Rakenteiden Mekaniikka (Journal of Structural Mechanics)
Vol. 59, No. 1, 2026, pp. 1–19
https://rakenteidenmekaniikka.journal.fi/index
https://doi.org/10.23998/rm.16333

VERTAISARVIOITU
KOLLEGIALT GRANSKAD
PEER-REVIEWED
www.tsv.fi/tunnus

Honey Badger Algorithm for optimizing truss structures with discrete design variables

Ouadie El Mrimar[1], Zakaria El Haddad, Bousselham Samoudi and Othmane Bendaou

Summary An optimization strategy utilizing the Honey Badger Algorithm (HBA) is formulated in this article for the optimal design of truss structures with discrete variables under multi-load scenarios. HBA is implemented to optimize fundamental design criteria that involve the cross-sectional areas of the truss members. Structural evaluations are conducted by analyzing different cross-sectional configurations, with the primary objective of minimizing the total mass of the structure while satisfying stress and displacement constraints. The HBA is used as a metaheuristic optimizer to explore the design domain and converge towards optimal configurations. This approach is applied to several benchmark examples, including planar and spatial trusses, and the results are compared with existing studies to assess the algorithm's performance. The optimization outcomes confirm the effectiveness and robustness of HBA in solving constrained structural optimization problems, as well as its ability to achieve rapid convergence toward high-quality solutions.

Keywords: HBA, optimization, truss structures, optimal design

Received: 16 July 2025. Accepted: 15 January 2026. Published online: 3 February 2026.

Introduction

The optimization of truss structures remains a fundamental challenge in structural engineering, aiming to design lightweight and cost-effective systems that satisfy stringent mechanical strength and stability constraints.This field has garnered significant attention in the computational mechanics community due to the nonlinear behavior exhibited by many real-world structures, often caused by complex geometric configurations and material properties [1, 2]. Truss systems are widely employed in engineering applications such as buildings, towers, bridges, and aerospace frameworks, consist of interconnected bars joined at nodes and are designed to efficiently carry loads while minimizing structural mass [3, 4, 5]. The optimal design involves determining the appropriate cross-sectional areas and spatial configuration of the elements, subject to constraints on stresses and displacements. The complexity of truss optimization stems from the presence of both continuous and discrete design variables. Traditional optimization techniques typically assume continuous variables, which may not be suitable for practical scenarios where

[1]Corresponding author: elmrimar.ouadie-etu@uae.ac.ma

cross-sectional areas are selected from a predefined set of standard profiles. Moreover, analytical and classical mathematical programming methods, including linear and nonlinear programming [6], although useful for simple problems, often fail to handle the nonlinearities and high-dimensionality associated with real-world truss structures. These methods usually require rigid problem formulations and may lack robustness when dealing with multiple constraints or nonconvex objective functions. In parallel with structural optimization, stochastic modeling techniques have proven to be relevant for understanding the dynamic and uncertain behavior of physical systems [7, 8]. Such an approach contributes to enhancing the robustness and reliability of truss designs.

To address these limitations, researchers have increasingly turned to evolutionary metaheuristic algorithms. Inspired by natural, biological, or physical processes, such algorithms include Particle Swarm Optimization (PSO) [9, 10, 11, 12], Genetic Algorithms (GA) [13, 14, 15, 16, 17], Harmony Search (HS) [18, 19], Ant Colony Optimization (ACO) [10, 20], Differential Evolution (DE) [21], Artificial Bee Colony (ABC) [22], Bees Algorithm (BA) [23], Firefly Algorithm [24], Cuckoo Search (CS) [25], and Symbiotic Organisms Search (SOS) [26], among others [28, **?**, 16]. These methods are especially useful for solving complex, multimodal problems by effectively exploring vast design spaces and avoiding premature convergence to local optima. Several studies [17] have shown that metaheuristic algorithms can outperform classical methods in terms of both accuracy and computational efficiency, especially for large-scale and highly nonlinear design problems.

In this context, the Honey Badger Algorithm (HBA) has newly emerged as a promising metaheuristic optimization method modeled by the intelligent foraging behavior of honey badgers, HBA balances exploration and exploitation using dynamic digging and hunting strategies [29, 30, 31]. The algorithm adapts its search patterns over iterations, making it particularly effective in navigating nonconvex search spaces and identifying global optima. Compared to traditional evolutionary algorithms, HBA offers a simple yet powerful mechanism for optimization, with competitive convergence performance and robustness across a variety of engineering applications.

This study explores the application of HBA for the optimization of truss structures with discrete design variables, where the primary goal is to minimize structural weight while adhering to stress and displacement constraints. The proposed method applies HBA to optimize the cross-sectional areas of truss members chosen from a predefined catalog, while also allowing for geometric variability in certain cases. To evaluate the effectiveness of HBA, we conduct simulations on several benchmark truss problems, including planar and spatial configurations (e.g., 10-bar, 15-bar, and 25-bar structures). The results are then compared with those reported in the literature using other metaheuristic techniques. While HBA shows promising results, its performance may vary depending on factors such as parameter tuning, initial population quality, and problem complexity. Moreover, like other metaheuristic methods, HBA lacks a strict mathematical proof of convergence, which limits the generalization of its observed performance. These limitations indicate that further testing on a wider range of structural problems is necessary to comprehensively assess the robustness and scalability of the algorithm.

Our findings demonstrate that the Honey Badger Algorithm is a viable and competitive alternative for structural optimization problems, particularly those involving discrete variables (i.e., variables that can only take specific predefined values, such as standard cross-sectional areas) and nonlinear constraints. The method exhibits relatively fast convergence, effective constraint handling, and the ability to identify high-quality solutions in the studied examples, while its performance may vary depending on the problem type,

parameter settings, and initial population. These observations highlight the potential of HBA for broader application in structural engineering optimization, although further studies are needed to confirm its general efficiency.

Honey Badger Algorithm

This section presents the biological foundation and mathematical modeling of the HBA, inspired by the remarkable foraging behavior of the honey badger in its natural habitat as described by [29, 30].

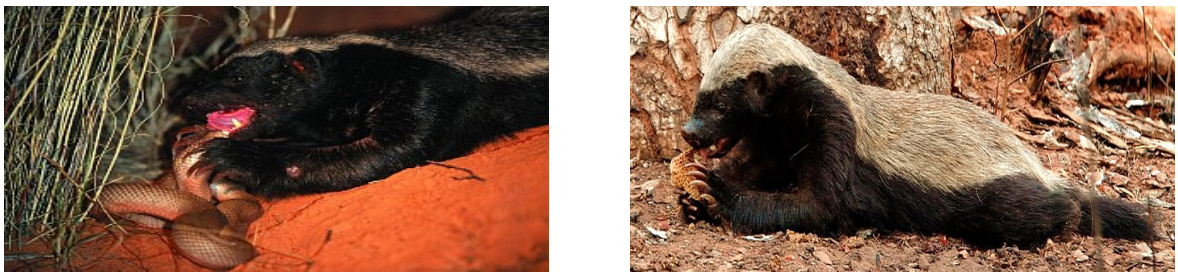

Figure 1. (a) Honey badger bites a venomous snake, and (b) Honey badger eats prey fearlessly on the ground.

