symmetry

MDPI

Article

Frequency-Constrained Optimization of a Real-Scale Symmetric Structural Using Gold Rush Algorithm

Sepehr Sarjamei, Mohammad Sajjad Massoudi * and Mehdi Esfandi Sarafraz

Department of Civil Engineering, West Tehran Branch, Islamic Azad University, Tehran 1468763785, Iran; sarjamei.sepehr@wtiau.ac.ir (S.S.); sarafraz.m@wtiau.ac.ir (M.E.S.)
* Correspondence: massoudi.ms@wtiau.ac.ir

Abstract: The optimal design of real-scale structures under frequency constraints is a crucial problem for engineers. In this paper, linear analysis, as well as optimization by considering natural frequency constraints, have been used for real-scale symmetric structures. These structures require a lot of time to minimize weight and displacement. The cyclically symmetric properties have been used for decreasing time. The structure has been decomposed into smaller repeated portions termed substructures. Only the substructure elements are needed when analyzing and designing with the concept of cyclic symmetries. The frequency constrained design of real-scale structures is a complex optimization problem that has many local optimal answers. In this research, the Gold Rush Optimization (GRO) algorithm has been used to optimize weight and displacement performances due to its effectiveness and robustness against uncertainties. The efficacy of the concept of cyclic symmetry to minimize the time calculated is assessed by three examples, including Disk, Silo, and Cooling Tower. Numerical results indicate that the proposed method can effectively reduce time consumption, and that the GRO algorithm results in a 14–20% weight reduction of the problems.

Keywords: structural optimization; frequency constraints; cyclic symmetry; Gold Rush Optimization algorithm

Citation: Sarjamei, S.; Massoudi, M.S.; Esfandi Sarafraz, M. Frequency-Constrained Optimization of a Real-Scale Symmetric Structural Using Gold Rush Algorithm. *Symmetry* **2022**, *14*, 725. https://doi.org/10.3390/sym14040725

Academic Editors: Jan Awrejcewicz and Sergei D. Odintsov

Received: 21 February 2022
Accepted: 31 March 2022
Published: 2 April 2022

1. Introduction

In vibrational analysis, the optimal design of real-scale symmetric structures under frequency constraints is a crucial problem. Since the modal properties of a structure determine its dynamic behavior, the frequency constraints and the capacity to adjust the values of natural frequencies are sensitive items in the analysis and design. Concerning the frequency constraints, including non-convex search spaces, sophisticated methods are needed [1]. Since the frequency constrained design of large-scale structures is a complex optimization problem with many local optima, an appropriate optimization technique is usually required. Among the research conducted to optimize the design of structures under frequency constraints, the following studies can be briefly reviewed.

Using laws of momentum and energy between collisions bodies, Kaveh and Mahdavi [2] introduced a new Colliding Bodies Optimization algorithm (CBO). Kaveh and Mahdavi [3] looked into the effectiveness of CBO for the problem and conducted parametric research on its internal characteristics. Enhanced Colliding Bodies Optimization (ECBO) introduced by Kaveh and Ilchi Ghazaan [4] improved the function of the CBO algorithm. ECBO uses memory to save some optimal solutions. Enhanced Colliding Bodies Optimization (ECBO) was used by Kaveh and Ilchi Ghazaan [5] to demonstrate the algorithm's efficiency in frequency-constrained structural optimization. Song and Zhang [6] assessed the wind deflection of a railway catenary in a crosswind under frequency constraints, based on wind tunnel tests and a nonlinear finite element model. Ho-Huu et al. [7] proposed a new version of the Differential Evolution (DE) method called Roulette Wheel Selection-Elitist-Differential Evolution (ReDE), which employs elitism in the selection phase using the

Roulette Wheel Selection technique. Lieu et al. [8] proposed the Adaptive hybrid Evolutionary Firefly Method by combining the differential evolution (DE) algorithm and the Firefly Algorithm (FA) (AHEFA). Tejani and Mirjalili [9] used Symbiotic Organisms Search (SOS) to optimize the size of space trusses. SOS is based on the biological interactions between organisms in an ecosystem. Kaveh and Dadras [10] have introduced a chaotic version of a newly-established metaheuristic algorithm called the Water Strider Algorithm (WSA) to tackle this problem. Kaveh and Ilchi Ghazaan [1] used the ECBO method to optimize large-scale dome trusses with frequency limitations, incorporating multi-stage cascading techniques. The possibilities of the Vibrating Particles System (VPS), an algorithm inspired by the damped oscillation of a single degree of freedom system, to cope with large-scale dome trusses were examined by Kaveh and Ilchi Ghazaan [11]. To handle a large number of variables, Kaveh and Ilchi Ghazaan [12] combined the VPS technique with multi-design variable configuration (Multi-DVC) cascade optimization, as well as employing an upper bound strategy (UBS) to reduce computing time. Weight optimization of truss structures with different frequency constraints was investigated by Carvalho et al. [13]. Rao [14] created the Teaching-Learning-Based Optimization (TLBO) algorithm based on a traditional school learning. Kar et al. [15] proposed a Craziness-based Particle Swarm Optimization (CRPSO), which they used. They employed cardinality constraints and frequency constraints to limit the maximum number of distinct cross-sectional areas, lowering the cost of selecting a different cross-section of elements and weights of structures. The Charged System Search (CSS) algorithm was introduced using principles from physics and mechanics [16]. Furthermore, they utilized a combination of governing Coulomb law from electrostatics and the Newtonian laws of mechanics. The (CSS) algorithm and its enhanced version (ECSS) are being used to optimize various truss structures [17]. To improve the CSS algorithm's convergence time, Jalili and Talatahari [18] devised a hybrid Charged System Search (CSS) method with a Migration-based Local Search (MBLS) mechanism. The effectiveness of the proposed hybrid approach was proved in their research by proving the optimum design of many benchmark truss instances with frequency constraints. For the best design of large-scale cyclically symmetric dome trusses with frequency constraints, Kaveh and Zolghadr [19] employed the Cyclical Parthenogenesis Algorithm (CPA). They used the block diagonalization technique to divide the domes' repeated patterns into smaller parts, lowering the computer time necessary for the analysis. Liu et al. [20] added the vision search radius for each fruit fly as well as an enhanced Deb (IDeb) rule to handle the limitations to the Fruit Fly Optimization Algorithm (FOA) utilizing a memory-based search strategy. They used this technique to optimize truss structures with frequency constraints, demonstrating that the new algorithm finds better answers. To modify the attractiveness and light absorption coefficients of FA, Kaveh, and Javadi [21] used two chaotic maps, namely Logistic and Gaussian maps. These chaotic algorithms were used to optimize large-scale domes that have various frequency constraints.

Real-scale structures need a lot of effort to find the modal parameters, whereas symmetric structures can be solved rapidly. Wang [22] optimized the real-scale bridge cables under frequency constraints. For tackling rotationally periodic structures, Williams [23] presented an accurate eigen solution technique. The component mode technique was used by Tran [24] for vibration analysis of cyclic symmetry systems. He used a scaled finite element approach for cyclically symmetric domain heat transport and structural mechanics difficulties [25,26]. For optimal structural analysis, graph theory [27,28] has been suggested as a helpful solution. Kaveh and Koohestani [29] created graph models for ordinary finite element meshes. In the free vibration analysis of cyclically repeated structures, Kaveh and Rahami [30] used block circulant matrices. Kaveh and Rahami [31] proposed a method for efficiently computing graph product-generated repeating structures. Using the force method, Koohestani [32] proposed an orthogonal self-stress matrix for quickly evaluating cyclically symmetric space truss designs. Koohestani [33] applied the properties of symmetry in graph theory to finite and boundary elements. For the free vibration analysis of cyclic symmetry, Koohestani [34] implemented the decomposition of extended Eigen problems.

This research aimed to optimize the design of real-scale symmetric structures under frequency constraints using the GRO metaheuristic algorithm. It has been practically impossible to optimize real-scale symmetric structures in the previous research using meta-heuristic algorithms due to the large volume and time of calculations. This study hypothesized that the abilities of the cyclic symmetric concept could reduce the time and volume of calculations. Using the metaheuristic algorithm and the concept of cyclic symmetry, real-scale structures such as a Disk, Silo, and Cooling Tower have been investigated. Furthermore, the results are discussed and compared with CSS and TLBO algorithms.

2. Materials and Methods

In this section, the methodology of the frequency constraint, cyclically symmetric formulation, and optimization algorithm are introduced.

