IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 17 January 2023, accepted 28 January 2023, date of publication 8 February 2023, date of current version 13 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3243164

RESEARCH ARTICLE

Size Optimization of Truss Structures Using Improved Grey Wolf Optimizer

HABES ALKHRAISAT[1], LAMEES MOHAMMAD DALBAH[2], MOHAMMED AZMI AL-BETAR[2,3], MOHAMMED A. AWADALLAH[4,5], KHALED ASSALEH[2], AND MOHAMED DERICHE[2]

[1]Department of Computer Science, University of York, YO10 5DD York, U.K.
[2]Artificial Intelligence Research Center (AIRC), College of Engineering and Information Technology, Ajman University, Ajman, United Arab Emirates
[3]Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, Irbid 1705, Jordan
[4]Department of Computer Science, Al-Aqsa University, Gaza 4051, Palestine
[5]Artificial Intelligence Research Center (AIRC), Ajman University, Ajman, United Arab Emirates

Corresponding author: Mohammed Azmi Al-Betar (m.albetar@ajman.ac.ae)

ABSTRACT The truss structure optimization problem is of substantial importance in diverse civil engineering applications. The ultimate goal is to determine the optimal cross-section (bar) areas of elements used in construction systems by minimizing structure weights. Such structure optimization problems can be categorized into three folds: sizing, shaping, and topology optimization. A number of optimization algorithms have recently been introduced to address truss structure with sizing constraints, including evolutionary algorithms, swarm-based algorithms, and trajectory-based algorithms. Here, the problem of size optimization in truss structures is solved using a modified Grey Wolf Optimizer (GWOM) using three different mutation operators. The Grey Wolf Optimizer, a swarm-based algorithm, was recently introduced to mitigate the wolves' natural behavior in encircling prey and in the hunting process. It has been successfully used to solve a number of optimization problems in both discrete and continuous spaces. Similarly to other optimization algorithms, the main challenge of the GWO is combinatorial and premature convergence. This is due to its navigating behavior over the search space, which is too greedy. One approach to handle greediness and proper balance between exploration and exploitation during the search is controlling mutation operators using appropriate rates. Here, this is achieved using two types of mutation approaches: 1) uniform mutation, and 2) nonuniform mutation. The proposed GWOM versions are evaluated using several benchmark examples of truss structures at 10-bars, 25-bars, 72-bars, and 200-bars. The results are compared with several state-of-the-art methods. The results show that the proposed Optimizer outperforms the comparative methods and fits well with the problem of optimization in truss structures.

INDEX TERMS Exploitation, exploration, grey wolf optimization, mutation, structural optimization, truss structure.

I. INTRODUCTION

In modern structural design practice, construction engineers are frequently faced with structural construction problems. They usually attempt to find an optimal structure design with minimum cost while keeping the highest performance characteristics, including joints displacement, stress on members, or bulking loads, within allowable limits [1]. Research regarding structural engineering and its optimization attracts the interest of both construction engineers and researchers [2]. Structural optimization problems are very challenging and difficult to implement in practice due to many constraints and design variables, non-linearity of the objective function(s) and constraints, and the volatile feasible region [3].

The demand for reliable, computationally inexpensive optimum structural design tools has motivated researchers to develop optimization methods to solve structural design problems more efficiently.

Conventionally, truss structure problems are classified into three interrelated categories: sizing optimization, shape optimization, and topology optimization [4]. The sizing

The associate editor coordinating the review of this manuscript and approving it for publication was Sotirios Goudos.

optimization problems are typically tackled by seeking optimal cross-section areas for the truss elements. So that the truss weight is minimized subject to inequality constraints related to stresses in the members, joint displacements, and buckling loads [5]. On the other hand, shape optimization is tackled by finding the optimal coordination among the joints in the design. Lastly, topology optimization is needed to find the optimal connectivity amongst joints [6].

Previously, gradient or calculus-based optimization algorithms showed excellent performance in tackling small-sized truss structure problems. However, due to the combinatorial nature and to non-linearity, the truss structural optimization problem cannot easily be solved using traditional gradient-based algorithms [7], and cellular automata [8]. Therefore, non-gradient optimization algorithms such as metaheuristic-based approaches emerged as more efficient techniques.

Metaheuristic-based (MH) approaches are general optimization techniques that are used to solve a wide range of optimization problems, including structural optimizations by means of exploiting the problem-specific knowledge represented using a given objective function and a solution formulation [9]. MH approaches to navigate the search spaces using intelligent operators. MH approaches make use of the accumulative knowledge together with proper learning mechanisms controlled by carefully selected parameters to ensure the right balance between the exploration and exploitation processes, thus accelerating convergence toward the optimal solution. They are generally classified into three categories: i) evolutionary algorithms, ii) trajectory-based algorithms, and iii) swarm-based intelligence [10].

Evolutionary algorithms (EA) are initiated with several individuals form a population. Generation after generation, a new population evolves using three types of operators: recombination, mutation, and selection. EA normally stops when the search converges [11]. Although they are very powerful in scanning many regions of the search space at the same time, they cannot go deeply into each direction and may fall into local optima [10]. Several EA algorithms have been successfully used for the truss structure optimization problem, including genetic algorithms [12], harmony search algorithms [13], Immune algorithms [14], and Differential Evolution algorithms [15], [16], [17], all with different degrees of success. Not limited to the truss problem, the following paper conducts a review of the antenna design problem for three EAs; Grey Wolf Optimizer (GWO), the Whale Optimization Algorithm (WOA), and the Salp Swarm Algorithm (SSA) [18].

The second category of MH algorithms includes trajectory-based techniques which start initially with a temporary solution. Iteration after iteration, this solution is improved using neighborhood search until a locally optimal solution in the same area is reached [10]. This type of algorithm is very efficient in exploiting the search area of the solution in which it converges. However, these types of approaches cannot explore several search space areas simultaneously. The most successful trajectory-based algorithms used for truss structure optimization include simulated annealing [19], and Tabu search [20].

Similar to other MH categories, swarm-based algorithms are also initiated with a swarm of several solutions. These solutions normally represent the positions or locations of the swarm packs. The swarm members normally fly/swim together in a collaborative manner, trying to hunt prey, seeking food sources, and avoiding attackers. This cooperation process empowers their behavior in searching and hunting. They are conventionally used as constructive-based approaches that build up the swarm members at each practice from scratch based on their accumulative historical values [21]. There are several swarm intelligence methods proposed for truss structure optimization problems such as Artificial Bee Colony [22], Ant Colony Optimization [23], Cuckoo search [5], Flower Pollination Algorithm [24], Particle Swarm Optimization [1], Mine blast algorithm [25], Red Deer Algorithm [26], etc.

Due to the complex nature of the truss structure search space shape, the attention of researchers has turned towards developing modified or hybridized versions of the MH approaches. This is achieved by considering the specific problem-based knowledge in the optimization framework [27]. Hence, several hybridized and modified versions of the metaheuristic algorithms have been developed and adapted to the different truss structure optimization problems. Kaveh and Talatahari [28] integrated Particle swarm optimizer, ant colony strategy, and harmony search scheme for optimum design of trusses. In another study, an improved version of harmony search and the adaptive harmony search algorithm was designed for sizing optimization of truss structures [29]. Similarly, an enhanced version of particle swarm optimizer for solving size, shape, and topology optimization of truss structures [30].

The Grey wolf optimizer (GWO), which is the recently introduced swarm intelligence method [31], imitates the behavior of grey wolves packs in their hunting and encircling processes. It is seen as an efficient optimization algorithm for a number of reasons: no derivative values are required in the initial search, and it is simple and adapts to a wide range of optimization problems. Due to its excellent characteristics, GWO has been successfully tailored for a wide variety of optimization problems such as economic load dispatch [32], [33], vehicle path planning [34], cervix lesion classification [35], multilevel image thresholding [36], template matching [37], Optimal scheduling workflows [38], parameter estimation [39], unit commitment problem [40], optimal power flow [41], dynamic scheduling in real-world welding industry [42], flow shop scheduling [43], appliances energy scheduling problem [44], and others are reported in [45] and [46].

Moreover, since our focus in this paper toward truss engineering optimization problem, following are some literature use of enhanced GWO for various engineering problems. For instance, the authors of [47], presented an enhanced grey wolf optimization (EGWO) for solving the pressure vessel design problem. In addition, a tension/compression spring design, welded beam design, and speed reducer design classical engineering problems were solved in [48] using the hybrid Grey Wolf Elephant Herding Optimization algorithm (GWOEHO).

Consequently, and based on its success and powerful outcomes in the previous implementations, this paper adopted the GWO. However, as with other swarm intelligence algorithms, the GWO has a chronic shortcoming related to exploration and exploitation balance, as it is more biased toward exploitation during the search process. The three best solutions in the GWO usually attract the other solutions in the hunting and encircling process, hence leading to a greedy search. The reason being is that at each iteration, the GWO depends on the best solutions and neglects the others. This behavior can result in premature convergence [49]. Hence, the GWO cannot ensure the best trade-off between exploration and exploitation during the search. A proposed solution to empower diversification is to utilize the mutation operators which was borrowed from other EA approaches [50], [51], [52].

In this paper, two types of mutation operators are combined within the framework of the Grey Wolf Optimizer (GWOM) to tackle truss structure optimization problems. The two mutation operators are (i) Uniform mutation, and (ii) Nonuniform mutation , which improves the diversity aspects of the GWO. Three versions of GWOM are then proposed which are GWOM1, GWOM2, and GWOM3. It should be noted that the uniform mutation is combined with GWO in GWOM1, while each version of nonuniform mutation is integrated with GWO in GWOM2 and GWOM3. The performance of GWOM versions is evaluated using a series of well-known truss structures benchmark circulated in the literature, including 10-bars, 25-bars, 72-bars, and 200-bars. For comparative evaluation, several state-of-the-art methods have been used to compare the results obtained by GWOM versions against their results. Interestingly, the versions of GWO reveals very successful results compared with well-regard methods using the same truss structure benchmark examples.

The remaining section of this paper is organized as follows: Section II provides the formulation of the Truss Structure Problem. Next, the fundamental concepts of GWO are illustrated in Section III. After that, the GWO hybridized version with mutation operators are established in Section V. The results and discussion are analyzed in Section V. Finally, the conclusion and possible future works are given in Section VI.

II. TRUSS STRUCTURE PROBLEMS

A truss is defined as a geometrically stable structure of assembled straight members (i.e., n bars) connected at pin joints (i.e., m nodes), forming triangular units that create a rigid structure [53]. While designing the trusses, it is crucial to consider the distribution of weights to handle changes in tension and compression on members, to avoid the fragility of the structure. As mentioned earlier, the three basic categories for structural optimization are cost (size), shape, and topology. However, the combinations between these categories are also addressed in the literature. For instance, in [54], the size and shape of truss structures were optimized using a modified simulated annealing algorithm. In addition, the genetic algorithm (GA) was tackled for size and topology optimization of structures [55]. Not limited to the previously mentioned combinations, many others were presented [56], [57], [58]. In this paper, the size structural optimization problem is considered for designing the truss structures.

A. PROBLEM FORMULATION

The solution of the truss structure problem is represented as a vector of cross-section variables $X = [\mathrm{A}_1, \mathrm{A}_2, \ldots, \mathrm{A}_\mathrm{n}]$ where the n cross-sectional areas (or members). The A_i is the cross-section for the i^{th} member, where A_i is assigned a value from the list of available profiles found for the number of n members in order to find the optimal design. Consequently, in order to measure the quality of the problem solution, the objective function is used as shown in Eq. (1).

$$\min \mathrm{W}(X) = \psi\rho \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{A}_\mathrm{i}\mathrm{L}_\mathrm{i} \tag{1}$$

where, $\mathrm{W}(X)$ is the total structure weight, ρ is the material density for each member, L_i is the length of i^{th} member, and A_i is the i^{th} member cross-sectional area. It should be noted that the structure weights are multiplied by the total penalty term (ψ), where $\psi = (1 + \phi_\sigma + \phi_\delta)^\varepsilon$, and ε is a positive penalty exponent set to 2 as recommended in [59].

