuned to run the algorithm.

3.3. Whale Optimization Algorithm

Whales, as the largest mammals, are known for their great hunting skills. Humpback whales are the largest species among them which hunts groups of tiny snails and fish. These whales create several bubbles in circles and search for prey around them with this amazing strategy. Since this behavior is similar to an optimization procedure, it can be simulated for this purpose. When these animals determine the position of the prey, they make circles around it. The whale considers the object as the best candidate, and then, other

agents reform their position based on the position of the best answer. The formulation of the procedure is as follows:

$$\vec{Q} = \left| \vec{T} \cdot \vec{X^*}(t) - \vec{X}(t) \right| \tag{15}$$

$$\vec{X}(t+1) = \vec{X^*}(t) - \vec{R} \cdot \vec{Q} \tag{16}$$

where $(\vec{\mathbf{R}})$ and $(\vec{\mathbf{T}})$ are coefficient vectors, $(\vec{\mathbf{X}^*})$ is the position vector of the best response, and $\vec{\mathbf{X}}$ represents the location vector. t represents the current iteration. $(\vec{\mathbf{X}^*})$ should be reformed in each iteration if there is a more optimal answer. The vectors R and T are determined by Equations (17) and (18):

$$\vec{R} = 2\vec{a} \cdot \vec{r} - \vec{a} \tag{17}$$

$$\vec{T} = 2 \cdot \vec{r} \tag{18}$$

In these relations, $(\vec{\mathbf{r}})$ is a random vector in the range of [0, 1], and $(\vec{\mathbf{a}})$ is linearly decreased from 2 to 0. The bubble production of whales during the exploration phase is modeled mathematically according to the following:

- The surrender mechanism contraction is determined by decreasing the value of $(\vec{\mathbf{a}})$ in Equation (17). The $(\vec{\mathbf{R}})$ vector is also decreased by changing $(\vec{\mathbf{a}})$ ($(\vec{\mathbf{R}})$ is a random value in the range [−a, a]). Using the random values of $(\vec{\mathbf{R}})$ in the range of [−1, +1], the new position of the agent is determined somewhere between the main position of the agent and the position of the best current agent.
- In the position reforming of spirals, the algorithm first measures the distance between the whale at the point (X, Y), and then, the prey position is calculated at the point (X^*, Y^*). A spiral equation is determined to simulate the spiral mode of the whales as the following equation:

$$\vec{X}(t+1) = \vec{Q'} \cdot e^b \cdot \cos(2\pi l) + \vec{X^*}(t) \tag{19}$$

where $\vec{Q'} = \left| \vec{\mathbf{X}^*}(t) - \vec{\mathbf{X}}(t) \right|$ represents the distance between the ith whale and the target (the former best answer), b is a constant representing the shape of the spiral, and l is a random number in the range of [−1, +1].

Based on the variations in $(\vec{\mathbf{R}})$, the whales' exploration is performed according to each other's locations. Consequently, random values out of the range [−1, 1] can oblige agents to scatter from the reference whale. This process, alongside $(|\vec{\mathbf{R}}| > 1)$, provides a global search with the following expressions:

$$\vec{Q} = \left| \vec{T} \cdot \vec{X}_{rand} - \vec{X} \right| \tag{20}$$

$$\vec{X}(t+1) = X_{rand} - \vec{R} \cdot \vec{Q} \tag{21}$$

where $(\mathbf{X_{rand}})$ is the position vector of the existing iteration.

Similar to other optimization algorithms, the WOA starts with a random population of answers. In each step, the agents reform their position due to two reasons: agents of random selections and the current best answer. The parameter a is reduced from 2 to 0. Depending on the value of b, the algorithm is able to create spiral movements. Overall, only (R) and (T) as internal input parameters must be tuned in this algorithm.

4. Sample Frames Investigation

In order to confirm the accuracy of the limit plastic theory and compare the responses of the optimization algorithms introduced in the previous sections, examples studied by Kaveh and Jahanshahi [29] are presented herein. The exact failure load factor and critical failure mode determined from the plastic analysis by combining pre-defined mechanisms were already available. Accordingly, the results of the mentioned study have been compared with the proposed method in the current research. A comparison is also made between the performance of the binary DE presented in Section 3.1 with the GWO algorithm and the WOA.

In each example, the plastic moment capacity of each member and loading condition is expressed in each figure. No specific unit is considered for these examples.

Ten analyses were performed for each algorithm in each example, and the results are presented in graphs and tables. Root mean squared error (RMSE) and maximum absolute error (MAE) values as two statistical indices were employed to compare the performance of these algorithms in each example. These values have been used previously in various papers to compare the performance of algorithms [44]. The relationships provided for these indices are denoted as follows:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(h_i{}^2 - t_i{}^2)}{n}} \tag{22}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|h_i - t_i| \tag{23}$$

where h_i and t_i are actual and calculated values, respectively, and n is the total number of observations.

4.1. Frame One: Three Story and Two Span

The geometry of this frame is according to Figure 6. The loadings and the plastic moment capacity of each member are depicted in the figure. The exact response of the plastic analysis of this frame was already available [29]. The actual failure load factor of this frame is 1.9756. Table 2 illustrates each independent mechanism and the corresponding failure load factor.

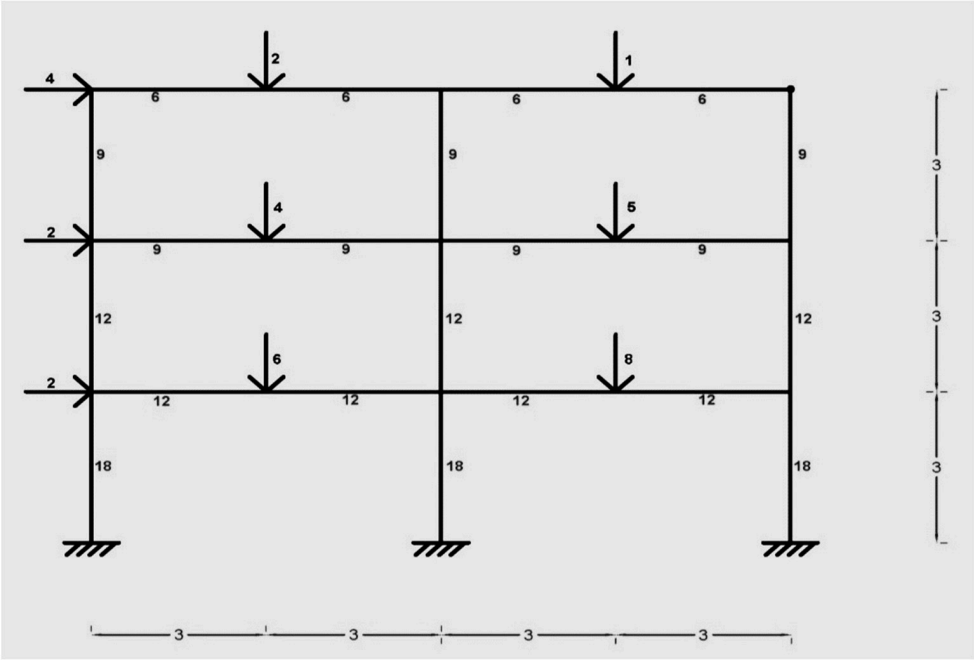


Figure 6. Frame of example one.

Table 2. Independent mechanisms of example one.

Mechanism No.	Mechanism Name/Location	Load Factor
1	Floor: Story 1	4.5
2	Beam: Story 1—left span	2.67
3	Beam: Story 1—right span	2.0
4	Floor: Story 2	4.0
5	Beam: Story 2—left span	3.0
6	Beam: Story 2—right span	2.4
7	Floor: Story 3	4.5
8	Beam: Story 3—left span	4.0
9	Beam: Story 3—right span	8.0

Each analysis was repeated ten times with the three algorithms. Figures 7–9 show the convergence diagrams of the DE algorithm, GWO algorithm, and WOA, respectively. Tables 3–5 describe the corresponding details of each diagram.