2.1. Methodology of the Frequency Constraint Optimization Problem

This problem aims to find the optimal design for structures with real-scale cyclic symmetry and multiple frequency constraints, where the cross-sectional area of the structural members is constantly changing in the search space. In such problems, size optimization minimizes weight while satisfying the constraints. The mathematical formula of the problem can be expressed as Equation (1).

$$\begin{array}{l} Find\ X = [x_1, x_2, \ldots, x_{nDV}],\ x_i \in R_i \\ \text{to minimize}\ P(X) = f_{penalty}(X) \times W(X); \\ \text{subject to}: \begin{cases} \omega_j \le \omega_j^* \ for\ some\ natural\ frequencies\ j \\ \omega_K \le \omega_k^* \ for\ some\ natural\ frequencies\ k \end{cases} \quad x_i^L \le x_i \le x_i^U \\ R_i = \left\{ x_i \,|\, x_i \in \left[x_i^L, x_i^U \right] \right\} \end{array} \tag{1}$$

where the vector X contains the design variables (sections), the ith design variable is x_i. (nDV) is the number of design variables according to the grouping of elements. $W(X)$ shows the weight of the structure. The penalty approach is used to consider the constraints, in which the cost function is defined as that which must be minimized. $f_{penalty}(X)$ is a penalty function. When certain constraints are violated in a particular solution, the penalty function $f_{penalty}(X)$ artificially increases the weight of the structure by taking values more significant than one. Also, ω_j is the jth natural frequency of the structure, ω_j^* is the upper limit, ω_k is the kth natural frequency of the structure, and ω_k^* is the lower limit. x_i^L and x_i^U are the lower and upper bounds of the design variable x_i. R_i is the allowable range of the ith design variable. The design variable x_i can be constantly changed in R_i. The weight of the structure is calculated from Equation (2).

$$W(X) = \sum_{i=1}^{nDV} x_i \sum_{j=1}^{nm\ (i)} \rho_j L_j \tag{2}$$

In which $nm(i)$ is the number of members allocated to the ith element. ρ_j and L_j are the material density and the length of the jth member, respectively. The penalty function is defined in Equation (3).

$$f_{penalty}(X) = (1 + \varepsilon_1 . \nu)^{\varepsilon 2} \tag{3}$$

For a particular solution, ν shows the sum of violations, defined in Equation (4).

$$\nu = \sum_{i=1}^{s} \nu_i \tag{4}$$

In which S is the number of frequency constraints. Values of ν_i can be considered as shown in Equation (5).

$$\nu_i = \begin{cases} 0 & \text{if the } i\text{th constraint is satisfied} \\ \left|1 - \frac{\omega_i}{\omega_i^*}\right| & else \end{cases} \quad (5)$$

In this study, ε_1 and ε_2 are calculated from Equation (6) to create a suitable balance between the algorithm's exploration and exploitation.

$$\begin{aligned} \varepsilon_1 &= 1.5 + 0.5 \times \frac{NSA}{MaxNSA} \\ \varepsilon_2 &= 1.5 + 1.5 \times \frac{NSA}{MaxNSA} \end{aligned} \quad (6)$$

where NSA is the current analysis number and $MaxNSA$ is the total number of structural analyses for the optimization process and is defined as the criterion for the optimization termination. As the value of NSA increases, the values of ε_1 and ε_2 grow. It can be inferred that the algorithm explores the search space in the early stages but in the final stages tends to choose solutions without violations. Equation (6) helps the algorithm search near a low-cost solution as a final design and converge to reduce errors.

2.2. Cyclically Symmetric Formulation

One of the general tasks in engineering is to determine the natural frequency of structures on a real scale. The system displays a specific pattern in structures with cyclic symmetry. The rotations of several repeating identical units (called substructures) along the central axis make up a cyclically symmetric structure [29]. Structural analysis is performed on only one of the substructures using the cyclic symmetry concept. Except at support nodes, points and elements are numbered in each substructure from top to bottom. The pattern of stiffness and mass matrices is obtained by using the right way of numbering nodes, which is the same as the Canonical Form F matrix as shown:

$$\mathrm{K} = \begin{bmatrix} \mathrm{K}_{11} & \mathrm{K}_{12} & & & \mathrm{K}_{21} \\ \mathrm{K}_{21} & \mathrm{K}_{11} & \mathrm{K}_{12} & & \\ & . & . & . & \\ & & \mathrm{K}_{21} & \mathrm{K}_{11} & \mathrm{K}_{12} \\ \mathrm{K}_{12} & & & \mathrm{K}_{21} & \mathrm{K}_{11} \end{bmatrix}_{n\times n} \quad \mathrm{M} = \begin{bmatrix} \mathrm{m}_{11} & \mathrm{m}_{12} & & & \mathrm{m}_{21} \\ \mathrm{m}_{21} & \mathrm{m}_{11} & \mathrm{m}_{12} & & \\ & . & . & . & \\ & & \mathrm{m}_{21} & \mathrm{m}_{11} & \mathrm{m}_{12} \\ \mathrm{m}_{12} & & & \mathrm{m}_{21} & \mathrm{m}_{11} \end{bmatrix}_{n\times n}$$

In which n is the number of repetitive substructures, $\mathrm{K}_{12} = \mathrm{K}_{21}^{\mathrm{t}}$, and $\mathrm{m}_{12} = \mathrm{m}_{21}^{\mathrm{t}}$. Given this, determining the natural frequency of the cyclically symmetric structure is a matter of total eigenvalue, as shown in Equation (7):

$$\mathrm{K}\phi = \mathrm{M}\phi\mathrm{D} \quad (7)$$

In Equation (7), the matrix D is diagonal. The values of matrix D are λ_i general eigenvalues. The T_i periods of the substructure are obtained from Equation (8).

$$T_i = 2\pi/\sqrt{\lambda_i} \qquad i = 1, \ldots, n \quad (8)$$

φ_i the eigenvector corresponds to the ith eigenvalues, and Equation (7) is rewritten as follows for each substructure Equation (9):

$$\mathrm{K}\varphi_i = \lambda_i \, \mathrm{M} \, \varphi_i \qquad i = 1, \ldots, n \quad (9)$$

The mass and stiffness matrix for each substructure is as follows in Equation (10) to reduce computing time using the canonical concept:

$$\begin{array}{l} K = I \otimes k_{11} + H \otimes k_{12} + H^t \otimes k_{21} \\ M = I \otimes m_{11} + H \otimes m_{12} + H^t \otimes m_{21} \end{array} \quad (10)$$

In Equation (10), I and H are an n × n identity matrix and a special and important matrix, respectively, with the desired pattern presented as follows:

$$I = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix}_{n \times n} \qquad H = \begin{bmatrix} 0 & 1 & & & & \\ & 0 & 1 & & & \\ & & \cdot & \cdot & & \\ & & & \cdot & 1 & \\ & & & & 0 & 1 \\ 1 & & & & & 0 \end{bmatrix}_{n \times n}$$

A full description of this section can be found in Kaveh [29].

2.3. Optimization Algorithm

In this study, the GRO, CSS, and TLBO algorithms are used for Equation (1). These algorithms have been adopted due to their ability to optimize truss structures in previous studies. The algorithms are stated below:

2.3.1. Gold Rush Optimization (GRO) Algorithm

Massoudi and Sarjamei created a GRO algorithm [35] based on the power of human thinking and decision making, and which will be called a Gold Rush Optimization. The GRO algorithm is a population-based evolutionary algorithm with a higher convergence speed than other optimization algorithms. The aim is to find the place of gold. Firstly, a group of people called operators stand in a random spot of search space. Every operator uses a device (metal detector) to find gold. In every stage, the operators move altogether and listen to the sound until they hear an increase in the sound and then stop at that point. Every operator would also listen to the sounds produced by other devices and constantly monitor if any other devices create a louder sound. At each stage, the group moves to the place of the loudest sound. In the end, the exact location of the gold is determined. Three parameters α, β, and γ indicate the probability of moving towards the loudest sound or moving away from it. The parameters α, β, and γ in the interval [0–1] are selected.

Level 1: Initialization

Each operator stands randomly in one spot inside the search space as represented in Equation (11). lb_i and ub_i are the lower and upper bounds of a domain (search space). *rand* in the interval [0–1] is a random number, and N is the number of operators.

$$location_i^{(0)} = lb_i + (ub_i - lb_i) * rand, i = 1, 2, \ldots, N \quad (11)$$

Level 2: Monitoring-Choosing the best locations

SOP is an operator who is successful in finding the optimal location. In this step, SOP should be generated. At the end of every iteration, the top ten percent of operators should be chosen and kept in the SOP.

Level 3: Fitness-distance

The analysis of the loudness of every sound (rate), operator with the most probability to extract gold, is calculated from Equation (12):

$$\text{rate}(i) = \frac{D_i}{\rho} * \frac{sound(highest\ volume) - sound(i)}{(sound(highest\ volume) - sound(lowest\ volume) + \varepsilon)} \quad (12)$$

The epsilon (ε) is a small positive number to avoid singularities. To prevent errors from environmental, the coefficients, ρ, and D_i represented in Equation (13) are used. The indices i and j indicate the current position of the two operators.

$$\rho = 2 - \frac{iter}{max_{iter}} \quad , \quad D_i = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + \ldots} \tag{13}$$

Level 4: Think-Decisions-move

In this step, every operator will create completely different selections based on a mix of sounds represented in Equation (14).

$$new\ location(i) = location(i) + md \times [(rate(j) - rate(i)) * (\ location(j) - location(i)) * rand] \tag{14}$$

The coefficients md means move direction determined from Equation (15):

$$md = \begin{cases} +1 \Rightarrow towards\ \ a\ \ loudest\ sound? & \alpha > rand \\ -1 \Rightarrow away\ from\ \ a\ \ loudest\ sound? & \alpha < rand \end{cases} \tag{15}$$

Level 5: Correct location

If the location obtained in Equation (14) does not meet the problem's constraints, Equation (16) is utilized to generate new locations. β and γ coefficients are selected as $0 < \beta < \gamma < 1$.

$$new\ \ location(i) = \begin{cases} choose\ \ a\ \ neighboring\ \ location & rand < \beta \\ select\ \ a\ \ new\ \ location\ \ randomly & \beta < \ \ rand < \gamma \\ do\ not\ move & \gamma < rand \end{cases} \tag{16}$$

Level 6: Termination

Steps 4 to 6 are eventually repeated in a loop until one of the following terminating conditions is met:

1. The maximum number of tries.
2. There has been no noticeable change in the optimal location.
3. The gap between the SOP function's values and the obtained most optimal answer is smaller than a pre-determined expected threshold. The parameters in the interval [0–1] are selected.
4. If the difference between the best and worst location's objective values is smaller than a given accuracy.

In this study, the GRO algorithm is used for Equation (1). The optimum amount of weight of an element is obtained. The algorithm is performed using MATLAB.

The flowchart of the GRO algorithm is illustrated in Figure 1.