1) THE STRUCTURAL BEHAVIOUR CONSTRAINTS

Members stress and nodal displacements are the first group of constraints where they are formulated in Eq. (2) and Eq. (4), respectively. Note that the stresses on all members must be within the allowable limits as shown in Eq. (2). Where σ_i is the compressing or tensile stress on the i^{th} element, $\sigma_\mathrm{i}^\mathrm{l}$ & $\sigma_\mathrm{i}^\mathrm{u}$ are the lower and upper limits of allowable compressing or tensile stress. If the stress constraint is satisfied, then no penalty will be added. $\phi_\sigma^\mathrm{i} = 0$ (ϕ_σ^i is the stress penalty of the i^{th} element). On the other hand, if the constraint is not satisfied, then ϕ_σ^i is computed using Eq. (3).

$$\sigma_\mathrm{i}^\mathrm{l} \leq \sigma_\mathrm{i} \leq \sigma_\mathrm{i}^\mathrm{u} \qquad \mathrm{i} = 1, 2, \ldots, \mathrm{n} \tag{2}$$

$$\phi_\sigma^\mathrm{i} = \left|\frac{\sigma_\mathrm{i} - \sigma_{\mathrm{l,u}}}{\sigma_{\mathrm{l,u}}}\right| \tag{3}$$

The third one in the same group is the nodal displacement constraint where the nodes' displacement restrictions are crucial in structural engineering. The structure is not allowed to deflect more than allowable limits as shown in Eq. (4). δ^l and δ^u are the lower and upper displacement limits, respectively. In case the nodal displacement constraint is satisfied, then the displacement penalty of the k^{th} node will be zero (ϕ_δ^k =0), otherwise, a displacement penalty will be added based on Eq. (5).

$$\delta^{l} \leq \delta_{k} \leq \delta^{u} \quad k = 1, 2, \ldots, m \tag{4}$$

$$\phi_{\delta}^{k} = \left|\frac{\delta_{k} - \delta_{L,U}}{\delta_{L,U}}\right| \tag{5}$$

In summary, the total members stress penalty ϕ_σ and the total nodal displacement penalty ϕ_δ are calculated using Eq. (6) and Eq. (7), respectively.

$$\phi_{\sigma} = \sum_{i=1}^{n} \phi_{\sigma}^{i} \tag{6}$$

$$\phi_{\delta} = \sum_{k=1}^{m} \phi_{\delta}^{k} \tag{7}$$

2) THE DESIGN VARIABLES CONSTRAINTS

This is the second group of constraints that deals with the design variables. In size optimization, two search spaces can be used, namely discrete and continuous search spaces. Starting with the discrete search space, the truss solution A elements' areas must only be from the available cross-sectional area set (A_{set}). On the other hand, with the continuous search space assumption, the algorithm constructs a truss where the elements' areas are selected within the allowed limits of the cross-sectional areas (A_l & A_u).

3) OBJECTIVE FUNCTION

The objective function (see Eq. (1)) and the penalty terms are merged together in a new objective function as shown in Eq. (8). We note that the structure weights are multiplied by the total penalty term (ψ), where $\psi = (1+\phi_\sigma+\phi_\delta)^{\varepsilon}$, and ε is a positive penalty exponent set to 2 as recommended in [59].

$$\min W(X) = \psi\rho \sum_{i=1}^{n} A_i L_i \tag{8}$$

III. THE GREY WOLF OPTIMIZER

The grey wolf optimizer (GWO) is a swarm intelligent algorithm established in [31], based on the mathematical modeling of the grey wolf leadership hierarchy and hunting behavior of prey. Given the GWO simplicity, ease of implementation, high search precision, and fast searching speed, the GWO has been used to solve several types of engineering problems. In brief, the GWO main steps are illustrated in the Algorithm 1. while the flowchart of the GWO is given in Fig. 1.

Referring to the GWO flowchart drawn in Fig 1, the first step is to construct the initial population. In this step and depending on the optimization problem at hand, the population needs to be defined. In the case of the truss size optimization problem, the population consists of N individuals, where each individual is basically denoted by the $\vec{X}$ vector with dimension n according to the truss dimensionality ($\vec{X} = [A_1, A_2, \ldots, A_n]$). Consequently, the developed population is a matrix shown in Eq. (9). Accordingly, the fitness value vector illustrated in Eq. (10) is

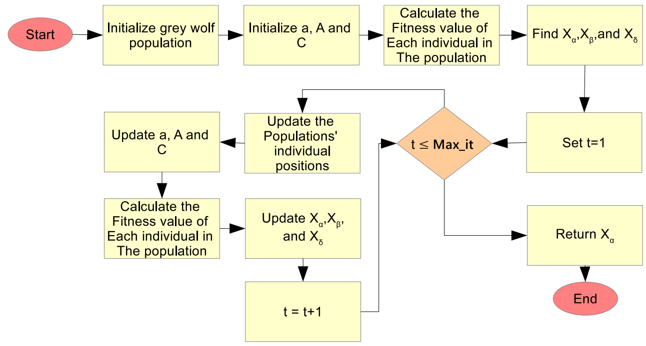

FIGURE 1. The flowchart of the grey wolf optimizer.

Algorithm 1 GWO Pseudo-Code

```
Initialize N & Max_it
Initialize GWO population X_i, ∀i = 1, 2, ..., N
Initialize GWO parameters a, A, C
Calculate the fitness value of each search agent in the population
Select the best, second best and third best solutions (i.e., X_α, X_β, and X_δ).
while (t ≤ Max_it) do
    for each search agent do
        Update the position of the current search agent by Eq (15-21)
    end for
    Update a, A, and C by Eq(13, 14)
    Calculate the fitness value of all search agents in population
    Update X_α, X_β, X_δ
    t = t + 1
end while
Return X_α
```

calculated for each population solution using Eq. (8). Where $W(X_1)$ refers to the structure total weight or quality for the first solution (Note that $\vec{X}$ subscript indicates the solution index in the population).

$$\text{Pop} = \begin{bmatrix} A_{1,1} & A_{1,2} & \ldots & A_{1,n} \\ A_{2,1} & A_{2,2} & \ldots & A_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ A_{N,1} & A_{N,2} & \ldots & A_{N,n} \end{bmatrix} \tag{9}$$

$$\text{Fitness} = \begin{bmatrix} W(X_1) \\ W(X_2) \\ \vdots \\ W(X_N) \end{bmatrix} \tag{10}$$

In GWO, the individuals are prioritized in a hierarchical manner based on their fitness values. Consequently, to model the wolves' leadership hierarchy, the wolves are categorized into four types: alpha (α), beta (β), delta (δ), and omega (ω). These represent the best individual, the second-best individual, the third-best individual, and the rest of the individuals, respectively. The GWO search process consists of three main stages, searching, encircling, and attacking the prey.

During the search process, the three best wolves (α, β, and δ) guide other wolves (ω) to the best areas in the search space. Where the prey possible position is assessed by α, β, and δ. Hencefore, others wolves locations are modified in order to encircle the prey and eventually attack it. Mathematically, this can be achieved using Eq (11) and Eq (12).

$$\vec{D} = \left|\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)\right| \tag{11}$$

$$\vec{X}(t+1) = \vec{X}(t) - \vec{A} \cdot \vec{D} \tag{12}$$

where t represents the t^{th} iteration, $\vec{A}$ and $\vec{C}$ are the coefficient vectors, $\vec{X}_p(t)$ is the position vector of the prey, $\vec{X}(t)$ represents the wolf position. $\vec{D}$ is the distance between the wolf position $\vec{X}$ and the prey position $\vec{X}_p(t)$ (best wolfs in the population). Lastly, The vector $\vec{A}$ and $\vec{C}$ can be expressed by:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \tag{13}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{14}$$

where, the coefficient $\vec{a}$ will linearly decrease from 2 to 0 with the increasing number of iterations, and its dimension as the solution dimension $\vec{X}$. Furthermore, $\vec{r}_1$ and $\vec{r}_2$ are randomly generated within the range of [0, 1].

The mathematical formulas shown in Eq (15), Eq (16), and Eq (17) provide more details about the expressions of Eq.11. These are used to calculate the distances between the position of the current individual and the best individuals represented by α, β, and δ.

$$\vec{D}_\alpha = \left|\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}\right| \tag{15}$$

$$\vec{D}_\beta = \left|\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}\right| \tag{16}$$

$$\vec{D}_\delta = \left|\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}\right| \tag{17}$$

where $\vec{X}$ represents the position vector of current individual, and $\vec{C}_1$, $\vec{C}_2$, $\vec{C}_3$ are randomly generated vectors. $\vec{X}_\alpha$, $\vec{X}_\beta$ and $\vec{X}_\delta$ are the position vectors of α, β, and δ, respectively. Consequently, utilizing the derived $\vec{D}_\alpha$, $\vec{D}_\beta$, and $\vec{D}_\delta$ the final positions of the current individual is updated using the following equations:

$$\vec{X}_1 = \vec{X}_\alpha(t) - \vec{A}_1 \cdot \vec{D}_\alpha \tag{18}$$

$$\vec{X}_2 = \vec{X}_\beta(t) - \vec{A}_2 \cdot \vec{D}_\beta \tag{19}$$

$$\vec{X}_3 = \vec{X}_\delta(t) - \vec{A}_3 \cdot \vec{D}_\delta \tag{20}$$

$$\vec{X}(t+1) = \frac{1}{3}\vec{X}_1 + \frac{1}{3}\vec{X}_2 + \frac{1}{3}\vec{X}_3 \tag{21}$$

After updating the individuals positions, the fitness values are recalculated. Subsequently, α, β and δ are reassigned. These steps are iteratively repeated until a maximum number of iterations (t) is reached Max_it. Once the termination condition is satisfied, α individual decision variables are considered to be the optimized cross-sectional area sizes for the truss members and its fitness value is used as the best achieved total structure weights.

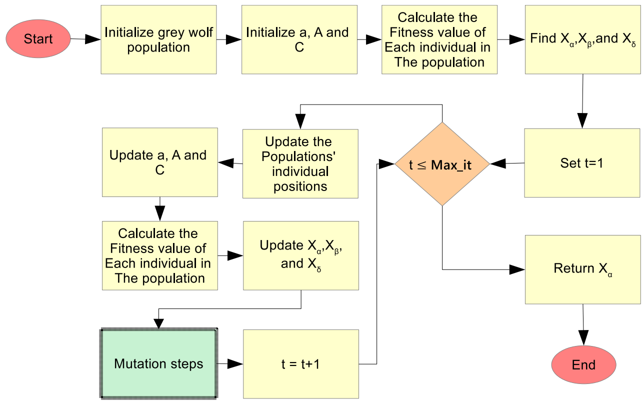

FIGURE 2. The flowchart of the proposed GWOM.

