VILNIUS TECH
Vilnius Gediminas Technical University

Journal of Civil Engineering and Management
ISSN 1392-3730 / eISSN 1822-3605
2023 Volume 29 Issue 8: 757–786
https://doi.org/10.3846/jcem.2023.20399

INTEGRATING ENHANCED OPTIMIZATION WITH FINITE ELEMENT ANALYSIS FOR DESIGNING STEEL STRUCTURE WEIGHT UNDER MULTIPLE CONSTRAINTS

Dinh-Nhat TRUONG[1,2], Jui-Sheng CHOU[1*]

[1]*Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan*
[2]*Department of Civil Engineering, University of Architecture Ho Chi Minh City, Ho Chi Minh City, Viet Nam*

Received 25 August 2023; accepted 30 October 2023

Abstract. Real-world optimization problems are ubiquitous across scientific domains, and many engineering challenges can be reimagined as optimization problems with relative ease. Consequently, researchers have focused on developing optimizers to tackle these challenges. The Snake Optimizer (SO) is an effective tool for solving complex optimization problems, drawing inspiration from snake patterns. However, the original SO requires the specification of six specific parameters to operate efficiently. In response to this, enhanced snake optimizers, namely ESO[1] and ESO[2], were developed in this study. In contrast to the original SO, ESO[1] and ESO[2] rely on a single set of parameters determined through sensitivity analysis when solving mathematical functions. This streamlined approach simplifies the application of ESOs for users dealing with optimization problems. ESO[1] employs a logistic map to initialize populations, while ESO[2] further refines ESO[1] by integrating a Lévy flight to simulate snake movements during food searches. These enhanced optimizers were compared against the standard SO and 12 other established optimization methods to assess their performance. ESO[1] significantly outperforms other algorithms in 15, 16, 13, 15, 21, 16, 24, 16, 19, 18, 13, 15, and 22 out of 24 mathematical functions. Similarly, ESO[2] outperforms them in 16, 17, 18, 22, 23, 23, 24, 20, 19, 20, 17, 22, and 23 functions. Moreover, ESO[1] and ESO[2] were applied to solve complex structural optimization problems, where they outperformed existing methods. Notably, ESO[2] generated solutions that were, on average, 1.16%, 0.70%, 2.34%, 3.68%, and 6.71% lighter than those produced by SO, and 0.79%, 0.54%, 1.28%, 1.70%, and 1.60% lighter than those of ESO[1] for respective problems. This study pioneers the mathematical evaluation of ESOs and their integration with the finite element method for structural weight design optimization, establishing ESO[2] as an effective tool for solving engineering problems.

Keywords: steel structural design, finite element analysis, metaheuristic algorithm, enhanced optimizer, benchmark functions.

Introduction

Real-world optimization problems pervade almost every scientific domain, and many engineering challenges can readily be transformed into optimization problems (Podolski & Sroka, 2019; Ji et al., 2020; Es-Haghi et al., 2022). These problems pose significant challenges, often involving nonlinearity, multiple objectives, discontinuities, high dimensions, uncertainties, and non-convex regions (Cheng & Cao, 2016). Metaheuristic optimization algorithms, constituting a branch of approximate optimization techniques, have become one of computer science's most active research areas in recent years (Xue & Chen, 2019; Chou & Truong, 2021). These techniques excel at finding optimal or near-optimal solutions for complex optimization problems, including nondeterministic polynomial time (NP) problems, all within a reasonable computational timeframe (Gao et al., 2022).

Metaheuristics have found diverse applications in engineering due to their ease of design and implementation, their independence from gradient information during the optimization process, and their applicability to a wide array of optimization problems (Yang, 2010b). The crux of metaheuristics lies in two fundamental factors: exploration and exploitation. Exploration signifies the search algorithm's capacity to unearth novel solutions in remote

*Corresponding author. E-mail: *jschou@mail.ntust.edu.tw*

corners of the search space. In contrast, exploitation involves the algorithm's prowess in discovering fresh solutions within regions that have demonstrated promise.

The Snake Optimizer, developed by Hashim and Hussien in 2022 (Hashim & Hussien, 2022), is rooted in the behavioral patterns of snakes, illustrated in Figure 1. Snakes engage in food hunting and mating during late spring and early summer when temperatures are low. However, mating is contingent upon factors beyond just temperature and food availability. In situations where the temperature is low and food is abundant, rival male snakes engage in combat to capture the attention of a female. The female, in turn, selects her mate. Upon successful mating, the female lays eggs in a nest or burrow before leaving promptly.

Snake optimization, a potent tool for solving complex optimization problems, grapples with the challenge of setting its algorithmic parameters, C_1, C_2, C_3, $Threshold_1$, $Threshold_2$ and $Threshold_3$ (Hashim & Hussien, 2022). This intricate parameter configuration impedes the practical application of SO. To surmount these limitations inherent in the basic SO algorithm, this study introduces two innovative variants: the Enhanced Snake Optimizers (ESO^1 and ESO^2), requiring only one parameter to be set. To accomplish this goal, ESO^1 employs a logistic map to initialize populations, while ESO^2 enhances ESO^1 further by integrating a Lévy flight, simulating snake movements during food searches. This parameter value was determined through sensitivity analysis conducted during the solution of mathematical functions.

To showcase the efficiency and prowess of the Enhanced Snake Optimizers (ESOs), we evaluate their performance on 24 mathematical functions (CEC-2022 functions) and five structural weight design problems taken from the literature. These design problems include the 47-bar power transmission tower, 72-bar tower, 672-bar double-layer grid, 1520-bar double-layer grid, and 1536-bar double-layer barrel vault problems. The outcomes are meticulously compared with results obtained using various other renowned optimizers. The findings demonstrate that the proposed ESOs outperform the standard Snake Optimizer (SO) and different well-known metaheuristic algorithms when solving mathematical benchmark functions and structure design problems.

The subsequent sections of this paper are structured as follows: Section 1 reviews the classification of metaheuristic algorithms, the original snake optimizer, refined techniques for metaheuristic optimization, and their applications in solving structural design problems. Section 2 introduces the Enhanced Snake Optimizer algorithm. Section 3 presents the results of numerical experiments using ESOs to solve mathematical functions. Section 4 explores the application of ESOs in addressing five specific structural engineering problems. Section 5 delves into the discussion of results from both the mathematical tests and structural applications. Final section draws conclusions based on the findings presented in this study.

1. Literature review

1.1. Classification of metaheuristic optimizer

Metaheuristic optimizers can be categorized into four classes, as shown in Figure 2: evolutionary-based algorithms, swarm intelligence-based algorithms, physical and chemical-based algorithms, and human-based algorithms (Zhao et al., 2019; Askari et al., 2020).

Evolutionary-based algorithms (EAs) are stochastic population-based algorithms inspired by nature and utilizing genetic principles, including selection, crossover, mutation, and elimination. Examples of EAs include the genetic algorithm (GA) (Holland, 1992), evolutionary strategies (ES) (Beyer & Schwefel, 2002), differential evolution (DE) (Rocca et al., 2011), and biogeography-based optimization (BBO) (Simon, 2008).

Swarm intelligence (SI) algorithms are characterized by decentralization, shape-formation, and self-organization, inspired by the natural behavior of colonies of birds, insects, fishes, horses, and other animals. The most renowned examples of SI algorithms include the particle swarm algorithm (PSO) (Kennedy & Eberhart, 1995), ant colony optimization (ACO) (Dorigo et al., 2006), artificial bee colony (ABC) (Karaboga & Basturk, 2007), cuckoo search (CS) (Gandomi et al., 2013), bat algorithm (BA) (Yang & Hossein, 2012), firefly algorithm (FA) (Yang, 2010a), and jellyfish search optimizer (JS) (Chou & Truong, 2021).

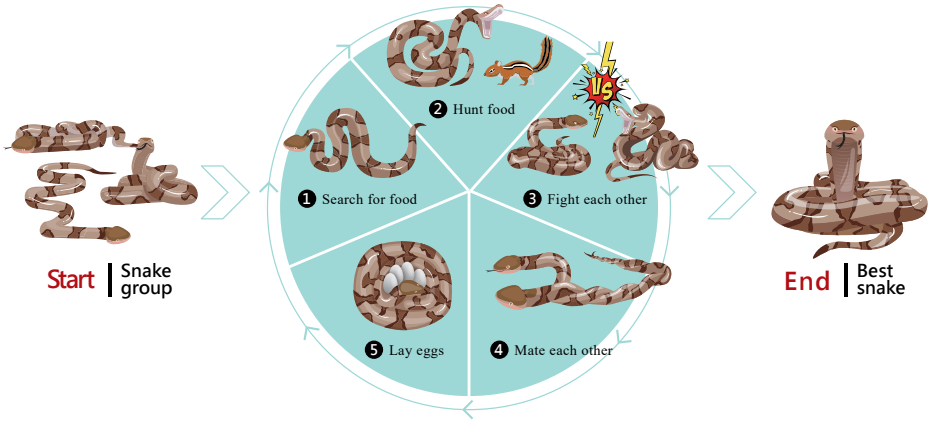

Figure 1. Life-cycle of a snake

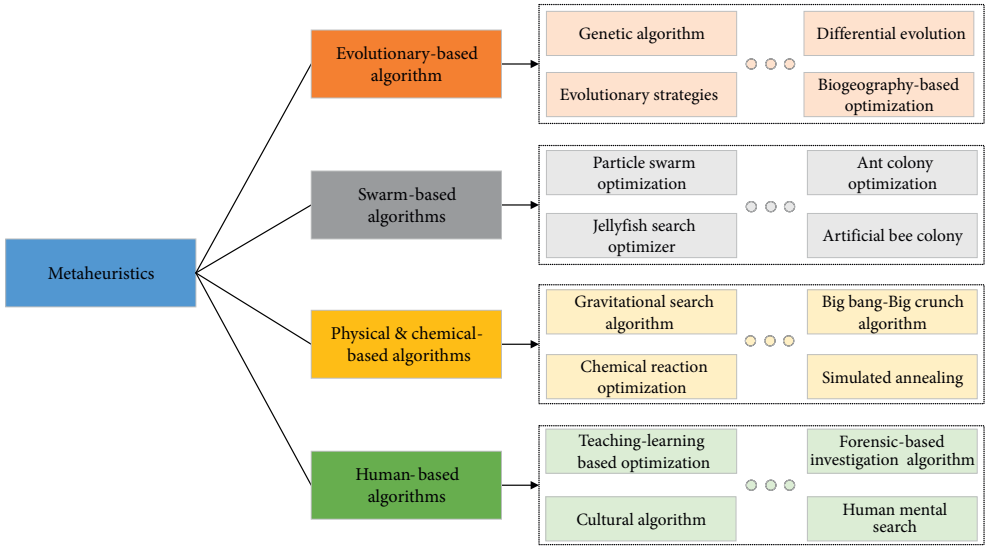

Figure 2. Classification of metaheuristic optimizers

Physical and chemical-based algorithms simulate natural physical phenomena or chemical laws. This class encompasses several algorithms, including simulated annealing (SA) (Kirkpatrick et al., 1983), the gravitational search algorithm (GSA) (Rashedi et al., 2009), chemical reaction optimization (CRO) (Lam & Li, 2010), big bang–big crunch (BBBC) (Erol & Eksin, 2006), charged system search (CSS) and magnetically charged system search (MCSS) (Kaveh & Talatahari, 2010), ray optimization (RO) (Kaveh & Khayatazad, 2012), atom search optimization (ASO) (Zhao et al., 2019), the vortex search algorithm (VSA) (Doğan & Ölmez, 2015), water evaporation optimization (WEO) (Kaveh & Bakhshpoori, 2016), the lighting search algorithm (LSA) (Shareef et al., 2015), and others.

The last category of metaheuristic algorithms under consideration is human-based algorithms, which draw inspiration from human activities, both physical and non-physical, encompassing cognitive processes and social behaviors. This class includes a variety of algorithms such as the teaching–learning based optimization (TLBO) algorithm (Rao et al., 2011), the ideology algorithm (IA) (Huan et al., 2017), the socio-evolution and learning optimization (SELO) algorithm (Kumar et al., 2018), the cognitive behavior optimization algorithm (COA) (Li et al., 2016), human mental search (HMS) (Mousavirad & Ebrahimpour-Komleh, 2017), the cultural algorithm (CA) (Omran, 2016), the forensic-based investigation (FBI) algorithm (Chou & Nguyen, 2020), poor and rich optimization (PRO) (Samareh Moosavi & Bardsiri, 2019), the student psychology-based optimization algorithm (SPBO) (Das et al., 2020), search and rescue optimization (SAR) (Shabani et al., 2020), and the arithmetic optimization algorithm (AOA) (Abualigah et al., 2021).

1.2. Snake optimization algorithm

Snake Optimization (SO) stands for the "Snake Optimizer", inspired by the mating behavior of snakes. Snakes mate when the temperature is low and food is available; otherwise, they focus solely on searching for food or consuming it (Hashim & Hussien, 2022). The search process in SO comprises two phases: exploration and exploitation. Exploration takes place only under specific environmental conditions, such as the right temperature and the presence of food. If these conditions are unmet, the snake focuses solely on searching for food. Exploitation involves several transitional phases aimed at reaching the optimal solution more efficiently.

If food is available but the temperature is high, snakes focus solely on consuming the available food. When food is available and the temperature is low, mating takes place. The mating process involves both fighting and mating. During the fighting stage, each male competes to attract the best female, while each female attempts to select the best male. In the mating stage, pairs mate based on the availability of food quantity. If successful mating occurs, the female may lay eggs that will eventually hatch. The subsequent subsections concisely introduce the algorithmic formulation presented by Hashim and Hussien (2022).

1.2.1. Initialization

SO starts by generating a random population with a uniform distribution to execute the optimization algorithm:

$$x_i = L_b + \text{rand}(0,1) \times (U_b - L_b), \quad (1)$$

where x_i is the position of the i^{th} individual; rand (0,1) is a random number between 0 and 1, and L_b and U_b are the lower and upper bounds on the problem.

1.2.2. Diving the swarm into two equal groups: males and females

SO assumes that 50% of the population is male and 50% is female:

$$N_{(m)} \approx \frac{N}{2}; \quad (2)$$

$$N_{(f)} \approx N - N_{(m)}, \tag{3}$$

where N is the number of individuals; $N_{(m)}$ denotes the numbers of males and $N_{(f)}$ represents the number of females.