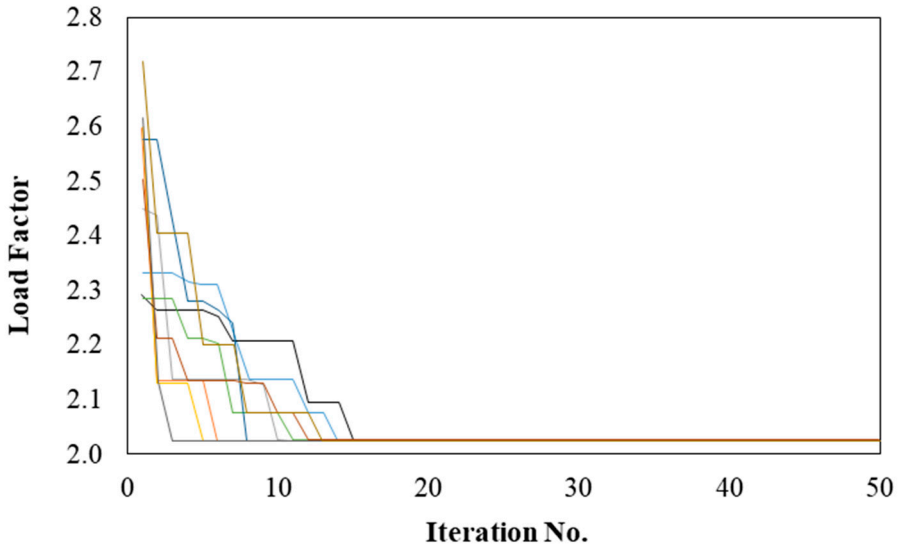


Figure 7. Results of ten analyses using binary DE in example one.

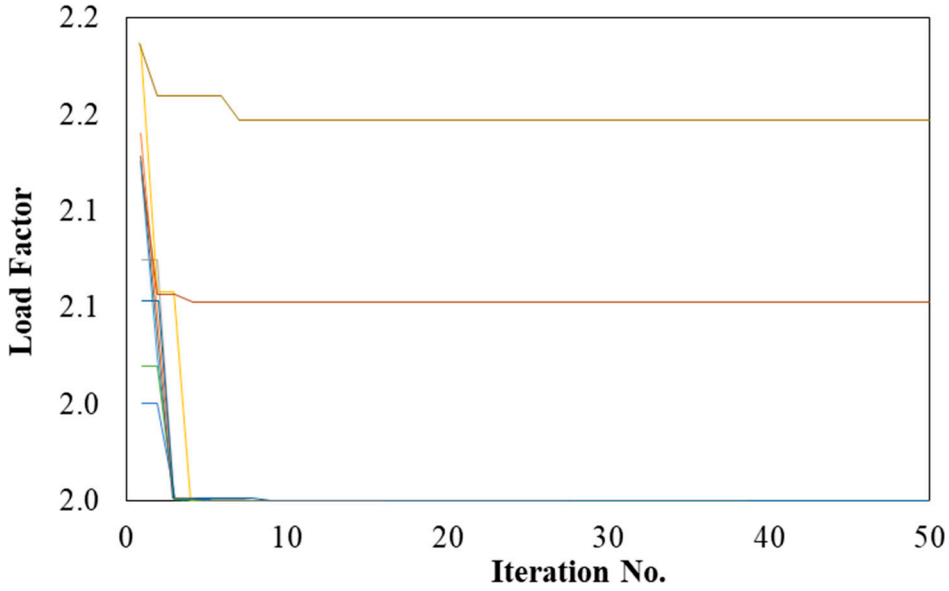


Figure 8. Results of ten analyses using GWO in example one.

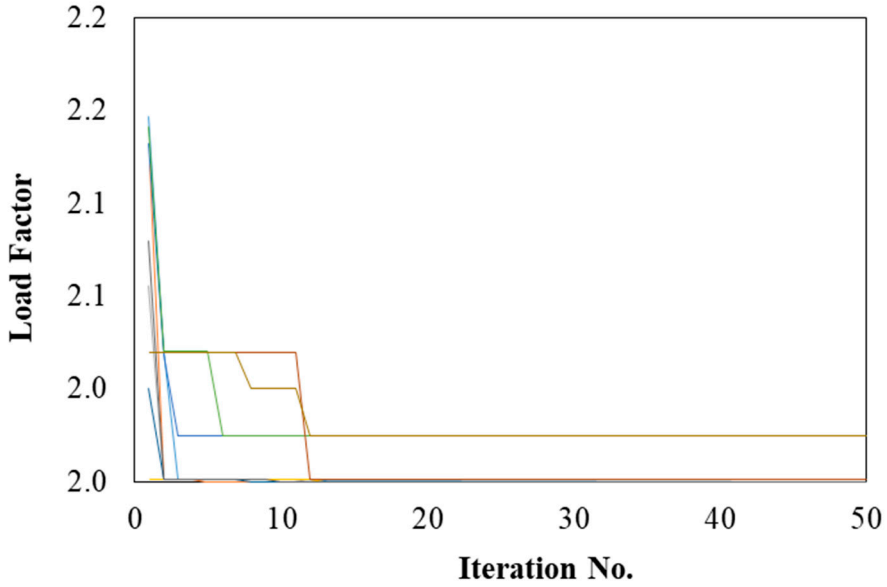


Figure 9. Results of ten analyses using WOA in example one.

Table 3. Details of Figure 7 results.

Analysis No.	Iteration No.	Failure Mode	Load Factor
1	14	{111111100}	1.9756
2	6	{111111100}	1.9756
3	10	{111111100}	1.9756
4	5	{111111100}	1.9756
5	15	{111111100}	1.9756
6	11	{111111100}	1.9756
7	8	{111111100}	1.9756
8	12	{111111100}	1.9756
9	3	{111111100}	1.9756
10	13	{111111100}	1.9756

Table 4. Details of Figure 8 results.

Analysis No.	Iteration No.	Failure Mode	Load Factor
1	3	{111111100}	1.9756
2	6	{111111100}	1.9756
3	8	{111111100}	1.9756
4	4	{111111100}	1.9756
5	6	{111111100}	1.9756
6	5	{111111100}	1.9756
7	9	{111111100}	1.9756
8	**4**	**{111110110}**	**2.0789**
9	5	{111111100}	1.9756
10	**7**	**{001111110}**	**2.1724**

This problem is defined in a binary format, and according to Tables 3–5, the failure mode is written in a set of binary codes. In each failure mode, zero means that the mechanism is not involved in the failure mode, and the number one represents its presence. Mechanisms 1 to 9 are identified in Table 2. The actual failure mode of example one is codified as {111111100}, which means that mechanisms one to seven exist in the failure mode and mechanisms eight and nine are absent.

Table 5. Details of Figure 9 results.

Analysis No.	Iteration No.	Failure Mode	Load Factor
1	13	{111111100}	1.9756
2	5	{111111100}	1.9756
3	12	{111111100}	1.9756
4	13	{111111100}	1.9756
5	**3**	**{001000000}**	**2.0000**
6	**6**	**{001000000}**	**2.0000**
7	8	{111111100}	1.9756
8	**12**	**{111111110}**	**1.9767**
9	10	{111111100}	1.9756
10	**12**	**{001000000}**	**2.0000**

The failure mode corresponding to the actual critical load factor is shown in Figure 10. The analysis results for the three algorithms revealed that the binary dolphin algorithm presented in this paper executed the exact answer in each of the ten analyses. Analyses 8 and 10 in the GWO algorithm presented a different failure mode even though the critical load factor was close to the actual value. In the WOA, four analyses, including numbers 5, 6, 8, and 10, did not have exact answers, and only analysis number 8 proposed a result very close to the actual response of the structure. By and large, in this particular example, the binary DE model illustrated zero error, the GWO model signified a 20% error, and the WOA had a 40% error in providing the exact failure response of the frame.