IV. MODIFIED GWO WITH MUTATION OPERATORS FOR TRUSS STRUCTURE

This section introduces a description of the proposed algorithm, called GWOM, for solving the truss size optimization problem. In GWOM, the mutation operators are integrated within the framework of the GWO. The primary motivation behind using the mutation operators is that mutation operators can help the GWO to diversify the search, hence enhancing the balance between exploration and exploitation abilities, as well as increasing the chance of reaching a good solution in the population space. Figure 2 illustrates the procedural steps of the proposed GWOM. Whereas noticed, it is the same as the original GWO flowchart (Fig. 1). However, the modification is basically with the newly green dashed box, which refers to the injection of the mutation operators. The proposed mutation operators will be invoked at the end of each iteration and before the α, β, and δ are reassigned.

In this paper, three independent hybridized GWO versions (GWOM1, GWOM2, GWOM3) are defined with three different mutations. Fig.3 highlights the proposed modified GWO versions based on the type of mutation used. Referring to the figure, two main mutation types are employed, which are (i) Uniform mutation, and (ii) Nonuniform mutation.

Note that the mutation process is conducted in a probabilistic fashion. In for each decision variable, a random number in the range [0,1] is generated and compared to a predefined mutation rate (R_m). Only if it is greater than or equal to R_m, then the mutation process will take place. Further, the mutated individual ($X' = [A'_1, A'_2, \ldots, A'_n]$) will replace the parent individual ($X = [A_1, A_2, \ldots, A_n]$) only if its fitness value is better than the fitness of its parent.

A. GWOM1: GWO WITH UNIFORM MUTATION

The first proposed version of the GWO is called GWOM1, in which the mutation operator used is the uniform mutation. The uniform mutation for $A_i (i = 1, 2, \ldots, n)$ depends on the search space(i.e., continuous or discrete). In the case of a continuous search space, Eq. (22) is used in which the new gene A'_i is randomly set between the lower and upper allowable cross-sectional areas $[A_l, A_u]$. In the case

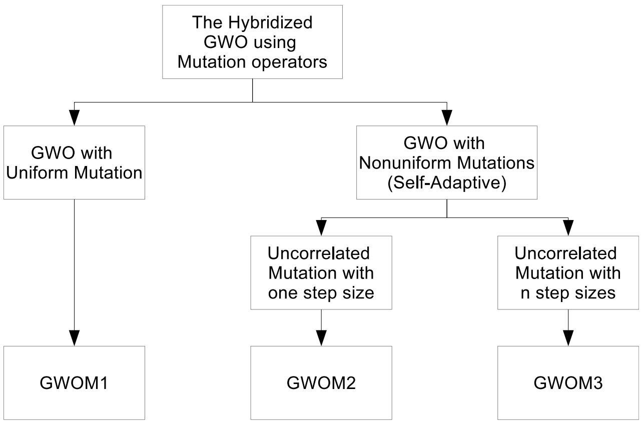

FIGURE 3. **The modified GWO versions according to the mutation types.**

of discrete search space, A_i' is chosen randomly to be one of the pre-defined cross-sectional areas in A_{Set} as illustrated in Eq.(23).

$$A_i' \in [A_l, A_u] \tag{22}$$

$$A_i' \in A_{Set} \tag{23}$$

B. NON-UNIFORM MUTATION

To keep things user-independent and accordingly reduce the user-error possibility of choosing the wrong step size value, the self-adaptive step size mutation will be utilized. Such that the change amount can be adjusted according to the convergence. The new gene (A_i') is calculated based on the old gene (A_i), and a step size σ as shown in Eq. (24) is set for each individual. Where the step size (σ), as well as the solution gene both, will be updated. This is known as a self-adaptive mutation, which is a type of nonuniform mutation. The self-adaptive mutation can further be classified into (i) Self-adaptive mutation with one step size (GWOM2), and (ii) Self-adaptive mutation with n step sizes (GWOM3).

$$A_i' = A_i + \sigma \tag{24}$$

1) GWOM2: GWO WITH NON-UNIFORM MUTATION (SELF-ADAPTIVE MUTATION WITH ONE STEP SIZE)

In the second modified version of the GWO (GWOM2), each solution is appended with one step size value σ. Such that the original individuals are represented as $X = [A_1, A_2, \ldots, A_n, \sigma]$. In which to apply the mutation, first the step size σ is updated before the new individual is generated.

Therefore, to update σ, Eq.(25) will be used (as stated in [60]). Where e stands for the exponential value, τ is the type of learning rate, and N(0, 1) is a normal distribution with a mean of 0 and a standard deviation of 1.

$$\sigma' = \sigma + e^{\tau \cdot N(0,1)} \tag{25}$$

Note that τ is a parameter that can be set by the user, or as a convention, it is inversely proportional to the square root of the number of decision variables (number of members in our problem n) as shown in Eq.(26) [60]. Furthermore, since the negative step size is unaccepted and no deviations in negative, then the newly developed σ needs to be verified to be not less than a pre-defined threshold ϵ. Otherwise, it will be set to the threshold value as illustrated in Eq.(27).

$$\tau \propto \frac{1}{\sqrt{n}} \tag{26}$$

$$\sigma' < \epsilon \rightarrow \sigma' = \epsilon \tag{27}$$

Subsequently, using the updated σ, the solution will be updated where each decision variable is calculated using Eq.(28), Where i = 1,2,...,n. Finally, the new solution becomes $X' = [A_1', A_2', \ldots, A_n', \sigma']$ where both the decision variables and the step size are updated.

$$A_i' = A_i + \sigma' \cdot N(0, 1) \tag{28}$$

2) GWOM3: GWO WITH NON-UNIFORM MUTATION (SELF-ADAPTIVE MUTATION WITH n STEP SIZE)

Next, the last modified version of GWO (GWOM3) is proposed. In this version, the same steps and concepts of GWOM2 are applied to GWOM3. However, instead of having one step size for all decision variables, n step sizes are generated for each variable. Since the population individuals contain n cross-sectional area sizes, we used n step sizes $X = [A_1, A_2, \ldots, A_n, \sigma_1, \sigma_2, \ldots, \sigma_n]$. Where each decision variable has its corresponding σ. This adaptation is used in order to modify each decision variable differently since one cross-sectional area may need to be adjusted with a bigger step size than others and vice versa.

Proceeding to the steps of implementation. Similar to GWOM2, initially, the step sizes will be updated using Eq.(29), but this time two learning rates will be involved. Which are (i) overall learning rate τ', and (ii) specific decision learning rate τ.

This basically generalizes the learning. Moreover, the relations between the learning rates and the number of decision variables are demonstrated in Eq.(30). Next, the generated $\sigma's$ will be checked not to exceed the threshold value Eq.(27).Note that σ will be more specific σ_i. Lastly, the decision variable A_i will be updated utilizing Eq.(31), Where i = 1,2,...,n.

$$\sigma' = \sigma + e^{\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)} \tag{29}$$

$$\tau \propto \frac{1}{\sqrt{2\sqrt{n}}} \quad \& \quad \tau' \propto \frac{1}{\sqrt{2n}} \tag{30}$$

$$A_i' = A_i + \sigma_i' \cdot N_i(0, 1) \tag{31}$$

V. EXPERIMENTS AND RESULTS

In order to evaluate the effectiveness and robustness of the proposed GWOM versions for solving the truss problem, four well-known truss optimization problems in discrete and continuous search spaces are considered: three case studies of 10-bars, four case studies of 25-bars, two case studies of 72-bars, and one case study of 200-bars truss structures. Moreover, the performance of the proposed GWOM versions is compared against the original version of GWO, as well as the other comparative methods taken from the literature.

The other comparative algorithms are, the harmony search algorithm (HS) [61], the weighted superposition attraction algorithm (WSA) [62], the modified teaching–learning-based optimization (TLBO) [59], the mine blast algorithm (MBA) [63], the chaotic swarming of particles (CSP) [64], the improved genetic algorithm (GA) [65], the hybridized ant colony-harmony search genetic algorithm (HACOHS-T) [66], the modified artificial bee colony algorithm (MABC) [67], the discrete heuristic particle swarm ant colony optimization (DHPSACO) [68], the ant lion optimizer (ALO) [69], the adaptive real-coded genetic algorithm (ARCGA) [70]. Furthermore, the particle swarm optimizer (PSO) [71], [72] and symbiotic organisms search algorithm (SOS) [73] along with their improved algorithms, the heuristic particle swarm optimizer (HPSO) [72], the particle swarm optimizer with passive congregation (PSOPC) [71], and the modified symbiotic organisms search algorithm (mSOS) [73].

It should be noted that the proposed GWOM versions and the original GWO were implemented using Matlab 2021 on a mid range PC Core-i7 with 16 GB RAM. Furthermore, The proposed algorithms were run 30 independent times with different initial populations and number of iterations. The experimental results obtained are reported in terms of the best optimal weight. Note that in the next sections the best result achieved for each case study is highlighted using **bold** fonts.

In order to fine-tune the best parameter settings of the proposed algorithms, we conducted a number of experimental scenarios in implementing different settings for the population size and the number of iterations. Three population sizes were tested 30, 50, and 100. These values were suggested randomly to study the influence of the population size on the performance of the proposed algorithm using low, mid, and high population dimensions. In addition, three maximum numbers of iterations are tested 500, 1000, and 1200. As a result, five different scenarios were designed as shown in table 1. That Sen#1 to Sen#3 were used to choose the best population size. Where the population size will vary and the number of iterations will be constant (set to 500). Subsequently, Sen#4 & Sen#5 uses the best-obtained population and vary the number of iteration to select the fittest number of iterations. The experimental scenario that achieved the best results on each case study of the truss problem is presented in table 2. The mean weight was utilized in the case where all scenarios were able to achieve the same best weight. The results of the best experimental scenarios are demonstrated in the following sections. Moreover, for mutation parameters, different values were tested. Eventually, we concluded to set P_m to 0.07 and mu to 0.2.

A. CASE STUDY 1: 10-BARS PLANAR TRUSS STRUCTURE

The 10-Bars planar truss consists of 6 nodes (joints) and 11 bars (members). Figure 4 illustrates a 10-Bars planar. The Young's modulus of member material is E = 10Msi and the material density of the members is $\rho = 0.1\text{lb/in.}^3$. The truss structure is subjected to two vertical loads conditions of P = 100 kips acting on joints 2 and 4. All truss members are subjected to symmetrical stress constraint, where tensile stress is $\sigma = +25$ksi and compression stress is $\sigma = -25$ksi. The displacement constraints of free joints are limited to ±2in in both directions (x and y). The cross-sectional areas of each structure A1,A2, . . . , A10 are defined to be between 0.1 in^2 and 35in^2 for the continuous search space. On the other hand, in discrete search space, two sets were tested. Case 1 set is A_{Set} = { 1.62, 1.8, 1.99, 2.13, 2.38, 2.62, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84,3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.94, 7.22, 7.97, 11.5, 13.50,13.90, 14.2, 15.5, 16.0, 16.9, 18.8, 19.9, 22.0, 22.9, 28.5, 30.0, 33.5 }, and case 2 set is A_{Set} = { 0.10, 0.50, 1, 1.50, 2, 2.50, 3, 3.50, 4, 4.50, 5, 5.50, 6, 6.50, 7, 7.50, 8, 8.50, 9, 9.50, 10, 10.5, 11, 11.5, 12, 12.5, 13, 13.5, 14, 14.5, 15, 15.5, 16, 16.5, 17, 17.5, 18, 18.5, 19, 19.5, 20, 20.5, 21, 21.5, 22, 22.5, 23, 23.5, 24, 24.5, 25, 25.5, 26, 26.5, 27, 27.5, 28, 28.5, 29, 29.5, 30, 30.5, 31, 31.5 }in.^2. Clearly, three versions of the 10-Bars planar truss structure are studied in this research, one continuous version, and two discrete versions.