2.3.2. Charged System Search (CSS) Algorithm

The Charged System Search (CSS) algorithm was created by Kaveh and Talatahari [16,36] as an efficient population-based metaheuristic using some physics and mechanics concepts, and it has been effectively applied to a variety of structural optimization problems [37–40]. CSS is based on the electrical Coulomb laws and the Newtonian rules of mechanics. Each agent in this algorithm is a charged particle (CP) with a fixed radius. The charge of a particle's magnitude q_i is calculated based on its quality from Equation (17):

$$q_i = \frac{fit(i) - fitworst}{fitbest - fitworst} \qquad i = 1, 2, \ldots, N \tag{17}$$

The best and the worst fitness of all the particles are $fitbest$ and $fitworst$, respectively; $fit(i)$ represents the fitness of the agent i, and N is the total number of CPs.

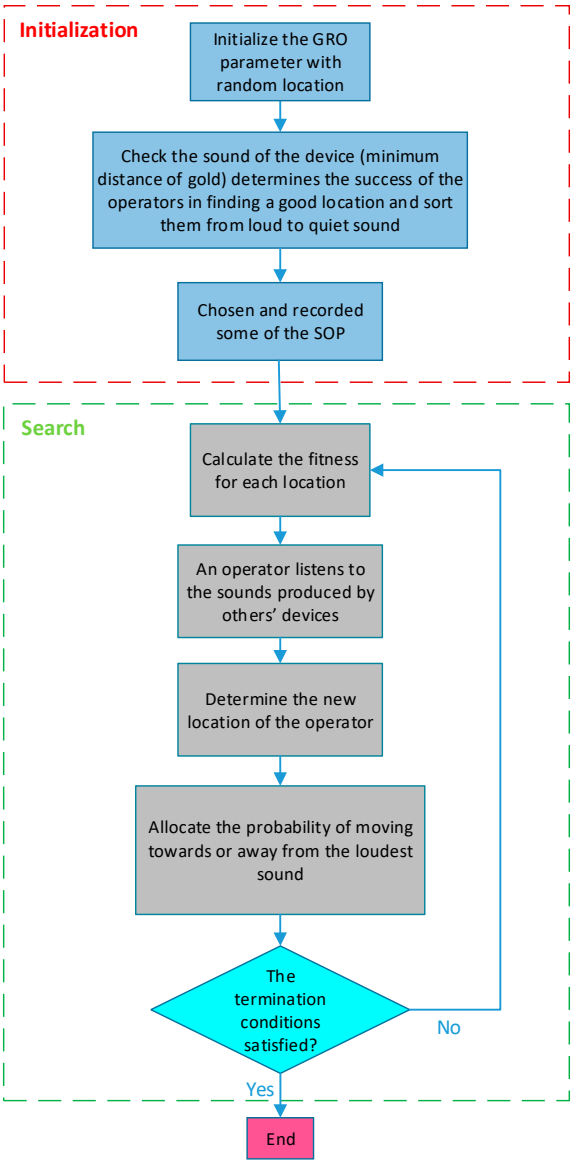

Figure 1. The Gold Rush Optimization algorithm flowchart.

Between two charged particles, the separation distance r_{ij} is defined from Equation (18):

$$r_{ij} = \frac{\|X_i - X_j\|}{\|(X_i - X_j)/2 - X_{best}\| + \varepsilon} \tag{18}$$

Here X_i and X_j are the positions of ith and the jth of charged particles. The position of the best charge particles is X_{best}. The epsilon (ε) is a small positive number to avoid singularities.

Each good particle generates an electric field that attracts other electrically charged things from Equation (19):

$$p_{ij} = \begin{cases} 1 & \frac{fit(i) - fitbest}{fit(j) - fit(i)} > rand \vee fit(j) > fit(i) \\ 0 & else \end{cases} \tag{19}$$

As a result, charged particles can interact with another one depending on their fitness values and separation distance from Equation (20):

$$F_j = q_j \sum_{i,i\neq j} \left(\frac{q_i}{a^3} r_{ij}.i_1 + \frac{q_i}{r_{ij}^2}.i_2 \right) p_{ij}(X_i - X_j) \qquad \left\langle \begin{array}{l} j = 1, 2, \ldots, N \\ i_1 = 1, i_2 = 0 \Leftrightarrow r_{ij} < a \\ i_1 = 0, i_2 = 1 \Leftrightarrow r_{ij} \geq a \end{array} \right. \tag{20}$$

The resultant force acting on the *j*th charged particle is F_j. (a) the radius of the charged sphere is set to unity.

The new position and velocity of each CP are computed by Equation (21):

$$\begin{array}{l} X_{j,new} = rand_{j1}.k_a.\frac{F_j}{m_j}.\Delta t^2 + rand_{j2}.k_v.V_{j,old}.\Delta t + X_{j,old} \\ V_{j,new} = \frac{X_{j,new} - X_{j,old}}{\Delta t} \end{array} \tag{21}$$

K_a and K_v are the acceleration coefficient and the velocity coefficient, respectively; $rand_{j1}$ and $rand_{j2}$ in the interval [0–1] are random numbers. m_j is the mass of the charged particles, and Δt is the time step set to one. Electrostatics laws are used to calculate the magnitude of the resultant force, whereas Newtonian mechanics laws are used to define the quality of the movement.

2.3.3. Teaching-Learning-Based Optimization (TLBO) Algorithm

Rao [14] created the Teaching-Learning-Based Optimization (TLBO) algorithm based on traditional school learning. The influence of a teacher on students and the effect of students on each other are the two stages of this algorithm. The population of random solutions in TLBO was dubbed students or learners at the start and initialized the population size (P_n). In TLBO, the regular distribution of marks received by pupils is considered as the performance of the class in learning or the teacher's performance in instructing. In each iteration, the best learner or most intelligent student with the best goal function is designated as the instructor. Students are updated iteratively to find the best solution in two phases: the first is based on the knowledge that transferred from a teacher (teacher phase), and the second is based on interaction with other students (interaction phase) (learner phase).

In the first phase (teacher phase), the mean of each design variable is calculated. The best solution that will act as a teacher is given by Equation (22).

$$X_{teacher} = X_{f(x)=\min} \tag{22}$$

$f\ (X)$ is the objective function, and X is a design variable.

The critical difference between the two normal distributions is the mean value (M), which means that a better instructor will teach pupils with higher average scores. In the teacher phase, TLBO enhances other pupils by utilizing the difference between the instructor's knowledge and the intermediate knowledge of all students. Modified solution based on the best solution is given by Equation (23):

$$X_{new,i} = X_{old,i} + r_i(M_{new} - T_F M_i) \tag{23}$$

r_i in the interval [0–1] is a random number. Determining the change in mean value is a teaching factor (T_F) and is explained in Equation (24).

$$T_F = round[1 + rand(0,1)\{2-1\}] \tag{24}$$

rand in the interval [0–1] is a random number.

In the second phase (learner phase), each student's knowledge is derived from their position in the search space. Students can also improve themselves by conversing with

another student after the teacher has finished teaching. The mathematical expression is explained as follows (25).

$$\begin{cases} X_{new,i} = X_{old,i} + r_i(X_i - X_j) & f(X_i) < f(X_j) \\ X_{new,i} = X_{old,i} + r_i(X_j - X_i) & else \end{cases} \quad (25)$$

TLBO increases each student's knowledge after contact with another randomly selected student throughout the learner phase

In this study, three numerical examples, including a Disk, Silo, and Cooling Tower, have been studied to evaluate the efficiency of the proposed method. In numerical examples, the results of optimal design by the GRO algorithm are evaluated and compared with two other famous algorithms.

3. Numerical Examples

In this section, the effectiveness of the concept of cyclic symmetry for minimizing the time required is assessed by three examples with continuous environments, including Disk, Silo, and Cooling tower. It is assumed that the number of frequencies in Equation (1) is equal to the number of degrees of freedom of the substructure. Structural properties which should be considered in all problems are listed in the following: the number of elements of the substructure is 30, the number of nodes of the substructure is 31, the number of repetitions of the substructure is 60, the number of elements of the structure is 1800, and the number of nodes of the structure is 1860. Material properties of these structures are: modulus of elasticity (E) = 2.4 $\times$ 10^7 (kN/m^2), mass per unit volume (ρ) = 2.4 (kNs2m-4), Poisson's ratio (ν) = 0.2. Examples have used a flat thin-shell element. This element is obtained by combining two elements, Q4 and DKQ. The isoperimetric four-node quadrilateral Q4 [41] element and the Discrete Kirchhoff Quadrilateral DKQ [42] element. The quadrilateral flat shell element has 24 degrees of freedom (6 for each node). The structure's weight is obtained from Equation (1). Furthermore, CSS and TLBO, two well-known algorithms in structural engineering problems, are used for optimization. The whole structure is then modeled in MATLAB using the approach mentioned above. In all examples, due to the random nature of meta-heuristic algorithms, each algorithm was run 20 times independently. The best result from 20 independent performances was reported as the best answer. Calculations were performed on the first-generation Intel Corei3 CPU. Frequency and displacement constraints were considered. Frequency constraints are considered to control the structure's dynamic behavior.

3.1. Disk

The first numerical example is a Disk, as shown in Figure 2. It is simply supported where R = 0.1 and R = 10.