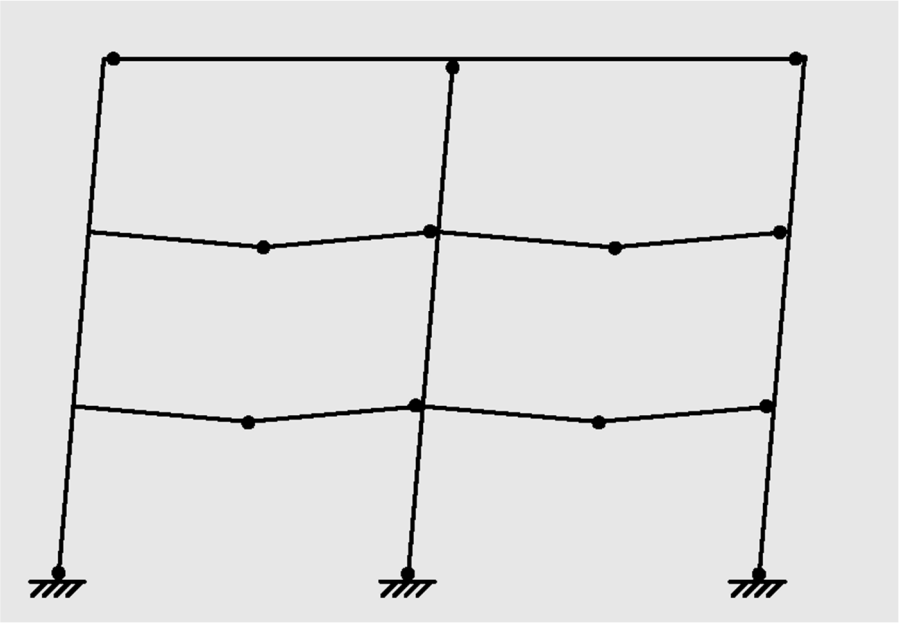


Figure 10. Critical failure mode of example one.

In the following example, a larger frame is considered to investigate the effect of independent mechanism increase in the error value of all algorithms.

4.2. Frame Two: Four Story and Four Span

Figure 11 shows the geometry, loadings, and plastic capacities of this frame. This frame is also one of the sample frames, and its actual load factor is 0.65. The corresponding critical failure mode of this frame is illustrated in Figure 12 [29].

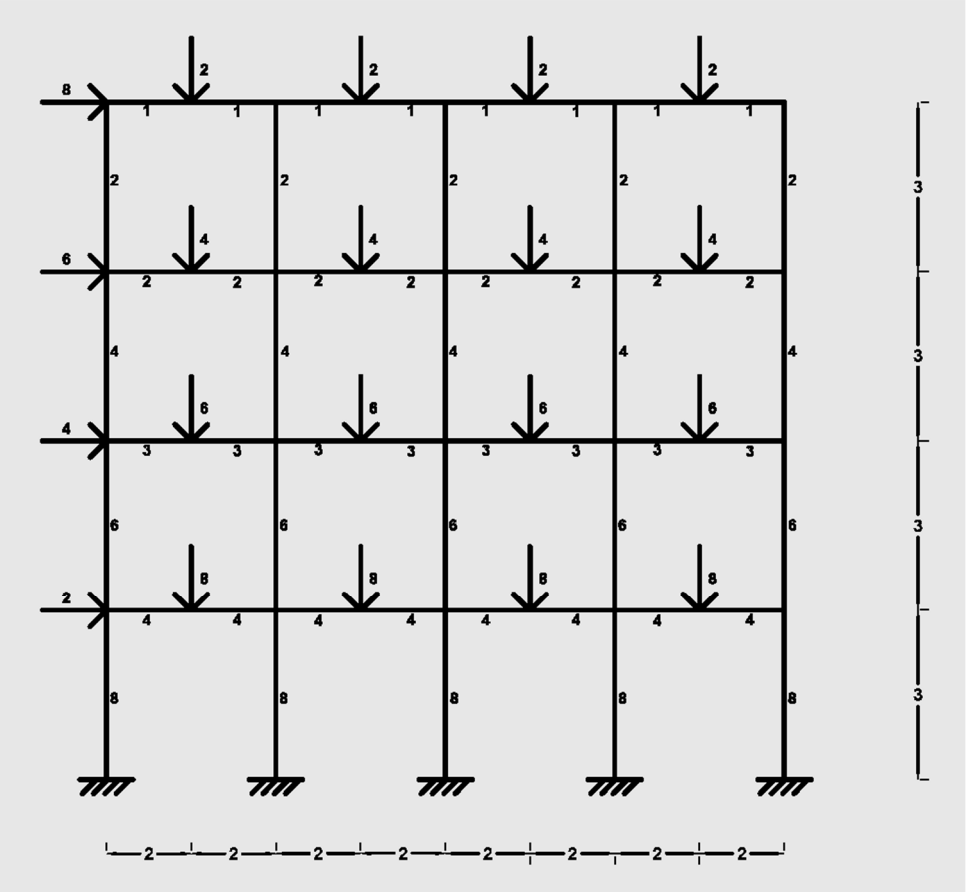


Figure 11. Frame of example two.

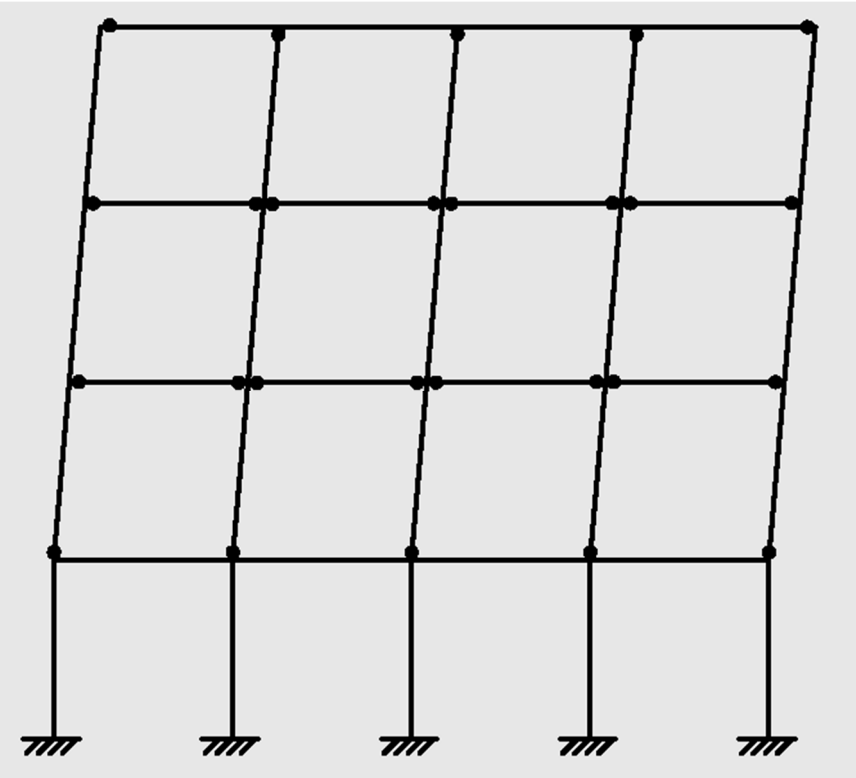


Figure 12. Critical failure mode of example two.

The mechanism identifications according to the code script are illustrated in Table 6. The span number in this table is from left to right.

Table 6. Independent mechanisms of example two.