1.2.3. Evaluate each group and define temperature and food quantity

Find the best individual in each group and identify the best male ($f_{(m)best}$), the best female ($f_{(f)best}$) and the position of the food (f_{food}). Temperature (Temp) is defined using the following equation:

$$Temp = \exp\left(\frac{-t}{T}\right), \tag{4}$$

where t denotes the current iteration and T is the maximum number of iterations.

Food quantity (Q) is defined by the following equation:

$$Q = C_1 \times \exp\left(\frac{t-T}{T}\right), \tag{5}$$

where C_1 is a constant and equals 0.5.

1.2.4. Exploration phase (no food)

Step 1: If Q < $Threshold_1$ ($Threshold_1$ = 0.25), then the snakes search for food by updating their position to a randomly chosen one. The exploration phase is modeled as follows:

$$\mathbf{x}_{(m)i}(t+1) = \mathbf{x}_{(m)rand}(t) \pm C_2 \times A_{(m)} \times \left((U_b - L_b) \times rand(0,1) + L_b\right), \tag{6}$$

where $\mathbf{x}_{(m)i}$ is the position of the i^{th} male; $\mathbf{x}_{(m)rand}$ is the position of a random male; rand (0,1) is a random number between 0 and 1, and $A_{(m)}$ is the ability of a male to find food, which is calculated as follows:

$$A_{(m)} = \exp\left(\frac{-f\left(\mathbf{x}_{(m)rand}\right)}{f\left(\mathbf{x}_{(m)i}\right)}\right), \tag{7}$$

where $f(\mathbf{x}_{(m)rand})$ is the fitness at $\mathbf{x}_{(m)rand}$; $f(\mathbf{x}_{(m)i})$ is the fitness of the i^{th} male and C_2 is a constant and equals 0.05.

$$\mathbf{x}_{(f)i}(t+1) = \mathbf{x}_{(f)rand}X(t) \pm C_2 \times A_{(f)} \times \left((U_b - L_b) \times rand(0,1) + L_b\right), \tag{8}$$

where $\mathbf{x}_{(f)i}$ is the position of the i^{th} female, $\mathbf{x}_{(f)rand}$ represents the position of a random female; rand (0,1) is a random number between 0 and 1, and $A_{(f)}$ is the ability of a female to find food, which is calculated as follows:

$$A_{(f)} = \exp\left(\frac{-f\left(\mathbf{x}_{(f)rand}\right)}{f\left(\mathbf{x}_{(f)i}\right)}\right), \tag{9}$$

where $f(\mathbf{x}_{(f)rand})$ is the fitness at $\mathbf{x}_{(f)rand}$ and $f(\mathbf{x}_{(f)i})$ is the fitness of the i^{th} female in the group.

1.2.5. Exploitation phase (food exists)

Step 2: If Q > $Threshold_1$ ($Threshold_1$ = 0.25) and if temperature Temp > $Threshold_2$ ($Threshold_2$ = 0.6), then the snakes are hot and will move toward food:

$$x_{i,j}(t+1) = x_{food,j}(t) \pm C_3 \times Temp \times rand(0,1) \times \left(x_{food,j}(t) - x_{i,j}(t)\right), \tag{10}$$

where $x_{i,j}$ is the position of the individual (male or female); $\mathbf{x}_{food}$ is the position of the best individuals, and C_3 is a constant and equals 2.0. If temperature Temp < $Threshold_2$ ($Threshold_2$ = 0.6), then the snakes are cold and will be in either fight mode or mating mode.

Step 3: If rand (0,1) > $Threshold_3$ ($Threshold_3$ = 0.6), then the snakes will be in fight mode:

$$\mathbf{x}_{(m)i}(t+1) = \mathbf{x}_{(m)i}(t) + C_3 \times FM \times rand(0,1) \times \left(Q \times \mathbf{x}_{(f)best}(t) - \mathbf{x}_{(m)i}(t)\right), \tag{11}$$

where $\mathbf{x}_{(m)i}$ is the position of the i^{th} male; $\mathbf{x}_{(f)best}$ is the position of the best female in the group, and FM is the ability of a male to fight:

$$\mathbf{x}_{(f)i}(t+1) = \mathbf{x}_{(f)i}(t) + C_3 \times FM \times rand(0,1) \times \left(Q \times \mathbf{x}_{(m)best}(t) - \mathbf{x}_{(f)i}(t)\right), \tag{12}$$

where $\mathbf{x}_{(f)i}$ is the position of the i^{th} female, $\mathbf{x}_{(m)best}$ represents the position of the best male in the group, and FF is the ability of the female to fight. FM and FF can be calculated as follows:

$$FM = \exp\left(\frac{-f\left(\mathbf{x}_{(f)best}\right)}{f\left(\mathbf{x}_{(m)i}\right)}\right); \tag{13}$$

$$FF = \exp\left(\frac{-f\left(\mathbf{x}_{(m)best}\right)}{f\left(\mathbf{x}_{(f)i}\right)}\right), \tag{14}$$

where $f(\mathbf{x}_{(f)best})$ is the fitness of the best female; $f(\mathbf{x}_{(m)best})$ is the fitness of the best male, and $f(\mathbf{x}_{(f)i})$ or $f(\mathbf{x}_{(m)i})$ is the fitness of an individual female or male, respectively.

Step 4: If rand (0,1) < $Threshold_3$ ($Threshold_3$ = 0.6), then the snakes are in mating mode:

$$\mathbf{x}_{(m)i}(t+1) = \mathbf{x}_{(m)i}(t) + C_3 \times M_{(m)} \times rand(0,1) \times \left(Q \times \mathbf{x}_{(f)i}(t) - \mathbf{x}_{(m)i}(t)\right); \tag{15}$$

$$\mathbf{x}_{(f)i}(t+1) = \mathbf{x}_{(f)i}(t) + C_3 \times M_{(f)} \times rand(0,1) \times \left(Q \times \mathbf{x}_{(m)i}(t) - \mathbf{x}_{(f)i}(t)\right), \tag{16}$$

where $\mathbf{x}_{(f)i}$ is the position of i^{th} female in the group; $\mathbf{x}_{(m)i}$ is the position of the i^{th} male in the group and $M_{(m)}$ & $M_{(f)}$ are the mating abilities of a male and a female, respectively, which calculated as follows:

$$M_{(m)} = \exp\left(\frac{-f\left(\mathbf{x}_{(f)i}\right)}{f\left(\mathbf{x}_{(m)i}\right)}\right); \quad (17)$$

$$M_{(f)} = \exp\left(\frac{-f\left(x_{(m)i}\right)}{f\left(x_{(f)i}\right)}\right). \quad (18)$$

Step 5: If an egg hatches, the worst male and female are replaced:

$$\mathbf{x}_{(m)worst} = L_b + rand(0,1)\times\left(U_b - L_b\right); \quad (19)$$

$$\mathbf{x}_{(f)worst} = L_b + rand(0,1)\times\left(U_b - L_b\right), \quad (20)$$

where $\mathbf{x}_{(m)worst}$ is the worst male and $\mathbf{x}_{(f)worst}$ is the worst female.

1.2.6. Checking terminating conditions

The process continues iteratively; if the stop criterion is satisfied, the process terminates.

1.3. Enhanced techniques for metaheuristic optimization

In recent decades, numerous metaheuristic optimizers have been developed to tackle optimization problems. However, these optimizers often require more coverage, impose constraints on the number of dimensions they can handle, and face challenges in balancing exploration and exploitation. As a result, many scholars have concentrated on enhancing these optimizers using various techniques to overcome these limitations. Below is a review of how prior literature has employed various efficient operational mechanisms to refine existing metaheuristic algorithms.

Arafa et al. (2014) enhanced the Differential Evolution (DE) algorithm by incorporating separate tuning processes for population size and a mutation-scaling factor; the crossover rate is implicitly handled during the crossover stage of the DE. A comparison between the original DE and the Success-History-based Adaptive DE (SHADE), a state-of-the-art DE algorithm, illustrates the superiority of the proposed approach for most of the considered functions.

Joshi and Arora (2017) introduced an Enhanced Grey Wolf Optimization (EGWO) algorithm, which incorporates an improved hunting mechanism compared to the original Grey Wolf Optimization (GWO). This enhancement effectively balances exploration and exploitation, optimizing the algorithm's performance and generating promising candidate solutions. Simulation results show that the proposed algorithm outperforms several well-known algorithms in mathematical tests. When applied to real-world problems with unknown search spaces, the results highlight the effectiveness of the proposed algorithm.

Chou and Ngo (2017) developed the Modified Firefly Algorithm (MFA) for multidimensional structural design optimization. In MFA, logistic chaotic maps generate a diverse initial population, and Gauss/mouse maps fine-tune the Firefly Algorithm's attractiveness parameter, adaptive inertia weight manages local exploitation and global exploration during the search process, and Lévy flight is utilized for exploitation. Experimental results demonstrated that the proposed MFA is more efficient and effective than the compared algorithms in solving mathematical functions and addressing three multidimensional structural design optimization problems.

Kamoona et al. (2018) introduced the Enhanced Cuckoo Search (ECS) algorithm, which integrates Gaussian diffusion random walks and greedy selection into its methodology. In contrast to employing Lévy flights random walks, ECS utilizes Gaussian diffusion random walks to enhance the local search process. Additionally, incorporating greedy selection ensures that ECS can find the optimal solution. Through mathematical tests, ECS has demonstrated outstanding performance, achieving optimal values with a higher convergence speed than the Cuckoo Search (CS) and recent Adaptive Cuckoo Search (ACS) algorithms.

Ramli et al. (2019) improved the Bat Algorithm by adjusting its number of dimensions and introducing an inertia weight. Mathematical tests showed that these modifications, involving changes in dimensionality and the addition of an inertia weight factor, significantly enhanced the effectiveness of the Bat Algorithm compared to its original version. These enhancements improved the results' quality and the algorithm's convergence speed.

Li and Han (2020) enhanced the Fruit Fly Optimization Algorithm for truss structure optimization by integrating the immune algorithm's self-non-self-antigen recognition mechanism and the immune system's learn-memory-forgetting knowledge processing mechanism. Optimization test results and comparisons with other algorithms demonstrated that the improved Fruit Fly Optimization Algorithm exhibited increased stability and exceptional efficiency in solving structural optimization problems.

Sabbah (2020) introduced an Enhanced Genetic Algorithm (EGA) inspired by Genetic Engineering and based on the Genetic Algorithm (GA). In EGA, the process of chromosome generation is modified based on the inter-correlation between genes, treating highly correlated genes differently. The proposed Enhanced Genetic Algorithm (EGA) was employed to optimize classifications using the Support Vector Machine (SVM) on the popular "Spambase" dataset.

Kaveh et al. (2021) developed the Enhanced Forensic-Based Investigation (EFBI) by modifying the original formulation of the FBI algorithm. EFBI enhances communication between investigation and pursuit teams to balance intensification and diversification tasks better. EFBI was applied to solve three structural optimization problems. The results confirmed that EFBI significantly outperforms the original FBI and exhibits performance superior to or comparable with other considered metaheuristics.

Makiabadi and Maheri (2021) introduced the Enhanced Symbiotic Organism Search (ESOS) algorithm to enhance the exploitative capabilities of the SOS algorithm. The ESOS algorithm was implemented with two constraint handling methods – the Mapping Strategy (MS) and the Penalty Function (PF) – resulting in two ESOS variants: ESOS-MS and ESOS-PF. These variants were employed to optimize the weights of four benchmark truss structures. The results demonstrated that the proposed ESOS variants outperformed other algorithms and provided faster solutions.

Yıldızdan and Baykan (2021) made modifications to the Artificial Jellyfish Search Algorithm. In their adapted version, two additional search strategies were incorporated, enhancing the active motion component of the Artificial Jellyfish Search Algorithm. This modification prolonged population diversity, resulting in superior performance in mathematical tests compared to the standard algorithm. The proposed algorithm demonstrated competitiveness with other algorithms in the existing literature.

Hu et al. (2022) developed an Enhanced Hybrid Arithmetic Optimization Algorithm (AOA) known as CSOAOA, which integrated the Point Set Strategy, Optimal Neighborhood Learning Strategy, and Crisscross Strategy to overcome deficiencies in the original AOA. These enhancements addressed challenges such as insufficient exploitation, susceptibility to local optima, and low convergence accuracy in large-scale applications. Experimental results and statistical comparisons demonstrated that CSOAOA outperformed other algorithms in terms of precision, convergence rate, and solution quality based on mathematical testing.

1.4. Use of metaheuristic in solving structural problem

Structural optimization problems involve highly non-convex and nonlinear design spaces with hundreds of design variables and constraints (Ficarella et al., 2021). Gradient-based optimizers may need improvement when applied to solve these problems, as they often get stuck in local optima. Moreover, evaluating gradients and defining search directions become computationally expensive when the optimization problem includes numerous design variables.

The random generation of trial designs enables the exploration of more significant fractions of the search space than local approximations in gradient-based optimization allow (Ficarella et al., 2021). However, a random search often leads to many analyses producing only marginal design improvements or even generating unfeasible trial designs. To streamline the search process, metaheuristic optimization methods have been developed, and these methods have found successful applications in various fields of science and engineering.

Comprehensive reviews of the applications of metaheuristic optimization algorithms to structural design problems and critical assessments of the relative advantages of various methods can be found elsewhere (Kaveh & Ilchi Ghazaan, 2018a, 2018c; Kaveh et al., 2021). Additionally, many studies have compared the performances of metaheuristic algorithms in solving specific optimization problems, including those involving large-scale and real-size truss structures and steel frames (Hasançebi et al., 2009, 2010), reinforced concrete frames (Aydogdu et al., 2017), cellular beams (Erdal et al., 2011), layout optimization, and optimization under frequency constraints (Truong & Chou, 2023), as well as tensegrity structures (Do et al., 2016).

According to the literature, metaheuristic algorithms have become standard for solving engineering problems (Ficarella et al., 2021). Their ease of implementation and the rapidly increasing computational power allow population-based algorithms to run on traditional computers. Since virtually all metaheuristic algorithms employ a population of candidate designs, their main limitation lies in the large number of function evaluations and structural analyses required in the search process. To overcome this limitation, researchers have developed hybrid formulations that combine either two or more metaheuristic algorithms (Hwang & He, 2006; Ficarella et al., 2021) or metaheuristic search with gradient-based optimization (Fesanghary et al., 2008) and approximate line search (Pholdee & Bureerat, 2012).