Biological overview of the Honey Badger

The honey badger (Mellivora capensis) is a small yet fearless mammal, recognizable by its distinctive black and white fur. It inhabits semi-arid regions and rainforests across Africa, Indian, and the Southwest Asia subcontinent. Despite its modest size measuring between 60 to 77 cm in length and weighing between 7 to 13 kg, the honey badger is known for its aggressive temperament and formidable hunting capabilities, preying on over sixty species, including venomous snakes. Highly intelligent, the honey badger has been observed using tools and can climb trees to access bird nests and beehives (See Fig.). It is a solitary animal that resides in self-dug burrows and only interacts with other badgers during mating. With 12 recognized subspecies, honey badgers reproduce year-round, without a fixed breeding season. Their bold behavior is legendary: when cornered, they do not hesitate to confront predators much larger than themselves. This resilience, along with their versatile foraging skills, makes them a compelling inspiration for metaheuristic algorithm design as noted by [29]).

Foraging behavior and its algorithmic analogy

In the wild, the honey badger primarily locates its prey by moving slowly and using its highly developed olfactory senses. Once a scent is detected, it estimates the prey's location and initiates a digging process to capture it. Remarkably, a single honey badger may dig up to fifty holes in a single day within a radius exceeding 40 kilometers during food search. Though it is fond of honey, the honey badger is not efficient at locating beehives. Interestingly, it forms a mutualistic relationship with the honeyguide bird, which can locate hives but cannot access the honey. The bird leads the badger to the hive, which the badger then opens using its powerful claws allowing both species to benefit from the cooperation [32].

The HBA mimics this dual foraging strategy through two operational modes:

- **Digging Mode:** The algorithm simulates the badger's use of olfactory tracking to locate prey and determine optimal digging points.
- **Honey Mode:** Inspired by its collaboration with the honeyguide bird, this mode models a more direct guidance toward optimal solutions.

These behaviors form the core of HBA's exploration and exploitation strategies, enabling it to effectively navigate complex optimization landscapes.

Mathematical modeling of the HBA

As previously described, the HBA operates based on two core behavioral strategies: the digging phase and the honey-guided phase. These phases correspond to the dual mechanisms of exploration and exploitation, making HBA a suitable method for global optimization tasks.

Algorithm framework

This subsection presents the mathematical foundation underlying the HBA. Conceptually, HBA is structured to balance global exploration with local exploitation, allowing it to effectively traverse the solution space and converge toward optimal solutions. The pseudocode of the algorithm, outlined in Algorithm 1, consists of three main stages: initialization of the population, fitness evaluation of each solution, and parameter update based on the foraging logic of honey badgers. The population of candidate solutions is modeled as a matrix, where each row corresponds to a distinct agent (honey badger), and each column represents a dimension of the search space:

$$\mathbf{X} = \begin{pmatrix} x_{11} & \cdots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{ND} \end{pmatrix} \tag{1}$$

Here, the position of the i^{th} agent is expressed as:

$$\mathbf{X}_i = [x_{i1},\, x_{i2},\, \ldots,\, x_{iD}], \quad i = 1, 2, \ldots, N \tag{2}$$

where D denotes the number of design variables (i.e., the dimensionality of the problem), and N represents the total number of agents in the population.

Step 1: Population initialization
The optimization process begins by generating an initial population composed of N honey badgers. The initial position of each individual, representing a potential solution, is randomly generated within the bounds of the search space according to the following equation:

$$x_i = lb_i + r_i \cdot (ub_i - lb_i) \tag{3}$$

where x_i is the position vector of the i^{th} honey badger, lb_i and ub_i define the lower and upper boundaries of the i^{th} dimension, respectively, and r_1 is a uniformly distributed random variable in the range [0, 1]. This initialization ensures a diverse initial distribution of solutions within the permissible domain.

Step 2: Intensity
In the HBA, agents are guided toward the best solution by mimicking the scent-tracking behavior of real honey badgers. The scent fades with the square of the distance, as defined by the inverse square law. Thus, the scent intensity I_i perceived by the i^{th} agent is modeled as:

$$I_i = \frac{r_2 \times (x_i - x_{i+1})^2}{4(x_{\text{prey}} - x_i)^2} \tag{4}$$

In this formulation, $(x_i - x_{i+1})^2$ represents the source intensity S emitted by the prey, $d_i = x_{\text{prey}} - x_i$ is the distance between the agent and the prey, and r_2 is a random value uniformly distributed in $[0, 1]$. The term x_{prey} denotes the position of the prey, which corresponds to the globally best solution identified so far. This mathematical modeling enables each agent to dynamically adjust its movement toward optimal regions in the solution space based on the perceived strength and proximity of the target.

Step 3: Density factor
The density factor α plays a critical role in controlling the randomness of the agents' movements over time. It enables a smooth shift from global exploration to local exploitation by gradually reducing its influence as the optimization progresses. In this implementation, it is computed using a Gaussian decay function defined as:

$$\alpha = C \cdot \exp\left(\frac{-t}{t_{\max}}\right) \tag{5}$$

where t is the current iteration count and t_{max} is the total number of iterations, and C is a constant value of ≥ 1 , set $C = 2$. This nonlinear decay mechanism accelerates the transition from exploration to exploitation, encouraging more intensive local search as the algorithm converges.

Step 4: Strategy to escape local optima
To enhance global search capability and avoid premature convergence, the algorithm introduces a control flag F that dynamically modifies the direction of exploration. This mechanism increases the likelihood of escaping local minima by diversifying the exploration patterns and reinforcing the search around unexplored or suboptimal areas of the solution space.

Step 5: Position updating mechanism
The evolution of each candidate solution (i.e., honey badger) is guided by two behaviorally inspired strategies: the digging phase and the honey phase. A random number determines which phase is activated at each iteration for each agent, promoting both exploration and exploitation. In addition to the position update mechanisms, it is crucial to ensure that candidate solutions satisfy structural constraints. Mechanical stress and displacement constraints are enforced using a penalty-based mechanism. Any candidate solution that violates a constraint is assigned a penalty proportional to the magnitude of the violation, which is incorporated into the objective function. Iterations are not stopped when constraints are exceeded; instead, the algorithm continues exploring the design space. The final solutions are those that achieve the minimum structural weight while satisfying all constraints.

Step 5.1: Digging phase
When the digging mode is triggered (with probability 0.5), the agent simulates a nonlinear local search behavior using the following update rule:

$$x_{\text{new}} = x_{\text{prey}} - Fd \cdot \cos(2\pi r) + \alpha \mathcal{N}(0,1) \tag{6}$$

where x_{prey} is the current best solution, $F \in [-1, +1]$ is a random directional factor, d is the estimated distance to the prey, $r \sim U(0,1)$ is a random scalar, α is the time-adaptive density factor, and $\mathcal{N}(0,1)$ is a standard normal random vector.

Step 5.2: Honey phase
If the honey phase is selected, the agent updates its position according to a more direct guidance strategy toward the prey:

$$x_{\text{new}} = x_{\text{prey}} - Fd + \alpha \mathcal{N}(0,1) \tag{7}$$

This mechanism encourages more straight forward convergence while maintaining randomness through α weighted noise. The alternation between these two strategies enhances the global search capabilities of the HBA and helps avoid premature convergence. This description clarifies the internal functioning of HBA, detailing how the alternation between the digging phase and the honey phase balances global exploration and local exploitation, and how the adaptive density factor α helps maintain solution diversity and avoid premature convergence.