Mechanism No.	Mechanism Name/Location	Load Factor
1	Floor: Story 1	1.33
2	Beam: Story 1—Span 1	1
3	Beam: Story 1—Span 2	1
4	Beam: Story 1—Span 3	1
5	Beam: Story 1—Span 4	1
6	Floor: Story 2	1.11
7	Beam: Story 2—Span 1	1
8	Beam: Story 2—Span 2	1
9	Beam: Story 2—Span 3	1
10	Beam: Story 2—Span 4	1
11	Floor: Story 3	0.95
12	Beam: Story 3—Span 1	1
13	Beam: Story 3—Span 2	1
14	Beam: Story 3—Span 3	1
15	Beam: Story 3—Span 4	1
16	Floor: Story 4	0.83
17	Beam: Story 4—Span 1	1
18	Beam: Story 4—Span 2	1
19	Beam: Story 4—Span 3	1
20	Beam: Story 4—Span 4	1

According to the procedure in the previous example, in this example, the convergence diagrams for each algorithm are given first, and after each diagram, its details are presented in tables. Figures 13–15 are the convergence diagrams for ten analyses using the binary DE algorithm, GWO algorithm, and WOA, respectively. The details of each diagram is also illustrated in Tables 7–9.

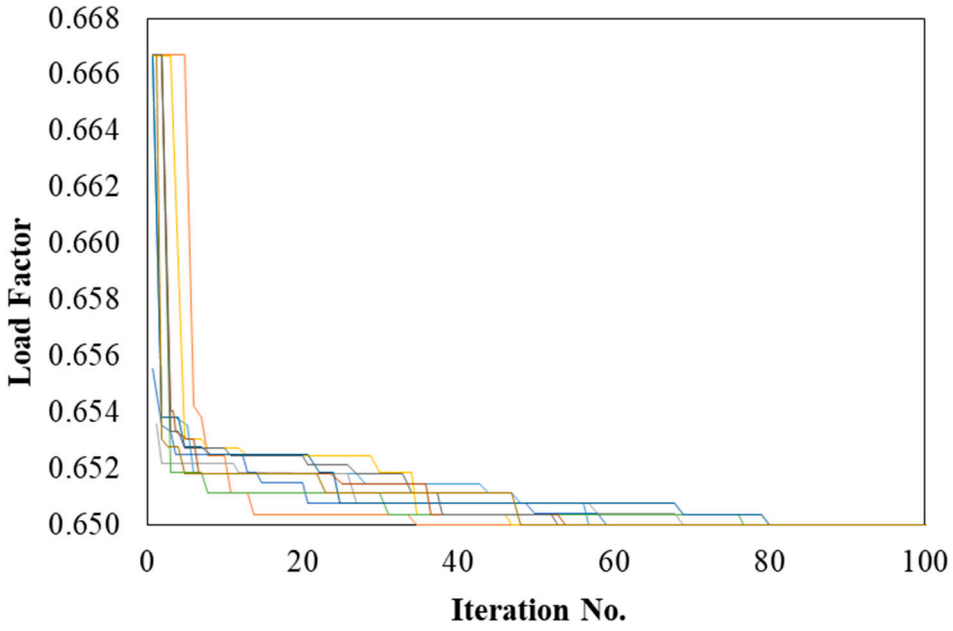


Figure 13. Results of ten analyses using binary DE in example two.

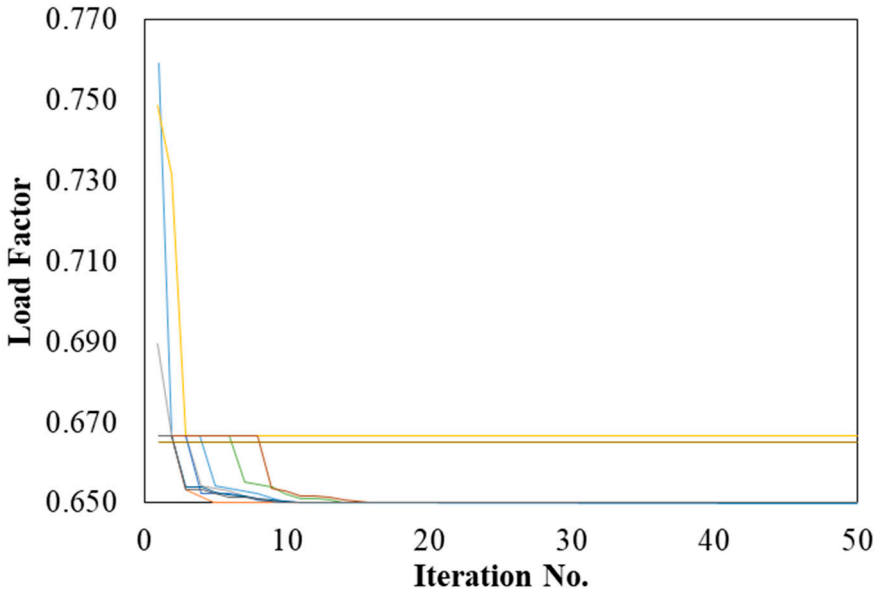


Figure 14. Results of ten analyses using GWO in example two.

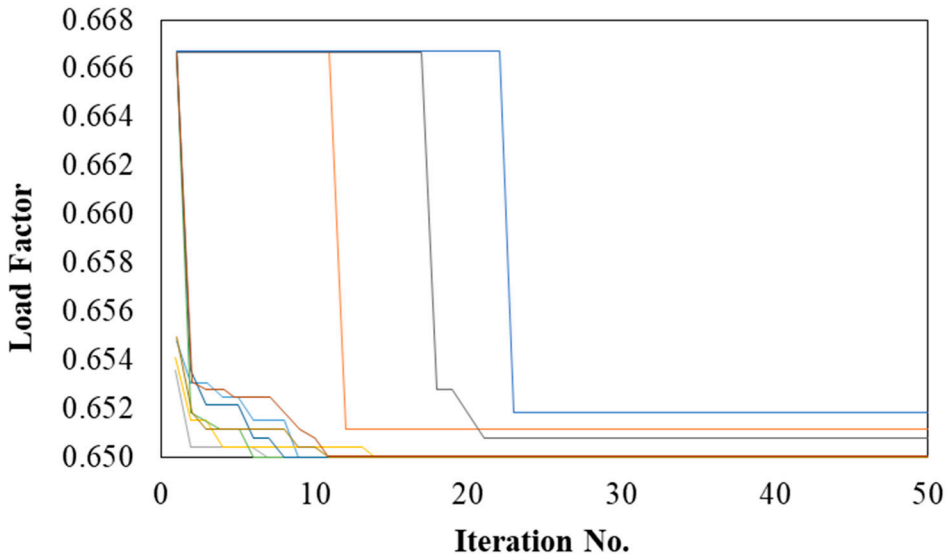


Figure 15. Results of ten analyses using WOA in example two.

Table 7. Details of Figure 13 results.

Analysis No.	Iteration No.	Failure Mode	Load Factor
1	57	{00000100001000010000}	0.6500
2	35	{00000100001000010000}	0.6500
3	58	{00000100001000010000}	0.6500
4	35	{00000100001000010000}	0.6500
5	59	{00000100001000010000}	0.6500
6	31	{00000100001000010000}	0.6500
7	69	{00000100001000010000}	0.6500
8	37	{00000100001000010000}	0.6500
9	38	{00000100001000010000}	0.6500
10	48	{00000100001000010000}	0.6500

Table 8. Details of Figure 14 results.

Analysis No.	Iteration No.	Failure Mode	Load Factor
1	11	{00000100001000010000}	0.6500
2	7	{00000100001000010000}	0.6500
3	10	{00000100001000010000}	0.6500
4	**3**	**{10000100111010010000}**	**0.6667**
5	10	{00000100001000010000}	0.6500
6	14	{00000100001000010000}	0.6500
7	11	{00000100001000010000}	0.6500
8	16	{00000100001000010000}	0.6500
9	10	{00000100001000010000}	0.6500
10	**1**	**{10000100111010010000}**	**0.6667**

Table 9. Details of Figure 15 results.