Truong and Chou (2023) enhanced a forensic-based investigation algorithm by integrating fuzzy logic to optimize frequency-constrained structural dome designs. This modified version, the Fuzzy Adaptive Forensic-Based Investigation (FAFBI), was applied to optimize the weight of steel dome structures under specific frequency constraints. Optimization results showed that FAFBI surpassed existing methods in the literature, enabling the identification of lighter domes that still met the required frequency constraints.

According to the "no free lunch" theorem, no metaheuristic algorithm outperforms all other algorithms in solving all optimization problems. Unlike gradient-based optimizers, which are still used as optimization tools in commercial software despite their formulations remaining essentially unchanged for almost 30 years, most newly developed metaheuristic algorithms have added very little to the practice of optimization, and their appeal disappears after a few years (Ficarella et al., 2021). This situation suggests that instead of proposing new metaheuristic algorithms that marginally improve upon existing methods, it is crucial to enhance the most common algorithms significantly. This enhancement aims to make them competitive with gradient-based optimizers in terms of computational cost while preserving the inherent ability of metaheuristic optimizers to explore the design space globally. Therefore, this work focuses on developing the enhanced snake optimization algorithm.

2. Enhanced snake optimization

2.1. Mechanism of enhanced snake optimization

2.1.1. Generating diverse initial populations using a logistic map

The initial location in the search space is commonly selected at random. The logistic map was initially developed by May (1976) and is used herein to initialize the initial population (Chou & Truong, 2021); it is given by Eqn (21):

$$x_{i+1} = \eta x_i \left(1 - x_i\right), 0 \le X_0 \le 1, \tag{21}$$

where x_i is the position of i^{th} member; x_0 is the initial population, $x_0 \in (0,1)$, $x_0 \notin \{0.0, 0.25, 0.75, 0.5, 1.0\}$; and η is set to 4.0.

2.1.2. Lévy flight to accelerate convergence

Lévy flight has been used to accelerate the convergence of metaheuristic optimization algorithms and escape from local optima (Yang & Deb, 2009). Equation (22) provides the step length of s in Lévy flight:

$$\text{Lévy} \sim s = \frac{u}{|v|^{1/\tau}}, 0 < \tau \le 2. \tag{22}$$

Suppose variables u and v are assumed to be normally distributed, then:

$$u \sim N\left(0, \sigma_u^2\right), v \sim N\left(0, \sigma_v^2\right);$$

$$\sigma_u = \left\{\frac{\Gamma(1+\tau)\sin(\pi\tau/2)}{\Gamma\left[\frac{1+\tau}{2}\right]\tau 2^{(\tau-1)/2}}\right\}^{1/\tau}, \sigma_v = 1, \tau \text{ is set to 1.5.}$$

Here, $\Gamma(z)$ is the Gamma function:

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt.$$

2.1.3. Evaluate each group and define temperature and quantity of food

Since the temperature and quality of food change randomly as determine by environmental factors, Eqns (4) and (5) are revised to Eqns (23) and (24), respectively:

$$\text{Temp} = \text{rand}(0,1) \times \exp\left(\frac{-t}{T}\right); \tag{23}$$

$$Q = \text{rand}(0,1) \times \exp\left(\frac{t-T}{T}\right), \tag{24}$$

where t is the current iteration; T represents the maximum number of iterations, and rand (0,1) is a random number between 0 and 1.

2.1.4. Exploration phase (no food)

Step 1: To switch between the behaviors of snakes, a threshold food quality is a randomly selected rand (0,1). When Q < rand (0,1), the snakes search for food by updating their position to any randomly selected one. Suppose this exploration phase is assumed to reduce by the time (t), Eqns (6) and (8) are adjusted to Eqns (25) and (26), respectively:

$$x_{(m)i}(t+1) = x_{(m)\text{rand}}(t) \pm A_{(m)} \times \text{rand}(0,1)^{\alpha \times t} \times \left(\left(U_b - L_b\right) \times \text{rand}(0,1) + L_b\right), \tag{25}$$

where α is the coefficient of motion of the snake, and the sensitivity of α from the mathematical function in Section 3.3 shows that $\alpha = 0.1$ is preferred for solving optimization problems; $x_{(m)i}$ is the position of the i^{th} male, $x_{(m)\text{rand}}$ is the position of a random male; rand (0,1) is a random number between 0 and 1, and $A_{(m)}$ is the ability of a male to find the food, which is calculated using Eqn (7), above:

$$x_{(f)i}(t+1) = x_{(f)\text{rand}}(t) \pm A_{(f)} \times \text{rand}(0,1)^{\alpha \times t} \times \left(\left(U_b - L_b\right) \times \text{rand}(0,1) + L_b\right), \tag{26}$$

where α is the coefficient of motion of the snake, and the sensitivity of α from the mathematical function in Section 3.3 shows that α =0.1 is the best value for solving optimization problem; $x_{(f)i}$ is the position of the i^{th} female; $x_{(f)\text{rand}}$ is the position of a random female; rand (0,1) is a random number between 0 and 1, and $A_{(f)}$ is the ability of a female to find food, which is calculated by Eqn (9) above.

2.1.5. Exploitation phase (food exists)

Step 2: Similar to Threshold_1, Threshold_2 for switching between hot and cool is set to a rand (0,1) number. If Q > rand (0,1) and the temperature Temp > rand (0,1), then the snake is hot and will only move toward food. This exploitation step is simulated by searching around the current best (x_{food}). As a result, Eqn (10) has been revised to Eqn (27):

$$x_{i,j}(t+1) = x_{\text{food},j} + 2 \times \left(\text{rand}(0,1) - 0.5\right) \times \left(x_{\text{food},j}(t) - x_{i,j}(t)\right). \tag{27}$$

Lévy flight simulates this motion to enhance exploitation (Zhou et al., 2018; Wu et al., 2023). Thus, Eqn (27) is revised to Eqn (28):

$$x_{i,j}(t+1) = x_{\text{food},j} + \left(x_{\text{food},j}(t) - x_{i,j}(t)\right) \otimes \text{Lévy}(s), \tag{28}$$

where $x_{i,j}$ is the position of an individual (male or female), and x_{food} is the position of the best individuals.

When the temperature (1 – Temp) > rand (0,1), the snake is in a cool state and can be either in fight mode or mating mode.

Step 3: Fight mode Eqn (11) and Eqn (12) are revised to Eqn (29) and Eqn (30), respectively, as follows:

$$x_{(m)i}(t+1) = x_{(m)i}(t) + \text{FM} \times \text{rand}(0,1) \times \left(Q \times x_{(f)\text{best}}(t) - x_{(m)i}(t)\right), \tag{29}$$

where $x_{(m)i}$ is the position of the i^{th} male; $x_{(f)best}$ is the position of the best female, and FM is the fighting ability of a male.

$$x_{(f)i}(t+1) = x_{(f)i}(t) + \text{FM} \times \text{rand}(0,1) \times \left(Q \times x_{(m)\text{best}}(t) - x_{(f)i}(t)\right), \tag{30}$$

where $x_{(f)i}$ is the position of the i^{th} female, $x_{(m)best}$ is the position of the best male, and FF is the fighting ability of a female. FM and FF can be calculated using Eqns (13) and (14), above.

Step 4: Mating mode Eqn (15) and Eqn (16) are revised to Eqn (31) and Eqn (32), respectively, as follows:

$$x_{(m)i}(t+1) = x_{(m)i}(t) + 2\times(\text{rand}(0,1) - 0.5)\times M_{(m)}\times\left(Q\times x_{(f)i}(t) - x_{(m)i}(t)\right); \quad (31)$$

$$x_{(f)i}(t+1) = x_{(f)i}(t) + 2\times(\text{rand}(0,1) - 0.5)\times M_{(f)}\times\left(Q\times x_{(m)i}(t) - x_{(f)i}(t)\right), \quad (32)$$

where $x_{(f)i}$ is the position of the i^{th} female; $x_{(m)i}$ is the position of the i^{th} male and $M_{(m)}$ & $M_{(f)}$ are the mating abilities of a male and a female, respectively, which can be calculated using Eqns (17) and (18), above.

Step 5: Finally, the egg hatches, and this phenomenon is simulated as Eqns (19) and (20), above.

2.2. Pseudo-code of enhanced snake optimization

The two primary phases of a metaheuristic algorithm are exploration and exploitation (Chou & Truong, 2021; Truong & Chou, 2023). In SO, the exploration phase obtains every snake's search for food, and when food exists, each snack moves toward food only. In the exploitation phase, the snakes fight, mate with each other, and lay eggs.

This study introduces two versions of ESO, namely ESO^1 and ESO^2. ESO^1 employs a logistic map to initialize diverse populations, while ESO^2 further incorporates a Lévy flight during the food searching (Step 2) to enhance ESO^1's exploitation ability. Compared to the original SO, ESO^1 and ESO^2 have only one specific parameter to configure: the coefficient of motion of the snake in Eqns (25) and (26), denoted as α, as depicted in Figure 3.

```
Begin
    Initialize Problem Setting: Number of dimensions (Dim); Objective function (f); Lower and upper bounds: (L_b, U_b); Population:
    (N); Maximum number of iterations: (T)
    Initialize the population using a logistic map according to Eqn (21) (Original equation is Eqn (1))
    Divide population N into two equal groups, N_(m) and N_(f) using Eqns (2) and (3)
    While (t ≤ T) do
        Evaluate each group N_(m) and N_(f)
        Find best male f_(m)best; best fermale f_(f)best
        Define temp using Eqn (23) (Original equation is Eqn (4))
        Define food quantity (Q) using Eqn (24) (Original equation is Eqn (5))
        If (Q < rand (0,1) then (Original formula is Q < 0.25)
            # Step 1:
            Calculate ability to find the food A_(m), A_(f) using Eqns. (7) and ( 9)
            Perform exploration using Eqns (25) and (26) (Original equations are Eqns (6) and (8))
        Else if (Temp (rand > 0,1)) then (Original formula is Temp > 0.6)
            # Step 2
                (ESO^1): Perform exploitation Eqn (27) (Original equation is Eqn (10))
                (ESO^2): Perform exploitation Eqn (28) (Original equation is Eqn (10))
        Else
            If (1-Temp)>rand(0,1), then (Original formula is rand > 0.6)
            # Step 3:
                Calculate the fighting abilities of male FM and female FF using Eqns (13) and (14)
                Snakes in fight mode by Eqns (29) and (30) (Original equations are Eqns (11) and (12))
            Else
                # Step 4:
                Calculate mating abilities of male M_(m) and female M_(f) using Eqns (17) and (18)
                Snakes in mating mode by Eqns (31) and (32) (Original equations are Eqns (15) and (16))
                # Step 5:
                Change the worst male and female using Eqns (19) and (20)
            End if
        End if
    End while
    Return the best solution
End
```

Figure 3. Pseudo-code of ESOs

3. Solving mathematical functions

3.1. Benchmarking test of optimization algorithms

The proposed algorithm is evaluated with 13 well-known metaheuristic optimizers (ABC (Karaboga & Basturk, 2007), CA (Omran, 2016), DE (Storn & Price, 1997), FA (Yang, 2010a), GA (Holland, 1992), HS (Zong Woo et al., 2001), JAYA (Rao, 2016), PSO (Kennedy & Eberhart, 1995), SA (van Laarhoven & Aarts, 1987), SO (Hashim & Hussien, 2022), SOS (Cheng & Prayogo, 2014), TLBO (Rao et al., 2011), and GWO (Mirjalili et al., 2014)), and 24 mathematical functions with ten and 20 dimensions that are taken from the IEEE CEC-2022 special section, detail of these functions are presented in Table A1 (Appendix, A) (Kumar et al., 2021). Thirty independent runs were carried out for each benchmark function to eliminate stochastic discrepancy (Chou & Truong, 2021).

3.2. Wilcoxon rank sum test

The performance of the proposed algorithm is compared with that of each peer algorithm at a significance level of 1%. If the performance of ESO^1 or ESO^2 is μ_1 and that of each peer algorithm is μ_2, then the hypotheses concerning the performance metrics are (Derrac et al., 2011):

$$\begin{cases} H_0 : \mu_1 \text{ is equivalent to } \mu_2 \\ H_1 : \mu_1 \text{ is better than } \mu_2 \end{cases}. \quad (33)$$

3.3. Motion coefficient (α) of enhanced snake optimization

The motion coefficient α (in Eqns (25) and (26)) affects the objective function value in solving optimization problems. Four mathematical functions (F3, F6, F9, and F12 with dimension = 20) were used to evaluate the appropriate value of the motion coefficient. In this investigation, α was set to various values (case 1 (α = 0.01), case 2 (α = 0.1), case 3 (α = 0.2), case 4 (α = 0.3), case 5 (α = 0.4), case 6 (α = 0.5), case 7 (α = 0.6), case 8 (α = 0.7), case 9 (α = 0.8), case 10 (α = 0.9), and case 11 (α = 1.0)).

The population size and the number of iterations are 50 and 1,000, respectively. ESO^1 and ESO^2 were executed 30 times for each case, generating the statistical box plots in Figures 4 and 5. In all four problems for ESO^1 and ESO^2, case 2 (α = 0.1) exhibits a smaller interquartile range and a lower median than the other cases. Figures 4 and 5 show that case 2 consistently yields a more stable global optimum. Consequently, α = 0.1 was chosen as the motion coefficient to solve these problems.