Formulation of the structural optimization problem

In this structural design optimization study, the primary objective is to minimize the total weight of the structure while satisfying specified constraints on nodal displacements and member stresses. The design variables correspond to the cross-sectional areas of the truss elements, which are selected from a predefined set of standard section sizes. The optimization problem is expressed as:

$$\min_{X} \; W(X) = \sum_{i=1}^{d} \rho_i A_i L_i \tag{8}$$

$$\text{subject to} \quad g_i(\{X\}) \leq 0, \quad i = 1, 2, \ldots, n_c \tag{9}$$

To handle these constraints in a metaheuristic framework such as the HBA, a penalty-based reformulation is applied. The penalized objective function becomes:

$$f_{\text{penalized}}(X) = W(X) + \lambda \sum_{j=1}^{n_c} \max(0, g_j(X))^2 \tag{10}$$

where, this formulation, $X = \{A_1, A_2, ..., A_d\}$ denotes the vector of design variables, where each A_i corresponds to the cross sectional area of the i^{th} truss member. The parameter d is the total number of structural members, and m is the number of nodes in the structure. For each element i, ρ_i, A_i, and L_i represent the material density, cross-sectional area, and length, respectively. The constraint functions $g_j(X)$ may include stress limits of the form $\sigma_j(X) - \sigma_{\max} \leq 0$, displacement limits such as $\delta_j(X) - \delta_{\max} \leq 0$, and bounds on the design variables $X_i^{\text{l}} \leq X_i \leq X_i^{\text{u}}$. Here, $\sigma_{\max}$ and $\delta_{\max}$ represent the allowable stress and

displacement thresholds, respectively, and λ is the penalty coefficient used to penalize constraint violations.

Algorithm 1 Pseudo-code of the simplified HBA

Set parameters: $t_{\max}$ (max iterations), N (population size)
Initialize population of N honey badgers with random positions in $[lb, ub]$
Evaluate the fitness of each agent using the objective function, assign to f_i
Save the best agent as x_{prey} with fitness f_{prey}
for $t = 1$ to $t_{\max}$ **do**
    Compute the density factor α
    for $i = 1$ to N **do**
        Generate random number r
        Compute distance vector $d = x_{\text{prey}} - x_i$
        Generate directional flag F, where $F \in \{-1, +1\}$
        if $r < 0.5$ **then**
            Update position: $x_i^{\text{new}} = x_{\text{prey}} - Fd + \alpha\mathcal{N}(0, 1)$
        else
            Update position: $x_i^{\text{new}} = x_{\text{prey}} - Fd \cdot \cos(2\pi r) + \alpha\mathcal{N}(0, 1)$
        end if
        Evaluate f_i^{new}
        if $f_i^{\text{new}} < f_i$ **then**
            $x_i \leftarrow x_i^{\text{new}}$
            $f_i \leftarrow f_i^{\text{new}}$
        end if
        if $f_i < f_{\text{prey}}$ **then**
            $x_{\text{prey}} \leftarrow x_i$
            $f_{\text{prey}} \leftarrow f_i$
        end if
    end for
end for
return x_{prey}

Results of numerical examples

In this section, we evaluate the performance of the HBA in solving structural optimization problems involving truss systems. Three benchmark design cases are considered, with the objective of minimizing the total structural weight using discrete cross sectional areas as design variables associated with the truss members. For each example, the HBA was executed over 50 independent runs to ensure statistical reliability of the results. The algorithm was implemented in Python, and all simulations were carried out on an HP workstation equipped with a 7th generation Intel Core i7 processor (3.4 GHz) and 64 GB of RAM. The performance of the HBA was assessed based on the best solution obtained and was compared with results reported in previous studies. In all benchmark truss structures analyzed, the following assumptions are made: the structures are considered weightless, experience small displacements, and are supported by hinged (pinned) supports.

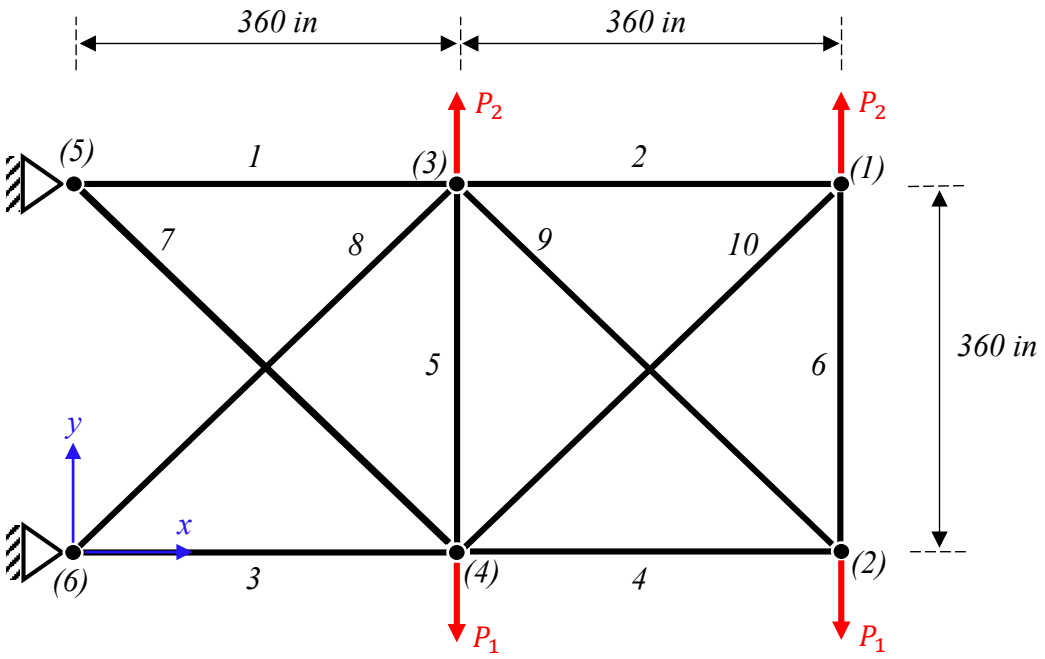

Figure 2. A 10-bar truss structure in a planar configuration.

The 10-bar plane truss structure

The 10-bar truss structure illustrated in Figure 2 serves as a classical benchmark in structural optimization. Reference solutions for this problem have been extensively documented in the literature, including studies by [33, 34, 35, 36]. The structure consists of a two-dimensional cantilever truss with fixed supports and externally applied loads.

The material properties include a Young's modulus of 10000 ksi and a a density of 0.1 lb/in^3. The maximum allowable nodal displacement is limited to ± 2 inches in both horizontal and vertical directions. The members are subjected to axial stress constraints of ± 25 ksi under both tension and compression. Two cases are considered: Case 1, $P_1 = 100$ kips and $P_2 = 0$; and Case 2, $P_1 = 150$ kips and $P_2 = 50$ kips.

The optimization problem involves 10 design variables, each representing the cross-sectional area of a truss member.

Discrete design scenarios are considered. In Case 1, allowable cross-sectional areas are selected from the set: L = [1,62, 1,80, 1,99, 2,13, 2,38, 2,62, 2,63, 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, 5,74, 7,22, 7,97, 11,50, 13,50, 13,90, 14,20, 15,50, 16,00, 16,90, 18,80, 19,90, 22,00, 22,90, 26,50, 30,00, 33,50] (in^2). In Case 2, The discrete variables were chosen from the following set: L = [0,1, 0,5, 1,0, 1,5, 2,0, 2,5, 3,0, 3,5, 4,0, 4,5, 5,0, 5,5, 6,0, 6,5, 7,0, 7,5, 8,0, 8,5, 9,0, 9,5, 10,0, 10,5, 11,0, 11,5, 12,0, 12,5, 13,0, 13,5, 14,0, 14,5, 15,0, 15,5, 16,0, 16,5, 17,0, 17,5, 18,0, 18,5, 19,0, 19,5, 20,0, 20,5, 21,0, 21,5, 22,0, 22,5, 23,0, 23,5, 24,0, 24,5, 25,0, 25,5, 26,0, 26,5, 27,0, 27,5, 28,0, 28,5, 29,0, 29,5, 30,0, 30,5, 31,0, 31,5] (in^2).