Table 3 illustrates the best designs obtained by the proposed GWOM versions in comparison with those of the original GWO and other comparative algorithms for both continuous and discrete cases of the 10-bar truss problem. From Table 3, it can be seen that the performance of the GWOM versions is similar to the GWO by obtaining almost the same designs satisfying all constraints according to allowable displacement and stress limits. However, GWOM1 outperforms the original GWO as well as the other two versions of the proposed GWOM (GWOM2 & GWOM3). The reason goes back to the high randomness nature that allows it to escape from local optima. On the other hand, the

TABLE 1. Population and maximum number of iterations experimental Scenarios.

	Pop_Size	Max_itr
Sen 1	30	500
Sen 2	50	
Sen 3	100	
Sen 4		1000
Sen 5		1200

TABLE 2. Best parameters' sittings for each version of the proposed GWOM algorithms according to truss Problem and problem space.

		GWO	GWOM1	GWOM2	GWOM3
10-bar	Continuous	Sen 5	Sen 5	Sen 4	Sen 5
	Discrete case 1	Sen 2	Sen 3	Sen 2	Sen 2
	Discrete case 2	Sen 4	Sen 4	Sen 3	Sen 3
		GWO	GWOM1	GWOM2	GWOM3
25-bar	Continuous	Sen 5	Sen 5	Sen 5	Sen 5
	Discrete case 1	Sen 4	Sen 5	Sen 4	Sen 5
	Discrete case 2	Sen 5	Sen 4	Sen 5	Sen 4
	Discrete case 3	Sen 4	Sen 5	Sen 4	Sen 4
		GWO	GWOM1	GWOM2	GWOM3
72-bar	Continuous	Sen 2	Sen 1	Sen 5	Sen 4
	Discrete case 1	Sen 5	Sen 5	Sen 5	Sen 3
		GWO	GWOM1	GWOM2	GWOM3
200-bar	Discrete	Sen 5	Sen 5	Sen 5	Sen 5

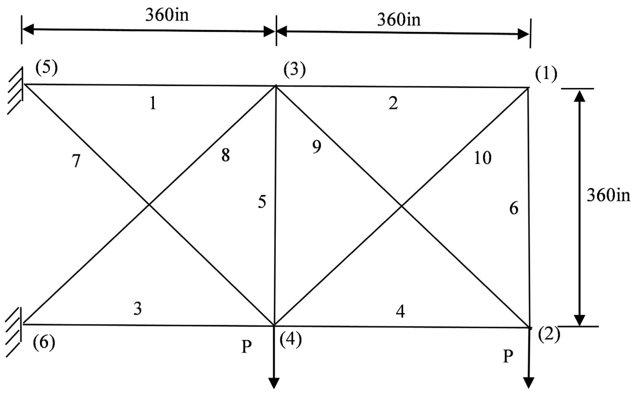

FIGURE 4. Planar 10-bar truss structure.

results of GWO and GWOM versions are exactly the same for the discrete cases.

Reading the results given in Table 3, once again, we also note that the performance of the HS is better than all algorithms in the case of continuous search space. However, it is observable that GWO and GWOM versions' results that satisfy all constraints considered in this research are somehow near the optimal result. Furthermore, in the case of discrete search space. The achievements of GWO and GWOM versions are the best with a weight of 5490.738 for a 10-bar structure - case 1. This weight was reported as well by other comparative algorithms, which are SOS and mSOS. Moreover, in discrete-case 2, GWO and GWOM obtained the second-best weight after the best weight reported by GA.

B. CASE STUDY 2: 25-BAR TRUSS STRUCTURE

The planar 25-bar tower consists of 10 nodes and 25 bars. The planar 25-bar tower is illustrated in Figure 5. The Young's modulus of member material is $E = 10$Msi and the density of member material is $\rho = 0.1$lb/in.^{3}. The 25 members are divided into 8 groups, as follows: G1) A1; G2) A2–A5; G3) A6–A9; G4) A10, A11; G5) A12, A13; G6) A14–A17; G7) A18–A21; and G8) A22–A25. Such that all members of a group have the same cross-sectional area size. Moreover, all truss members are subject to symmetrical stress constraints, where tensile stress is $\sigma = +40$ksi and compression stress is $\sigma = -40$ksi, while displacement constraints of the free nodes are limited to ± 0.35in. in all directions.

Furthermore, there are two independent loading cases that are implemented as shown in Table 4. It should be noted that the P_x, P_y and P_z are the loads along x, y and z axes, respectively. With respect to the members' values, the members can have a cross-sectional area size between 0.01in^2 and 3.4in^2 for the continuous search space. However, with discrete search space, three cases were considered. For Case 1 the following cross-section set are considered $A_{Set} =$ {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.6, 2.8, 3.0, 3.2, 3.4} (in.$_2$), for Case 2, we used $A_{Set} =$ {0.01, 0.4, 0.8, 1.2, 1.6, 2.0, 2.4, 2.8, 3.2, 3.6, 4.0, 4.4, 4.8, 5.2, 5.6, 6.0} (in.$_2$). Lastly, Case 3 set is in Table 5 using the American Institute

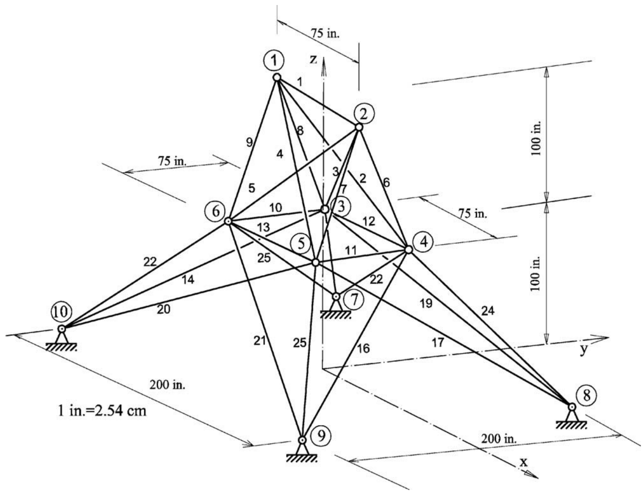

FIGURE 5. Planar 25-bar tower structure.

of Steel Construction (AISC) data. Clearly, four cases of the 25-Bars planar truss structure are studied in this research, one continuous version, and three discrete versions.

The best designs are accomplished by the proposed GWOM versions. The original GWO, and other comparative algorithms on the different cases of the 25-bar structure are recorded in Table 6. It can be seen that the performance of the proposed GWOM algorithms is similar to the original version of GWO in two cases of the 25-bar truss structure (discrete - case 1 and discrete - case 2). Furthermore, the performance of the original GWO, GWOM2, and GWOM3 are almost the same by getting the same best results on case 3 of discrete variables. The performance of original GWO performs better than the proposed GWOM algorithms on the continuous case of a 25-bar truss structure.

Reading the results recorded in Table 6, once more time, it can be seen the proposed GWOM versions, as well as the original GWO, are ranked first by achieving the best results on the 25-bar truss structure for the first discrete case. In addition, the GWO, GWOM2, and GWOM3 obtained the first rank by achieving the best results on the 25-bar truss structure on the third discrete case. The proposed GWOM algorithms and the original GWO are ranked second on the 25-bar truss structure on the second discrete case, while the first best algorithm was HPSACO for the same case study. Finally, the performance of the proposed GWOM algorithms is very competitive with the other competitors in the continuous case of the 25-bar structure.

C. CASE STUDY 3: 72-BAR TRUSS STRUCTURE

This case study includes 72 bars with 20 nodes as shown in Figure 6. The Young's modulus of member material is E = 10Msi and the density of member material is $\rho = 0.1\text{lb}/\text{in.}^3$. The 72 truss members are divided into 16 groups, as listed in Table 7. Moreover, all members are subjected to symmetrical stress, where tensile stress is $\sigma = +25$ksi and compression stress is $\sigma = -25$ksi, while the displacement constraints

TABLE 3. Performance of the proposed GWOM algorithms compared to other comparative algorithms for the 10-bar truss structure.

10-bar for Continuous											
Algorithm	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	W(A)
GWO	30.313	0.104	23.185	15.324	0.100	0.549	21.167	7.466	0.101	21.487	5061.486
GWOM1	30.669	0.101	23.266	15.116	0.100	0.563	21.088	7.468	0.100	21.389	5061.221
GWOM2	30.635	0.102	23.295	15.057	0.100	0.516	21.161	7.473	0.101	21.397	5061.703
GWOM3	30.629	0.102	23.048	15.426	0.100	0.494	21.030	7.474	0.100	21.459	5061.6
HS	30.150	0.102	22.710	15.270	0.102	0.544	7.541	21.560	21.450	0.100	5057.880
HPSO	30.704	0.100	23.167	15.183	0.100	0.551	7.460	20.978	21.508	0.100	5060.920
WSA	30.538	0.100	23.176	15.248	0.100	0.554	7.458	21.027	21.522	0.100	5060.850
TLBO	30.668	0.100	23.158	15.223	0.100	0.542	21.026	7.465	0.100	21.466	5060.973
ALO	30.101	0.100	23.480	15.559	0.100	0.568	7.417	21.016	21.454	0.100	5061.530
10-bar for Discrete case 1											
Algorithm	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	W(A)
GWO	33.50	1.62	22.90	14.20	1.62	1.62	22.90	7.97	1.62	22	5490.738
GWOM1	33.5	1.62	22.90	14.2	1.62	1.62	22.9	7.97	1.62	22	5490.738
GWOM2	33.5	1.62	22.90	14.2	1.62	1.62	22.9	7.97	1.62	22	5490.738
GWOM3	33.5	1.62	22.90	14.2	1.62	1.62	22.9	7.97	1.62	22	5490.738
MBA	30.00	1.62	22.90	16.90	1.62	1.62	7.97	22.90	22.90	1.62	5507.750
HPSO	30.00	1.62	22.90	13.50	1.62	1.62	7.97	26.50	22.00	1.80	5531.980
mSOS	33.50	1.62	22.90	14.20	1.62	1.62	7.97	22.90	22.00	1.62	5490.738
SOS	33.50	1.62	22.90	14.20	1.62	1.62	7.97	22.90	22.00	1.62	5490.738
TLBO	33.50	1.62	22.90	14.20	1.62	1.62	22.90	7.97	1.62	22.00	5490.740
PSO	30.00	1.62	30.00	13.50	1.62	1.80	11.50	18.80	22.00	1.80	5581.760
10-bar for Discrete case 2											
Algorithm	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	W(A)
GWO	31.5	0.1	23	15	0.1	0.5	20.5	7.5	0.1	21.5	5052.42
GWOM1	30.5	0.1	23	16	0.1	0.5	21	7.5	0.1	21	5052.42
GWOM2	31.5	0.1	23.5	14.5	0.1	0.5	21	7.5	0.1	21	5052.42
GWOM3	31	0.1	23	15.5	0.1	0.5	20.5	7.5	0.1	21.5	5052.42
MBA	29.5	0.1	24.0	15.0	0.1	0.5	7.5	21.5	21.5	0.1	5067.33
PSO	24.5	0.1	22.5	15.5	0.1	1.5	8.5	21.5	27.5	0.1	5243.71
PSOPC	25.5	0.1	23.5	18.5	0.1	0.5	7.5	21.5	23.5	0.1	5133.16
HPSO	31.5	0.1	24.5	15.5	0.1	0.5	7.5	20.5	20.5	0.1	5073.51
GA	30.5	0.5	16.5	15	0.1	0.1	0.5	18	19.5	0.5	4,217.30
FA	25.5	1.5	24.5	13.5	0.1	2	12	19	20	2.5	5139.37
AFA	31	0.1	23	15	0.1	0.5	7.5	21	21.5	0.1	5,059.87

TABLE 4. 25-bar truss loading cases.