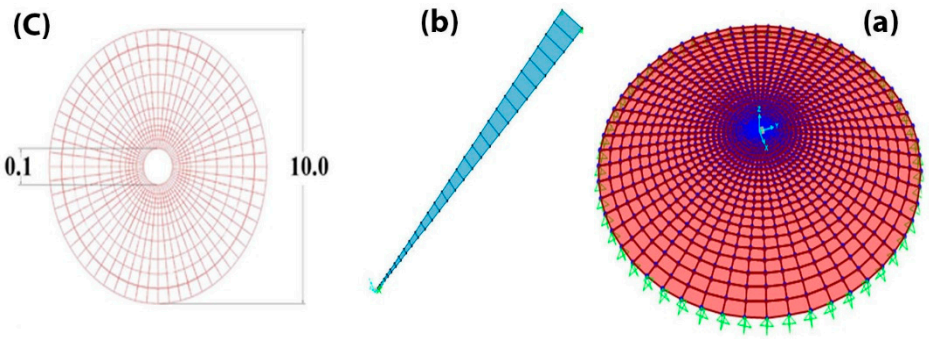

Figure 2. (**a**) A Disk discretized by quadrilateral shell finite elements, (**b**) substructure of Disk, (**c**) Disk dimensions.

The coordinate of the nodes of the Disk's substructure is presented in Table 1. This substructure is rotated 60 times around the center to create a Disk. The whole structure uses

a four-node quadrilateral flat shell element. In each element, the active mass is considered at the nodes.

Table 1. Coordinates of the nodes of the Disk.

Node Number	Coordinates (x, y, z)	Node Number	Coordinates (x, y, z)	Node Number	Coordinates (x, y, z)
1	(0.1, 0, 0)	12	(1.5052, 0, 0)	23	(5.4865, 0, 0)
2	(0.1213, 0, 0)	13	(1.7607, 0, 0)	24	(5.9762, 0, 0)
3	(0.1639, 0, 0)	14	(2.0374, 0, 0)	25	(6.4871, 0, 0)
4	(0.2277, 0, 0)	15	(2.3355, 0, 0)	26	(7.0194, 0, 0)
5	(0.3129, 0, 0)	16	(2.6548, 0, 0)	27	(7.5729, 0, 0)
6	(0.4194, 0, 0)	17	(2.9955, 0, 0)	28	(8.1478, 0, 0)
7	(0.5471, 0, 0)	18	(3.3574, 0, 0)	29	(8.7439, 0, 0)
8	(0.6961, 0, 0)	19	(3.7407, 0, 0)	30	(9.3613, 0, 0)
9	(0.8665, 0, 0)	20	(4.1452, 0, 0)	31	(10.0, 0, 0)
10	(1.0581, 0, 0)	21	(4.5710, 0, 0)		
11	(1.2710, 0, 0)	22	(5.0181, 0, 0)		

In this example, the optimization is performed once by considering the frequency constraint and next by considering both frequency and displacement constraints to evaluate the efficiency of the proposed method:

First constraint: first and third frequencies assumed to be less than 0.155 and 0.149, respectively ($\omega_1 < 0.155$, $\omega_3 < 0.149$).

Second constraint: first and third frequencies assumed to be less than <0.155 and <0.149, respectively ($\omega_1 < 0.155$, $\omega_3 < 0.149$), and the displacement of node number two under a load of 10,000 kN in the Y direction is less than <0.0103.

The thickness member of the substructure elements in this example is the main variable obtained according to the constraints considered by meta-heuristic algorithms. The specified range for the thickness of structural elements is (0.25–0.35 m) in both cases of ten and thirty variables.

In this example, to analyze a cyclically symmetric structure, only the matrix with dimensions 186, equal to the substructures, is calculated. Only once the calculation for the frequency of the structure without using the concept of cyclic symmetry was stopped after about 4 h without any result. The calculation time using the proposed method is significantly reduced.

For Disk evaluation, the coefficient of the GRO algorithm was $\alpha = 0.7, \beta = 0.5, \gamma = 0.8$. For 100 operators working in the Disk, a maximum of 500 repetitions was determined as a termination condition in the example. For this purpose, the structure was optimized with ten elements or variables (three elements in one group) and once with thirty variables, as shown in Figure 3.

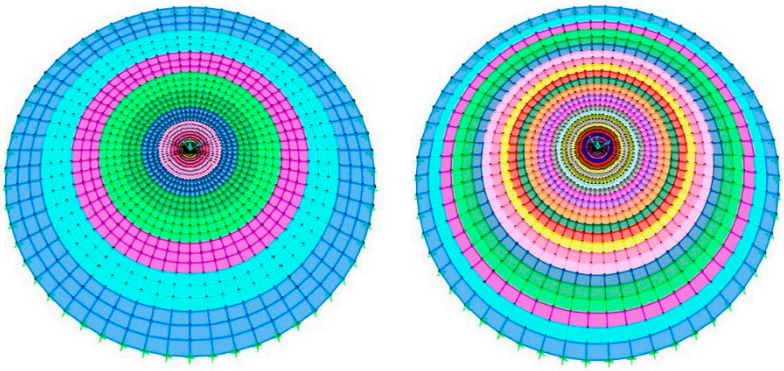

Figure 3. Disk discretized by ten and thirty quadrilateral shell finite elements.

The structure's weight with an initial thickness of 0.35 m is 263.3873 kN. The results are presented in Tables 2 and 3. The results in Table 2 show that, under the first and second constraint, the structure's weight is 210.6800 kN (20.0113% weight reduction) and 225.1152 kN (14.5307% weight reduction) by using the GRO algorithm, respectively, indicating the GRO algorithm's very intimate performance and better efficacy compared with the other two algorithms.

Table 2. Optimal design results for Disk with ten variables.

Group of Element	Constraint One			Constraint Two		
	GRO	CSS	TLBO	GRO	CSS	TLBO
1	0.3223	0.25432	0.32055	0.32432	0.31181	0.30816
2	0.31358	0.26921	0.28618	0.29656	0.32971	0.32527
3	0.35	0.30995	0.28575	0.31167	0.32477	0.33627
4	0.27218	0.28793	0.27218	0.30292	0.3174	0.30797
5	0.29664	0.27156	0.31867	0.27249	0.26037	0.30335
6	0.28693	0.30716	0.27789	0.25351	0.27004	0.27181
7	0.27102	0.3094	0.27254	0.32775	0.25913	0.27129
8	0.30973	0.30414	0.34035	0.30485	0.29696	0.32472
9	0.27882	0.29335	0.2535	0.32963	0.29731	0.33487
10	0.26383	0.25606	0.28665	0.27671	0.33245	0.28385
Weight (kN)	210.68	214.0883	215.5785	225.1152	227.3797	227.8022
Weight reduction (percentage)	20.0113	18.7173	18.1515	14.5307	13.671	13.5106

Table 3. Optimal design results for Disk with thirty variables.

Group of Element	Constraint One			Constraint Two		
	GRO	CSS	TLBO	GRO	CSS	TLBO
1	0.32895	0.34863	0.3236	0.27153	0.28859	0.31611
2	0.3392	0.27333	0.27333	0.29466	0.26096	0.28818
3	0.2831	0.2831	0.31302	0.27043	0.31731	0.27222
4	0.30319	0.2946	0.34495	0.29997	0.33555	0.29043
5	0.25	0.28761	0.28761	0.29314	0.30404	0.25802
6	0.34779	0.32129	0.34832	0.32587	0.33136	0.34974
7	0.30548	0.30548	0.29013	0.33109	0.3306	0.28823
8	0.26898	0.26898	0.33433	0.3434	0.2793	0.27152
9	0.29968	0.33482	0.28475	0.32096	0.29732	0.25
10	0.2951	0.30222	0.29329	0.3207	0.27777	0.34179
11	0.34994	0.29251	0.28194	0.33587	0.32934	0.32385
12	0.28303	0.30634	0.25284	0.32524	0.34933	0.28928
13	0.34992	0.31479	0.25787	0.33255	0.3445	0.30287
14	0.33866	0.28642	0.26022	0.33715	0.34862	0.31572
15	0.31011	0.27181	0.25	0.25431	0.30349	0.28225
16	0.2935	0.2935	0.29923	0.25576	0.27672	0.33805
17	0.30241	0.34981	0.31378	0.25793	0.29611	0.30806
18	0.32113	0.33979	0.28981	0.29638	0.32797	0.30032
19	0.28916	0.32681	0.35	0.31963	0.25141	0.3022
20	0.258	0.258	0.25008	0.31484	0.2888	0.26151
21	0.33324	0.30111	0.35	0.26566	0.3161	0.35
22	0.28162	0.30381	0.31155	0.30963	0.29594	0.26372
23	0.30371	0.30371	0.33676	0.35	0.29573	0.25857
24	0.29568	0.29568	0.29316	0.26058	0.31643	0.34493
25	0.26007	0.26007	0.26432	0.3004	0.34709	0.28111

Table 3. *Cont.*

Group of Element	Constraint One			Constraint Two		
	GRO	**CSS**	**TLBO**	**GRO**	**CSS**	**TLBO**
26	0.27469	0.29141	0.27469	0.33481	0.27642	0.3143
27	0.25478	0.27388	0.29988	0.28506	0.3321	0.33521
28	0.34821	0.34837	0.32301	0.34921	0.32673	0.28613
29	0.26503	0.26503	0.26624	0.30049	0.30046	0.32646
30	0.30911	0.32668	0.33041	0.25829	0.25107	0.31654
Weight (kN)	220.815	225.397	225.4451	226.4353	227.5151	231.0375
Weight reduction (percentage)	16.1634	14.4237	14.4055	14.0295	13.6196	12.2822

The results in Table 3 show that, under the first constraint, the structure's weight is 220.8150 kN (16.1634% weight reduction) by using the GRO algorithm, indicating its very intimate performance and better efficacy compared with the other two algorithms. However, under the second constraint, the structure's weight is 226.4353 kN (14.0295% weight reduction) by using the GRO and 227.5151 kN (13.6196% weight reduction) by using the CSS, indicating the GRO and CSS algorithms' very intimate performances, and a better efficacy compared with the TLBO algorithms. Tables 2 and 3 show the calculated thicknesses obtained by all algorithms. As can be seen, the thicknesses are in a suitable range, indicating the correct operation of the algorithms in finding the optimal answers in the search space.