Analysis No.	Iteration No.	Failure Mode	Load Factor
1	9	{00000100001000010000}	0.6500
2	**12**	**{00000100011000010000}**	**0.6512**
3	7	{00000100001000010000}	0.6500
4	14	{00000100001000010000}	0.6500
5	**23**	**{00000110001000010101}**	**0.6519**
6	6	{00000100001000010000}	0.6500
7	8	{00000100001000010000}	0.6500
8	11	{00000100001000010000}	0.6500
9	**21**	**{00000100001010010000}**	**0.6508**
10	11	{00000100001000010000}	0.6500

Regarding the failure mechanisms, the GWO predicted {10000100111010010000} twice, which demonstrates that the pre-defined mechanisms 1, 6, 9, 10, 11, 13, and 16 are present in the final failure mode. However, according to the actual mechanism set, {00000100001000010000}, the failure mode contains the mechanisms of 6, 11, and 16. It shows that the GWO proposed one floor and three beams other than the critical failure mode. The binary DE determined an accurate answer in each of the ten analyses. It can be concluded that no error has been proposed in an example with a high number of pre-defined mechanisms. However, errors were observed again occasionally in the GWO algorithm. In this example, only two analyses (4 and 10) led to answers with different failure mechanisms, although the load factor had a proper answer compared to the actual value. In the WOA, three errors were observed, which was the highest error among these three algorithms, similar to the previous example.

4.3. Frame Three: Six Story and Two Span

This frame with 18 floors and two spans contains 18 pre-defined failure mechanisms. The frame with specifications is shown in Figure 16, and the identification of pre-defined mechanisms is also given in Table 10. The span number is from left to right in this table. The exact answer determined from the plastic analysis of this frame is also available [29], with the failure load factor of 1.2889. The corresponding critical failure mode is depicted in Figure 17. According to this figure and the identification of pre-defined mechanisms in Table 10, mechanisms number 1, 4, 7, and 10 exist in the actual failure mode.

According to the previous examples, each algorithm was analyzed ten times. The results are shown in Figures 18–20, the details of which are tabulated after each convergence diagram for the three algorithms in Tables 11–13.

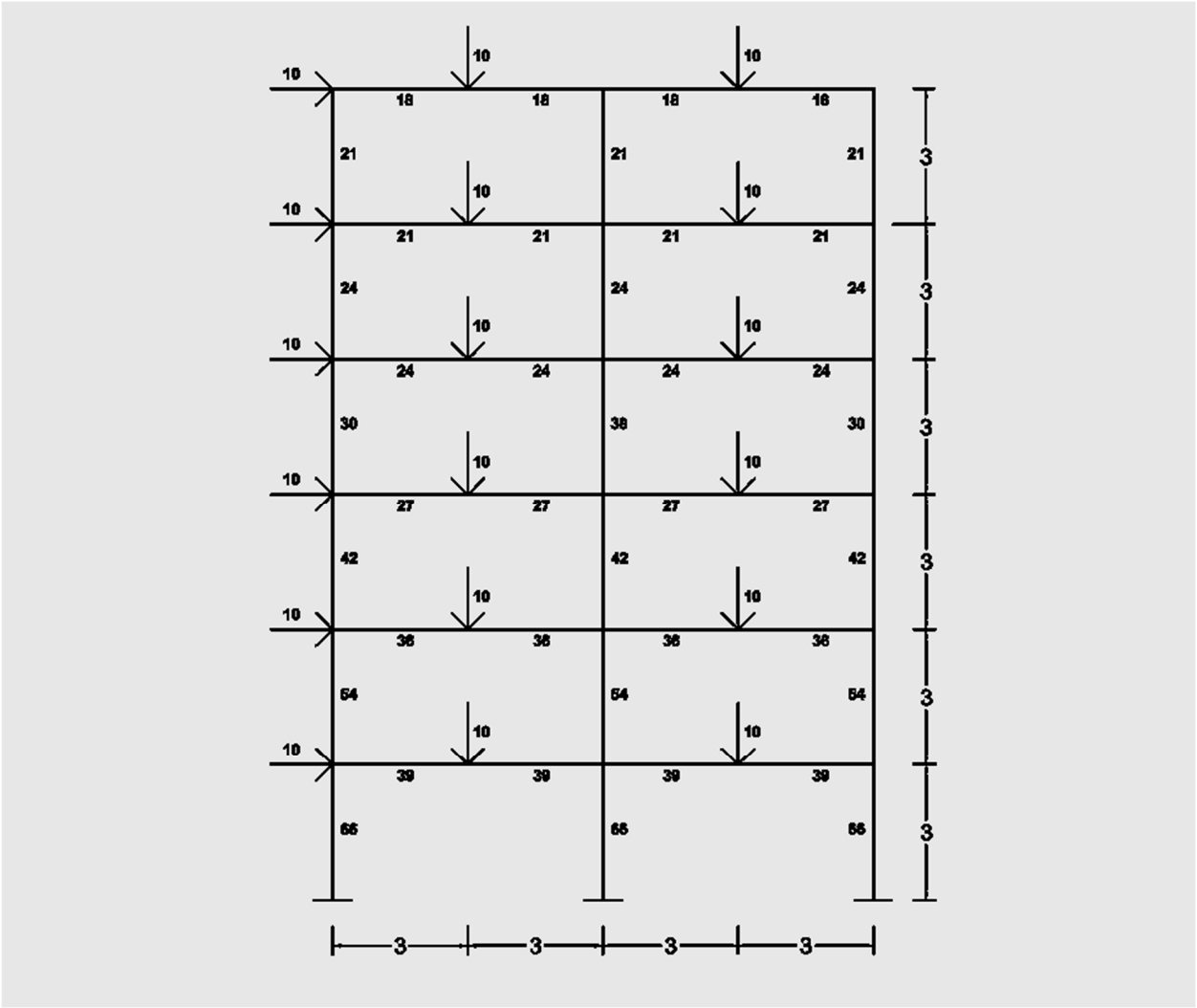


Figure 16. Frame of example three.

Table 10. Independent mechanisms of example three.

Mechanism No.	Mechanism Name/Location	Load Factor
1	Floor: Story 1	2.20
2	Beam: Story 1—Span 1	5.20
3	Beam: Story 1—Span 2	5.20
4	Floor: Story 2	2.16
5	Beam: Story 2—Span 1	4.80
6	Beam: Story 2—Span 2	4.80
7	Floor: Story 3	2.10
8	Beam: Story 3—Span 1	3.60
9	Beam: Story 3—Span 2	3.60
10	Floor: Story 4	2.00
11	Beam: Story 4—Span 1	3.20
12	Beam: Story 4—Span 2	3.20
13	Floor: Story 5	1.60
14	Beam: Story 5—Span 1	2.80
15	Beam: Story 5—Span 2	2.80
16	Floor: Story 6	4.20
17	Beam: Story 6—Span 1	2.40
18	Beam: Story 6—Span 2	2.40