Load Cases	Nodes	P_x (kips)	P_y (kips)	P_z (kips)
Case 1	1	1.0	10.0	-5.0
	2	0.0	10.0	-5.0
	3	0.5	0.0	0.0
	6	0.5	0.0	0.0
Case 2	1	0.0	20.0	-5.0
	2	0.0	-20.0	-5.0

of the free nodes are limited to ±0.25in in both x and y directions.

Table 8 illustrates the two loading cases that are used to solve this testing problem. The P_x, P_y and P_z are the loads along x, y, and z axis, respectively. Furthermore, concerning the problem search space, the cross-sectional size in the continuous search space ranges between 0.1in^2 and 3in^2. On the other hand, the members are chosen to be from $A_{Set} = \{0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3, 3.1, 3.2\}$ in the case of discrete search space.

The best designs for the 72-bar truss structure obtained using the proposed GWOM versions and other algorithms are recorded in Table 9. It can be observed that the proposed GWOM versions, original GWO, MHA, and DHPSACO perform better than the other comparative algorithms by finding the best design with minimal weight on the discrete search space case of this case study. Studying the performance of the competitors in the continuous search space case, it can be seen that the HPSO is ranked first by finding the best design with minimal weight. While the proposed GWOM2 is ranked second by achieving the design with the second-best weight. In addition, the GWOM3, GWOM1, and GWO got the third, fourth, and fifth rankings, respectively. This proves the efficiency of the proposed GWOM versions against the original version of GWO and other comparative algorithms.

TABLE 5. The cross-section areas following the AISC norm.

No.	in^2	No.	in^2	No.	in^2	No.	in^2
1	0.111	17	1.563	33	3.84	49	11.5
2	0.141	18	1.62	34	3.87	50	13.5
3	0.196	19	1.8	35	3.88	51	13.9
4	0.25	20	1.99	36	4.18	52	14.2
5	0.307	21	2.13	37	4.22	53	15.5
6	0.391	22	2.38	38	4.49	54	16
7	0.442	23	2.62	39	4.59	55	16.9
8	0.563	24	2.63	40	4.8	56	18.8
9	0.602	25	2.88	41	4.97	57	19.9
10	0.766	26	2.93	42	5.12	58	22
11	0.785	27	3.09	43	5.74	59	22.9
12	0.994	28	3.13	44	7.22	60	24.5
13	1	29	3.38	45	7.97	61	26.5
14	1.228	30	3.47	46	8.53	62	28
15	1.266	31	3.55	47	9.3	63	30
16	1.457	32	3.63	48	10.85	64	33.5

TABLE 6. Performance of the proposed GWOM algorithms compared to other comparative algorithms for the 25-bar truss structure.

25-bar for Continuous									
Algorithm	G1	G2	G3	G4	G5	G6	G7	G8	W(A)
GWO	1.626	2.036	3.001	0.015	0.014	0.695	1.629	2.657	557.3405
GWOM1	0.013	1.939	3.060	0.013	0.014	0.696	1.687	2.627	545.3739
GWOM2	0.014	2.005	2.957	0.019	0.010	0.686	1.680	2.668	545.4201
GWOM3	0.014	1.982	2.995	0.022	0.012	0.681	1.682	2.664	545.4542
HS	0.047	2.022	2.950	0.010	0.014	0.688	1.657	2.663	544.380
HPSO	0.010	1.970	3.016	0.010	0.010	0.694	1.681	2.643	545.190
WSA	0.010	1.983	3.000	0.010	0.010	0.683	1.678	2.661	545.163
CSP	0.010	1.910	2.798	0.010	0.010	0.708	1.836	2.645	545.090
ALO	0.010	1.994	2.983	0.010	0.010	0.684	1.676	2.665	545.160
25-bar for Discrete case 1									
Algorithm	G1	G2	G3	G4	G5	G6	G7	G8	W(A)
GWO	0.1	0.6	3.4	0.1	2.3	1	0.2	3.4	482.2136
GWOM1	0.1	0.6	3.4	0.1	2.3	1	0.2	3.4	482.2136
GWOM2	0.1	0.6	3.4	0.1	2.3	1	0.2	3.4	482.2136
GWOM3	0.1	0.6	3.4	0.1	2.3	1	0.2	3.4	482.2136
MBA	0.1	0.3	3.4	0.1	2.1	1	0.5	3.4	484.850
GA	0.1	0.3	3.4	0.1	2.0	1	0.5	3.4	483.354
TLBO	0.1	0.3	3.4	0.1	2.1	1	0.5	3.4	484.850
PSO	0.4	0.6	3.5	0.1	1.7	1	0.3	3.4	486.540
PSOPC	0.1	1.1	3.1	0.1	2.1	1	0.1	3.5	490.160
HPSO	0.1	0.3	3.4	0.1	2.1	1	0.5	3.4	484.850
25-bar for Discrete case 2									
Algorithm	G1	G2	G3	G4	G5	G6	G7	G8	W(A)
GWO	0.01	1.60	3.2	0.01	0.01	0.80	2	2.40	553.813
GWOM1	0.01	1.60	3.2	0.01	0.01	0.80	2	2.40	553.813
GWOM2	0.01	1.60	3.2	0.01	0.01	0.80	2	2.40	553.813
GWOM3	0.01	1.60	3.2	0.01	0.01	0.80	2	2.40	553.813
MBA	0.01	2	3.60	0.01	0.01	0.80	1.60	2.40	560.590
PSO	0.01	2	3.60	0.01	0.40	0.80	1.60	2.40	566.440
PSOPC	0.01	2	3.60	0.01	0.01	0.80	1.60	2.40	560.590
HPSO	0.01	2	3.60	0.01	0.01	0.80	1.60	2.40	560.590
HPSACO	0.01	1.60	3.20	0.01	0.01	0.80	2.00	2.40	551.610
25-bar for Discrete case 3									
Algorithm	G1	G2	G3	G4	G5	G6	G7	G8	W(A)
GWO	0.111	2.130	2.880	0.111	0.111	0.766	1.620	2.620	551.140
GWOM1	0.010	1.600	3.200	0.01	0.010	0.800	2	2.400	553.8131
GWOM2	0.111	2.130	2.880	0.111	0.111	0.766	1.620	2.620	551.140
GWOM3	0.111	2.130	2.880	0.111	0.111	0.766	1.620	2.620	551.140
MBA	0.111	2.130	2.880	0.111	0.111	0.766	1.620	2.620	551.140
PSO	1	2.620	2.620	0.250	0.307	0.602	1.457	2.880	567.490
PSOPC	0.111	1.563	3.380	0.111	0.111	0.766	1.990	2.380	556.900
HPSO	0.111	2.130	2.880	0.111	0.111	0.766	1.620	2.620	551.140

TABLE 7. 72-bar truss structure element grouping.

Group Index	Members	Group Index	Members
G1	A1 - A4	G9	A37 - A40
G2	A5 - A12	G10	A41 - A48
G3	A13 - A16	G11	A49 - A52
G4	A17 - A18	G12	A53 - A54
G5	A19 - A22	G13	A55 - A58
G6	A23 - A30	G14	A59 - A66
G7	A31 - A34	G15	A67 - A70
G8	A35 - A36	G16	A71 - A72

TABLE 8. 72-bar truss Load cases.

Load Cases	Nodes	P_x (kips)	P_y (kips)	P_z (kips)
Case 1	17	0	0	-5
	18	0	0	-5
	19	0	0	-5
	20	0	0	-5
Case 2	17	5	5	-5

This success could not achieved without reaching a state of balance between the exploration and exploitation during the search process.

D. CASE STUDY 4: 200-BAR TRUSS STRUCTURE

The performance of the proposed GWOM versions were further evaluated using large-scale truss problems with 200 bars and 77 nodes as shown in Figure 7. The same material with E = 30Msi and ρ = 0.268lb/in^3 is used for all truss members. The 200 truss members are divided into 29 groups, as listed in Table 10. All truss members are subject to symmetrical stress, where tensile stress is σ = +10ksi and compression stress is σ = −10ksi, while there are no displacement constraints on truss nodes.

For this case study, only the discrete search space was considered in this research, such that the values of the member of this problem are selected from the set A_{Set} ={0.10, 0.347, 0.440, 0.539, 0.954, 1.081, 1.174, 1.333, 1.488, 1.764, 2.142, 2.697, 2.80, 3.13 3.565, 3.813, 4.805, 5.952, 6.572, 7.192, 8.525, 9.30, 10.850, 13.330, 14.29, 17.17, 19.18, 23 28.08, 33.70} (in.$_2$). The structure is subjected to the following three independent loading conditions: i) 1.0 kip acting in the

TABLE 9. Performance of the proposed GWOM algorithms compared to other comparative algorithms for the 72-bar truss structure.

72-bar for Continuous																	
Algorithm	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	W(A)
GWO	1.885	0.497	0.100	0.100	1.308	0.502	0.100	0.100	0.681	0.499	0.103	0.100	0.136	0.536	0.407	0.562	377.982
GWOM1	1.882	0.502	0.100	0.100	1.309	0.500	0.100	0.100	0.688	0.504	0.100	0.111	0.135	0.531	0.403	0.549	377.9663
GWOM2	1.895	0.499	0.100	0.100	1.308	0.497	0.101	0.100	0.682	0.507	0.100	0.100	0.136	0.535	0.401	0.553	377.926
GWOM3	1.893	0.500	0.100	0.100	1.321	0.502	0.100	0.100	0.662	0.505	0.100	0.106	0.137	0.535	0.398	0.545	377.9596
HS	1.790	0.521	0.100	0.100	1.229	0.522	0.100	0.100	0.517	0.504	0.100	0.101	0.156	0.547	0.442	0.590	379.270
HPSO	1.857	0.505	0.100	0.100	1.255	0.503	0.100	0.100	0.496	0.506	0.100	0.100	0.100	0.524	0.400	0.534	369.650
WSA	1.885	0.514	0.100	0.100	1.271	0.511	0.100	0.100	0.526	0.516	0.100	0.100	0.156	0.545	0.412	0.568	379.618
HACOHS-T	1.563	0.563	0.111	0.111	1.266	0.563	0.111	0.111	0.391	0.563	0.111	0.111	0.196	0.563	0.391	0.602	390.180
TLBO	1.881	0.514	0.100	0.100	1.271	0.515	0.100	0.100	0.532	0.513	0.100	0.100	0.157	0.543	0.408	0.573	379.632
CSP	1.945	0.503	0.100	0.100	1.268	0.510	0.100	0.100	0.507	0.517	0.108	0.100	0.156	0.540	0.422	0.579	379.970
72-bar for Discrete																	
Algorithm	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	W(A)
GWO	1.9	0.5	0.1	0.1	1.3	0.5	0.1	0.1	0.5	0.5	0.1	0.1	0.2	0.6	0.4	0.6	385.540
GWOM1	1.9	0.5	0.1	0.1	1.3	0.5	0.1	0.1	0.5	0.5	0.1	0.1	0.2	0.6	0.4	0.6	385.540
GWOM2	1.9	0.5	0.1	0.1	1.3	0.5	0.1	0.1	0.5	0.5	0.1	0.1	0.2	0.6	0.4	0.6	385.540
GWOM3	1.9	0.5	0.1	0.1	1.3	0.5	0.1	0.1	0.5	0.5	0.1	0.1	0.2	0.6	0.4	0.6	385.540
MBA	2.0	0.6	0.4	0.6	0.5	0.5	0.1	0.1	1.4	0.5	0.1	0.1	1.9	0.5	0.1	0.1	385.540
DHPSACO	1.9	0.5	0.1	0.1	1.3	0.5	0.1	0.1	0.6	0.5	0.1	0.1	0.2	0.6	0.4	0.6	385.540
PSO	2.6	1.5	0.3	0.1	2.1	1.5	0.6	0.3	2.2	1.9	0.2	0.9	0.4	1.9	0.7	1.6	1089.880
PSOPC	3.0	1.4	0.2	0.1	2.7	1.9	0.7	0.8	1.4	1.2	0.8	0.1	0.4	1.9	0.9	1.3	1069.790
HPSO	2.1	0.6	0.1	0.1	1.4	0.5	0.1	0.1	0.5	0.5	0.1	0.1	0.2	0.5	0.3	0.7	388.940