3.2. Silo

The second numerical example is a Silo, as shown in Figure 4. It is simply supported at Z = 0.

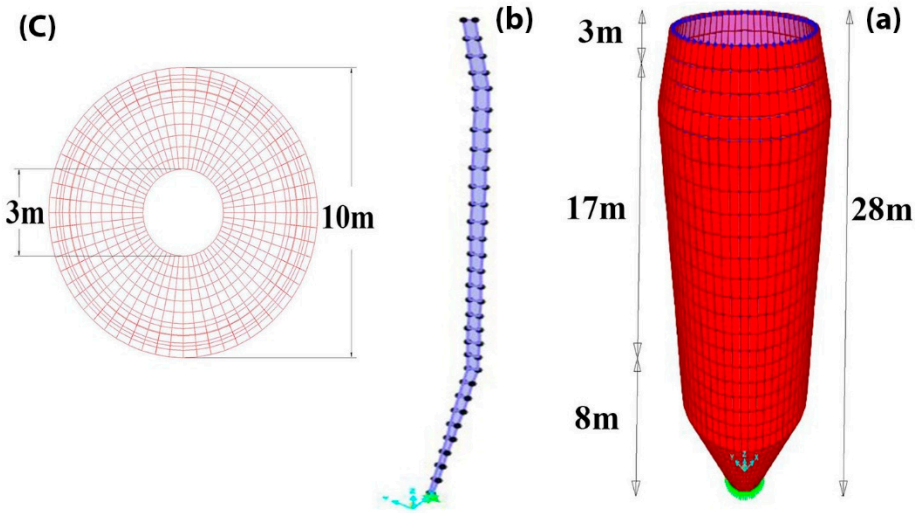

Figure 4. (**a**) A Silo discretized by quadrilateral shell finite elements, (**b**) substructure of Silo (**c**), Silo dimensions.

The coordinate of the nodes of Silo's substructure is presented in Table 4. This substructure is rotated 60 times around the center to create a Silo. The whole structure uses a four-node quadrilateral flat shell element. In each element, the active mass is considered at the nodes.

In this example, the optimization is performed once by considering the frequency constraint and next by considering both frequency and displacement constraints to evaluate the efficiency of the proposed method:

First constraint: first and third frequencies assumed to be less than 0.49 and 0.3, respectively ($\omega_1 < 0.49$, $\omega_3 < 0.3$).

Second constraint: first and third frequencies assumed to be less than <0.49 and <0.3, respectively ($\omega_1 < 0.49$, $\omega_3 < 0.3$), and the displacement of node number one under a load of 10,000 kN in the Y direction is less than <0.327.

Table 4. Coordinates of the nodes of the Silo.

Node Number	Coordinates (x, y, z)	Node Number	Coordinates (x, y, z)	Node Number	Coordinates (x, y, z)
1	(4, 0, 28)	12	(5, 0, 18)	23	(4.61111, 0, 7.111)
2	(4.25, 0, 27.25)	13	(5, 0, 17)	24	(4.22222, 0, 6.22222)
3	(4.5, 0, 26.5)	14	(5, 0, 16)	25	(3.83333, 0, 5.33333)
4	(4.75, 0, 25.75)	15	(5, 0, 15)	26	(3.4444, 0, 4.44444)
5	(5, 0, 25)	16	(5, 0, 14)	27	(3.05556, 0, 3.55556)
6	(5, 0, 24)	17	(5, 0, 13)	28	(2.66667, 0, 2.66667)
7	(5, 0, 23)	18	(5, 0, 12)	29	(2.27778, 0, 1.77778)
8	(5, 0, 22)	19	(5, 0, 11)	30	(1.88889, 0, 0.88889)
9	(5, 0, 21)	20	(5, 0, 10)	31	(1.5, 0, 0)
10	(5, 0, 20)	21	(5, 0, 9)		
11	(5, 0, 19)	22	(5, 0, 8)		

The thickness member of the substructure elements in this example is the main variable obtained according to the constraints considered by meta-heuristic algorithms. The specified range for the thickness of structural elements is (0.25–0.35 m) in both cases of ten and thirty variables.

In this example, to analyze a cyclically symmetric structure, only the matrix with dimensions 186, equal to the dimensions of the substructure, is calculated. Only once was the calculation for the frequency of the structure without using the concept of cyclic symmetry was stopped after about 4 h without any result. The calculation time using the proposed method is significantly reduced.

For Silo evaluation, the coefficient of the GRO algorithm was $\alpha = 0.7, \beta = 0.5, \gamma = 0.8$. For 120 operators working in the Silo, a maximum of 600 repetitions was determined as a termination condition in the example. For this purpose, the structure was optimized with ten elements or variables (three elements in one group) and once with thirty variables, as shown in Figure 5.

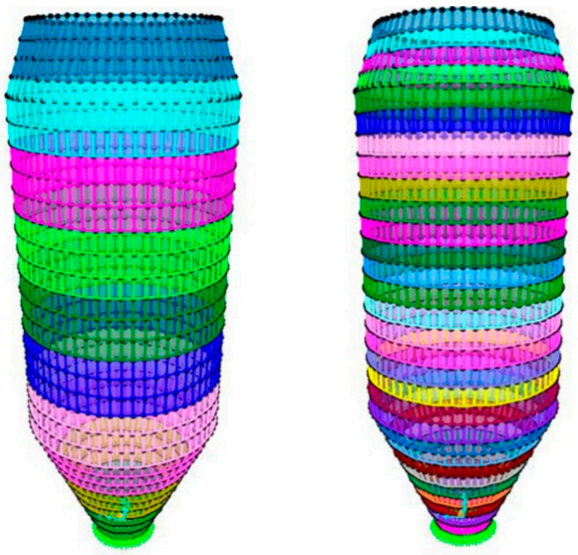

Figure 5. Silo discretized by ten and thirty quadrilateral shell finite elements.

The structure's weight with an initial thickness of 0.35 m is 673.2002 kN. The results are presented in Tables 5 and 6. Table 5 shows that the optimal weights obtained by GRO and CSS algorithms were almost similar. Under the first constraint, the structure's weight is 555.6478 kN (17.7588% weight reduction) by using the GRO algorithm and 553.9053 kN (17.7206% weight reduction) by using the CSS algorithm, indicating their very similar performance and a better efficacy compared with the TLBO algorithms. Under the second constraint, the GRO algorithm obtained a weight of 584.8135 kN (13.1293% weight reduction), indicating its very intimate performance and also a better efficacy compared with the CSS and TLBO algorithms. Moreover, the calculated thicknesses obtained by all algorithms are in a suitable range indicating the correct operation of the algorithms in finding the optimal answers in the search space.

Table 5. Optimal design results for Silo with ten variables.

Group of Element	Constraint One			Constraint Two		
	GRO	CSS	TLBO	GRO	CSS	TLBO
1	0.32025	0.32399	0.33612	0.29212	0.34293	0.27209
2	0.27501	0.27337	0.28823	0.33717	0.32757	0.33174
3	0.26468	0.27171	0.28095	0.3301	0.29867	0.33572
4	0.27503	0.27634	0.27058	0.2539	0.29358	0.27366
5	0.29596	0.29512	0.30204	0.33178	0.29467	0.35
6	0.30478	0.30104	0.30587	0.29637	0.28063	0.29641
7	0.26461	0.26172	0.2746	0.2635	0.30085	0.26175
8	0.32331	0.32247	0.32201	0.34564	0.30107	0.32959
9	0.25977	0.25875	0.27422	0.27857	0.33176	0.31459
10	0.31475	0.3161	0.32023	0.31055	0.32948	0.30645
Weight (kN)	553.6478	553.9053	567.7535	584.8135	589.7962	591.6707
Weight reduction (percentage)	17.7588	17.7206	15.6635	13.1293	12.3892	12.1107

Table 6. Optimal design results for Silo with thirty variables.

Group of Element	Constraint One			Constraint Two		
	GRO	CSS	TLBO	GRO	CSS	TLBO
1	0.25297	0.34543	0.32236	0.32116	0.30579	0.28185
2	0.33252	0.32762	0.3451	0.26957	0.32116	0.3034
3	0.30635	0.33772	0.29503	0.31626	0.35	0.25899
4	0.34707	0.25129	0.30513	0.32934	0.35	0.26117
5	0.28937	0.32143	0.29793	0.27499	0.34743	0.26362
6	0.28224	0.30234	0.34672	0.27977	0.27941	0.31786
7	0.26436	0.29812	0.25715	0.34766	0.31038	0.29951
8	0.25862	0.28594	0.32884	0.33849	0.25	0.26897
9	0.25809	0.3413	0.34648	0.33029	0.30293	0.2995
10	0.34392	0.2599	0.34856	0.28625	0.25	0.26476
11	0.32459	0.33119	0.29741	0.26383	0.32394	0.25549
12	0.32583	0.30706	0.32901	0.31237	0.33197	0.33507
13	0.29277	0.33953	0.27285	0.33736	0.25	0.30605
14	0.29757	0.26708	0.26982	0.26647	0.25	0.34296
15	0.2946	0.33958	0.30887	0.3001	0.35	0.31966
16	0.29474	0.25971	0.27594	0.31944	0.34774	0.30827
17	0.33942	0.32193	0.26614	0.30722	0.32049	0.33153
18	0.28089	0.31204	0.30539	0.26862	0.25	0.3379
19	0.3167	0.30271	0.25261	0.27952	0.28465	0.34889
20	0.31116	0.25531	0.26994	0.26276	0.25	0.25005
21	0.3238	0.29538	0.29521	0.34741	0.31815	0.33654