The HBA was implemented in Python and applied to both cases. The key parameters were selected according to the specific structure optimization problem. The population size was set to $N = 20$, and the number of iterations was fixed at $t_{\max} = 5000$ to ensure

sufficient convergence. The density factor α was adapted dynamically during the search process using a nonlinear decay function defined in equation 5.

As a stochastic optimization method, the performance of HBA may vary slightly between runs. The quality of the initial population and the selected parameters (population size, number of iterations, etc.) can influence the convergence speed and the quality of the solutions. In the studied examples, HBA generally demonstrated stable and reliable convergence, with minor variations observed across multiple runs.

To promote diversity during the search, random variables r_1 to r_4 were sampled from a uniform distribution in $[0, 1]$. The agents' positions were iteratively updated based on the two behavioral phases of the HBA: the digging phase and the honey phase. The simplified adaptive mechanism, governed by the density factor α, a random directional flag F, and a distance-based attraction term d, effectively guided the population toward feasible and high-quality solutions while avoiding premature convergence.

Tables 1 and 2 summarize the comparative results of the HBA against those reported in previous studies. In Case 1, the HBA obtained a best structural weight of 5490.73 lb within 5,000 structural analyses, consistent with the best results reported by other metaheuristic approaches such as HHS [19], ABC [36], and SAR [33]. However, unlike these approaches which required over 20,000 structural evaluations to reach comparable solutions, HBA achieved competitive efficiency, converging with significantly fewer computations in the studied example. In comparison, other techniques such as HPSO [35] and GA [34] produced slightly higher structural weights of 5531.8 lb and 5613.84 lb, respectively.

Table 1. Comparison of optimization results obtained using HBA with those from the literature for the 10-bar truss (Case 1).

Design Variables	HPSO [35]	ABC [36]	GA [34]	HHS [19]	SAR [33]	HBA
A_1	30.00	33.50	33.50	33.50	33.50	33.50
A_2	1.62	1.62	1.62	1.62	1.62	1.62
A_3	22.90	22.90	22.00	22.90	22.90	22.90
A_4	13.50	14.20	15.00	14.20	14.20	14.20
A_5	1.62	1.62	1.62	1.62	1.62	1.62
A_6	1.62	1.62	1.62	1.62	1.62	1.62
A_7	7.97	7.97	14.20	7.97	7.97	7.97
A_8	26.50	22.90	19.90	22.90	22.90	22.90
A_9	22.00	22.00	19.90	22.00	22.00	22.00
A_{10}	1.80	1.62	2.62	1.62	1.62	1.62
Best weight (lb)	5531.98	5490.74	5613.84	5490.74	5490.74	5490.74
No. of analyses	50000	25800	10000	5000	10000	5000

Table 2. Comparison of optimization results obtained using HBA with those from the literature for the 10-bar truss (Case 2).

Variables (in^2)	Ringertz [37]	HPSO [35]	HHS [19]	HBA
A_1	30.50	31.50	30.50	31.50
A_2	0.10	0.10	0.10	0.10
A_3	23.00	24.50	24.00	23.00
A_4	15.50	15.50	14.00	15.00
A_5	0.10	0.10	0.10	0.10
A_6	0.50	0.50	0.50	0.50
A_7	7.50	7.50	7.50	7.50
A_8	21.00	20.50	21.50	20.50
A_9	21.50	20.50	21.50	21.50
A_{10}	0.10	0.10	0.10	0.10
Best weight (lb)	5059.90	5073.51	5067.33	5070.41
No. of analyses	N/A	50,000	5,000	5,000

In Case 2, the HBA demonstrated competitive performance compared to established methods such as Ringertz [37], HHS [19], and HPSO [35]. The best weight obtained by HBA was 5070.41 lb, positioning it among the top designs between those produced by HHS (5067.33 lb), HPSO (5073.51 lb), and Ringertz (5059.90 lb). Remarkably, HBA required only 5,000 structural analyses to achieve this result, whereas HHS and HPSO required approximately 5,000 and 50,000 evaluations, respectively.

Figures 3 and 4 illustrate the convergence behavior of the HBA across both case studies, demonstrating rapid progression toward optimal solutions and efficient constraint satisfaction.

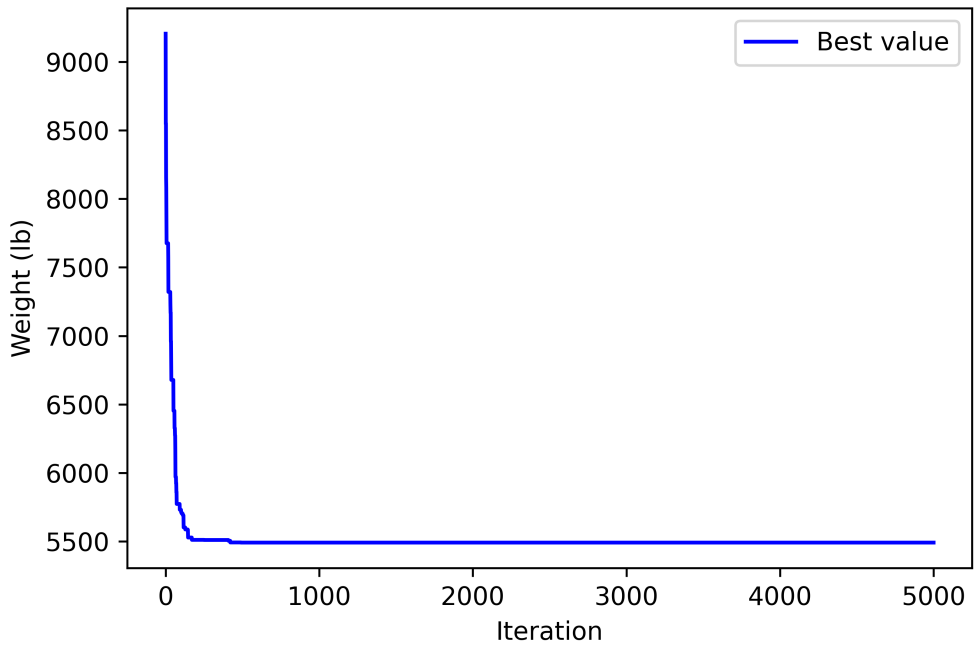

Figure 3. Evolution of convergence for the 10-bar truss structure (Case 1).