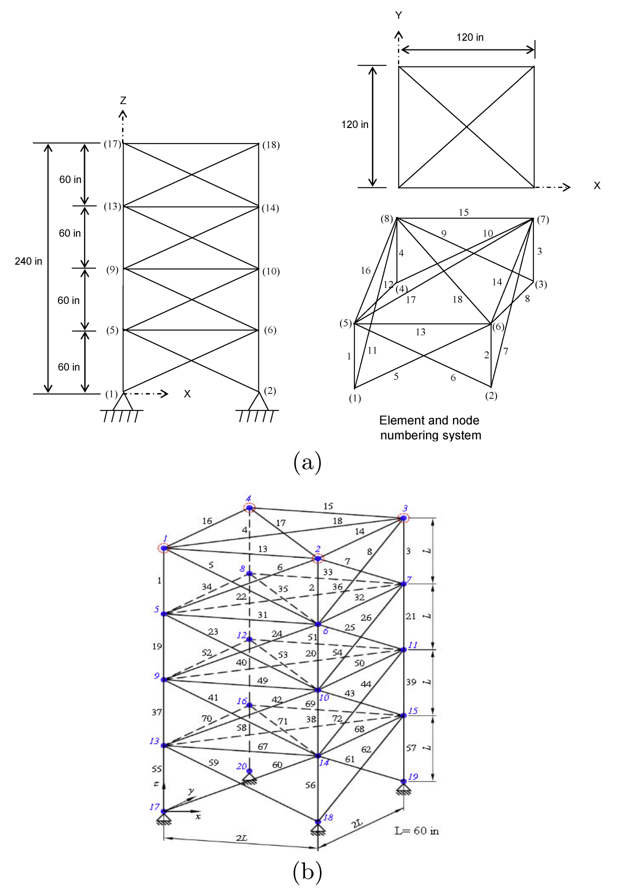

FIGURE 6. 72-bar truss structure.

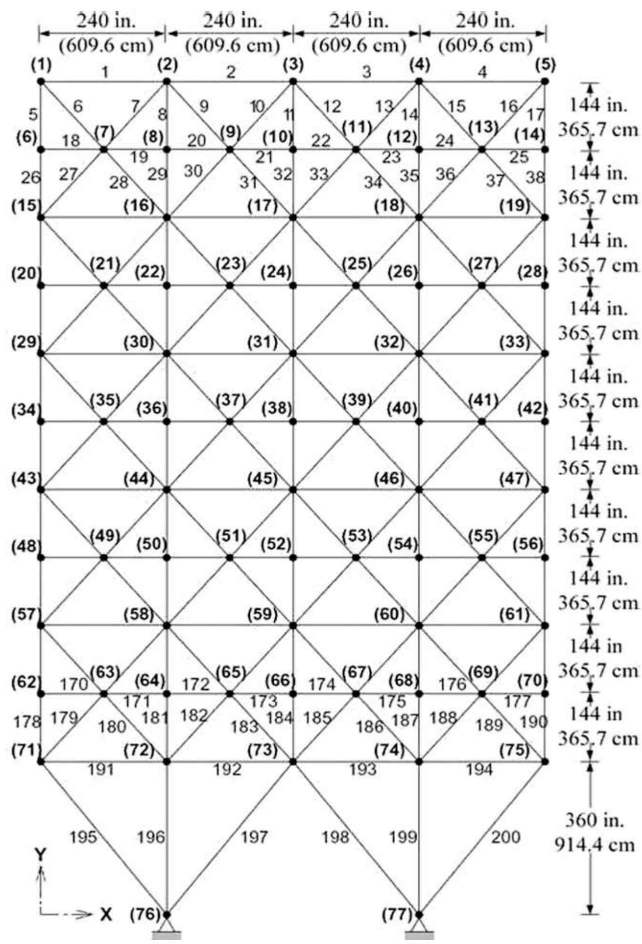

FIGURE 7. Planar 200-bar truss structure.

positive x-direction at nodes 1, 6, 15, 20, 29, 34, 43, 48, 57, 62, and 71; ii) 10 kips acting in the negative y-direction at nodes 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 15, 16, 17, 18, 19, 20, 22, 24, 26, 28, 29, 30, 31, 32, 33, 34, 36, 38, 40, 42, 43, 44, 45, 46, 47, 48, 50, 52, 54, 56, 57, 58, 59, 60, 61, 62, 64, 66, 68, 70, 71, 72, 73, 74, 75; and iii) the previous two loading are acting together.

Table 11 illustrates the best designs obtained by the original GWO and GWOM versions in comparison with those of

TABLE 10. 200-bar truss structure element grouping.

Group	Members
G1	A1, A2, A3, A4
G2	A5, A8, A11, A14, A17
G3	A19, A20, A21, A22, A23, A24
G4	A18, A25, A56, A63, A94, A101, A132, A139, A170, A177
G5	A26, A29, A32, A35, A38
G6	A6, A7, A9, A10, A12, A13, A15, A16, A27, A28, A30, A31, A33, A34, A36, A37
G7	A39, A40, A41, A42
G8	A43, A46, A49, A52, A55
G9	A57, A58, A59, A60, A61, A62
G10	A64, A67, A70, A73, A76
G11	A44, A45, A47, A48, A50, A51, A53, A54, A65, A66, A68, A69, A71, A72, A74, A75
G12	A77, A78, A79, A80
G13	A81, A84, A87, A90, A93
G14	A95, A96, A97, A98, A99, A100
G15	A102, A105, A108, A111, A114
G16	A82, A83, A85, A86, A88, A89, A91, A92, A103, A104, A106, A107, A109, A110, A112, A113
G17	A115, A116, A117, A118
G18	A119, A122, A125, A128, A131
G19	A133, A134, A135, A136, A137, A138
G20	A140, A143, A146, A149, A152
G21	A120, A121, A123, A124, A126, A127, A129, A130, A141, A142, A144, A145, A147, A148, A150, A151
G22	A153, A154, A155, A156
G23	A157, A160, A163, A166, A169
G24	A171, A172, A173, A174, A175, A176
G25	A178, A181, A184, A187, A190
G26	A158, A159, A161, A162, A164, A165, A167, A168, A179, A180, A182, A183, A185, A186, A188, A189
G27	A191, A192, A193, A194
G28	A195, A197, A198, A200
G29	A196, A199

TABLE 11. Performance of the proposed GWOM algorithms compared to other comparative algorithms for the 200-bar truss structure.

	GWO	GWOM1	GWOM2	GWOM3	HACOHS-T	GA	MABC	ARCGA	SOS	mSOS
G1	0.100	0.100	1.081	0.100	0.100	0.347	0.100	0.100	0.100	0.100
G2	0.954	0.954	1.488	1.333	1.081	1.081	1.333	1.081	0.954	0.954
G3	0.954	0.347	0.100	0.100	0.347	0.100	0.100	0.100	0.100	0.440
G4	0.347	0.347	0.347	0.44	0.100	0.100	0.100	0.100	0.100	0.100
G5	2.142	2.142	2.142	2.142	2.142	2.142	2.697	2.142	2.142	2.142
G6	0.44	0.347	0.347	0.347	0.347	0.347	0.347	0.347	0.347	0.347
G7	1.081	0.954	0.44	0.100	0.100	0.100	0.100	0.100	0.100	0.100
G8	4.805	3.131	3.131	3.565	3.131	3.565	3.131	3.131	3.131	3.131
G9	0.44	0.100	0.100	0.100	0.100	0.347	0.100	0.100	0.100	0.100
G10	5.952	4.805	4.805	4.805	4.805	4.805	4.805	4.805	4.805	4.805
G11	0.44	0.539	0.44	0.539	0.440	0.440	0.440	0.347	0.440	0.440
G12	0.100	0.100	0.347	0.100	0.100	0.440	0.539	0.100	0.100	0.440
G13	5.952	5.952	5.952	6.572	5.952	5.952	5.952	5.952	5.952	5.952
G14	1.764	0.100	0.100	0.539	0.100	0.347	0.100	0.100	0.100	0.100
G15	6.572	6.572	6.572	6.572	6.572	6.572	6.572	6.572	6.572	6.572
G16	0.954	0.539	0.539	0.954	0.539	0.954	1.081	0.539	0.954	0.954
G17	0.954	0.44	2.142	1.174	1.174	0.347	0.347	1.081	0.347	0.347
G18	8.525	8.525	8.525	8.525	8.525	8.525	8.525	7.192	8.525	8.525
G19	0.44	0.100	0.100	0.100	0.100	0.100	0.100	0.539	0.100	0.100
G20	9.300	9.300	9.300	9.300	9.300	9.300	9.300	8.525	9.300	9.300
G21	1.488	1.081	1.488	1.333	1.333	0.954	0.954	1.333	0.954	0.954
G22	0.100	1.764	0.347	0.539	0.539	1.764	1.764	1.081	1.764	1.764
G23	13.33	10.85	13.33	13.33	13.330	13.330	13.330	10.850	13.330	13.330
G24	1.174	0.100	1.333	1.081	1.174	0.347	0.440	0.100	0.440	0.440
G25	14.29	13.33	13.33	13.33	13.330	13.330	13.330	13.330	13.330	13.330
G26	1.764	1.488	2.142	2.142	2.697	2.142	2.142	1.488	2.142	2.142
G27	3.565	7.192	3.565	3.565	3.565	4.805	3.813	5.952	3.813	3.813
G28	8.525	10.85	8.525	8.525	8.525	9.300	8.525	13.330	8.525	8.525
G29	17.170	14.29	17.170	17.170	17.170	17.170	19.180	14.290	17.170	17.170
W(A)	29318.23	28110.74	28476.4	28397.15	28030.200	28544.014	28366.365	28347.594	27544.191	27544.191

literature algorithms. Apparently, it can clearly be seen that the performance of three GWOM versions performs better than the GWO by finding the best design with a lower weight.

Reading the results recorded in Table 11, it can be seen that the SO and mSOS are ranked first by obtaining the best design with the minimum weights. The proposed GWOM1 got the second rank by obtaining the second-best design with the second-minimum weights, while the other two versions of the proposed algorithm GWOM3 and GWOM2 obtained the sixth and seventh rankings, respectively. Lastly, the original version of GWO is ranked last by getting the worst design with the highest weight against the other competitors. This proves the efficiency of the proposed GWOM by using the mutation operators within the framework of the GWO to avoid trapping in local optima problems and thus enhance a balance between the exploration and exploitation during the search process.