3.4. Comparison of algorithms for solving benchmark functions

In this test, the population size was set to 50, and each test function was solved within 5 seconds. All other internal parameters were kept at their default values. The statisti-

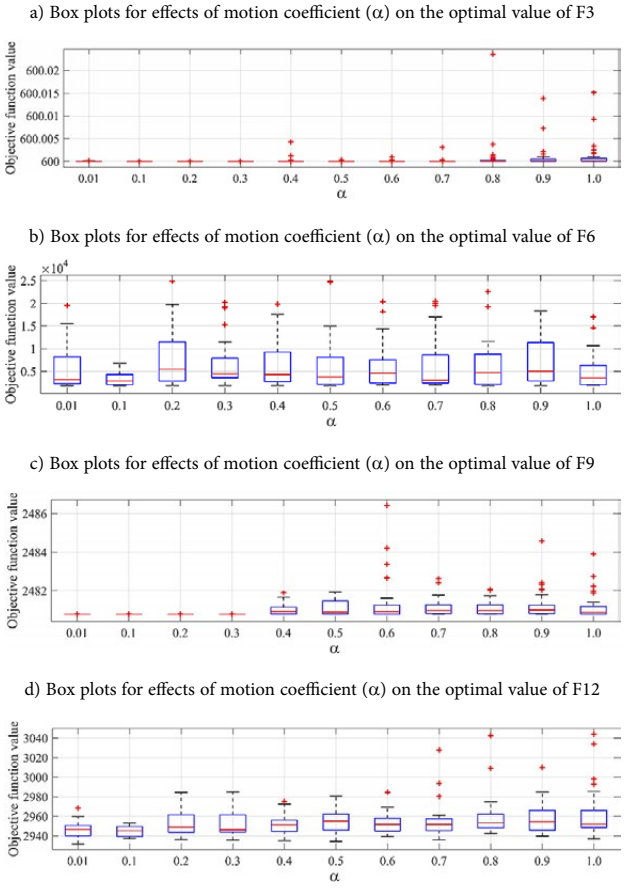

Figure 4. Box plots for effects of motion coefficient (α) on the optimal value for ESO^1

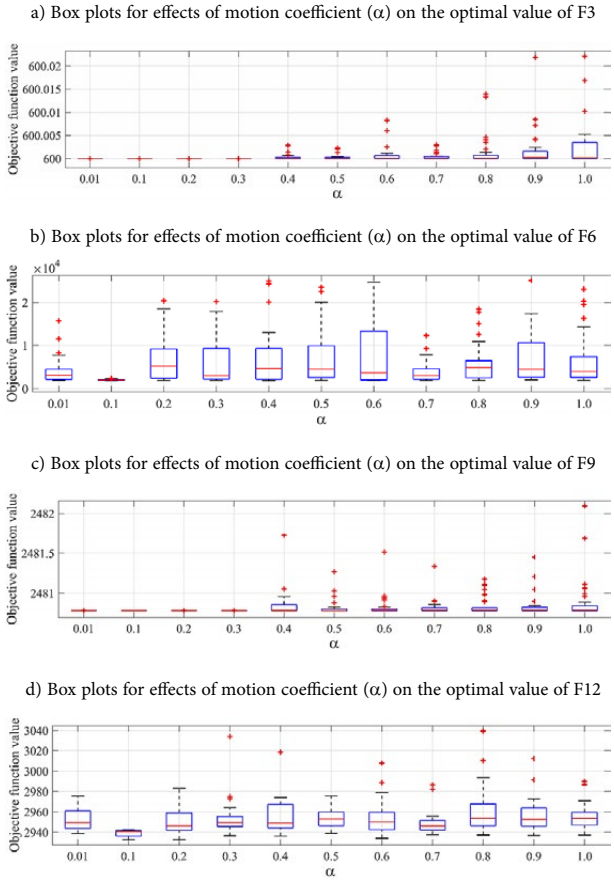

Figure 5. Box plots for effects of motion coefficient (α) on the optimal value for ESO^2

cal optimal values of all optimizers are presented in Tables B.1 and B.2 (Appendix, B). Concurrently, the results of the Wilcoxon rank sum tests for ESO^1, ESO^2, and 13 well-known algorithms are depicted in Figures 6 and 7, respectively. The analysis reveals that ESO^1 significantly outperforms ABC, CA, DE, FA, GA, HS, JAYA, PSO, SA, SO, SOS, TLBO, and GWO in 15, 16, 13, 15, 21, 16, 24, 16, 19, 18, 13, 15, and 22 out of 24 functions. Similarly, ESO^2 outperforms these algorithms in 16, 17, 18, 22, 23, 23, 24, 20, 19, 20, 17, 22, and 23 functions, demonstrating its superior performance.

Thus, ESO^1 and ESO^2 are more effective and robust than the original SO and other optimizers in the mathematical tests. Figure 8 illustrates this, with ESO^2 consistently outperforming ESO^1 across 0 to 7 functions, showcasing its superior ability to find optimal values compared to other algorithms.

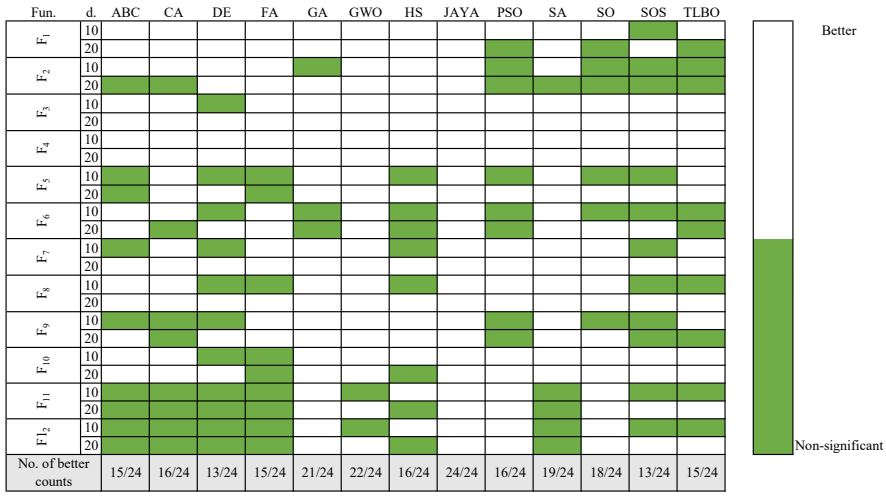

Figure 6. Results of Wilcoxon rank sum test of ESO^1 vs. other optimization algorithms

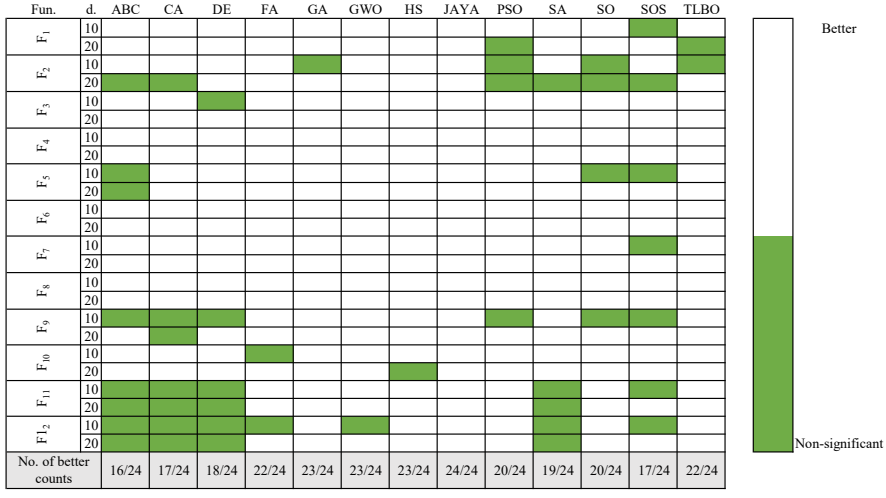

Figure 7. Results of Wilcoxon rank sum test of ESO^2 vs. other optimization algorithms

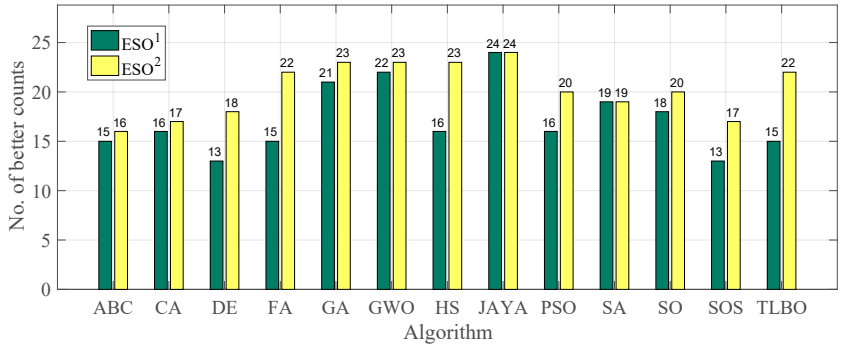

Figure 8. Comparison of ESO^1 and ESO^2 vs. other optimization algorithms

4. Designing large space double-layer grid structures

4.1. Formulation of the optimization problem

In a truss optimization problem with multiple constraints, the objective is to minimize the total weight of the truss structure while satisfying various conditions. The sizing optimization problem for a truss structure is mathematically formulated (Degertekin et al., 2018):

$$\text{Find } \{\mathbf{A}\} = [a_1, a_2, \cdots, a_d] \text{ that minimizes}$$

$$f_{cost}(\{\mathbf{A}\}) = \sum_{i=1}^{d} a_{i,j} \sum_{j=1}^{nm(i)} \rho_j L_j, \tag{34}$$

$$\text{Subject to: } \begin{cases} g_j(\{\mathbf{A}\}) \le 0 & j = 1,2,\ldots,n_c \\ a_i^{min} \le a_i \le a_i^{max} & i = 1,2,\ldots,d \end{cases}, \tag{35}$$

where $f(\{\mathbf{A}\})$ is the objective function, which is taken as the weight of the structure; **A** is the vector of design variables, whose elements are the cross-sectional areas of structural members; d is the number of sizing design variables; nm is the number of structural members; $a_{i,j}$, ρ_j and L_j are the cross-sectional area, material density, and length of the j^{th} structural member, respectively; a_i^{min} and a_i^{max} are the lower and upper bounds on i^{th} design variable, respectively; $g_j(\{\mathbf{A}\})$ denote design constraints; n_c is the number of design constraints.

Among the various strategies for handling constraints in optimization problems, some common ones involve penalizing processes. The general idea behind penalization is to transform the constrained optimization problem into an unconstrained one. As demonstrated above, this is achieved by incorporating a penalty term into the objective function:

$$\text{Minimize } F(\{\mathbf{A}\}) = f_{cost}(\{\mathbf{A}\}) \times f_p(\{\mathbf{A}\}), \tag{36}$$

where $F(\{\mathbf{A}\})$ is the penalized objective function and $f_p(\{\mathbf{A}\})$ is the penalty function. Degertekin et al. (2018) proposed a penalty function $f_p(\{\mathbf{A}\})$ to handle the constraint conditions of the objective function $f_{cost}(\{\mathbf{A}\})$ (Degertekin et al., 2018).

$$f_p(\{\mathbf{A}\}) = (1+\phi)^{\varepsilon}, \tag{37}$$

where ϕ is the sum of penalties, defined as

$$\phi = \sum_{i=1}^{q} \phi_i; \tag{38}$$

$$\phi_i = \left| 1 - \frac{p_i}{p_i^*} \right|, \tag{39}$$

where ϕ_i is the degree of constraint violation with the bound set to p_i^*, and q is the number of active constraints. The exponent in the penalty function ε is a function of the number of iterations and is given by

$$\varepsilon = \varepsilon_o \left(1 + \frac{t}{Max_{iter}} \right), \tag{40}$$

where the initial value ε_o can be set between 1.001 and 10,000 (Degertekin et al., 2018) and is set to 2 herein.

The finite element method (FEM) is applied to calculate the axis forces (or element stresses) of the 3D truss structure, which is presented as a global stiffness equation below:

$$\{F\} = [K]\{d\}. \tag{41}$$

Hooke's law calculates the axial stress of each element (σ_x) as follows (Logan, 2016):

$$\sigma_x = E\varepsilon_x. \tag{42}$$

Thus, each member's axial force (T) is calculated by Eqn (42):

$$T = A\sigma_x, \tag{43}$$

where {F} denotes global nodal forces; [K] is the global stiffness matrix; {d} is global nodal displacements; E is the modulus of elasticity, ε_x is axial strain; A is cross-sectional area.

The SO and ESOs are integrated with the finite element method (FEM) to find the minimum weights of steel structures. Figure 9 shows the analytical procedure in this application.

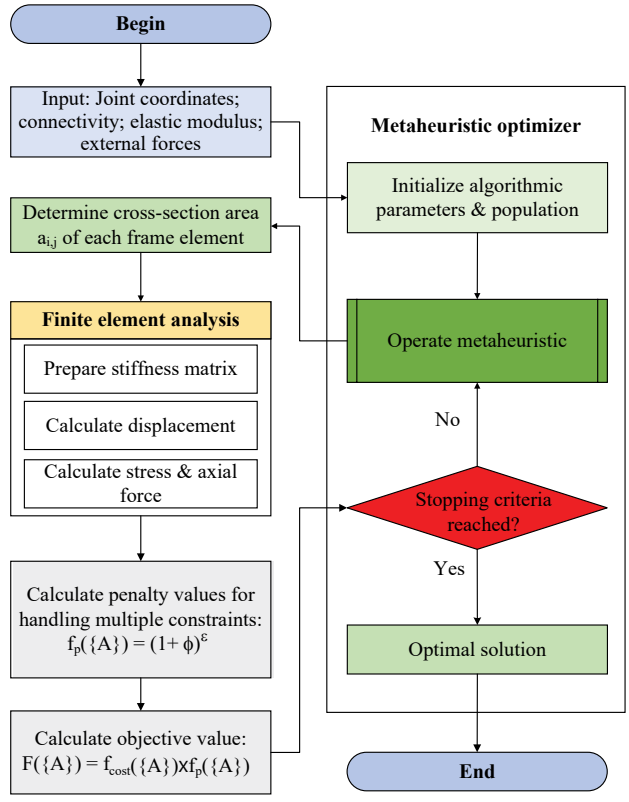

Figure 9. Framework of metaheuristic optimizers integrated with FEM for solving structural design problems

4.2. 47-bar power transmission tower

The design example, depicted in Figure 10, consists of 47 members and 22 nodes, as presented in the work of Degertekin et al. (2019). The cross-sectional areas of the members are divided into 27 groups and were selected from the 64 discrete values listed in Table 1. The members' material has Young's modulus of 30,000 ksi and a density of 0.3 lb/in^3 (Lee et al., 2005).

The structure's nodes are subjected to three loading cases: (1) 6.0 kips in the positive x-direction and 14.0 kips in the negative y-direction at nodes 17 and 22, (2) 6.0 kips in the positive x-direction and 14.0 kips in the negative y-direction at node 17, and (3) 6.0 kips in the positive x-direction and 14.0 kips in the negative y-direction at node 22. The first loading case corresponds to applying a load by the two power lines at an angle to the tower, while the second and third loading cases correspond to snapping one of the two lines. All members of the tower must satisfy both stress and buckling constraints. Allowable tensile and compressive stresses are set at 20 ksi and 15.0 ksi, respectively. The Euler buckling compressive stress of a member with a cross-sectional area of A_i is calculated as follows:

$$\sigma_i^{cr} = \frac{-KEA_i}{L_i^2}, \; i = 1, 2, 3, \ldots, 47, \tag{44}$$

where K is a constant parameter that should be selected according to the cross-sectional geometry; E is Young's modulus of the material; and L_i is the length of member i. The buckling constant K is considered 3.96 (Lee et al., 2005).