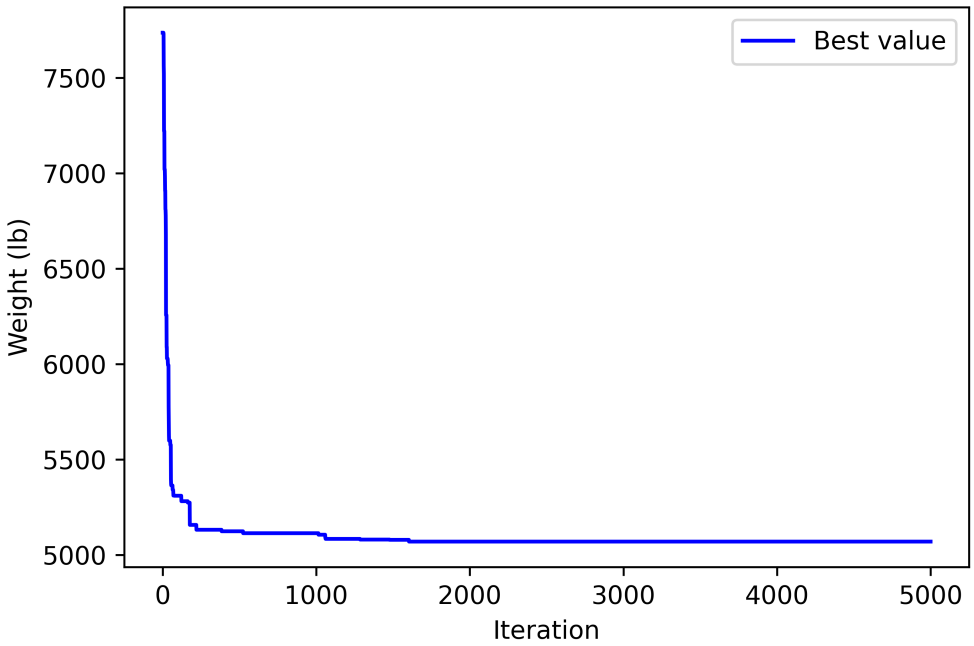

Figure 4. Evolution of convergence for the 10-bar truss structure (Case 2).

The 15-bar plane truss structure

This optimization case study considers a 15-bar planar truss structure subjected to a transverse load of $P = 10$ kip, as illustrated in Figure 5. The material properties are defined by a density of $\rho = 0.1$ lb/in^3 and a Young's modulus of $E = 10^4$ ksi.

The design problem includes 15 discrete variables corresponding to the cross-sectional areas of the truss elements. The cross-sectional areas are selected from the following discrete set in in^2: L = {0.111, 0.141, 0.174, 0.220, 0.270, 0.287, 0.347, 0.440, 0.539, 0.954, 1.081, 1.174, 1.333, 1.488, 1.764, 2.142, 2.697, 2.800, 3.131, 3.565, 3.813, 4.805, 5.952, 6.572, 7.192, 8.525, 9.300, 10.850, 13.330, 14.290, 17.170, 19.180 }.

The continuous design variables correspond to the coordinates x_2–x_6, x_3–x_7, and $y_2, y_3, y_4, y_6, y_7, y_8$, and are subject to the following domain constraints: $100 < x_2 < 140$, $220 < x_3 < 260$, $100 < y_2 < 140$, $100 < y_3 < 140$, $50 < y_4 < 90$, $-20 < y_6 < 20$, $-20 < y_7 < 20$, and $20 < y_8 < 60$.

These bounds ensure geometric feasibility while allowing flexibility in the optimization process. All truss members are constrained to a maximum allowable tensile stress limitation of ± 25 ksi. To ensure an effective exploration and exploitation balance during the optimization process, the population size is set to $n_{\text{agent}} = 20$. The HBA is executed for a maximum of 6000 iterations. The dynamic exploration factor is computed at each iteration according to the nonlinear decay formula α.

Figure 6 displays the convergence profile of the HBA when applied to the 15-bar truss optimization problem. The algorithm demonstrates a rapid convergence trend, with notable improvements observed between iterations 1900 and 2000, and stabilization occurring around the 6000 iterations.

Tables 3 present a comparative summary between the optimal design obtained using HBA and those reported by alternative optimization techniques in the literature. HBA

achieved an optimal structural weight of 74.993 lb, outperforming several existing methods under similar computational budgets. The structural weights reported by D-ICDE [38], GA [41], and GA [40], were 80.5688 lb, 79.82 lb, and 76.685 lb, respectively.

Additionally, the PSO variant [39] resulted in 82.2344 lb, while CPSO and SCPSO achieved 77.6153 lb and 72.5143 lb, respectively. These findings highlight the potential of HBA in handling discrete optimization problems with structural constraints. However, as a stochastic algorithm, its performance may vary across different problem instances.

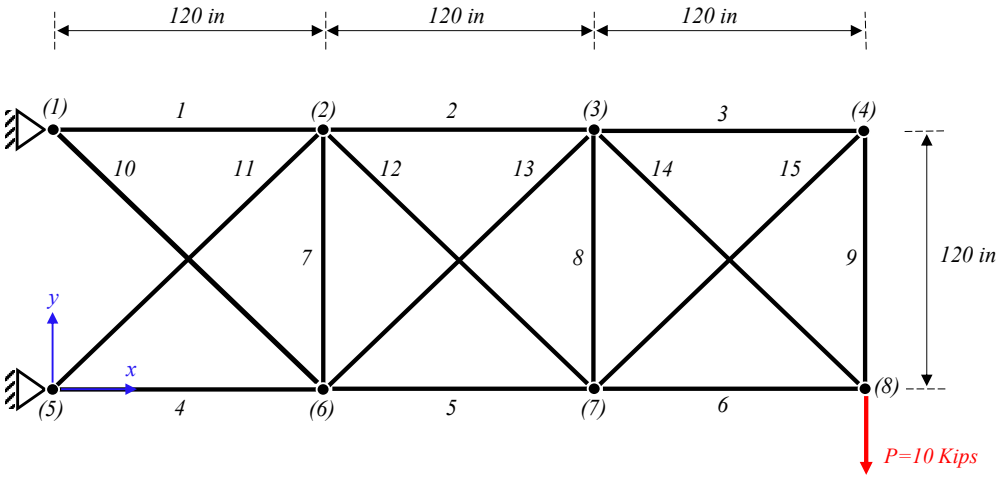

Figure 5. A 15-bar truss structure in a planar configuration.

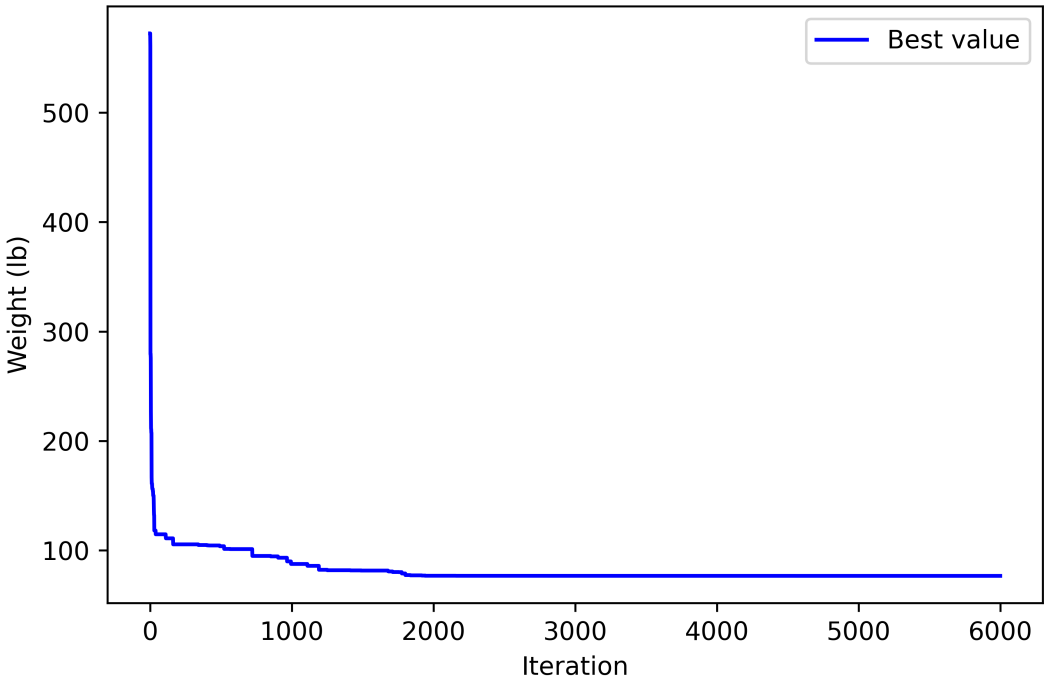

Figure 6. Evolution of convergence for the 15-bar truss structure

Table 3. Comparison of optimization results obtained using HBA with those from the literature for the 15-bar truss.