E. RESULTS ANALYSIS

In summary, four truss structure problems were solved (i) 10-bars problem with a continuous case and two discrete cases, (ii) 25-bars problem with a continuous case and three discrete cases, (iii) 72-bars problem with a continuous case and a discrete case, and (iv) 200-bars problem with a discrete case. Overall, comparing the problems' findings of the proposed GOWM versions against the original GWO, the results are varying to be either the same or GWOM1 outperforms others. Where this achievement was due to the high rate of randomness in GWOM1. On the other hand, the results of GWO and GWOM versions were analyzed and compared against others that exist in the literature. Eventually conclude the following, GWOM versions were able to find the best-published weight in (i) 10-bars problem discrete case 1, (ii) 25-bars problem discrete case 1 & case 3, and (iii) 72-bars problem discrete case. In addition, the GWOM findings were ranked as the second-best in the following problems (i) 10-bars discrete case2, and (ii) 25-bars discrete case2. Furthermore, in the 200-bar problem, the GWOM was ranked as the third-best algorithm according to its findings. Lastly, the results of the remaining problems' cases are acceptable and near other algorithms' results.

VI. CONCLUSION AND FUTURE WORK

This study presented a modified versions of the grey wolf optimizer (GWO), called the GWOM, for the sizing optimization of truss structures with discrete and continuous design variables. This problem is complicated where multiple constraints need to be satisfied. The main goal was to find the best members' sittings (cross-sectional area sizes), for the truss structure, that minimize the total weight. In this paper, three modified versions of the GWO are proposed (i.e., GWOM1, GWOM2, and GWOM3). These versions were obtained by modifying the original GWO algorithm by integrating two types of mutation operators, namely Uniform and Nonuniform mutations, in GWO optimization loop.

The performance of the proposed GWOM versions is evaluated using four truss optimization problems with different study cases including three cases of 10-bar, four cases of 25-bar, two cases of 72-bar, and one discrete case of 200-bar. We compared the best experimental results obtained by the proposed GWOM versions with those of the other competitors published in the literature to prove the effectiveness and robustness of the proposed algorithm. The experimental results demonstrated the effectiveness of the proposed GWOM versions by finding the best design with a minimum weight as similar to some of the other competitors in one case study of 10-bar, two case studies of 25-bar, and one case study of 72-bar. In addition, the GWOM versions achieved the second-best results for 200-bar, one case study of 72-bar, and one case of 25-bar. However, the performance of the proposed GWOM is very competitive with the other competitors in the remaining three case studies. In other perspectives, it can be seen that the performance of the proposed GWOM versions is better than or similar to the original GWO in almost all case studies. This proves the effectiveness of integrating the mutation operators within the framework of GWO in order to enhance the balance between the exploration and exploitation abilities during the search process and avoid trapping in local optima.

With the promising results we obtained in truss structure optimization, future research will focus on expanding the proposed algorithm for other engineering optimization problems with more complex search spaces. Furthermore, the multi-objective GWO and GWOM will then be extended to solve other truss structural optimization problems such as shape and topology. In addition, embedding mutation parameters to the GWO results in notable enhancements in the performance. As a result, implementing the hybridized GWOM versions for other optimization problems might be beneficial in terms of exploration and exploitation balancing.

REFERENCES

[1] A. Kaveh and S. Talatahari, "A particle swarm ant colony optimization for truss structures with discrete variables," *J. Constructional Steel Res.*, vol. 65, nos. 8–9, pp. 1558–1568, Aug. 2009.
[2] V. Ho-Huu, D. Duong-Gia, T. Vo-Duy, T. Le-Duc, and T. Nguyen-Thoi, "An efficient combination of multi-objective evolutionary optimization and reliability analysis for reliability-based design optimization of truss structures," *Expert Syst. Appl.*, vol. 102, pp. 262–272, Jul. 2018.
[3] M. Stolpe, "Truss optimization with discrete design variables: A critical review," *Struct. Multidisciplinary Optim.*, vol. 53, no. 2, pp. 349–374, Feb. 2016.
[4] S. Zargham, T. A. Ward, R. Ramli, and I. A. Badruddin, "Topology optimization: A review for structural designs under vibration problems," *Struct. Multidisciplinary Optim.*, vol. 53, no. 6, pp. 1157–1177, 2016.
[5] A. H. Gandomi, S. Talatahari, X.-S. Yang, and S. Deb, "Design optimization of truss structures using cuckoo search algorithm," *Struct. Design Tall Special Buildings*, vol. 22, no. 17, pp. 1330–1349, Dec. 2013.
[6] H. Assimi, A. Jamali, and N. Nariman-Zadeh, "Sizing and topology optimization of truss structures using genetic programming," *Swarm Evol. Comput.*, vol. 37, pp. 90–103, Dec. 2017.
[7] U. T. Ringertz, "A branch and bound algorithm for topology optimization of truss structures," *Eng. Optim.*, vol. 10, no. 2, pp. 111–124, Jan. 1986.
[8] M. Afshar and A. Faramarzi, "Size optimization of truss structures by cellular automata," *J. Comput. Sci. Eng.*, vol. 3, no. 1, pp. 1–9, 2010.
[9] I. H. Osman and G. Laporte, "Metaheuristics: A bibliography," *Ann. Oper. Res.*, vol. 63, no. 5, pp. 511–623, 1996.
[10] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Comput. Surv.*, vol. 35, no. 3, pp. 268–308, Sep. 2003.
[11] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI, USA: Univ. Michigan Press, 1975.
[12] M. Ohsaki, "Genetic algorithm for topology optimization of trusses," *Comput. Struct.*, vol. 57, no. 2, pp. 219–225, Oct. 1995.
[13] K. S. Lee and Z. W. Geem, "A new structural optimization method based on the harmony search algorithm," *Comput. Struct.*, vol. 82, nos. 9–10, pp. 781–798, 2004.
[14] G.-C. Luh and C.-H. Chueh, "Multi-objective optimal design of truss structure with immune algorithm," *Comput. Struct.*, vol. 82, nos. 11–12, pp. 829–844, May 2004.
[15] V. Ho-Huu, T. Nguyen-Thoi, T. Vo-Duy, and T. Nguyen-Trang, "An adaptive elitist differential evolution for optimization of truss structures with discrete design variables," *Comput. Struct.*, vol. 165, pp. 59–75, Mar. 2016.

[16] V. Ho-Huu, T. Nguyen-Thoi, M. H. Nguyen-Thoi, and L. Le-Anh, "An improved constrained differential evolution using discrete variables (D-ICDE) for layout optimization of truss structures," *Expert Syst. Appl.*, vol. 42, no. 20, pp. 7057–7069, Nov. 2015.
[17] V. Ho-Huu, T. Nguyen-Thoi, T. Truong-Khac, L. Le-Anh, and T. Vo-Duy, "An improved differential evolution based on roulette wheel selection for shape and size optimization of truss structures with frequency constraints," *Neural Comput. Appl.*, vol. 29, no. 1, pp. 167–185, Jan. 2018.
[18] A. D. Boursianis, M. S. Papadopoulou, M. Salucci, A. Polo, P. Sarigiannidis, K. Psannis, S. Mirjalili, S. Koulouridis, and S. K. Goudos, "Emerging swarm intelligence algorithms and their applications in antenna design: The GWO, WOA, and SSA optimizers," *Appl. Sci.*, vol. 11, no. 18, p. 8330, Sep. 2021.
[19] L. Lamberti, "An efficient simulated annealing algorithm for design optimization of truss structures," *Comput. Struct.*, vol. 86, nos. 19–20, pp. 1936–1953, Oct. 2008.
[20] W. A. Bennage and A. K. Dhingra, "Optimization of truss topology using Tabu search," *Int. J. Numer. Methods Eng.*, vol. 38, no. 23, pp. 4035–4052, Dec. 1995.
[21] T. Blickle and L. Thiele, "A comparison of selection schemes used in evolutionary algorithms," *Evol. Comput.*, vol. 4, no. 4, pp. 361–394, Dec. 1996.
[22] M. Sonmez, "Artificial bee colony algorithm for optimization of truss structures," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2406–2418, 2011.
[23] A. Kaveh and S. Talatahar, "A hybrid particle swarm and ant colony optimization for design of truss structures," *ASIAN J. Civil Eng. Building Housing*, vol. 9, no. 4, pp. 329–348, 2008.
[24] G. Bekdaş, S. M. Nigdeli, and X.-S. Yang, "Sizing optimization of truss structures using flower pollination algorithm," *Appl. Soft Comput.*, vol. 37, pp. 322–331, Dec. 2015.
[25] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, "Mine blast algorithm for optimization of truss structures with discrete variables," *Comput. Struct.*, vols. 102–103, pp. 49–63, Jul. 2012.
[26] L. M. Dalbah, H. S. Alshamsi, M. A. Al-Betar, and M. A. Awadallah, "Solving truss structures problem by size optimizing using red deer algorithm," in *Proc. Palestinian Int. Conf. Inf. Commun. Technol. (PICICT)*, Sep. 2021, pp. 1–8.
[27] W. Tang, L. Tong, and Y. Gu, "Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables," *Int. J. Numer. Methods Eng.*, vol. 62, no. 13, pp. 1737–1762, Apr. 2005.
[28] A. Kaveh and S. Talatahari, "Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures," *Comput. Struct.*, vol. 87, no. 3, pp. 267–283, 2009.
[29] S. O. Degertekin, "Improved harmony search algorithms for sizing optimization of truss structures," *Comput. Struct.*, vols. 92–93, pp. 229–241, Feb. 2012.
[30] A. Mortazavi and V. Toğan, "Simultaneous size, shape, and topology optimization of truss structures using integrated particle swarm optimizer," *Struct. Multidisciplinary Optim.*, vol. 54, no. 4, pp. 715–736, Oct. 2016.
[31] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
[32] M. A. Al-Betar, M. A. Awadallah, and M. M. Krishan, "A non-convex economic load dispatch problem with valve loading effect using a hybrid grey wolf optimizer," *Neural Comput. Appl.*, vol. 32, no. 16, pp. 12127–12154, Aug. 2020.
[33] V. K. Kamboj, S. K. Bath, and J. S. Dhillon, "Solution of non-convex economic load dispatch problem using grey wolf optimizer," *Neural Comput. Appl.*, vol. 27, no. 5, pp. 1301–1316, Jul. 2015.
[34] S. Zhang, Y. Zhou, Z. Li, and W. Pan, "Grey wolf optimizer for unmanned combat aerial vehicle path planning," *Adv. Eng. Softw.*, vol. 99, pp. 121–136, Sep. 2016.
[35] A. Sahoo and S. Chandra, "Multi-objective Grey Wolf Optimizer for improved cervix lesion classification," *Appl. Soft Comput.*, vol. 52, pp. 64–80, Mar. 2017.
[36] L. Li, L. Sun, J. Guo, J. Qi, B. Xu, and S. Li, "Modified discrete grey wolf optimizer algorithm for multilevel image thresholding," *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–16, Jan. 2017.
[37] S. Zhang and Y. Zhou, "Template matching using grey wolf optimizer with lateral inhibition," *Optik*, vol. 130, pp. 1229–1243, Feb. 2017.
[38] A. Khalili and S. M. Babamir, "Optimal scheduling workflows in cloud computing environment using Pareto-based grey wolf optimizer," *Concurrency Comput., Pract. Exper.*, vol. 29, no. 11, Jun. 2017, Art. no. e4044.
[39] X. Song, L. Tang, S. Zhao, X. Zhang, L. Li, J. Huang, and W. Cai, "Grey Wolf Optimizer for parameter estimation in surface waves," *Soil Dyn. Earthquake Eng.*, vol. 75, pp. 147–157, Aug. 2015.
[40] L. K. Panwar, S. Reddy, A. Verma, B. K. Panigrahi, and R. Kumar, "Binary Grey Wolf Optimizer for large scale unit commitment problem," *Swarm Evol. Comput.*, vol. 38, pp. 251–266, Feb. 2018.
[41] A. A. El-Fergany and H. M. Hasanien, "Single and multi-objective optimal power flow using grey wolf optimizer and differential evolution algorithms," *Electr. Power Compon. Syst.*, vol. 43, no. 13, pp. 1548–1559, 2015.
[42] C. Lu, L. Gao, X. Li, and S. Xiao, "A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry," *Eng. Appl. Artif. Intell.*, vol. 57, pp. 61–79, Jan. 2017.
[43] G. M. Komaki and V. Kayvanfar, "Grey Wolf Optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time," *J. Comput. Sci.*, vol. 8, pp. 109–120, May 2015.
[44] S. N. Makhadmeh, A. K. Abasi, M. A. Al-Betar, M. A. Awadallah, I. A. Doush, Z. A. A. Alyasseri, and O. A. Alomari, "A novel link-based multi-objective grey wolf optimizer for appliances energy scheduling problem," *Cluster Comput.*, vol. 25, no. 6, pp. 4355–4382, Dec. 2022.
[45] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: A review of recent variants and applications," *Neural Comput. Appl.*, vol. 30, no. 2, pp. 413–435, 2018.
[46] V. Kumar, I. Sharma, and S. Sharma, "A comprehensive survey on grey wolf optimization," *Recent Adv. Comput. Sci. Commun.*, vol. 15, no. 3, pp. 323–333, Mar. 2022.
[47] H. Joshi and S. Arora, "Enhanced grey wolf optimisation algorithm for constrained optimisation problems," *Int. J. swarm Intell.*, vol. 3, nos. 2–3, pp. 126–151, 2017.
[48] Z. Hoseini, H. Varaee, M. Rafieizonooz, and J.-H. J. Kim, "A new enhanced hybrid Grey Wolf Optimizer (GWO) combined with elephant herding optimization (EHO) algorithm for engineering optimization," *J. Soft Comput. Civil Eng.*, vol. 6, no. 4, pp. 1–42, 2022.
[49] M. A. Al-Betar, M. A. Awadallah, H. Faris, I. Aljarah, and A. I. Hammouri, "Natural selection methods for grey wolf optimizer," *Expert Syst. Appl.*, vol. 113, pp. 481–498, Dec. 2018.
[50] N. Mittal, U. Singh, and B. S. Sohi, "Modified grey wolf optimizer for global engineering optimization," *Appl. Comput. Intell. Soft Comput.*, vol. 2016, pp. 1–16, Mar. 2016.
[51] J. Luo and Z. Liu, "Novel grey wolf optimization based on modified differential evolution for numerical function optimization," *Int. J. Speech Technol.*, vol. 50, no. 2, pp. 468–486, Feb. 2020.
[52] H. Xu, X. Liu, and J. Su, "An improved grey wolf optimizer algorithm integrated with Cuckoo search," in *Proc. 9th IEEE Int. Conf. Intell. Data Acquisition Adv. Comput. Syst., Technol. Appl. (IDAACS)*, vol. 1, Sep. 2017, pp. 490–493.
[53] L. Weiwei and T. Yoda, *Bridge Engineering: Classifications, Design Loading, and Analysis Methods*. Butterworth-Heinemann, 2017, doi: 10.1016/C2015-0-00602-5.
[54] C. Millan-Paramo and J. E. Abdalla Filho, "Size and shape optimization of truss structures with natural frequency constraints using modified simulated annealing algorithm," *Arabian J. Sci. Eng.*, vol. 45, no. 5, pp. 3511–3525, May 2020.
[55] I. Delyová, P. Frankovský, J. Bocko, P. Trebuňa, J. Živčák, B. Schürger, and S. Janigová, "Sizing and topology optimization of trusses using genetic algorithm," *Materials*, vol. 14, no. 4, p. 715, Feb. 2021.
[56] G. G. Tejani, V. J. Savsani, V. K. Patel, and P. V. Savsani, "Size, shape, and topology optimization of planar and space trusses using mutation-based improved metaheuristics," *J. Comput. Des. Eng.*, vol. 5, no. 2, pp. 198–214, Apr. 2018.
[57] A. Mortazavi, V. Togan, A. Daloglu, and A. Nuhoglu, "Comparison of two metaheuristic algorithms on sizing and topology optimization of trusses and mathematical functions," *Gazi Univ. J. Sci.*, vol. 31, no. 2, pp. 416–435, 2018.
[58] A. Mortazavi, "The performance comparison of three metaheuristic algorithms on the size, layout and topology optimization of truss structures," *Mugla J. Sci. Technol.*, vol. 5, pp. 28–41, Dec. 2019.
[59] C. V. Camp and M. Farshchin, "Design of space trusses using modified teaching–learning based optimization," *Eng. Struct.*, vols. 62–63, pp. 87–97, Mar. 2014.
[60] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, vol. 53. Cham, Switzerland: Springer, 2003.