To solve this problem, the snake population is set to 50, and the number of structural analyses is 10,000. Table 2 compares the results concerning the optimal designs obtained using different methods. The original SO, ESO[1], and ESO[2] generated the lightest designs, weighing 2,292.06 lb, 2,281.61 lb, and 2,279.19 lb, respectively. The average optimized weight and the standard deviation of the average weight for that design (2,324.52 lb and 30.96 lb) by SO, (2,315.91 lb and 21.84 lb) by ESO[1], and (2,297.67 lb and 12.80 lb) by ESO[2] are less than those obtained using any other method. SO and ESOs require 10,000 structural analyses to find the optimal solution design, whereas HS, CBO, DAJA, and MDVC-UVPS require 45,557, 25,000, 8,046, and 4,867 structural analyses.

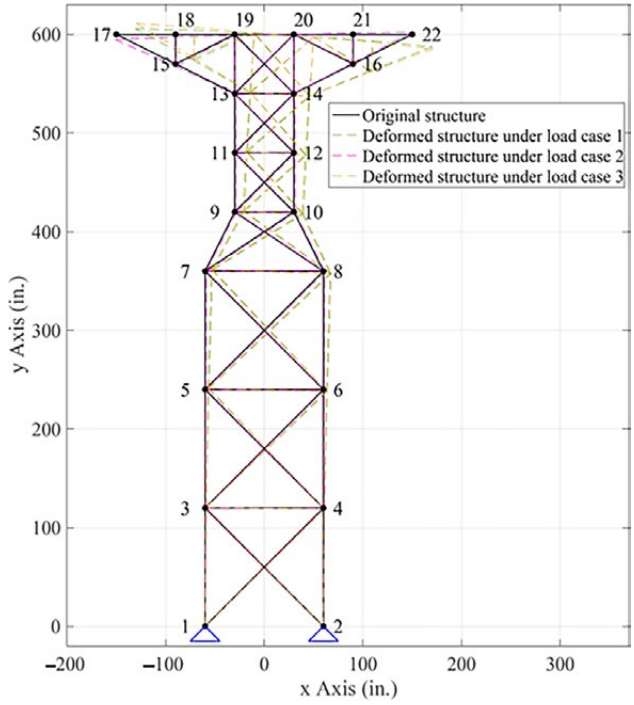

Figure 10. Schematic of 47-bar power transmission tower

Table 1. List of available cross-sectional areas in 47-bar power transmission tower problem (Kaveh & Ilchi Ghazaan, 2018c)

No.	Area (in.^{2})	No.	Area (in.^{2})	No.	Area (in.^{2})	No.	Area (in.^{2})
1	0.111	17	1.563	33	3.840	49	11.5001
2	0.141	18	1.620	34	3.870	50	13.500
3	0.196	19	1.800	35	3.880	51	13.900
4	0.25	20	1.990	36	4.180	52	14.200
5	0.307	21	2.130	37	4.220	53	15.500
6	0.391	22	2.380	38	4.490	54	16.000
7	0.442	23	2.620	39	4.5899	55	16.900
8	0.563	24	2.630	40	4.800	56	18.800
9	0.602	25	2.880	41	4.970	57	19.900
10	0.766	26	2.930	42	5.120	58	22.000
11	0.785	27	3.090	43	5.740	59	22.900
12	0.994	28	1.130	44	7.220	60	24.500
13	1.000	29	3.380	45	7.970	61	26.500
14	1.228	30	3.470	46	8.530	62	28.000
15	1.266	31	3.550	47	9.300	63	30.000
16	1.457	32	3.630	48	10.85	64	33.500

Table 2. Weight design comparison for 47-bar power transmission tower

Design variables a_i (in^2)	HS (Lee et al., 2005)	CBO (Kaveh & Mahdavi, 2014)	DAJA (Degertekin et al., 2019)	MDVC-UVPS (Kaveh & Ilchi Ghazaan, 2018b)	SO	ESO[1]	ESO[2]
					(Present study)		
a_1	3.84	3.84	3.84	3.84	2.93	2.93	2.88
a_2	3.38	3.38	3.38	3.38	2.38	2.38	2.38
a_3	0.766	0.785	0.766	0.766	0.766	0.766	0.766
a_4	0.141	0.196	0.111	0.111	0.111	0.111	0.111
a_5	0.785	0.994	0.785	0.785	0.785	0.785	0.785
a_6	1.99	1.8	1.99	1.99	1.99	1.99	1.99
a_7	2.13	2.13	2.13	2.13	2.13	2.13	2.13
a_8	1.228	1.228	1.228	1.228	0.994	0.994	0.994
a_9	1.563	1.563	1.563	1.563	1.228	1.228	1.228
a_{10}	2.13	2.13	2.13	2.13	2.13	2.13	2.13
a_{11}	0.111	0.111	0.111	0.111	0.785	0.785	0.785
a_{12}	0.111	0.111	0.111	0.111	0.111	0.111	0.111
a_{13}	1.8	1.8	1.8	1.8	1.8	1.8	1.8
a_{14}	1.8	1.8	1.8	1.8	1.8	1.8	1.8
a_{15}	1.457	1.563	1.457	1.457	1.457	1.457	1.457
a_{16}	0.442	0.442	0.563	0.563	0.442	0.563	0.442
a_{17}	3.63	3.63	3.63	3.63	3.38	3.38	3.38
a_{18}	1.457	1.457	1.457	1.457	1.13	1.13	1.13
a_{19}	0.442	0.307	0.25	0.25	0.307	0.307	0.307
a_{20}	3.63	3.09	3.09	3.09	3.09	3.09	3.09
a_{21}	1.457	1.266	1.266	1.228	1.266	1.266	1.266
a_{22}	0.196	0.307	0.307	0.391	0.442	0.391	0.391
a_{23}	3.84	3.84	3.84	3.84	3.84	3.84	3.84
a_{24}	1.563	1.563	1.563	1.563	1.563	1.563	1.563
a_{25}	0.196	0.111	0.141	0.111	0.111	0.111	0.111
a_{26}	4.59	4.59	4.59	4.59	4.5899	4.5899	4.5899
a_{27}	1.457	1.457	1.457	1.457	1.563	1.457	1.457
Weight (lb)	2,396.8	2,386	2,376.019	2,374.09	2,292.06	2,281.61	2,279.19
Average (lb)	N/A	2,405.91	2,399.92	2,413.46	2,324.52	2,315.91	2,297.67
Std. (lb)	N/A	19.61	13.15	38.21	30.96	21.84	12.80
NSA	45,557	25,000	8,046	4,867	10,000	10,000	10,000

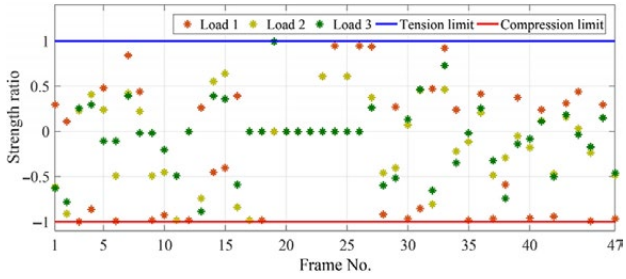

Figure 11. Violation check of strength constraint of 47-bar power transmission tower

All frame members' strength constraints are satisfied by the solution generated by ESO[2], as shown in Figure 11. Although SO and ESOs require a higher number of structural analyses (NSA) compared to DAJA and MDVC-UVPS, they need a lower NSA than HS and CBO while being able to find lighter solutions than the other methods. Hence, both SO and ESOs effectively address the initial problem, with ESO[2] consistently discovering solutions that are, on average, 1.16% and 0.79% lighter than those found by SO and ESO[1], respectively.

4.3. 72-bar tower

This problem, previously studied by Wu and Chow (1995), Lee et al. (2005), and da Silva et al. (2011), involves seven members and 20 nodes and is presented in Figure 12. The material density is 0.1 lb/in.^{3}, and the modulus of elasticity is 10,000 ksi. Stress limitations for all members are within ±25 ksi. The uppermost nodes are regulated with a tolerance of ±0.25 in. in the x and y directions. The loads applied to the structure include load case 1: $F_{1x} = F_{1y} = 5$ kips and $F_{1z} = -5$ kips and load case 2: $F_{1z} = F_{2z} = F_{3z} = F_{4z} =$

−5 kips. The frame members are divided into 16 groups as follows: (a_1) A1–A4, (a_2) A5–A12, (a_3) A13–A16, (a_4) A17–A18, (a_5) A19–A22, (a_6) A23–A30, (a_7) A31–A34, (a_8) A35–A36, (a_9) A37–A40, (a_{10}) A41–A48, (a_{11}) A49–A52, (a_{12}) A53–A54, (a_{13}) A55–A58, (a_{14}) A59–A66, (a_{15}) A67–A70, (a_{16}) A71–A72. The design variables are the cross-sectional areas of the bars, each with a minimum value of 0.1 in^2.

To solve this problem, a population of 50 snakes and 20,000 structural analyses were utilized. Table 3 presents the optimal designs obtained by the original SO, ESOs, and methods from the literature. Among all methods, only ESO^2 produced the lightest design, weighing 378.429 lb. Additionally, the average optimized weight and its standard deviation for ESO^2 (378.607 lb and 0.515 lb) were lower than those of all other methods. The best designs found by GA+APM, DE, DUVDE+APM, and GAOS were 387.036 lb, 379.880 lb, 379.667 lb, and 383.120 lb, respectively. SO and ESOs required 20,000 structural analyses to find the optimal solution, whereas GA+APM, DE, DUVDE+APM, and GAOS required 35,000 or 30,000. Figure 13 illustrates the constraint violation check for

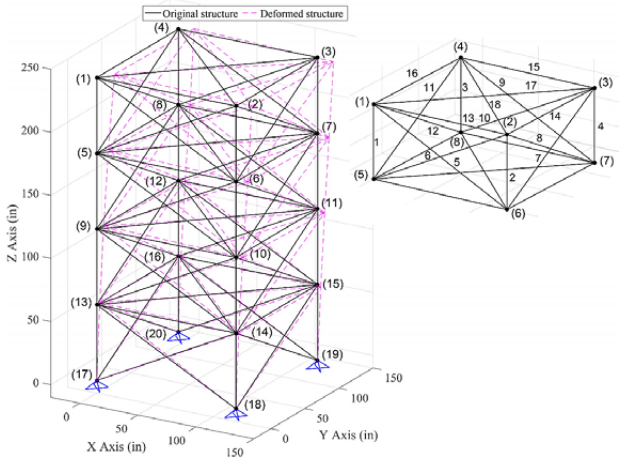

Figure 12. 72-bar tower problem

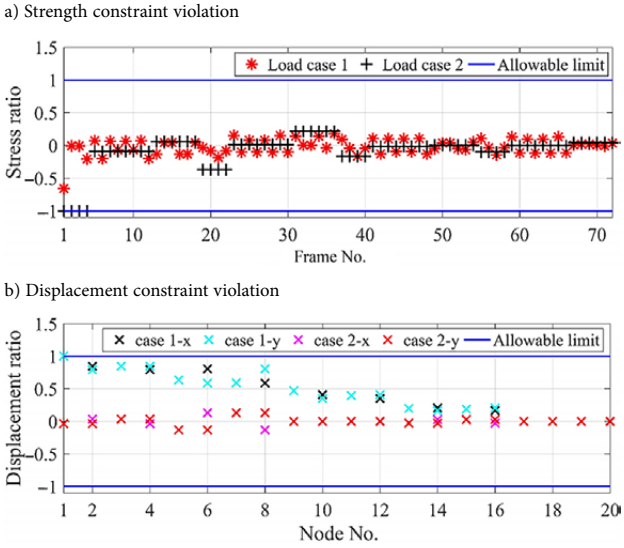

Figure 13. Constraint violation check in the 72-bar tower

Table 3. Weight design comparison for the 72-bar tower

Design variables a_i (in^2)	GA+APM (Lemonge & Barbosa, 2004)	DE (da Silva et al., 2011)	DUVDE +APM (da Silva et al., 2011)	GAOS (Erbatur et al., 2000)	SO	ESO^1	ESO^2
					(Present study)		
a_1	0.15500	0.15665	0.15621	0.161	0.15609	0.15528	0.15558
a_2	0.54534	0.54617	0.55147	0.544	0.52633	0.55283	0.55215
a_3	0.27496	0.38993	0.41361	0.379	0.35323	0.39651	0.39637
a_4	0.51853	0.58564	0.56267	0.521	0.52405	0.53918	0.53737
a_5	0.60365	0.53008	0.53063	0.535	0.62609	0.56421	0.55861
a_6	0.66607	0.52951	0.51932	0.535	0.52554	0.51884	0.51557
a_7	0.10159	0.10000	0.10002	0.103	0.10185	0.10000	0.10002
a_8	0.13008	0.10000	0.10052	0.111	0.11811	0.10897	0.10550
a_9	1.19954	1.27713	1.27057	1.310	1.18824	1.19375	1.18999
a_{10}	0.47368	0.52173	0.50926	0.498	0.52803	0.52433	0.52087
a_{11}	0.10059	0.10000	0.10012	0.110	0.10188	0.10000	0.10002
a_{12}	0.10945	0.10000	0.10000	0.103	0.10409	0.10001	0.10000
a_{13}	1.95307	1.82352	1.86245	1.910	1.97753	1.97362	1.96867
a_{14}	0.51653	0.50643	0.51098	0.525	0.52109	0.49679	0.49323
a_{15}	0.10000	0.10000	0.10000	0.122	0.10000	0.10000	0.10000
a_{16}	0.10105	0.10000	0.10000	0.103	0.10095	0.10002	0.10001
Weight (lb)	387.036	379.880	379.667	383.120	380.543	380.127	378.429
Average (lb)	402.58	380.24	380.42	N/A	381.271	380.651	378.607
Std. (lb)	N/A	0.26517	0.57266	N/A	0.553	0.456	0.515
NSA	35,000	35,000	35,000	30,000	20,000	20,000	20,000

the stress and displacement constraints in the solution obtained using ESO^2, confirming that all frame members or nodes satisfy these constraints. Thus, ESO^2 is an effective method for solving the second problem, consistently providing solutions that are, on average, 0.70% and 0.54% better than those found by SO and ESO[1], respectively.