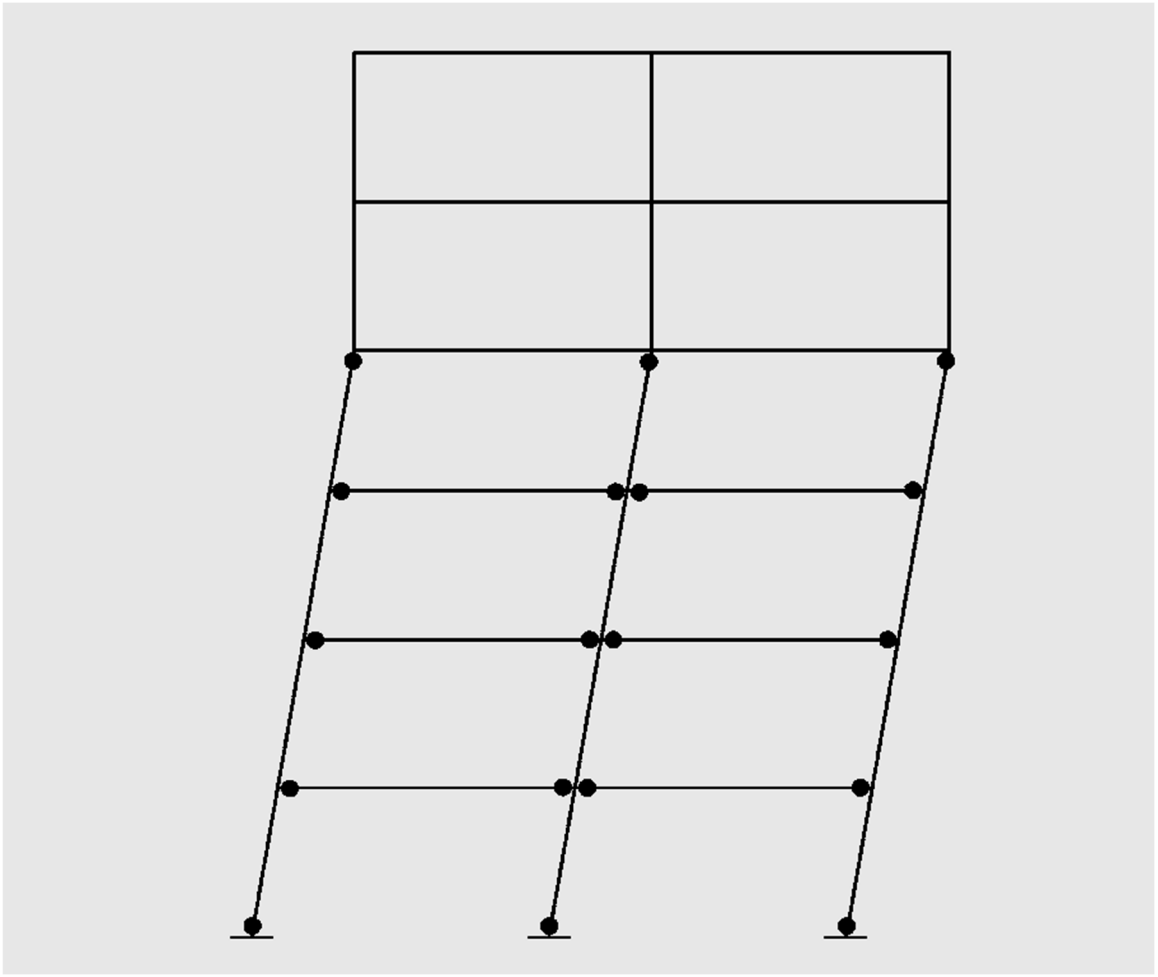


Figure 17. Critical failure mode of example three.

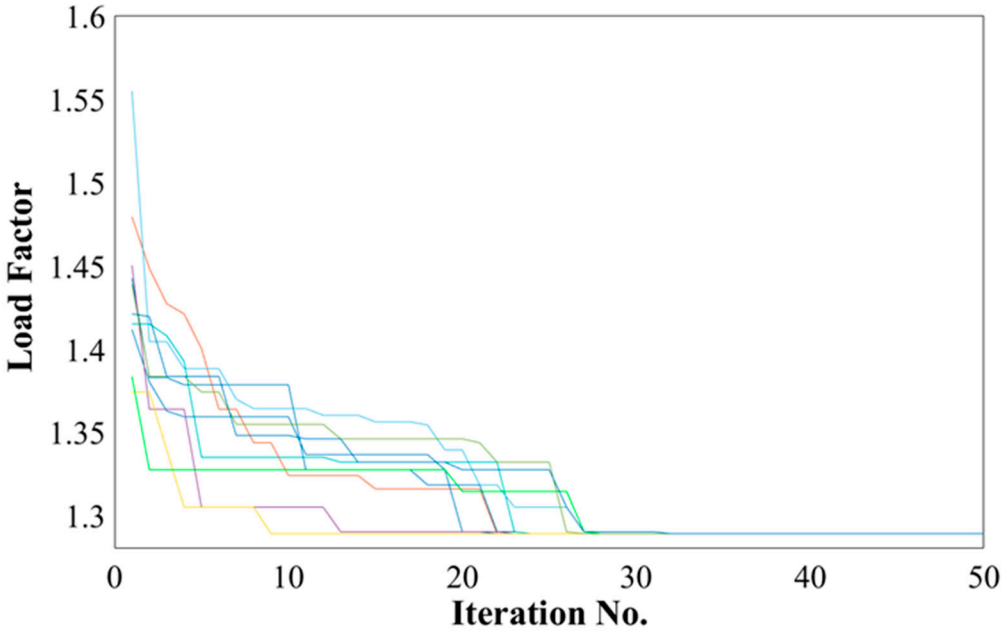


Figure 18. Results of ten analyses using binary DE in example three.

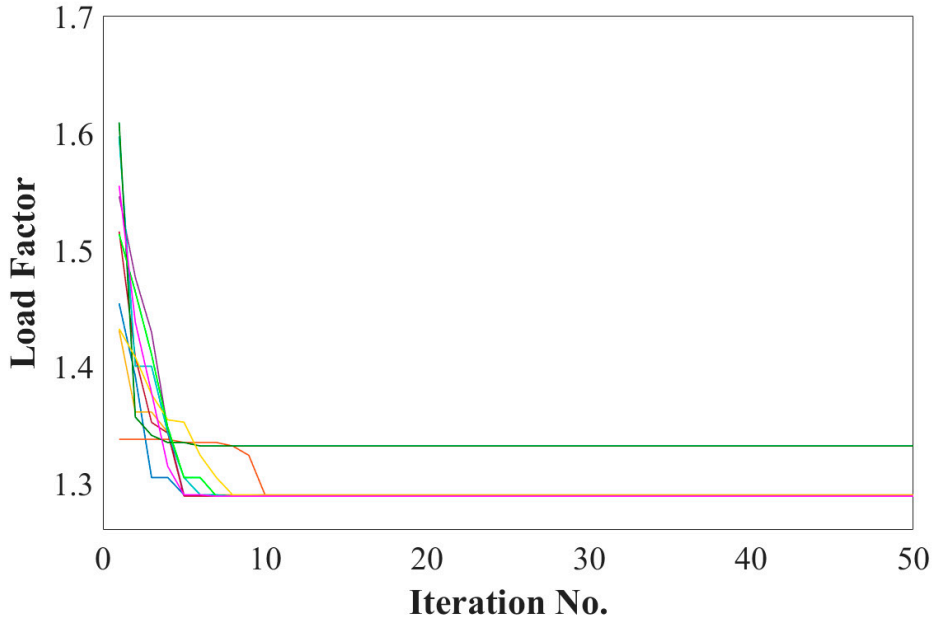


Figure 19. Results of ten analyses using GWO in example three.

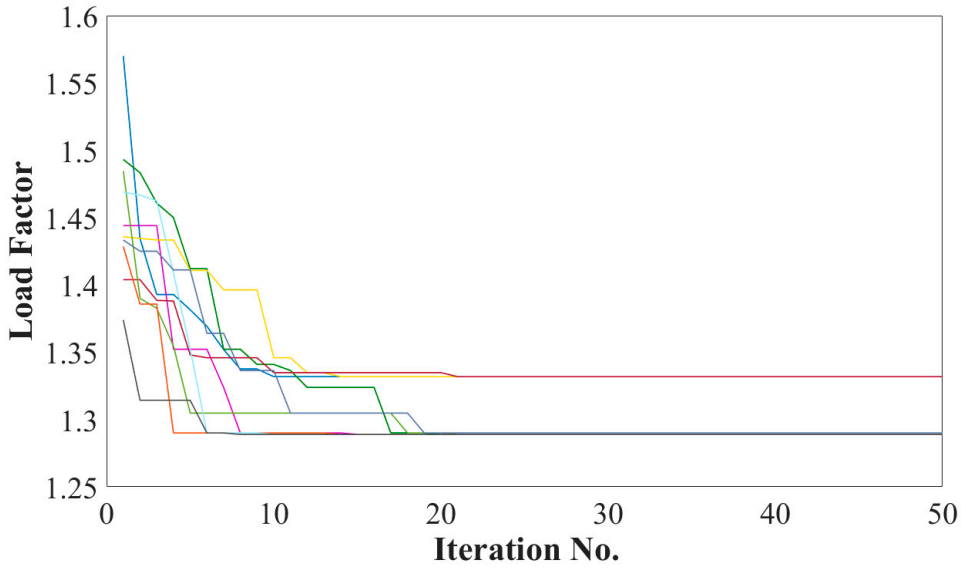


Figure 20. Results of ten analyses using WOA in example three.