[61] K. S. Lee, "Standard harmony search algorithm for structural design optimization," in *Harmony Search Algorithms for Structural Design Optimization*. Cham, Switzerland: Springer, 2009, pp. 1–49.

[62] B. Adil and B. Cengiz, "Optimal design of truss structures using weighted superposition attraction algorithm," *Eng. With Comput.*, vol. 36, no. 3, pp. 965–979, Jul. 2020.

[63] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, "Mine blast algorithm for optimization of truss structures with discrete variables," *Comput. Struct.*, vols. 102–103, pp. 49–63, Jul. 2012.

[64] A. Kaveh, R. Sheikholeslami, S. Talatahari, and M. Keshvari-Ilkhichi, "Chaotic swarming of particles: A new method for size optimization of truss structures," *Adv. Eng. Softw.*, vol. 67, pp. 136–147, Jan. 2014.

[65] V. Toğan and A. T. Daloğlu, "An improved genetic algorithm with initial population strategy and self-adaptive member grouping," *Comput. Struct.*, vol. 86, no. 11, pp. 1204–1218, 2008.

[66] M. Talebpour, A. Kaveh, and V. Kalatjari, "Optimization of skeletal structures using a hybridized ant colony-harmony search-genetic algorithm," *Iranian J. Sci. Technol. Trans. Civil Eng.*, vol. 38, no. C1, pp. 1, 2014.

[67] A. Hadidi, S. K. Azad, and S. K. Azad, "Structural optimization using artificial bee colony algorithm," in *Proc. 2nd Int. Conf. Eng. Optim.*, 2010, pp. 6–9.

[68] A. Kaveh and S. Talatahari, "A particle swarm ant colony optimization for truss structures with discrete variables," *J. Constructional Steel Res.*, vol. 65, nos. 8–9, pp. 1558–1568, Aug. 2009.

[69] M. Salar and B. Dizangian, "Sizing optimization of truss structures using ant lion optimizer," in *Proc. 2nd Int. Conf. Civil Eng., Archit. Urban Manage. Iran*, Aug. 2019, pp. 1–14.

[70] K. Koohestani and S. K. Azad, "An adaptive real-coded genetic algorithm for size and shape optimization of truss structures," in *Proc. 1st Int. Conf. Soft Comput. Technol. Civil, Struct. Environ. Eng.*, B. H. V. Topping and Y. Tsompanakis, Eds. Stirlingshire, U.K.: Civil-Comp Press, 2009, Paper 13, doi: 10.4203/ccp.92.13.

[71] L. J. Li, Z. B. Huang, and F. Liu, "A heuristic particle swarm optimization method for truss structures with discrete variables," *Comput. Struct.*, vol. 87, nos. 7–8, pp. 435–443, Apr. 2009.

[72] L. Li and F. Liu, "Harmony particle swarm algorithm for structural design optimization," in *Harmony Search Algorithms for Structural Design Optimization*. Cham, Switzerland: Springer, 2009, pp. 121–157.

[73] D. T. T. Do and J. Lee, "A modified symbiotic organisms search (mSOS) algorithm for optimization of pin-jointed structures," *Appl. Soft Comput.*, vol. 61, pp. 683–699, Dec. 2017.

HABES ALKHRAISAT received the B.A. degree from Al-Balqa Applied University, in 2001, the master's degree in computer science from The University of Jordan, in 2003, and the Ph.D. degree from Saint Petersburg Electrotechnical University, in 2008. Since 2018, he has been an Associate Professor with the Department of Computer Science, University of York, U.K. His research interests include optimization, power systems, civil engineering, and truss structure.

LAMEES MOHAMMAD DALBAH received the bachelor's degree in computer engineering from Ajman University, where she is currently pursuing the master's degree in artificial intelligence. She worked as a Research Assistant for two years with the Artificial Intelligence Research Center, where she has published a number of research papers in various conferences and journals.

MOHAMMED AZMI AL-BETAR received the Ph.D. degree in artificial intelligence from the University of Science, Malaysia (USM), in 2010. He was a Postdoctoral Research Fellow at USM for three years and invited as a Visiting Researcher two times. He is currently the Head of the Artificial Intelligence Research Center (AIRC), Ajman University. He has been a full-time Faculty Member with the Master of Artificial Intelligence (MSAI) Program, Ajman University, since 2020. He is also the Head of the Evolutionary Computation Research Group (ECRG). He has published more than 175 scientific publications in high-quality and well-reputed journals and conferences. He ranked in Stanford's study of the world's top 2% of scientists. He has more than 15 years of teaching experience in higher education institutions. He has taught several courses in computer science and artificial intelligence fields. His main research interests include scheduling and optimization.

MOHAMMED A. AWADALLAH was the Dean of the Faculty of Computers and Information Sciences, Al-Aqsa University, Gaza, Palestine, from July 2018 to November 2021. In addition, he was appointed as an Adjunct Research Associate (ARA) at the Artificial Intelligence Research Center (AIRC), Ajman University, United Arab Emirates, in February 2021. He is currently an Associate Professor of artificial intelligence with the Department of Computer and Information Sciences, Al-Aqsa University. He has published more than 80 research papers in international journals and conferences. His main research interests include applied computing, where many real-world complex problems are being solved using metaheuristic-based optimization methods. He was listed as one of the top 2% of top-cited scientists in the world according to Stanford University indicator, in 2022. He serves as a reviewer and an editor for numerous conferences and journals of high repute.

KHALED ASSALEH received the B.Sc. degree in electrical engineering from The University of Jordan, Amman, Jordan, in 1988, the M.S. degree in electronic engineering from Monmouth University, West Long Branch, NJ, USA, in 1990, and the Ph.D. degree in electrical engineering from Rutgers, The State University of New Jersey, New Brunswick, NJ, USA, in 1993. From 2002 to 2017, he was with the American University of Sharjah (AUS). Prior to joining AUS, he had a nine-year of research and development career in the Telecom Industry in the USA with Rutgers, Motorola, and Rockwell/Skyworks. He is currently the Vice Chancellor for Academic Affairs and a Professor of electrical engineering with Ajman University. He holds 11 U.S. patents and has published over 130 articles on signal/image processing and machine learning and their applications. His research interests include bio-signal processing, biometrics, speech and image processing, and pattern recognition. He has served on organization committees of several international conferences, including ICIP, ISSPA, ICCSPA, MECBME, and ISMA. He has also served as a guest editor for several special issues of journals.

MOHAMED DERICHE received the Engineering degree from the National Polytechnic School of Algiers, Algeria, and the M.Sc. and Ph.D. degrees from the University of Minnesota, Minneapolis, MN, USA. He joined the Queensland University of Technology (QUT), Australia, in 1994. In 2001, he joined the King Fahd University of Petroleum and Minerals, Saudi Arabia, where he led the DSP Group. In 2021, he joined Ajman University, United Arab Emirates (UAE), as a Professor of AI/ML. He has supervised over 40 graduate thesis. He has published over 300 papers in different aspects of signal/image processing. His research interests include multimedia signal/image processing and AI/ML applications. He was a recipient of numerous awards.