Table 6. *Cont.*

Group of Element	Constraint One			Constraint Two		
	GRO	**CSS**	**TLBO**	**GRO**	**CSS**	**TLBO**
22	0.25836	0.31413	0.33671	0.28007	0.32062	0.31125
23	0.27941	0.26399	0.30183	0.31973	0.35	0.34899
24	0.26723	0.27624	0.28075	0.32454	0.26019	0.30276
25	0.34343	0.28986	0.32634	0.33977	0.33982	0.29795
26	0.35	0.26216	0.25405	0.25345	0.25	0.33013
27	0.33301	0.25092	0.33797	0.27689	0.31054	0.27278
28	0.25373	0.33171	0.27102	0.28472	0.35	0.2998
29	0.26444	0.30078	0.27342	0.29952	0.35	0.34008
30	0.31457	0.32356	0.34425	0.31369	0.30959	0.30746
Weight (kN)	576.8934	577.6467	579.7768	580.8192	583.9202	584.0356
Weight reduction (percentage)	14.3058	14.1939	13.8775	13.7227	13.262	13.2449

Table 6 shows that under the first and second constraint, the structure's weight is 576.8934 kN (14.3058% weight reduction) and 580.8192 kN (13.7227% weight reduction), respectively, by using the GRO algorithms, indicating the GRO algorithms' very intimate performance and better efficacy compared with the other two algorithms. The CSS and TLBO algorithms had a lower ability to optimize the structure's weight. Moreover, the calculated thicknesses obtained by all algorithms are in a suitable range indicating the correct operation of the algorithms in finding the optimal answers in the search space.

3.3. Cooling Tower

The third numerical example is a Cooling Tower, as shown in Figure 6. It is simply supported at Z = 0.

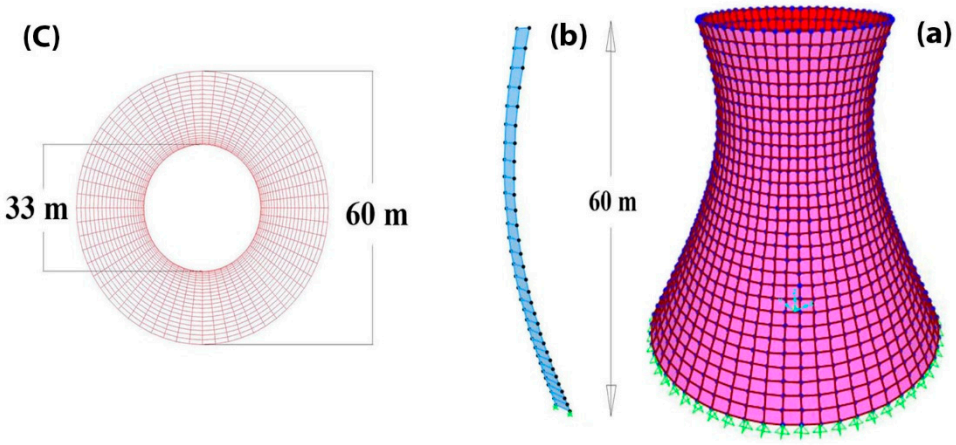

Figure 6. (**a**) A Cooling Tower discretized by quadrilateral shell finite elements, (**b**) substructure of Cooling Tower, (**c**) Cooling Tower dimensions.

The coordinates of the nodes of the Cooling Tower's substructure are presented in Table 7. This substructure is rotated 60 times around the center to create the Cooling Tower. The whole structure uses a four-node quadrilateral flat shell element. In each element, the active mass is considered at the nodes.

In this example, the optimization is performed once by considering the frequency constraint and next by considering both frequency and displacement constraints to evaluate the efficiency of the proposed method:

First constraint: first and third frequencies assumed to be less than 0.3 and 0.28, respectively ($\omega_1 < 0.3, \omega_3 < 0.28$).

Second constraint: first and third frequencies assumed to be less than <0.3 and <0.28, respectively ($\omega_1 < 0.3$, $\omega_3 < 0.28$), and the displacement of node number one under a load of 10,000 kN in the Y direction is less than <0.024.

Table 7. Coordinates of the nodes of the Cooling Tower.

Node Number	Coordinates (x, y, z)	Node Number	Coordinates (x, y, z)	Node Number	Coordinates (x, y, z)
1	(30, 0, 0)	12	(19.4492, 0, 22)	23	(13.9331, 0, 44)
2	(28.9589, 0, 2)	13	(18.6435, 0, 24)	24	(13.9331, 0, 46)
3	(27.929, 0, 4)	14	(17.8796, 0, 26)	25	(14.0328, 0, 48)
4	(26.9115, 0, 6)	15	(17.163, 0, 28)	26	(14.2302, 0, 50)
5	(25.9079, 0, 8)	16	(16.5, 0, 30)	27	(14.5214, 0, 52)
6	(24.9199, 0, 10)	17	(15.8972, 0, 32)	28	(14.9007, 0, 54)
7	(23.9493, 0, 12)	18	(15.3616, 0, 34)	29	(15.3616, 0, 56)
8	(22.9985, 0, 14)	19	(14.9007, 0, 36)	30	(15.8972, 0, 58)
9	(22.0699, 0, 16)	20	(14.5214, 0, 38)	31	(16.5, 0, 60)
10	(21.1665, 0, 18)	21	(14.2302, 0, 40)		
11	(20.2916, 0, 20)	22	(14.0328, 0, 42)		

The thickness member of the substructure elements in this example is the main variable obtained according to the constraints considered by meta-heuristic algorithms. The specified range for the thickness of structural elements is (0.25–0.35 m) in both cases of ten and thirty variables.

In this example, to analyze a cyclic symmetric structure, only the matrix with dimensions 186, equal to the dimensions of the substructure, is calculated. Only once was the calculation of the frequency of the structure without using the concept of cyclic symmetry stopped after about 4 h without any result. The calculation time using the proposed method is significantly reduced.

For Cooling Tower evaluation, the coefficient of the GRO algorithm was $\alpha = 0.7$, $\beta = 0.5$, $\gamma = 0.8$. For 150 operators working in the Cooling Tower, a maximum of 750 repetitions was determined as a termination condition in the example.

For this purpose, the structure was optimized with ten elements or variables (three elements in one group) and once with thirty variables, as shown in Figure 7.

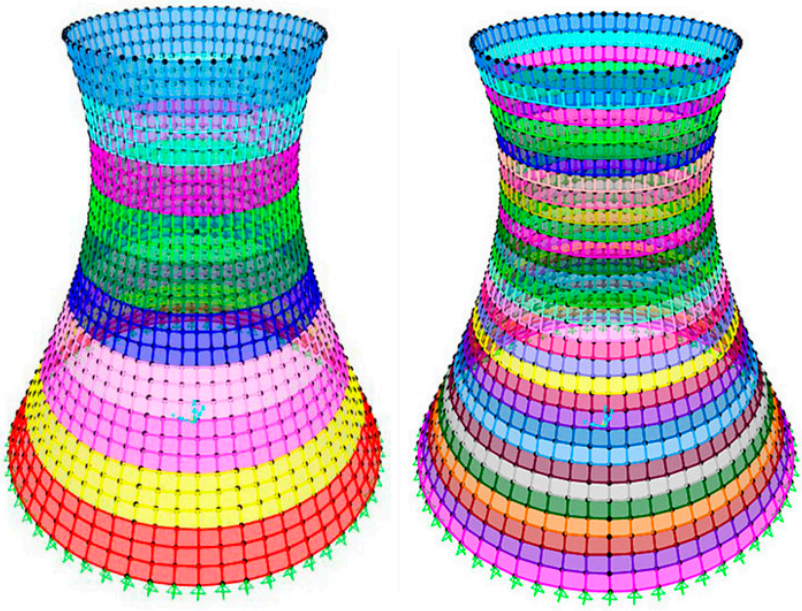

Figure 7. Cooling Tower discretized by ten and thirty quadrilateral shell finite elements.

The structure's weight with an initial thickness of 0.35 m is 6.3602 $\times$ 10^3 kN. The results are presented in Tables 8 and 9. Table 8 shows that the optimal weights obtained by GRO and CSS algorithms were almost similar. Under the first constraint, the structure's weight is 5.1639 $\times$ 10^3 kN (18.8092% weight reduction) using the GRO algorithm and 5.1791 $\times$ 10^3 kN (18.5702% weight reduction) by using the CSS algorithm, indicating their very close performances and better efficacy compared with the TLBO algorithms. Under the second constraint, the structure's weight is 5.2833 $\times$ 10^3 kN (16.9319% weight reduction) by using the GRO algorithm, indicating the GRO algorithm's very intimate performance and a better efficacy compared with the other two algorithms. Moreover, the calculated thicknesses obtained by all algorithms are in a suitable range indicating the correct operation of the algorithms in finding the optimal answers in the search space.

Table 8. Optimal design results for Cooling Tower with ten variables.

Group of Element	Constraint One			Constraint Two		
	GRO	**CSS**	**TLBO**	**GRO**	**CSS**	**TLBO**
1	0.33742	0.33688	0.33612	0.34202	0.32676	0.33282
2	0.2702	0.26562	0.28823	0.32097	0.31465	0.34823
3	0.29306	0.28943	0.28095	0.31782	0.27892	0.28108
4	0.27632	0.27001	0.27058	0.30344	0.26085	0.27296
5	0.32973	0.33358	0.30204	0.3077	0.2742	0.29914
6	0.32264	0.32705	0.30587	0.31144	0.34497	0.32055
7	0.25114	0.25375	0.2746	0.26305	0.26624	0.29247
8	0.30594	0.31358	0.32201	0.25248	0.31324	0.26423
9	0.25114	0.25	0.27422	0.26438	0.3196	0.30321
10	0.25238	0.25322	0.32023	0.28449	0.2583	0.34176
Weight (kN)	5.16 $\times$ 10^3	5.18 $\times$ 10^3	5.43 $\times$ 10^3	5.28 $\times$ 10^3	5.36 $\times$ 10^3	5.57 $\times$ 10^3
Weight reduction (percentage)	18.8092	18.5702	14.6017	16.9319	15.674	12.4084

Table 9. Optimal design results for Cooling Tower with thirty variables.