4.4. 672-bar double-layer grid problem

The structure spans 40×40 m with a height of 3 m, as outlined in Kaveh and Ilchi Ghazaan (2018b). This configuration comprises 672 frame members and 205 nodes, with the bottom layer supported at the nodes illustrated in Figure 14 (Kaveh & Ilchi Ghazaan, 2018b). The cross-sectional areas of the members are divided into 22 groups. Each top layer joint experiences a concentrated vertical load of 30 kN, as shown in Figure 14.

The design variables involve the cross-sectional areas of the bar elements, which are chosen from a selection of steel pipe sections specified in AISC-LRFD and detailed in Table 4 (American Institute of Steel Construction, 1994). In this table, ST, EST, and DEST represent standard weight, extra strong, and double-extra robust, respectively. The steel's modulus of elasticity, yield stress, and density are considered 205 GPa, 248.2 MPa, and 7833.413 kg/m^3, respectively. As mentioned, the strength and slenderness constraints adhere to the AISC-LRFD regulations. Additionally, displacement limitations of span/600 were applied to all nodes in the vertical direction (Kaveh & Ilchi Ghazaan, 2018b). Figure 14 provides a 3D view of a diagonal member within the grid. The constraint conditions for grid structures are as follows (American Institute of Steel Construction, 1994):

Displacement constraint:

$$\delta_i \le \delta_i^{max},\ i = 1,2,\ldots,nn. \tag{45}$$

Tension member constraint:

$$P_u \le P_r;\ P_r = \min\begin{cases} \phi_t F_y A_g;\ \phi_t = 0.9 \\ \phi_t F_u A_e;\ \phi_t = 0.75 \end{cases}. \tag{46}$$

Compression member constraint:

$$P_u \le P_r;\ P_r = \phi_c F_{cr} A_g;\ \phi_c = 0.85; \tag{47}$$

$$F_{cr} = \begin{cases} \left(0.658^{\frac{F_y}{F_e}}\right) F_y; & \frac{KL}{r} \le 4.71\sqrt{\frac{E}{F_y}} \\ 0.877 F_e; & \frac{KL}{r} > 4.71\sqrt{\frac{E}{F_y}} \end{cases};\ F_e = \frac{\pi^2 E}{\left(\frac{KL}{r}\right)^2}. \tag{48}$$

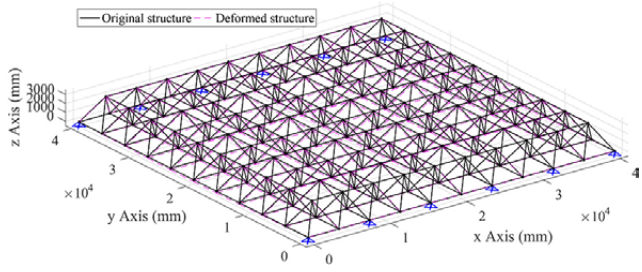

Figure 14. 672-bar tower problem

Table 4. List of steel pipe sections from American Institute of Steel Construction (1994)

No.	Type	Area (cm^2)	Gyration radius (cm)	No.	Type	Area (cm^2)	Gyration radius (cm)
1	[a]ST ½	1.6129	0.662432	20	EST 3½	23.741888	3.318002
2	[b]EST ½	2.064512	0.635	21	DEST 2½	25.999948	2.143506
3	ST ¾	2.129028	0.846582	22	ST 5	27.74188	4.775454
4	EST ¾	2.774188	0.818896	23	EST 4	28.451556	3.749548
5	ST 1	3.161284	1.066038	24	DEST 3	35.290252	2.65811
6	EST 1	4.129024	1.034542	25	ST 6	35.999928	5.700014
7	ST1 ¼	4.322572	1.371346	26	EST 5	39.419276	4.675124
8	ST1 ½	5.16128	1.582166	27	DEST 4	52.25796	3.490976
9	EST 1¼	5.677408	1.331214	28	ST 8	54.19344	7.462012
10	EST 1½	6.903212	2.003806	29	EST 6	54.19344	5.577332
11	ST 2	6.903212	1.53543	30	DEST 5	72.90308	4.379976
12	EST 2	9.548368	1.945132	31	ST 10	76.77404	9.342628
13	ST 2½	10.96772	2.41681	32	EST 8	82.58048	7.309358
14	ST 3	14.387068	2.955798	33	ST 12	94.19336	11.10361
15	EST 2½	14.5161	2.346452	34	DEST 6	100.64496	5.236464
16	[c]DEST 2	17.161256	1.782572	35	EST 10	103.87076	9.216898
17	ST 3½	17.290288	3.395726	36	EST 12	123.87072	11.028934
18	EST 3	19.483832	2.882646	37	DEST 8	137.41908	7.004812
19	ST 4	20.451572	3.835908				

Note: [a]ST standard weight; [b]EST extra strong; [c]DEST double-extra strong.

Slenderness ratio constraints:

$$\frac{KL}{r} \le 200 \quad \text{for compression member;} \tag{49}$$

$$\frac{KL}{r} \le 300 \quad \text{for tension member.} \tag{50}$$

To tackle this problem, a population of 50 snakes was used, and the structural analyses were performed 10,000 times. Table 5 presents the optimal designs obtained through SO, ESOs, and various methods from the literature. Among these methods, ESO^2 identified the lightest structure, weighing 53,381 kg. The average optimized weight and its standard deviation for SO (57,354 kg and 2,910 kg), ESO^1 (56,735 kg and 3,615 kg), and ESO^2 (56,011 kg and 3,603 kg) were lower than those obtained by any other method. Specifically, CBO, ECBO, VPS, and MDVC-UVPS resulted in average optimized weight of 62,287 kg, 59,164 kg, 60,850 kg, and 58,589 kg, respectively.

Figure 15 illustrates the constraint violations related to strength, displacement, and slenderness ratio for the solution identified by ESO^2. Additionally, ESOs required 10,000 structural analyses to reach the optimum solution, while CBO required 4,360 NSA, MDVC-UVPS required 3,142 NSA, ECBO necessitated 18,000 NSA, and VPS demanded 12,120 NSA. Consequently, ESOs are effective methods for resolving this problem, with ESO^2 consistently producing solutions that are, on average, 2.34% and 1.28% lighter than those found by SO and ESO^1, respectively.

4.5. 1520-bar double-layer grid problem

This structure comprises 1520 members and 401 nodes; Figure 16 provides a 3D view of a diagonal on a grid. Like the previous 672-bar problem, this structure spans 40×40 m with a height of 3 m, as detailed in Kaveh and Ilchi Ghazaan (2018b). Each top-layer joint bears a concentrated vertical load of 16 kN. The members' cross-sectional areas are grouped into 31 categories. The constraint conditions for grid structures are outlined in Eqns (45) to (50), above.

To address this problem, a population of 50 snakes was employed, and structural analyses were conducted 10,000 times. Table 6 provides a comparison of the optimal designs obtained using different methods. Both SO and ESOs

Table 5. Weight design comparison in 672-bar double-layer grid problem

Design variables a_i	CBO (Kaveh & Ilchi Ghazaan, 2018b)	ECBO (Kaveh & Ilchi Ghazaan, 2018b)	VPS (Kaveh & Ilchi Ghazaan, 2018b)	MDVC-UVPS (Kaveh & Ilchi Ghazaan, 2018b)	SO	ESO^1	ESO^2
					(Present study)		
a_1	ST 4	ST 4	ST 5	ST 4	ST 5	ST 5	ST 5
a_2	ST 6	ST 5	ST 4	ST 5	ST 5	ST 5	ST 5
a_3	ST 2½	ST 3½	ST 4	ST 3½	EST 2	EST 2	EST 2
a_4	ST 2½	EST 1½	EST 1½	EST 1½	ST 1¼	ST 1¼	ST 1¼
a_5	ST 2½	ST 4	ST 6	ST 3	EST 4	EST 4	ST 5
a_6	EST 6	EST 4	EST 6	DEST 4	DEST 5	DEST 5	EST 6
a_7	EST 4	EST 6	EST 6	DEST 4	EST 4	DEST 3	DEST 3
a_8	EST 6	DEST 4	DEST 5	EST 6	DEST 4	DEST 4	ST 8
a_9	EST 5	ST 6	EST 5	ST 6	EST 5	EST 5	EST 5
a_{10}	ST 4	ST 3½	ST 3½	ST 3½	ST 4	ST 4	ST 4
a_{11}	ST 6	EST 6	ST 5	DEST 4	EST 4	EST 4	ST 6
a_{12}	EST 5	EST 4	EST 4	ST 5	EST 4	EST 4	EST 4
a_{13}	EST 5	DEST 4	EST 6	EST 6	ST 8	DEST 4	ST 8
a_{14}	DEST 4	DEST 4	ST 6	EST 5	EST 5	EST 5	EST 5
a_{15}	EST 8	EST 6	DEST 4	DEST 4	DEST 4	DEST 4	DEST 4
a_{16}	DEST 4	DEST 4	EST 6	EST 5	DEST 4	DEST 4	DEST 4
a_{17}	ST 5	DEST 5	ST 8	DEST 5	EST 6	EST 6	EST 6
a_{18}	ST 4	ST 4	ST 4	ST 4	ST 4	ST 4	ST 4
a_{19}	ST 4	ST 4	ST 3½	ST 3½	ST 3½	ST 3½	ST 3½
a_{20}	ST 3½	ST 3½	ST 3½	ST 3½	ST 3½	ST 3½	ST 3½
a_{21}	ST 3½	ST 3	ST 3½	EST 2½	ST 3	ST 3	ST 3
a_{22}	ST 2½	ST 2½	ST 2½	ST 2½	ST 2½	ST 2½	ST 2½
Weight (kg)	55,621	54,569	53,704	53,552	53,575	53,537	53,381
Average (kg)	62,287	59,164	60,850	58,589	57,354	56,735	56,011
Std. (kg)	9,853	5,597	5,985	3,626	2,910	3,615	3,603
NSA	4,360	18,000	12,120	3,142	10,000	10,000	10,000

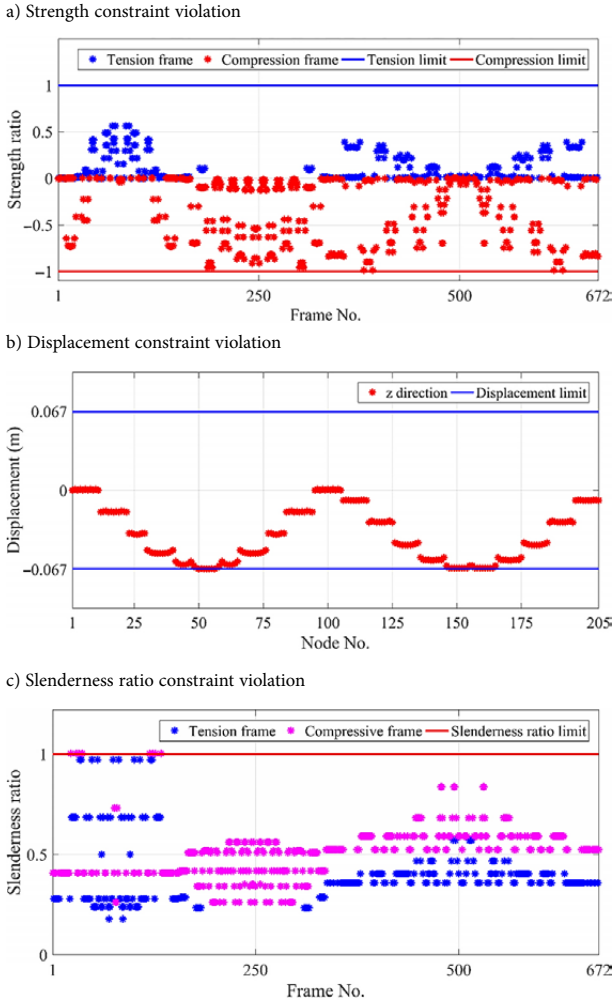

Figure 15. Constraint violation check in the 672-bar double-layer grid problem

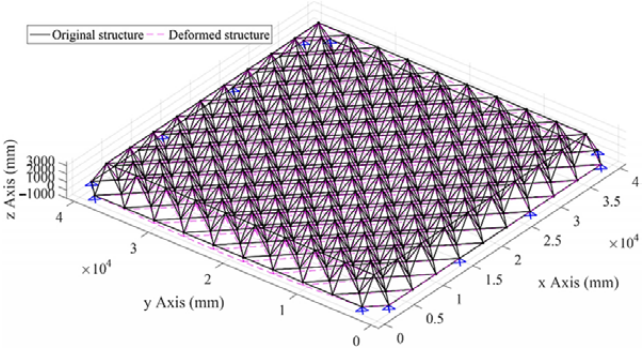

Figure 16. 3D view of a 1520-bar double-layer grid

produced designs lighter than any other method, resulting in a weight of 79,426 kg, ESO[1] at 79,423 kg, and ESO[2] at 79,421 kg. The average optimized weight and its standard deviation for SO (84,681 kg and 4,050 kg), ESO[1] (82,979 kg and 4,012 kg), and ESO[2] (81,565 kg and 3,672 kg) were lower than those achieved by all other methods.

The SO and ESOs required 10,000 structural analyses to find the optimal solution, while CBO, ECBO, VPS, and MDVC-UVPS required 4,360, 18,000, 12,120, and 3,142 structural analyses. Although SO and ESOs needed more

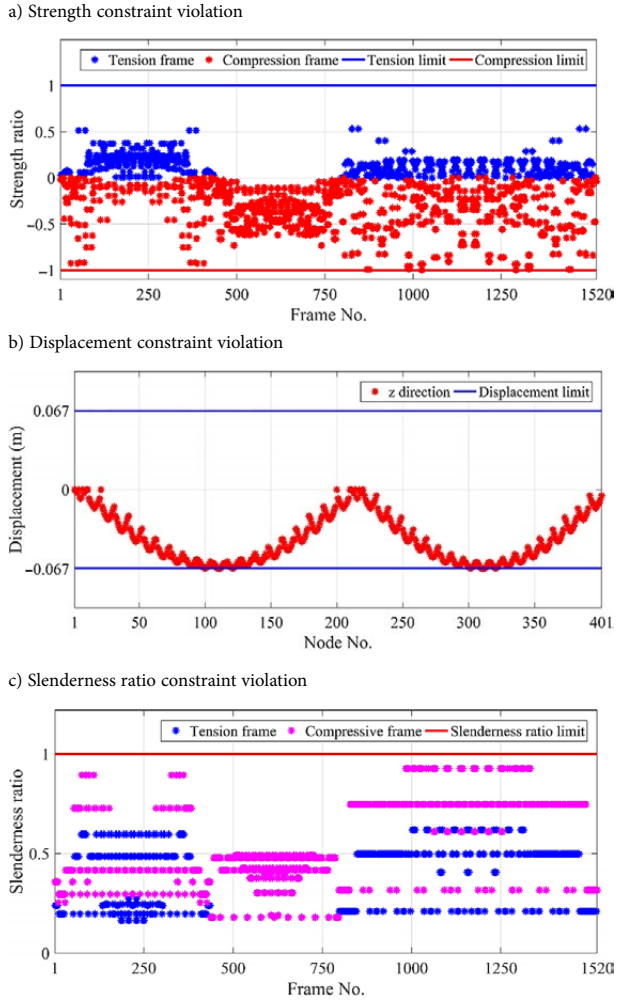

Figure 17. Constraint violation check in the 1520-bar double-layer grid problem

structural analyses (NSA) than CBO and MDVC-UVPS, the solutions generated by CBO or MDVC-UVPS were heavier than those obtained using SO and ESOs. ESO[2] produced designs that were 3.68% and 1.70% lighter on average than SO and ESO[1], respectively. The solution found by ESO[2] satisfied all strength, displacement, and slenderness ratio constraints, as depicted in Figure 17. Consequently, ESO[2] effectively solved this problem.