Design Variables	GA [39]	GA [40]	PSO [41]	CPSO [41]	SCPSO [41]	D-ICDE [38]	HBA
A_1	1.081	1.081	0.954	1.174	0.954	1.081	0.954
A_2	0.539	0.539	1.081	0.539	0.539	0.539	0.539
A_3	0.287	0.287	0.270	0.347	0.270	0.141	0.111
A_4	0.954	0.954	1.081	0.954	0.954	0.954	0.954
A_5	0.954	0.539	0.539	0.954	0.539	0.539	0.539
A_6	0.220	0.141	0.287	0.141	0.174	0.287	0.347
A_7	0.111	0.111	0.141	0.141	0.111	0.111	0.111
A_8	0.111	0.111	0.111	0.111	0.111	0.111	0.111
A_9	0.287	0.539	0.347	0.174	0.287	0.141	3.565
A_{10}	0.220	0.440	0.440	0.141	0.347	0.347	0.440
A_{11}	0.440	0.539	0.270	0.440	0.347	0.440	0.440
A_{12}	0.440	0.270	0.111	0.440	0.220	0.270	0.174
A_{13}	0.111	0.220	0.347	0.141	0.220	0.270	0.270
A_{14}	0.220	0.141	0.440	0.141	0.174	0.287	0.347
A_{15}	0.347	0.287	0.220	0.347	0.270	0.174	0.111
x_2	133.612	101.577	106.052	102.287	137.221	100.039	116.006
x_3	234.752	227.911	239.024	240.505	259.909	238.701	229.352
y_2	100.449	134.798	130.355	112.584	123.500	132.847	134.202
y_3	104.738	128.221	114.273	108.042	110.002	125.366	121.041
y_4	73.762	54.863	51.987	57.795	59.935	60.307	46.986
y_6	10.067	16.448	1.814	-6.430	-5.180	-10.665	-19.350
y_7	1.339	13.301	9.183	-1.801	4.219	-12.248	-11.086
y_8	50.402	54.857	46.909	57.799	57.883	59.993	46.964
Best weight (lb)	79.820	76.685	83.234	77.615	72.514	74.681	74.993
No. of analyses	8000	8000	4500	7980	7980	–	5000

The 25-bar spatial truss structure

Figure 7 presents the configuration of a spatial truss composed of 25 elements. This benchmark problem has been widely used in the literature to evaluate the performance of structural optimization algorithms. All truss members share common mechanical properties: a Young's modulus of 10,000 ksi and a material density of 0.1 lb/in^3. The loading conditions applied to the structure are detailed in Table 4.

Due to the structural symmetry, the 25 bars are grouped into eight design categories to reduce the number of independent design variables. These groups are defined as follows: Group 1 contains A1; Group 2 includes A2 to A5; Group 3 includes A6 to A9; Group 4 includes A10 and A11; Group 5 includes A12 and A13; Group 6 includes A14 to A17; Group 7 includes A18 to A21; and Group 8 includes A22 to A25.

This grouping simplifies the optimization problem while preserving the mechanical behavior and constraints of the spatial structure. All free nodes are constrained to displacement limits of ±35 in in all directions, and each bar group must satisfy an allowable axial stress constraint of ±40 ksi. The discrete set of admissible cross-sectional areas is L = { 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 .

Table 4. Loading conditions for the 25-bar truss problem.

Node	X (kips)	Y (kips)	Z (kips)
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

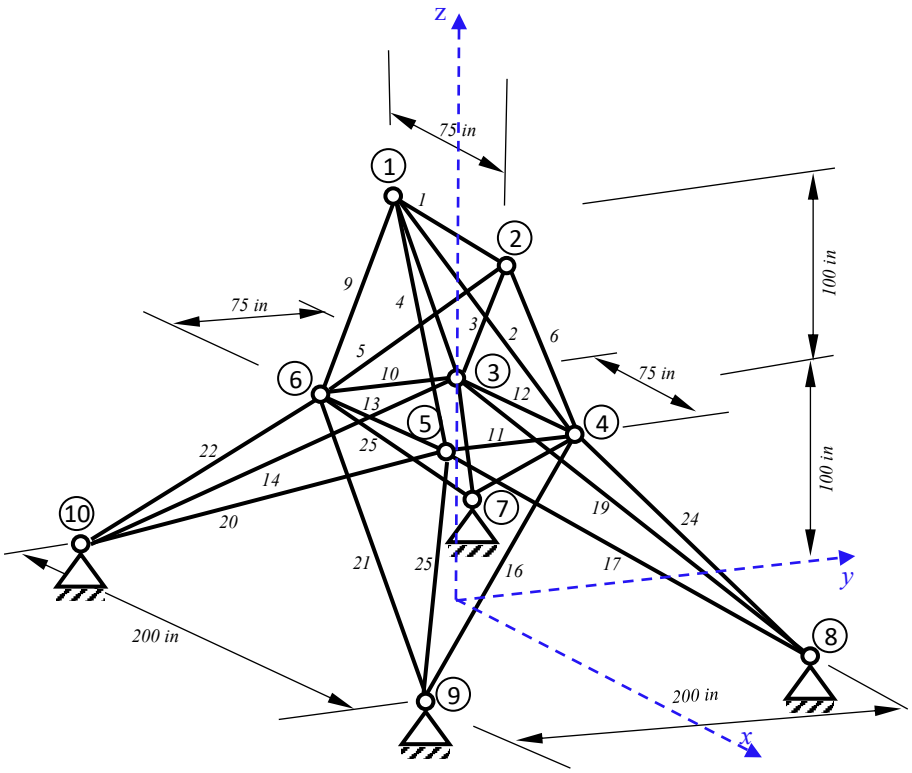

Figure 7. A 25-bar truss structure in a spatial configuration.

The optimization performance of various algorithms for the 25-bar spatial truss structure is outlined in Table 5. The HBA was benchmarked against seven established metaheuristic techniques, including Colliding Bodies Optimization (CBO) [42], Ant Colony Optimization (ACO) [20], Big Bang-Big crunch (BB-BC)[43], Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [44], Harmony Search-based HHS [19], and Hybrid PSO (HPSO) [35]. Among them, HBA produced the lightest optimal weight of 482.49 lb, surpassing all the alternatives, which converged to a weight of 484.85 lb. Notably, HBA achieved this result after just 1000 structural analyses, whereas competing methods required significantly more computational effort such as 25,000 evaluations for HPSO, 9090 for BB-BC, and 5000 for both HHS and CMA-ES.