Group of Element	Constraint One			Constraint Two		
	GRO	**CSS**	**TLBO**	**GRO**	**CSS**	**TLBO**
1	0.3459	0.34619	0.34586	0.30839	0.31638	0.32904
2	0.33143	0.33243	0.3321	0.29527	0.33477	0.34493
3	0.33312	0.33098	0.33066	0.2964	0.27562	0.28275
4	0.34452	0.34429	0.34489	0.32947	0.2517	0.31712
5	0.27742	0.27807	0.2784	0.348	0.27984	0.29386
6	0.28809	0.2895	0.29123	0.29577	0.3173	0.33335
7	0.25415	0.25776	0.25804	0.33183	0.25603	0.32688
8	0.26287	0.26121	0.26145	0.34692	0.29207	0.26672
9	0.28618	0.28212	0.28212	0.34079	0.30397	0.33619
10	0.27722	0.27711	0.27741	0.33078	0.32918	0.34898
11	0.27621	0.27786	0.27854	0.25715	0.34789	0.30144
12	0.34381	0.34673	0.34718	0.25984	0.25573	0.33842
13	0.32408	0.32499	0.32521	0.28145	0.25559	0.3088
14	0.30331	0.30355	0.304	0.3264	0.3405	0.26547
15	0.29603	0.29459	0.2941	0.26143	0.27699	0.26998
16	0.3446	0.34786	0.3489	0.33107	0.28078	0.29069
17	0.343	0.34412	0.344	0.2536	0.27745	0.32487
18	0.32094	0.32502	0.32554	0.33921	0.30914	0.33255
19	0.27975	0.27827	0.27781	0.32665	0.25043	0.32899
20	0.26122	0.25572	0.25603	0.25729	0.25043	0.28185

Table 9. *Cont.*

Group of Element	Constraint One			Constraint Two		
	GRO	**CSS**	**TLBO**	**GRO**	**CSS**	**TLBO**
21	0.26689	0.26755	0.26834	0.26037	0.33234	0.3034
22	0.2644	0.26331	0.26222	0.25246	0.25907	0.25899
23	0.2775	0.27809	0.27915	0.34578	0.34298	0.26117
24	0.25	0.25061	0.25003	0.25425	0.26278	0.26362
25	0.25827	0.25381	0.25396	0.26124	0.343	0.31786
26	0.25265	0.25273	0.25312	0.26699	0.28866	0.29951
27	0.25904	0.2615	0.26037	0.25202	0.33224	0.26897
28	0.27787	0.28072	0.28161	0.27124	0.34494	0.2995
29	0.25462	0.2533	0.25552	0.25193	0.29461	0.26476
30	0.27682	0.28003	0.28003	0.27306	0.33866	0.25549
Weight (kN)	5.20×10^3	5.206×10^3	5.210×10^3	5.23×10^3	5.47×10^3	5.38×10^3
Weight reduction (percentage)	18.2321	18.1472	18.0718	17.729	14.0341	15.4067

Table 9 shows that the optimal weights obtained by GRO and CSS algorithms were almost similar. Under the first constraint, the structure's weight is 5.2006×10^3 kN (18.2321% weight reduction) using the GRO algorithm and 5.2060×10^3 kN (18.1472% weight reduction) by using the CSS algorithm, indicating their very close performances and a better efficacy compared to the TLBO algorithms. Under the second constraint, the structure's weight is 5.2326×10^3 kN (17.7290% weight reduction) by using the GRO algorithm, indicating the GRO algorithm's very intimate performance, and a better efficacy compared with the other two algorithms. Moreover, the calculated thicknesses obtained by all algorithms are in a suitable range indicating the correct operation of the algorithms in finding the optimal answers in the search space.

4. Checking the Frequencies and Mode Shapes

To ensure that the frequency constraints are satisfied, limited frequencies with optimal results were considered and are presented in Table 10. In all examples, the constraints were adequately satisfied, and the frequencies were approximately close to the limit values with an average difference of 0.02%.

Table 10. Constrained natural frequencies of structures (Hz).

Structure	Frequency	Limited Frequencies	Ten Variable	Thirty Variable
			GRO	**GRO**
Disk	ω_1	0.29	0.2900	0.2900
	ω_2	0.27	0.2700	0.2700
Silo	ω_1	0.49	0.4900	0.4900
	ω_3	0.3	0.3000	0.3000
Cooling Tower	ω_1	0.3	0.3000	0.3000
	ω_3	0.28	0.2800	0.2800003

Mode shapes of the frequencies and the optimal schemes obtained by the GRO algorithm in the first and third modes are shown in Figures 8–10. In structures, the first mode showed a sway-type shape, and the third mode had a vertical displacement.

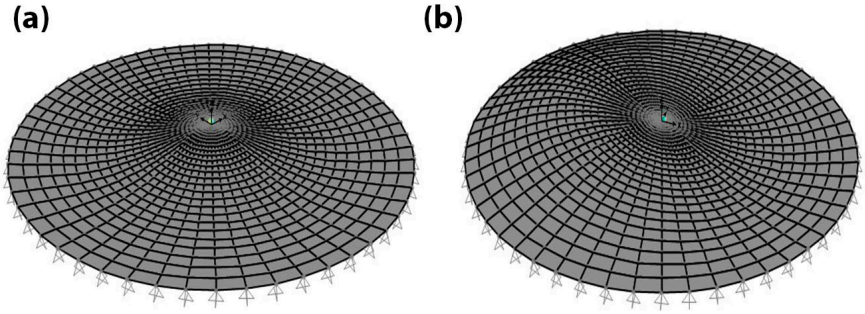

Figure 8. The mode shapes of the Disk. (**a**) First mode shape. (**b**) Third mode shape.

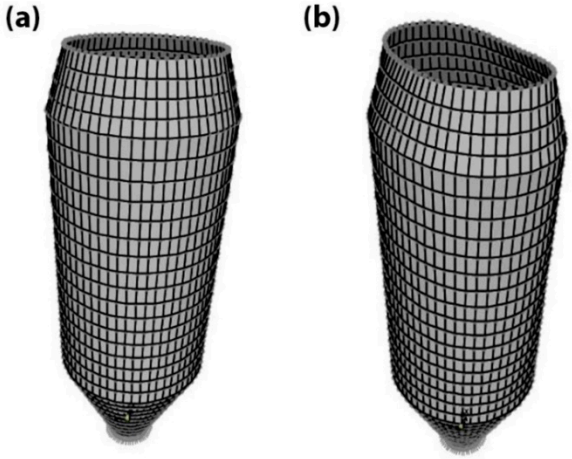

Figure 9. The mode shapes of the Silo. (**a**) First mode shape. (**b**) Third mode shape.

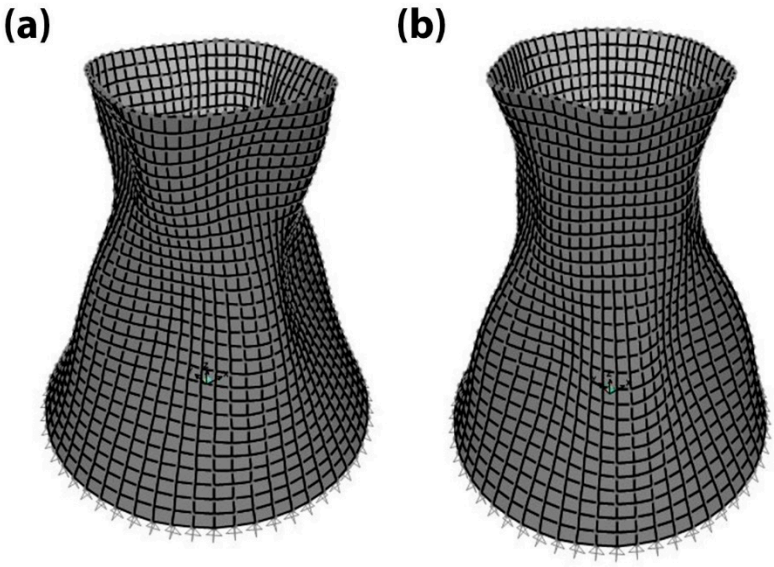

Figure 10. The mode shapes of the Cooling Tower. (**a**) First mode shape. (**b**) Third mode shape.

5. Conclusions

Problems of optimizing structures with cyclic symmetry, considering the frequency constraints and multiple displacements, including non-convex search spaces, are among the most challenging issues in civil engineering. In this study, a design optimization was performed using GRO meta-heuristic algorithms to deal with this problem. It is practically impossible to use algorithms to optimize real-scale structures due to the large volume and the calculations time. The concept of cyclic symmetry was used to reduce the

volume and time of analyses. By decomposing the main structure into repetitive patterns called substructures, the number of calculations was significantly reduced. To evaluate the performance of the proposed method, Disk, Silo, and Cooling Tower were modeled at real scale to confirm the ability of the concept of cyclic symmetry and the GRO algorithm. To better evaluate the performance of this algorithm, two well-known and widely used algorithms, CSS and TLBO, were used for comparison. According to the obtained results, the GRO algorithm was stable and well performed in finding optimal answers. Results show GRO reduces the structure's weight by 14–20% with good accuracy in finding global optimal designs. The algorithm correctly satisfies the constraints and shows that using the concept of cyclic symmetry is an efficient and useful solution in reducing computation time for analyzing symmetric structures. However, it does not apply to nonlinear analyses. As a new application of the concept of cyclic symmetry, it can be used to ellipsoid symmetry problems. The authors intend to implement the proposed scheme for optimizing ellipsoid structures, considering the frequency constraints and multiple displacements.