Table 11. Details of Figure 18 results.

Analysis No.	Iteration No.	Failure Mode	Load Factor
1	22	{100100100100000000}	1.2889
2	24	{100100100100000000}	1.2889
3	22	{100100100100000000}	1.2889
4	24	{100100100100000000}	1.2889
5	23	{100100100100000000}	1.2889
6	28	{100100100100000000}	1.2889
7	9	{100100100100000000}	1.2889
8	28	{100100100100000000}	1.2889
9	27	{100100100100000000}	1.2889
10	32	{100100100100000000}	1.2889

Table 12. Details of Figure 19 results.

Analysis No.	Iteration No.	Failure Mode	Load Factor
1	11	{100100100100000000}	1.2889
2	13	{100100100100000000}	1.2889
3	**6**	**{100100100100100000}**	**1.2900**
4	7	{100100100100000000}	1.2889
5	5	{100100100100000000}	1.2889
6	8	{100100100100000000}	1.2889
7	**6**	**{100100100100100110}**	**1.3318**
8	10	{100100100100000000}	1.2889
9	7	{100100100100000000}	1.2889
10	7	{100100100100000000}	1.2889

Table 13. Details of Figure 20 results.

Analysis No.	Iteration No.	Failure Mode	Load Factor
1	**10**	**{100100100100100110}**	**1.3318**
2	15	{100100100100000000}	1.2889
3	20	{100100100100000000}	1.2889
4	**14**	**{100100100100100110}**	**1.3318**
5	**21**	**{100100100100100110}**	**1.3318**
6	21	{100100100100000000}	1.2889
7	14	{100100100100000000}	1.2889
8	10	{100100100100000000}	1.2889
9	19	{100100100100000000}	1.2889
10	**8**	**{100100100100100000}**	**1.2900**

According to Table 13, the highest error of prediction by the WOA relates to the load factor of 1.3318, which occurred three times. The failure mode is {100100100100100110}, illustrating that mechanisms number 13, 16, and 17 were formed against the actual mechanism ({100100100100000000}). These mechanisms are related to the floor of story 5, floor of story 6, and beam of story 6 within span 1, respectively, according to Table 10. In this example, the binary DE algorithm provided the exact load factor and failure mode in all analyses. The GWO algorithm failed only twice for the exact load factor prediction. Stating the result up to two decimal numbers results in only one error. Nevertheless, the corresponding failure mode is different in each case. The number of errors in the WOA was four in this example, revealing a higher error than the other two algorithms.

Last but not least, to compare the performance of each algorithm in the problem of plastic method of structures with the help of the statistical values stated at the beginning of this section, Table 14 is presented. In this table, binary DE shows no errors in each of the ten analyses performed, while the GWO and WOA illustrated some errors in determining the collapse load factor and the corresponding collapse mechanism of all examples.

Table 14. RMSE and MAE values for each algorithm.

Algorithm	Example One			Example Two			Example Three		
	Error No.	RMSE	MAE	Error No.	RMSE	MAE	Error No.	RMSE	MAE
Binary DE	0	0	0	0	0	0	0	0	0
GWO	2	0.07029	0.03000	2	0.007468	0.00334	2	0.01357	0.0044
WOA	4	0.01337	0.00743	3	0.000753	0.00039	4	0.02350	0.0130

It is noteworthy to mention that although the error frequency of the WOA was more than that of the GWO algorithm in all three examples, the MAE and RMSE were less than those of the GWO algorithm (except for the third example). This result demonstrates that

although the frequency of the WOA error was higher, these error values were closer to the actual response than those of the GWO algorithm.

5. Conclusions

This study addresses the challenge of analyzing structures to determine failure modes, which typically requires significant computational resources and costs. By optimizing the analysis process, this methodology enables engineers to obtain accurate solutions in a shorter time frame, potentially saving computational resources and reducing analysis costs. In this research, the analysis of steel moment frames was investigated using the theory of plastic analysis by merging pre-defined mechanisms. Initially, a new algorithm was proposed based on the principles of the dolphin echolocation algorithm, which is specified for binary problems. Hence, through modification of the DE algorithm, binary DE was introduced. Then, using the existing principles in the plastic method of steel frames, the critical failure load factor and its corresponding failure mechanism were determined with the aid of the proposed optimization algorithm. In order to compare the performance of this algorithm with other similar algorithms in this field, two of the most popular algorithms, called the grey wolf optimizer and whale optimization algorithm, were used for this problem, and their results were compared with those of binary DE. By examining three sample frames from the literature, this study established that binary DE provided the answers with very high accuracy. Accordingly, the GWO was also accurate in failure mode determination but with less precision than binary DE. Nonetheless, the WOA yielded the lowest accuracy. Although binary DE could properly work for this specific problem in structural engineering, a deeper investigation is required for other binary optimization problems to extend it to other issues.

Author Contributions: Conceptualization, A.A.; methodology, A.A., H.H.A. and B.Y.; validation, H.H.A.; formal analysis, A.A. and B.Y.; writing—original draft preparation, H.H.A. and B.Y.; writing—review and editing, A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The datasets used and analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. He, J.; Zhu, S.; Luo, C.; Niu, X.; Wang, Q. Size effect in fatigue modelling of defective materials: Application of the calibrated weakest-link theory. *Int. J. Fatigue* **2022**, *165*, 107213. [CrossRef]
2. Niu, X.; Zhu, S.; He, J.; Liao, D.; Correia, J.A.F.O.; Berto, F.; Wang, Q. Defect tolerant fatigue assessment of AM materials: Size effect and probabilistic prospects. *Int. J. Fatigue* **2022**, *160*, 106884. [CrossRef]
3. Zhai, S.; Lyu, Y.; Cao, K.; Li, G.; Wang, W.; Chen, C. Seismic behavior of an innovative bolted connection with dual-slot hole for modular steel buildings. *Eng. Struct.* **2023**, *279*, 115619. [CrossRef]
4. Yao, Y.; Huang, H.; Zhang, W.; Ye, Y.; Xin, L.; Liu, Y. Seismic performance of steel-PEC spliced frame beam. *J. Constr. Steel Res.* **2022**, *197*, 107456. [CrossRef]
5. Ghasemi, M.; Zhang, C.; Khorshidi, H.; Zhu, L.; Hsiao, P. Seismic upgrading of existing RC frames with displacement-restraint cable bracing. *Eng. Struct.* **2023**, *282*, 115764. [CrossRef]
6. Huang, H.; Huang, M.; Zhang, W.; Yang, S. Experimental study of predamaged columns strengthened by HPFL and BSP under combined load cases. *Struct. Infrastruct. Eng.* **2021**, *17*, 1210–1227. [CrossRef]
7. Liao, D.; Zhu, S.; Keshtegar, B.; Qian, G.; Wang, Q. Probabilistic framework for fatigue life assessment of notched components under size effects. *Int. J. Mech. Sci.* **2020**, *181*, 105685. [CrossRef]
8. Yang, Y.; Lin, B.; Zhang, W. Experimental and numerical investigation of an arch–beam joint for an arch bridge. *Arch. Civ. Mech. Eng.* **2023**, *23*, 101. [CrossRef]
9. Asteris, P.G.; Chountalas, A.T.; Sophianopoulos, D.S.; Alam, N.; Maraveas, C. Fire resistance prediction of slim-floor asymmetric steel beams using single hidden layer ANN models that employ multiple activation functions. *Steel Compos. Struct.* **2022**, *44*, 755–774.