Figure 8 shows the convergence profile of the HBA applied to the 3D truss optimization problem. The algorithm was executed with a population size of $N = 30$ and a maximum of 1000 iterations. The adaptive density factor α was computed using a nonlinear Gaussian decay function, while the movement of each agent was guided by a randomized directional flag $F \in [-1, 1]$ and a stochastic attraction term d, where $\beta \sim \mathcal{U}(0, 1)$. The search alternated between the digging and honey phases. Over the course of optimization, HBA effectively improved the structure, demonstrating robustness and efficiency for solving high-dimensional, constraint-based structural problems.

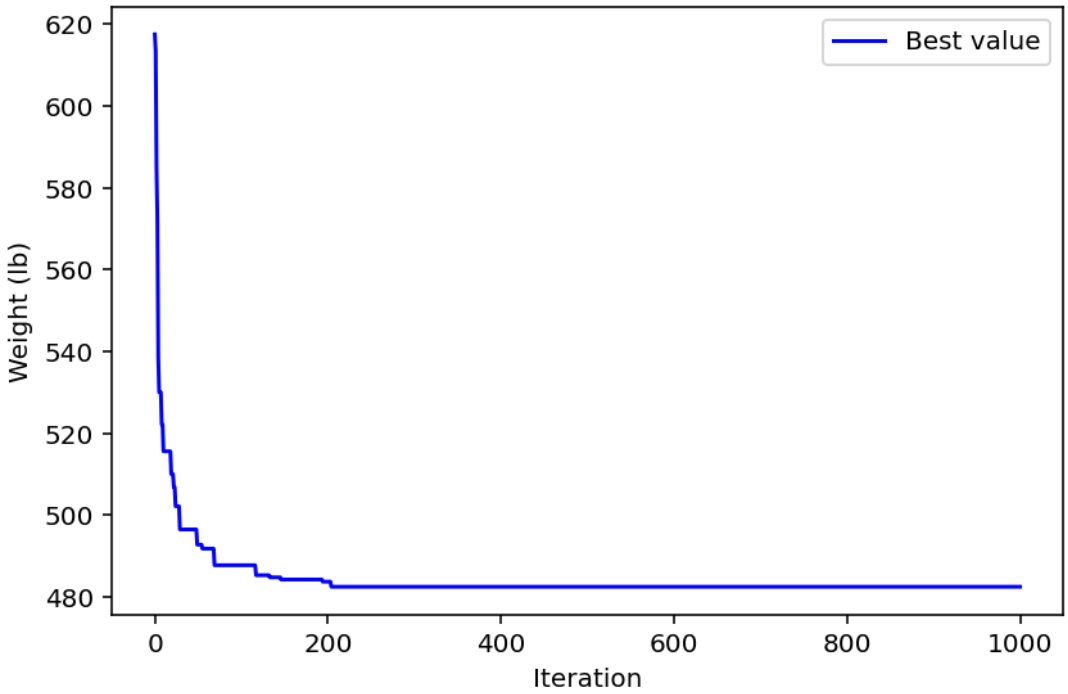

Figure 8. Evolution of convergence for the 25-bar truss structure

Table 5. Comparison of optimization results obtained using HBA with those from the literature for the 25-bar truss.

Group	CBO [42]	ACO [20]	ABC [36]	HPSO [35]	BB-BC [43]	HHS [19]	CMA-ES [44]	**HBA**
1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
2	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.7
3	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4
4	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
5	2.1	2.1	2.1	2.1	2.1	2.1	2.1	2.0
6	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9
7	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.3
8	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4
Best Weight (lb)	484.85	484.85	484.85	484.85	484.85	484.85	484.85	**482.49**
No. of analyses	20000	–	24250	25000	9090	5000	5000	**1000**

Conclusions

In this work, the Honey Badger Algorithm (HBA) was employed to address the structural optimization of both planar and spatial truss systems. The algorithm was implemented in Python, incorporating a penalty-based mechanism to effectively handle design constraints. This dynamic constraint handling approach ensures the feasibility of candidate solutions while maintaining optimization efficiency. Experimental evaluations on various truss configurations subjected to different loading conditions demonstrated the algorithm's ability to deliver accurate and robust solutions. A key strength of HBA lies in its bio-inspired search mechanism, which adaptively balances global exploration and local exploitation through its two main strategies: the digging and honey phases. Furthermore, HBA is a derivative-free algorithm, making it particularly suitable for structural problems where

gradient information is difficult or costly to obtain. Although this approach requires a considerable number of structural analyses, it remains advantageous for problems involving discrete variables or nonlinear constraints, where gradient-based methods are impractical or unreliable. While its convergence speed may vary depending on problem complexity, its robustness, simplicity, and constraint-handling effectiveness make HBA a promising tool for structural optimization tasks. It should be noted that the performance of HBA also depends on several algorithmic parameters, including population size, quality of the initial population, number of optimization runs, number of iterations per run, and penalty parameters. These choices can significantly influence convergence speed and solution quality, and should be carefully selected based on the specific problem considered.

References

[1] K. Deb. Optimization for engineering design: Algorithms and examples. PHI Learning Pvt. Ltd., 2012.

[2] X. S. Yang. A new metaheuristic bat-inspired algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, Springer, 2010. https://doi.org/10.1007/978-3-642-12538-6_6

[3] M. Jahanbakhsh and A. Ferrantelli. Computer vision framework for crack detection and estimation of air leakage through the straight-through cracks in buildings envelopes. *Rakenteiden Mekaniikka*, 58(2):71–85, 2025. https://doi.org/10.23998/rm.152481

[4] A. Sundararajan, N. Pandiyan, and S. Mohan. Topology optimization of 2D trusses using teaching learning-based optimization algorithm. *Engineering Optimization*, 53(4):602–617, 2021. https://doi.org/10.1080/0305215X.2020.1784665

[5] M. Heinisuo. Buckling analysis of members restrained by sandwich panels. *Rakenteiden Mekaniikka*, 54(2):95–116, 2021. https://doi.org/10.23998/rm.85711

[6] G. I. N. Rozvany. A critical review of established methods of structural topology optimization. *Structural and Multidisciplinary Optimization*, 37:217–237, 2009. https://doi.org/10.1007/s00158-007-0217-0

[7] O. El Mrimar, O. Bendaou, and B. Samoudi. The perturbation method for dynamic analysis of pole vaulting. *Proceedings of Interdisciplinarity in Engineering*, pages 641–650, 2021. https://doi.org/10.1007/978-3-030-93817-8_57

[8] O. El Mrimar and O. Bendaou. A perturbation method for the stochastic dynamic analysis of mechanical systems: Application to pole vaulting. *Proceedings of the 2nd International Conference on Intelligent Systems and Renewable Energy Technologies (IRASET)*, pages 1–5, 2022. https://doi.org/10.1109/IRASET52964.2022.9738395

[9] J. Kennedy and R. Eberhart. Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, 4:1942–1948, 1995. https://doi.org/10.1109/ICNN.1995.488968

[10] A. Kaveh and S. Talatahari. A particle swarm ant colony optimization for truss structures with discrete variables. *Journal of Constructional Steel Research*, 65:1558–1568, 2009. https://doi.org/10.1016/j.jcsr.2009.04.021

[11] O. El Mrimar, O. Bendaou, and B. Samoudi. Optimization of pole vault parameters using particle swarm optimization and genetic algorithm. *Series on Biomechanics*, 37(4):93–106, 2023. https://doi.org/10.7546/SB.12.04.2023

[12] P. C. Fourie and A. A. Groenwold. The particle swarm optimization algorithm in size and shape optimization. *Structural and Multidisciplinary Optimization*, 23:259–267, 2002. https://doi.org/10.1007/s00158-002-0188-0

[13] A. Kaveh and S. Talatahari. A hybrid particle swarm and ant colony optimization for design of truss structures. *Asian Journal of Civil Engineering (Building and Housing)*, 11(1):79–94, 2010.