Author Contributions: S.S.: Conceptualization, methodology, software, validation, analysis, investigation, resources, data curation, writing—original draft preparation, writing—review. M.S.M.: Conceptualization, methodology, resources, data curation, review—editing, supervision, project administration. M.E.S.: resources, review—editing. 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.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kaveh, A.; Ilchi Ghazaan, M. Optimal design of dome truss structures with dynamic frequency constraints. *Struct. Multidiscip. Optim.* **2016**, *53*, 605–621. [CrossRef]
2. Kaveh, A.; Mahdavi, V.R. Colliding Bodies Optimization method for optimum discrete design of truss structures. *Comput. Struct.* **2014**, *70*, 1–12. [CrossRef]
3. Kaveh, A.; Mahdavi, V.R. Colliding-Bodies Optimization for Truss Optimization with Multiple Frequency Constraints. *J. Comput. Civ. Eng.* **2015**, *29*, 4014078. [CrossRef]
4. 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]
5. Kaveh, A.; Mahdavi, V.R. A hybrid CBO–PSO algorithm for optimal design of truss structures with dynamic constraints. *Appl. Soft Comput.* **2015**, *34*, 260–273. [CrossRef]
6. Song, Y.; Zhang, M.; Øiseth, O.; Rønnquist, A. Wind deflection analysis of railway catenary under crosswind based on nonlinear finite element model and wind tunnel test. *Mech. Mach. Theory* **2022**, *168*, 104608. [CrossRef]
7. Ho-Huu, V.; Nguyen-Thoi, T.; Truong-Khac, T.; Le-Anh, L.; Vo-Duy, T. An improved differential evolution based on roulette wheel selection for shape and size optimization of truss structures with frequency constraints. *Neural Comput. Appl.* **2018**, *29*, 167–185. [CrossRef]
8. Lieu, Q.X.; Do, D.T.T.; Lee, J. An adaptive hybrid evolutionary firefly algorithm for shape and size optimization of truss structures with frequency constraints. *Comput. Struct.* **2018**, *195*, 99–112. [CrossRef]
9. Tejani, G.; Savsani, V.J.; Mirjalili, S.; Patel, V.K. Truss optimization with natural frequency bounds using improved symbiotic organisms search. *Knowl.-Based Syst.* **2018**, *143*, 162–178. [CrossRef]
10. Kaveh, A.; Dadras Eslamlou, A. Water strider algorithm: A new metaheuristic and applications. *Structures* **2020**, *25*, 520–541. [CrossRef]
11. Kaveh, A.; Ilchi Ghazaan, M. Vibrating particles system algorithm for truss optimization with multiple natural frequency constraints. *Acta Mech.* **2017**, *228*, 307–322. [CrossRef]
12. Kaveh, A.; Ilchi Ghazaan, M. A new hybrid meta-heuristic algorithm for optimal design of large-scale dome structures. *Eng. Optim.* **2018**, *49*, 235–252. [CrossRef]
13. Carvalho, J.P.G.; Lemonge, A.C.C.; Carvalho, É.C.R.; Hallak, P.H.; Bernardino, H.S. Truss optimization with multiple frequency constraints and automatic member grouping. *Struct. Multidiscip. Optim.* **2018**, *56*, 547–577. [CrossRef]

14. Rao, R.V. *Teaching Learning Based Optimization Algorithm and Its Engineering Applications*; Springer: London, UK, 2016; ISBN 978-3-319-22731-3.
15. Kar, R.; Mandal, D.; Mondal, S.; Ghoshal, S.P. Craziness based Particle Swarm Optimization algorithm for FIR band stop filter design. *Swarm Evol. Comput.* **2012**, *7*, 58–64. [CrossRef]
16. Kaveh, A.; Talatahari, S. A novel heuristic optimization method: Charged system search. *Acta Mech.* **2010**, *3*. [CrossRef]
17. Kaveh, A.; Zolghadr, A. Shape and Size Optimization of Truss Structures With Frequency Constraints Using Enhanced Charged System Search Algorithm. *Asian J. Civ. Eng.* **2011**, *12*, 487–509.
18. Jalili, S.; Talatahari, S. Optimum Design of Truss Structures Under Frequency Constraints using Hybrid CSS-MBLS Algorithm. *KSCE J. Civ. Eng.* **2018**, *22*, 1840–1853. [CrossRef]
19. Kaveh, A.; Zolghadr, A. Optimal design of cyclically symmetric trusses with frequency constraints using cyclical parthenogenesis algorithm. *Adv. Struct. Eng.* **2018**, *21*, 739–755. [CrossRef]
20. Liu, S.; Zhu, H.; Chen, Z.; Cao, H. Frequency-constrained truss optimization using the fruit fly optimization algorithm with an adaptive vision search strategy. *Eng. Optim.* **2020**, *52*, 777–797. [CrossRef]
21. Kaveh, A.; Javadi, S.M. Chaos-based firefly algorithms for optimization of cyclically large-size braced steel domes with multiple frequency constraints. *Comput. Struct.* **2019**, *214*, 28–39. [CrossRef]
22. Wang, D.; Sun, M.; Ma, R.; Shen, X. Numerical Modeling of Ice Accumulation on Three-Dimensional Bridge Cables under Freezing Rain and Natural Wind Conditions. *Symmetry* **2022**, *14*, 396. [CrossRef]
23. Williams, F.W. An algorithm for exact eigenvalue calculations for rotationally periodic structures. *Int. J. Numer. Methods Eng.* **1986**, *23*, 609–622. [CrossRef]
24. Tran, D.M. Component mode synthesis methods using partial interface modes: Application to tuned and mistuned structures with cyclic symmetry. *Comput. Struct.* **2009**, *87*, 1141–1153. [CrossRef]
25. He, Y.; Yang, H.; Deeks, A.J. On the use of cyclic symmetry in SBFEM for heat transfer problems. *Int. J. Heat Mass Transf.* **2014**, *71*, 98–105. [CrossRef]
26. He, Y.; Yang, H.; Xu, M.; Deeks, A.J. A scaled boundary finite element method for cyclically symmetric two-dimensional elastic analysis. *Comput. Struct.* **2013**, *120*, 1–8. [CrossRef]
27. Kaveh, A. *Optimal Analysis of Structures by Concepts of Symmetry and Regularity*; Springer: Vienna, Austria, 2013; ISBN 9783709115657.
28. Kaveh, A. *Computational Structural Analysis and Finite Element Methods*; Springer: Vienna, Austria, 2014; ISBN 978-3-319-02963-4.
29. Kaveh, A.; Koohestani, K. Formation of graph models for regular finite element meshes. *Comput. Assist. Mech. Eng. Sci.* **2009**, *16*, 101–115.
30. Kaveh, A.; Rahami, H. Block circulant matrices and applications in free vibration analysis of cyclically repetitive structures. *Acta Mech.* **2011**, *217*, 51–62. [CrossRef]
31. Kaveh, A.; Rahami, H. An efficient analysis of repetitive structures generated by graph products. *Int. J. Numer. Methods Eng.* **2010**, *84*, 108–126. [CrossRef]
32. Koohestani, K. An orthogonal self-stress matrix for efficient analysis of cyclically symmetric space truss structures via force method. *Int. J. Solids Struct.* **2011**, *48*, 227–233. [CrossRef]
33. Koohestani, K. Exploitation of symmetry in graphs with applications to finite and boundary elements analysis. *Int. J. Numer. Methods Eng.* **2012**, *90*, 152–176. [CrossRef]
34. Koohestani, K. On the decomposition of generalized eigenproblems for the free vibration analysis of cyclically symmetric finite elementmodels. *Int. J. Numer. Methods Eng.* **2010**, *82*, 359–378. [CrossRef]
35. Sarjamei, S.; Massoudi, M.S.; Esfandi Sarafraz, M. Gold Rush Optimization Algorithm. *Iran Univ. Sci. Technol.* **2021**, *11*, 291–327.
36. Kaveh, A.; Talatahari, S. Optimal design of skeletal structures via the charged system search algorithm. *Struct. Multidiscip. Optim.* **2010**, *41*, 893–911. [CrossRef]
37. Kaveh, A.; Talatahari, S. Charged system search for optimum grillage system design using the LRFD-AISC code. *J. Constr. Steel Res.* **2010**, *66*, 767–771. [CrossRef]
38. Kaveh, A.; Talatahari, S. Geometry and topology optimization of geodesic domes using charged system search. *Struct. Multidiscip. Optim.* **2011**, *43*, 215–229. [CrossRef]
39. Talatahari, S.; Kaveh, A.; Mohajer Rahbari, N. Parameter identification of Bouc-Wen model for MR fluid dampers using adaptive charged system search optimization. *J. Mech. Sci. Technol.* **2012**, *26*, 2523–2534. [CrossRef]
40. Kaveh, A.; Talatahari, S. Charged system search for optimal design of frame structures. *Appl. Soft Comput.* **2012**, *12*, 382–393. [CrossRef]
41. Cook, R.D.; Malkus, D.S.; Plesha, M.E.; Witt, R.J. *Concepts and Applications of Finite Element Analysis*, 4th ed.; John Wiley & Sons: New York, NY, USA, 2001; ISBN 978-0-471-35605-9.
42. Batoz, J.; Tahar, M.B. Evaluation of a new quadrilateral thin plate bending element. *Int. J. Numer. Methods Eng.* **1982**, *18*, 1655–1677. [CrossRef]