10. Jiang, J.; Ye, M.; Chen, L.Y.; Zhu, Z.W.; Wu, M. Study on static strength of Q690 built-up K-joints under axial loads. *Structures* **2023**, *51*, 760–775. [CrossRef]
11. Fu, Q.; Gu, M.; Yuan, J.; Lin, Y. Experimental Study on Vibration Velocity of Piled Raft Supported Embankment and Foundation for Ballastless High Speed Railway. *Buildings* **2022**, *12*, 1982. [CrossRef]
12. Huang, H.; Li, M.; Yuan, Y.; Bai, H. Theoretical analysis on the lateral drift of precast concrete frame with replaceable artificial controllable plastic hinges. *J. Build. Eng.* **2022**, *62*, 105386. [CrossRef]
13. Neal, B.G.; Symonds, P.S. The calculation of collapse loads for framed structures. *J. Inst. Civ. Eng.* **1950**, *35*, 21–40. [CrossRef]
14. Neal, B.G.; Symonds, P.S. The rapid calculation of the plastic collapse load for a framed structure. *Proc. Inst. Civ. Eng.* **1952**, *1*, 58–71. [CrossRef]
15. Watwood, V.B. Mechanism Generation for Limit Analysis of Frames. *J. Struct. Div.* **1979**, *105*, 1–15. [CrossRef]
16. Deeks, A.J. Automatic computation of plastic collapse loads for frames. *Comput. Struct.* **1996**, *60*, 391–402. [CrossRef]
17. Chen, W.-F.; Sohal, I. *Plastic Design and Second-Order Analysis of Steel Frames Buch*; Springer: New York, NY, USA, 1995.
18. Wong, M.B. *Plastic Analysis and Design of Steel Structures*; Butterworth-Heinemann: Burlington, MA, USA, 2009.
19. Samanta, R.K.; Sadhukhan, B.; Samaddar, H.; Sarkar, S.; Koner, C.; Ghosh, M. Scope of machine learning applications for addressing the challenges in next-generation wireless networks. *CAAI Trans. Intell. Technol.* **2022**, *7*, 395–418. [CrossRef]
20. Wang, X.; Wang, S.; Chen, P.Y.; Lin, X.; Chin, P. Block switching: A stochastic approach for deep learning security. *arXiv* **2020**, arXiv:2002.07920. [CrossRef]
21. Palizi, S.; Toufigh, V. Fire-induced damage assessment of cementless alkali-activated slag-based concrete. *Constr. Build Mater.* **2023**, *393*, 132002. [CrossRef]
22. Palizi, S.; Toufigh, V. Bond strength prediction of timber-FRP under standard and acidic/alkaline environmental conditions based on gene expression programming. *Eur. J. Wood Wood Prod.* **2022**, *80*, 1457–1471. [CrossRef]
23. Toufigh, V.; Palizi, S. Performance evaluation of slag-based concrete at elevated temperatures by a novel machine learning approach. *Constr. Build. Mater.* **2022**, *358*, 129357. [CrossRef]
24. Daryan, A.S.; Salari, M.; Palizi, S.; Farhoudi, N. Size and layout optimum design of frames with steel plate shear walls by metaheuristic optimization algorithms. *Structures* **2023**, *48*, 657–668. [CrossRef]
25. Lu, H.; Zhu, Y.; Yin, M.; Yin, G.; Xie, L. Multimodal Fusion Convolutional Neural Network with Cross-Attention Mechanism for Internal Defect Detection of Magnetic Tile. *IEEE Access* **2022**, *10*, 60876–60886. [CrossRef]
26. Palizi, S.; Toufigh, V.; Ramezanpour Kami, M. Ultrasonic pulse velocity for mechanical properties determination of wood. *Wood Mater. Sci. Eng.* **2023**, 1–12. [CrossRef]
27. Wang, J.; Yang, M.; Liang, F.; Feng, K.; Zhang, K.; Wang, Q. An Algorithm for Painting Large Objects Based on a Nine-Axis UR5 Robotic Manipulator. *Appl. Sci.* **2022**, *12*, 7219. [CrossRef]
28. Mafarja, M.M.; Mirjalili, S. Hybrid Whale Optimization Algorithm with simulated annealing for feature selection. *Neurocomputing* **2017**, *260*, 302–312. [CrossRef]
29. Kaveh, A.; Jahanshahi, M. Plastic limit analysis of frames using ant colony systems. *Comput. Struct.* **2008**, *86*, 1152–1163. [CrossRef]
30. Kohama, Y.; Takada, T.; Kozawa, N.; Miyamura, A. Collapse analysis of rigid frames by genetic algorithm. *Trans. Built Environ.* **1997**, *28*, 193–202.
31. Kaveh, A.; Jahanshahi, M. Plastic design of frames using heuristic algorithms. In Proceedings of the Eighth International Conference on Computational Structures Technology, Las Palmas, Spain, 12–15 September 2006; Paper No. 108. Topping, B.H.V., Montero, G., Montenegro, R., Eds.; Civil-Comp Press: Stirlingshire, Scotland, 2006.
32. Jahanshahi, M.; Pouraghajan, M.; Pouraghajan, M. Enhanced ACS algorithms for plastic analysis of planar frames. *Comput. Methods Civil Eng.* **2013**, *4*, 65–82.
33. Kaveh, A.; Jahanshahi, M.; Khanzadi, M. Plastic analysis of frames using genetic algorithm and ant colony algorithm. *Asian J. Civil Eng.* **2008**, *9*, 227–246.
34. Jahanshahi, M.; Maleki, E.; Ghiami, A. On the efficiency of artificial neural networks for plastic analysis of planar frames in comparison with genetic algorithms and ant colony systems. *Neural Comput. Appl.* **2016**, *28*, 3209–3227. [CrossRef]
35. Kaveh, A.; Ghafari, M.H. Plastic analysis of planar frames using CBO and ECBO algorithms. *Int. J. Optim. Civil Eng.* **2015**, *5*, 479–492.
36. Kaveh, A.; Ilchi Ghazaan, M. Enhanced colliding bodies optimization for design problems with continuous and discrete variables. *Adv. Eng. Softw.* **2014**, *77*, 66–75. [CrossRef]
37. Daryan, A.S.; Palizi, S.; Farhoudi, N. Optimization of plastic analysis of moment frames using modified dolphin echolocation algorithm. *Adv. Struct. Eng.* **2019**, *22*, 2504–2516. [CrossRef]
38. Saedi Daryan, A.; Palizi, S. New Plastic Analysis Procedure for Collapse Prediction of Braced Frames by Means of Genetic Algorithm. *J. Struct. Eng.* **2020**, *146*, 04019168. [CrossRef]
39. Palizi, S.; Saedi Daryan, A. Plastic Analysis of Braced Frames by Application of Metaheuristic Optimization Algorithms. *Int. J. Steel Struct.* **2020**, *20*, 1135–1150. [CrossRef]
40. Palizi, S.; Saedi Daryan, A. Critical Temperature Evaluation of Moment Frames by Means of Plastic Analysis Theory and Genetic Algorithm. *Iran. J. Sci. Technol. Trans. Civ. Eng.* **2021**, *46*, 843–856. [CrossRef]
41. Kaveh, A.; Farhoudi, N. A new optimization method: Dolphin echolocation. *Adv. Eng. Softw.* **2013**, *59*, 53–70. [CrossRef]

42. Daryan, A.S.; Salari, M.; Farhoudi, N.; Palizi, S. Seismic Design Optimization of Steel Frames with Steel Shear Wall System Using Modified Dolphin Algorithm. *Int. J. Steel Struct.* **2021**, *21*, 771–786. [CrossRef]
43. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
44. Gandomi, A.H.; Mohammadzadeh, S.D.; Pérez-Ordóñez, J.L.; Alavi, A.H. Linear genetic programming for shear strength prediction of reinforced concrete beams without stirrups. *Appl. Soft Comput.* **2014**, *19*, 112–120. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.