[14] P. Ponterosso and D. S. J. Fox. Heuristically seeded genetic algorithms applied to truss optimisation. *Engineering with Computers*, 15:345–355, 1999. https://doi.org/10.1007/s003660050029

[15] T. Dede, S. Bekiroglu, and Y. Ayvaz. Weight minimization of trusses with genetic algorithm. *Applied Soft Computing*, 11:2565–2575, 2011. https://doi.org/10.1016/j.asoc.2010.10.006

[16] A. Javanmiri and J. Makinen. Weight optimization of truss structures by using genetic algorithms. *Rakenteiden Mekaniikka*, 55(2):42–54, 2022. https://doi.org/10.23998/rm.111471

[17] A. Nicholas, B. Kamran, and F. Zouheir. Applicability and viability of a GA based finite element analysis architecture for structural design optimization. *Computers and Structures*, 81:2259–2271, 2003. https://doi.org/10.1016/S0045-7949(03)00255-4

[18] S. O. Degertekin. Improved harmony search algorithms for sizing optimization of truss structures. *Computers and Structures*, 92–93:229–241, 2012. https://doi.org/10.1016/j.compstruc.2011.10.022

[19] M. Y. Cheng, D. Prayogo, Y. W. Wu, and M. M. Lukito. A hybrid Harmony Search algorithm for discrete sizing optimization of truss structure. *Automation in Construction*, 69:21–33, 2016. https://doi.org/10.1016/j.autcon.2016.05.023

[20] C. V. Camp and B. J. Bichon. Design of space trusses using ant colony optimization. *Journal of Structural Engineering*, 130(5):741–751, 2004. https://doi.org/10.1061/(ASCE)0733-9445(2004)130:5(741)

[21] R. Storn and K. Price. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997. https://doi.org/10.1023/A:1008202821328

[22] D. Karaboga. An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[23] D. T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi. The bee algorithm: A novel tool for complex optimisation problems. *Proceedings of the 2nd International Virtual Conference on Intelligent Production Machines and Systems (IPROMS 2006)*, pages 454–459, 2006. https://doi.org/10.1016/B978-008045157-2/50081-X

[24] X. S. Yang. Firefly algorithms for multimodal optimization. In *5th International Conference on Stochastic Algorithms: Foundations and Applications*, Springer-Verlag, pages 169–178, 2009. https://doi.org/10.1007/978-3-642-04944-6_14

[25] X. S. Yang and S. Deb. Cuckoo search via Levy flights. *World Congress on Nature and Biologically Inspired Computing (NaBIC 2009)*, pages 210–214, 2009. https://doi.org/10.1109/NABIC.2009.5393690

[26] M. Y. Cheng and D. Prayogo. Symbiotic organisms search: A new metaheuristic optimization algorithm. *Computers and Structures*, 139:98–112, 2014. https://doi.org/10.1016/j.compstruc.2014.03.007

[27] T. Jokinen, K. Mela, T. Tiainen, and M. Heinisuo. Optimization of tubular trusses using intumescent coating in fire. *Rakenteiden Mekaniikka*, 49(4):160–175, 2016.

[28] T. Jokinen, K. Mela, T. Tiainen, and M. Heinisuo. Optimization of tubular trusses using intumescent coating in fire. *Rakenteiden Mekaniikka*, 49(4):160–175, 2016.

[29] F. A. Hashim et al. Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems. *Mathematics and Computers in Simulation*, 192:84–110, 2022. https://doi.org/10.1016/j.matcom.2021.08.013

[30] S. W. Zhang, J. S. Wang, Y. X. Li, et al. Improved honey badger algorithm based on elementary function density factors and mathematical spirals in polar coordinate system. *Artificial Intelligence Review*, 57:55, 2024. https://doi.org/10.1007/s10462-023-10658-2

[31] T. Han, T. Li, Q. Liu, Y. Huang, and H. Song. A multi-strategy improved honey badger algorithm for engineering design problems. *Algorithms*, 17(12):573, 2024. https://doi.org/10.3390/a17120573

[32] Honey Badger Team. Honey Badger Algorithm Official Website. http://www.honeybadger.com/index.html

[33] A. Shabani, A. Behrouz, and S. Miguel. Search and rescue optimization algorithm for size optimization of truss structures with discrete variables. *Numerical Methods in Civil Engineering*, 3(3):28–40, 2019. https://doi.org/10.29252/nmce.3.3.28

[34] S. Rajeev and C. Krishnamoorthy. Discrete optimization of structures using genetic algorithms. *Journal of Structural Engineering*, 118(5):1233–1250, 1992. https://doi.org/10.1061/(ASCE)0733-9445(1992)118:5(1233)

[35] L. J. Li, Z. B. Huang, and F. Liu. A heuristic particle swarm optimization method for truss structures with discrete variables. *Computers and Structures*, 87(7–8):435–443, 2009. https://doi.org/10.1016/j.compstruc.2009.01.004

[36] M. Sonmez. Discrete optimum design of truss structures using artificial bee colony algorithm. *Structural and Multidisciplinary Optimization*, 43:85–97, 2011. https://doi.org/10.1007/s00158-010-0551-5

[37] U. L. F. T. Ringertz. On methods for discrete structural optimization. *Engineering Optimization*, 13(1):47–64, 1988.

[38] 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 Systems with Applications*, 42(20):7057–7069, 2015. https://doi.org/10.1016/j.eswa.2015.04.072

[39] W. Tang, L. Tong, and Y. Gu. Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables. *International Journal for Numerical Methods in Engineering*, 62:1737–1762, 2005. https://doi.org/10.1002/nme.1244

[40] H. Rahami, A. Kaveh, and Y. Gholipour. Sizing, geometry and topology optimization of trusses via force method and genetic algorithm. *Engineering Structures*, 30:2360–2369, 2008. https://doi.org/10.1016/j.engstruct.2008.01.012

[41] S. Gholizadeh. Layout optimization of truss structures by hybridizing cellular automata and particle swarm optimization. *Computers and Structures*, 125:86–99, 2013. https://doi.org/10.1016/j.compstruc.2013.04.024

[42] A. Kaveh and M. I. Ghazaan. A comparative study of CBO and ECBO for optimal design of skeletal structures. *Computers and Structures*, 153:137–147, 2015. https://doi.org/10.1016/j.compstruc.2015.02.028

[43] C. V. Camp. Design of space trusses using big bang–big crunch optimization. *Journal of Structural Engineering*, 133(7):999–1008, 2007. https://doi.org/10.1061/(ASCE)0733-9445(2007)133:7(999)

[44] O. El Mrimar, Z. El Haddad, O. Bendaou, and B. Samoudi. Covariance matrix adaptation evolution strategy for optimizing truss structures with discrete variables. *Mechanics of Solids*, 60(2):1194–1207, 2025. https://doi.org/10.1134/S0025654424607055

Ouadie El Mrimar, Zakaria El Haddad, Bousselham Samoudi, Othmane Bendaou
Optics, Materials and Systems Team, Faculty of Sciences, Abdelmalek Essaadi University
B.P.2121, M'Hannech II, 93002 Tetouan, Morocco
elmrimar.ouadie-etu@uae.ac.ma, zakaria-elhaddad@hotmail.fr, b.samoudi@uae.ac.ma, o.bendaou@uae.ac.ma