4.6. 1536-bar double-layer barrel vault

In the final example, the optimization focuses on sizing a double-layer barrel vault with 1536 bars, 413 joints, a span of 40 m, a length of 50 m, and 35 independent variable groups (Kaveh & Ilchi Ghazaan, 2018b). Figure 18 and Table 7 provide detailed geometric information and member groupings. The material properties include a modulus of elasticity (E) of 30,450 ksi, yield stress of steel (F_y) at 58.0 ksi, and material density (ρ) of 0.288lb/in^3. Design variables pertain to the cross-sectional areas of the bar elements, as listed in Table 4. The structure's top layer is fixed at its external edges, with all joints of the top layer subjected to concentrated vertical loads of 5 kips. Nodal displacements are constrained within ±0.1969 in. (5 mm) in all directions.

Table 6. Weight design comparison in 1520-bar double-layer grid problem

Design variables a_i	CBO (Kaveh & Ilchi Ghazaan, 2018b)	ECBO (Kaveh & Ilchi Ghazaan, 2018b)	VPS (Kaveh & Ilchi Ghazaan, 2018b)	MDVC-UVPS (Kaveh & Ilchi Ghazaan, 2018b)	SO	ESO[1]	ESO[2]
					(Present study)		
a_1	EST 5	DEST 5	EST 5	ST 6	EST 6	EST 6	EST 6
a_2	ST 5	EST 5	ST 5	DEST 3	ST 5	ST 5	ST 5
a_3	ST 2½	ST 2½	EST 2½	ST 2½	ST 3½	ST 3½	ST 3½
a_4	EST 1½	EST 1½	EST 1½	ST 2½	EST 2	EST 2	EST 2
a_5	ST 1½	ST 2	ST 1½	ST 2½	ST 1½	ST 1½	ST 1½
a_6	EST 1¼	ST 2	ST 1¼	EST 2½	EST 2	EST 2	EST 2
a_7	ST 3	EST 2½	ST 8	EST 2½	EST 2	EST 2	EST 2
a_8	ST 4	EST 3	EST 5	EST 2½	ST 4	ST 4	ST 4
a_9	ST 5	DEST 2½	EST 3½	EST 4	ST 5	ST 5	ST 5
a_{10}	EST 10	EST 4	DEST 3	DEST 3	ST 6	ST 6	ST 6
a_{11}	ST 6	ST 6	EST 5	DEST 3	DEST 4	DEST 4	DEST 4
a_{12}	DEST 6	DEST 5	EST 10	DEST 6	ST 12	ST 12	ST 12
a_{13}	EST 4	ST 2½	ST 2½	ST 3½	ST 3	ST 3	ST 3
a_{14}	ST 3½	ST 3½	ST 2½	ST 3½	ST 3½	ST 3½	ST 3½
a_{15}	ST 3	ST 4	EST 2½	ST 3½	ST 3	ST 3	ST 3
a_{16}	ST 4	ST 5	ST 3	EST 3	EST 3	EST 3	EST 3
a_{17}	EST 10	DEST 2½	ST 4	DEST 2½	EST 3½	EST 3½	EST 3½
a_{18}	EST 3	EST 3½	DEST 3	DEST 3	EST 4	EST 4	EST 4
a_{19}	DEST 2½	ST 6	DEST 3	EST 5	EST 5	EST 5	EST 5
a_{20}	EST 4	DEST 5	DEST 4	ST 6	DEST 4	DEST 4	DEST 4
a_{21}	DEST 6	EST 6	EST 10	ST 10	ST 8	ST 8	ST 8
a_{22}	EST 5	EST 5	ST 6	ST 6	ST 6	ST 6	ST 6
a_{23}	EST 3½	ST 3	ST 2½	ST 2½	ST 2½	ST 2½	ST 2½
a_{24}	ST 3	ST 2½	EST 2½	ST 2½	ST 2½	ST 2½	ST 2½
a_{25}	ST 2½	ST 2½	ST 3	ST 2½	ST 2½	ST 2½	ST 2½
a_{26}	EST 1½	EST 1½	EST 1½	EST 2	ST 2½	ST 2½	ST 2½
a_{27}	ST 3	EST 2	ST 2½	EST 2	EST 2	EST 2	EST 2
a_{28}	ST 2½	EST 2	ST 3	EST 2	EST 2	EST 2	EST 2
a_{29}	ST 2½	ST 2½	ST 2½	EST 1½	ST 3	EST 2½	ST 3
a_{30}	ST 3	EST 1½	ST 1½	EST 2	EST 2	EST 2	EST 2
a_{31}	ST 3½	EST 2	ST 4	ST2 ½	ST 2½	EST 2	EST 2
Weight (kg)	93,174	82,254	82,357	79,571	79,426	79,423	79,421
Average (kg)	97,823	90,752	89,607	85,398	84,681	82,979	81,565
Std. (kg)	9,226	5,995	5,188	4,407	4,050	4,012	3,672
NSA	4,360	18,000	12,120	3,142	10,000	10,000	10,000

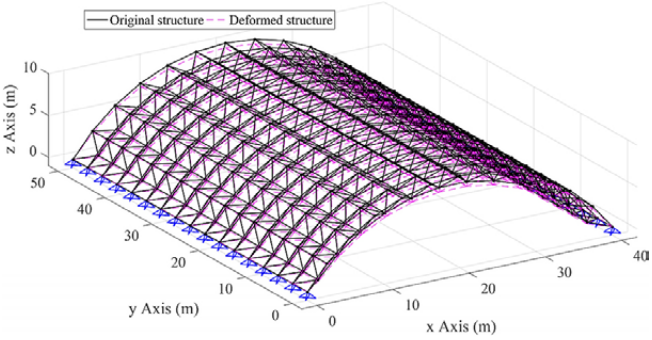

Figure 18. 3D view of 1536-bar double-layer barrel vault

Allowable tensile and compressive stresses adhere to AISC-ASD specifications (American Institute of Steel Construction, 1989; Kaveh & Ilchi Ghazaan, 2018b):

$$\begin{cases} \sigma_i^+ = 0.6F_y & \sigma_i \geq 0 \\ \sigma_i^- & \sigma_i < 0 \end{cases}, \tag{51}$$

where σ_i^- depends on the slenderness ratio:

$$\sigma_i^- = \begin{cases} \dfrac{\left[\left(1-\dfrac{\lambda_i^2}{2C_c^2}\right)F_y\right]}{\dfrac{5}{3}+\dfrac{3\lambda_i}{8C_c}-\dfrac{\lambda_i^3}{8C_c^3}} & \text{for } \lambda_i < C_c \\ \dfrac{12\pi^2 E}{23\lambda_i^2} & \text{for } \lambda_i \geq C_c \end{cases}, \tag{52}$$

C_c is the slenderness ratio that divides the elastic from the inelastic buckling regions, which is given by:

$$C_c = \sqrt{\frac{2\pi^2 E}{F_y}}. \quad (53)$$

For the i^{th} member, λ_i is the slenderness ratio $\left(\lambda_i = \frac{KL_i}{r_i}\right)$; K is the effective length factor (K = 1); L_i is the length; r_i is the minimum radius of gyration, and A_i is the cross-sectional area.

AISC-ASD recommends a maximum slenderness ratio of 300 and 200 for tension and compression members, respectively (American Institute of Steel Construction, 1989; Kaveh & Ilchi Ghazaan, 2018b).

To tackle this problem, a snake population of 50 was employed, and a total of 10,000 structural analyses were conducted. Table 7 displays the optimal designs achieved through various optimization algorithms. The weight of the best configuration obtained by SO is 122,843 lb, while ESO^1 and ESO^2 resulted in 121,920 lb and 121,840 lb, respectively, marking the lowest values among all compared methods.

Table 7. Weight design comparison in 1536-bar barrel vault problem

Design variables a_i (in^2)	ABC (Kaveh & Mahjoubi, 2018)	CS (Kaveh & Mahjoubi, 2018)	PSO (Kaveh & Mahjoubi, 2018)	LPOA (Kaveh & Mahjoubi, 2018)	MDVC-UVPS (Kaveh & Ilchi Ghazaan, 2018b)	SO	ESO^1	ESO^2
						(Present study)		
a_1	EST 6	ST 6	EST 6	DEST 4	DEST 4	EST 5	EST 6	EST 5
a_2	DEST 2½	EST 4	EST 3	EST 3	ST 4	EST 4	EST 4	EST 4
a_3	ST 4	EST 3½	EST 3	EST 3	EST 3½	ST 5	EST 3½	ST 5
a_4	EST 4	EST 4	ST 6	ST 6	EST 5	EST 5	EST 5	EST 5
a_5	EST 5	DEST 3	ST 6	DEST 3	EST 5	DEST 4	EST 5	DEST 4
a_6	DEST 3	EST 6	EST 6	DEST 3	EST 5	DEST 4	ST 8	DEST 4
a_7	EST ½	EST ¾	EST ½	ST ½	ST ½	ST 1½	ST 1½	ST 1½
a_8	ST 3	DEST 2	EST 1½	EST 1½	ST 1½	ST 1½	ST 1½	ST 1½
a_9	ST 3½	ST 3	EST 2½	ST 1½	ST 1½	ST 1½	ST 1½	ST 1½
a_{10}	EST 2	DEST 2	ST 3	EST 1½	ST 1¼	ST 1	ST 1	ST 1
a_{11}	ST 3½	ST 2½	ST 3	EST 2½	ST 2½	EST 1¼	ST 1¼	EST 1¼
a_{12}	EST 3½	ST 3½	EST 2	ST 1½	EST 2½	ST 2	EST 1½	ST 2
a_{13}	EST 2	ST 5	ST 3½	EST 1½	EST 3	EST 2½	EST 2	ST 3
a_{14}	ST 5	ST 6	EST 3½	EST 4	DEST 3	ST 5	ST 6	ST 5
a_{15}	EST 5	EST 5	DEST 4	ST 6	ST 6	EST 5	EST 5	EST 5
a_{16}	ST 5	EST 4	ST 5	DEST 3	ST 5	DEST 3	ST 6	DEST 3
a_{17}	ST 5	EST 3½	EST 3	ST 5	DEST 2½	ST 4	EST 3½	EST 3½
a_{18}	ST 3½	ST 3	ST 3	EST 2	EST 2	ST 2½	ST 2½	ST 2½
a_{19}	ST 3½	EST 3	ST 3½	EST 2	EST 2	EST 2	ST 2½	EST 2
a_{20}	EST 2½	DEST 2	EST 1½	ST 1½	ST 1½	ST 1½	ST 1½	ST 1½
a_{21}	DEST 2½	ST 3½	EST 2	ST 1½	ST 1½	ST 1	ST 1	ST 1
a_{22}	EST 2	EST 3½	EST 2	EST 1½	ST 1½	ST 1½	ST 1½	ST 1½
a_{23}	DEST 2½	ST 5	EST 2	ST 1½	EST 2	ST 1½	ST 1½	ST 1½
a_{24}	ST 3½	EST 2	ST 2½	EST 1½	ST 1½	ST 1½	ST 1½	ST 1½
a_{25}	EST 2½	EST 3½	EST 2	EST 1½	EST 2	ST 1½	ST 2	ST 1½
a_{26}	EST 3	DEST 2	ST 3	ST 4	ST 2½	ST 2½	ST 2½	ST 2½
a_{27}	ST 2½	ST 1½	ST 1½	EST 1½	EST 1½	ST 1½	ST 2	ST 1½
a_{28}	ST 1½	EST 2	EST 1½	ST 1½	ST 1½	ST 1½	ST 1¼	ST 1½
a_{29}	ST 4	EST 2	EST 1½	ST 2½	ST 1½	ST 1½	ST 1½	ST 1½
a_{30}	EST 1½	EST 2½	ST 1½	EST 1½	EST 1½	ST 1½	ST 1¼	ST 1½
a_{31}	EST 2½	ST 4	EST 1½	ST 2½	EST 1½	ST 2½	ST 2	ST 2½
a_{32}	EST 1½	EST 1½	ST 1½	EST 1½	ST 2	ST 1½	ST 2	ST 1½
a_{33}	EST 2½	ST 2½	EST 2	EST 1½	EST 1½	ST 2	ST 1¼	ST 2
a_{34}	EST 1½	EST 2½	EST 1½	ST 1½	ST 2	ST 1½	ST 1¼	ST1½
a_{35}	EST 2	EST 2	EST 1½	ST 1½	ST 2	ST 1½	ST 1¼	ST 1½
Weight (lb)	158,936	161,126	133,331	125,665	122,852	122,843	121,920	121,840
Average (lb)	167,500.58	170,331.45	147,812.18	135,156.90	146,229	133,556	126,630	124,600
Std. (lb)	4,863.51	2,902.44	15,645.08	11,208.05	14,552	8,632	6,884	5,144
NSA	15,000	15,000	15,000	15,000	4,762	10,000	10,000	10,000

The average optimized weight with SO was 133,556 lb, whereas ESO[1] and ESO[2] produced 126,630 lb and 124,600 lb, respectively.

The weight obtained through this method is lower than that achieved using other methods. While SO and ESOs required 10,000 structural analyses to find the optimal solution, ABC, CS, PSO, and LPOA required 15,000 structural analyses, and MDVC-UVPS needed 4,762. Despite MDVC-UVPS employing a lower NSA than ESOs to solve this problem, it resulted in a heavier solution than ESOs. Moreover, the solution provided by ESO[2] fulfills all strength, displacement, and slenderness ratio constraints, as depicted in Figure 19. Consequently, ESOs effectively solve this problem, with ESO[2] finding an average weight of 6.71% and 1.60% lighter than SO and ESO[1], respectively.

5. Discussion

This subsection discusses five critical aspects of the robust Enhanced Snake Optimizers (ESOs): exploring the search space, exploiting promising solutions, converging to the optimal solution, insights derived, and user convenience.

5.1. Exploratory capacity

Exploration, or diversification, involves exhaustively searching the solution space to identify promising areas. An algorithm's ability to locate the global optimum site signifies its strong exploration capability (Alorf, 2023). Multimodal functions, which feature numerous local optima, serve as effective benchmarks for evaluating the exploration capabilities of optimization algorithms.

Based on the optimization results of IEEE CEC 2022 functions (F2–F5) for dimensions 10 and 20, as illustrated in Figures 6 and 7, the p-values from Wilcoxon's rank-sum tests indicate that the ESO[1] optimizer outperformed the ABC algorithm (5/8), CA (7/8), DE (6/8), FA (6/8), GA (7/8), GWO (8/8), HS (7/8), JAYA (8/8), PSO (5/8), SA (7/8), SO (5/8), SOS (5/8), and TLBO (6/8). Similarly, the ESO[2] optimizer outperformed the ABC algorithm (5/8), CA (7/8), DE (7/8), FA (8/8), GA (7/8), GWO (8/8), HS (8/8), JAYA (8/8), PSO (6/8), SA (7/8), SO (5/8), SOS (6/8), and TLBO (7/8). Consequently, the proposed ESOs outperform the other selected optimization algorithms in exploring the search space.

5.2. Exploitation capability

Exploitation, or intensification, involves searching within a promising region to find the best solution. An algorithm with this capability seeks the optimum within a potential area (Alorf, 2023). Consequently, unimodal functions can assess an optimization algorithm's exploitation capability. In this study, IEEE CEC 2022 functions (F1 and F6-F8) were utilized to evaluate the exploitation capabilities of the ESOs optimizer and 13 other optimization algorithms, as depicted in Figures 6 and 7.

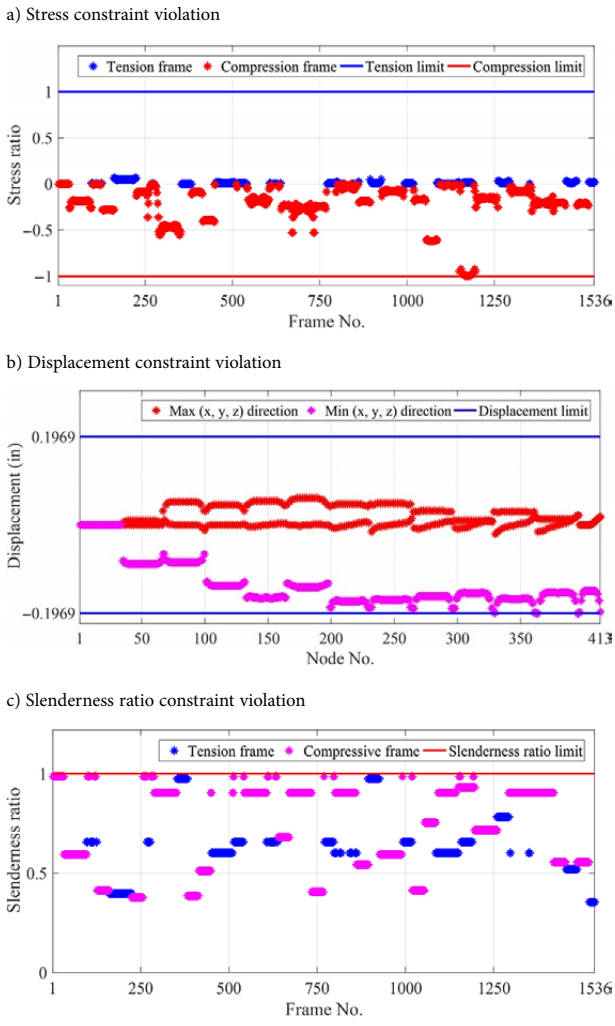

Figure 19. Constraint violation check in the 1536-bar double-layer barrel vault

The p-values obtained from Wilcoxon's rank-sum tests indicated that the ESO[1] optimizer was superior or equivalent to the ABC algorithm (7/8), CA (7/8), DE (5/8), FA (7/8), GA (6/8), GWO (8/8), HS (4/8), JAYA (8/8), PSO (5/8), SA (8/8), SO (6/8), SOS (4/8), and TLBO (4/8). Additionally, the ESO[2] optimizer outperformed the ABC algorithm (8/8), CA (8/8), DE (8/8), FA (8/8), GA (8/8), GWO (8/8), HS (8/8), JAYA (8/8), PSO (7/8), SA (8/8), SO (8/8), SOS (6/8), and TLBO (7/8). Therefore, the proposed ESOs perform better than the other selected optimization algorithms in exploring the search space. Specifically, the ESOs exhibit excellent exploitation capability, with ESO[2] displaying exceptional exploitation abilities compared to ESO[1]. These outcomes are attributed to the Lévy flight implemented in step 2. Moreover, ESO[2] outperforms ESO[1] in finding optimal values in mathematical tests, as illustrated in Figures 6 and 7.

5.3. Convergence efficiency

In the context of the IEEE CEC 2022 functions, convergence represents the state where the characteristics of all solutions in population-based metaheuristics become

similar, indicating that the solutions have ceased to improve. An algorithm lacking this ability might need help effectively exploiting promising regions (Alorf, 2023). According to the p-values from Wilcoxon's rank-sum tests, the ESO[1] optimizer outperformed ABC, CA, DE, FA, GA, GWO, HS, JAYA, PSO, SA, SO, SOS, and TLBO in 15/24, 16/24, 13/24, 15/24, 21/24, 22/24, 16/24, 24/24, 16/24, 19/24, 18/24, 13/24, and 15/24 functions, respectively. Similarly, ESO[2] demonstrated superiority in 16/24, 17/24, 18/24, 22/24, 23/24, 23/24, 23/24, 24/24, 20/24, 19/24, 20/24, 17/24, and 22/24 functions. The ESOs effectively identified optimal values for unimodal, multimodal, separate, non-separable, and hybrid composite functions. Remarkably, concerning hybrid composite functions (F6-F12), ESO[1] outperformed ABC, CA, DE, FA, GA, GWO, HS, JAYA, PSO, SA, SO, SOS, and TLBO in 8/14, 7/14, 5/14, 7/14, 12/14, 12/14, 7/14, 14/14, 10/14, 10/14, 12/14, 7/14, and 8/14 functions, respectively. Similarly, ESO[2] exhibited superiority in 9/14, 8/14, 9/14, 12/14, 14/14, 13/14, 13/14, 14/14, 13/14, 10/14, 13/14, 10/14, and 14/14 functions. Given that all optimizers were set to solve mathematical functions within a similar timeframe (5s), these results highlight the higher convergence abilities of ESOs compared to the comparison algorithms.

5.4. Insights derived

The performance of the proposed ESOs was evaluated by solving five structural weight design problems: the 47-bar power transmission tower, 72-bar tower, 672-bar double-layer grid, 1520-bar double-layer grid, and 1536-bar double-layer barrel vault, all of which encompassed multiple constraints. The results obtained by applying ESOs were compared with those documented in existing literature. ESO[1] exhibited favorable outcomes, while ESO[2] excelled. ESO[2] demonstrated significantly superior opti-

Table 8. Comparison of parameters setting for SO vs. ESOs

No.	Stages of algorithm	SO	ESO[1]	ESO[2]
1.	**Initialization**			
	Equation	Eqn (1)	Eqn (21)	
	Specific parameters to operate	Uniform distribution	Logistic map	
2.	**Specific definition**			
2.1	***Temperature***			
	Equation	Eqn (4)	Eqn (23)	
	Specific parameters to operate	Decrease over time	Randomly decrease over time	
2.2	***Food quantity***			
	Equation	Eqn (5)	Eqn (24)	
	Specific parameters to operate	Increase over time ☑ Constant C_1= 0.5	Randomly increase over time –	
3.	**Exploration phase** (Step 1 and 5)			
3.1	***Step 1:*** *The snakes search for food*			
	Equation	Eqns (6) and (8)	Eqns (25) and (26)	
	Specific parameters to operate	☑ Threshold$_1$ = 0.25 Constant C_2 = 0.05	☑ Coefficient of motion α=0.1	
3.2	***Step 5:*** *If an egg hatches, the worst male and female are replaced*			
	Equation	Eqns (19) and (20)		
	Specific parameters to operate	–	–	
4.	**Exploitation phase** (Steps 2, 3 and 4)			
4.1	***Step 2:*** *The snakes are hot and move toward food*			
	Equation	Eqn (10)	Eqn (27)	Eqn (28)
	Specific parameters to operate	☑ Threshold$_2$ = 0.6 Constant C_3 = 2	–	Lévy flight to enhance the exploitation phase
4.2	***Step 3:*** *The snakes are in fight mode*			
	Equation	Eqn (11) and (12)	Eqns (29) and (30)	
	Specific parameters to operate	☑ Threshold$_3$ = 0.6 Constant C_3 = 2	–	
4.3	***Step 4:*** *The snakes are in mating mode*			
	Equation	Eqns (15) and (16)	Eqns (31) and (32)	
	Specific parameters to operate	☑ Threshold$_3$ = 0.6 Constant C_3 = 2	–	
5.	**Checking terminating conditions**			
	Specific parameters to operate	–	–	–

mization results, highlighting its stability and robustness in addressing these complex structural issues. Specifically, ESO^2 generated solutions that were, on average, 1.16%, 0.70%, 2.34%, 3.68%, and 6.71% lighter than those produced by SO, and 0.79%, 0.54%, 1.28%, 1.70%, and 1.60% lighter than those produced by ESO^1 for these respective problems. Consequently, ESO^2 can be regarded as a viable alternative to other metaheuristic algorithms, particularly for tackling high-dimensional challenges.

5.5. User convenience

This study introduces enhanced versions of the snake optimizer, namely Enhanced Snake Optimizer 1 (ESO^1) and Enhanced Snake Optimizer 2 (ESO^2). ESO^1 incorporates a logistic map to generate the initial population, while ESO^2 builds upon ESO^1 by employing a Lévy flight to simulate snake movement during the food search. ESO^1 and ESO^2 require only specific parameter settings (the motion coefficient parameter (α)), unlike the original Snake Optimizer (SO), which necessitates six particular parameters (C_1, C_2, C_3 $Threshold_1$, $Threshold_2$ and $Threshold_3$). Table 8 illustrates the differences between the SO algorithm and the ESO algorithms. The exact parameters of ESOs are determined through sensitivity analysis when solving mathematical functions ($\alpha = 0.1$). Therefore, users need to configure fewer parameters for ESOs compared to the original SO algorithm.

Conclusions

This study developed two new enhanced versions of the snake optimizer, ESO^1, and ESO^2, which require setting only one specific parameter for the operator (the motion coefficient, α), compared to the six parameters of the SO algorithm (C_1, C_2, C_3 $Threshold_1$, $Threshold_2$ and $Threshold_3$). Thus, the proposed ESOs (ESO^1 and ESO^2) are easier to use than the original SO. In ESO^1, a logistic map is used to generate diverse initial populations, and the optimal value of the motion coefficient parameter (α) is determined. Notably, Lévy flight is employed to simulate the movement of snakes in ESO^2, enhancing the exploitation ability of ESO^1.

The Enhanced Snake Optimizers (ESOs) were rigorously tested against 13 established optimization algorithms using 24 mathematical benchmark functions (CEC-2022). The results of Wilcoxon rank sum tests comparing the ESOs with these algorithms demonstrate that both ESO^1 and ESO^2 significantly outperform ABC, CA, DE, FA, GA, GWO, HS, JAYA, PSO, SA, SO, SOS, and TLBO in the mathematical tests. Furthermore, ESO^2 proves superior to ESO^1 in identifying optimal values within these mathematical tests.

The ESOs were then applied with the finite element method to solve five distinct structural design problems. The ESOs efficiently optimized all structural weights, notably discovering solutions lighter than those documented in the existing literature. The performance of ESO^2 is particularly noteworthy, which consistently found solutions lighter than those obtained by the standard Snake Optimizer (SO) and even more lightweight than those achieved by ESO^1 for these specific problems. Consequently, ESO^2 emerges as a potent computational technique for addressing optimization challenges in structural engineering.

Future research should concentrate on delving deeper into the performance of ESO^2 with additional structural problems and its application in various machine learning methods across diverse scenarios. The development of multiple objective-enhanced snake optimizer (MOESO) or many objective-enhanced snake optimizer (MAESO) should also be pursued. Creating a user-friendly platform that integrates ESO^2 with the Finite Element Method (FEM) will empower designers to conveniently utilize the system for solving engineering problems.

Data availability

The datasets, codes, and replication of results that are generated and analyzed in this study are available from the corresponding author upon reasonable request.

Acknowledgements

The authors thank the National Science and Technology Council (110-2221-E-011-080-MY3), Taiwan, for financially supporting this research.

Disclosure statement

We declare no known conflict of interest associated with this publication and that no significant financial support for this work could have influenced its outcome.

References

American Institute of Steel Construction. (1989). *Manual of steel construction: allowable stress design.* Chicago, USA.

American Institute of Steel Construction. (1994). *Manual of steel construction load resistance factor design*. Chicago, USA.

Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H. (2021). The arithmetic optimization algorithm. *Computer Methods in Applied Mechanics and Engineering*, *376*, 113609. https://doi.org/10.1016/j.cma.2020.113609

Alorf, A. (2023). A survey of recently developed metaheuristics and their comparative analysis. *Engineering Applications of Artificial Intelligence*, *117*, 105622. https://doi.org/10.1016/j.engappai.2022.105622

Arafa, M., Sallam, E. A., & Fahmy, M. M. (2014). An enhanced differential evolution optimization algorithm. In *2014 Fourth International Conference on Digital Information and Communication Technology and its Applications* (*DICTAP*), Bangkok, Thailand. https://doi.org/10.1109/DICTAP.2014.6821685

Askari, Q., Younas, I. & Saeed, M. (2020). Political Optimizer: A novel socio-inspired meta-heuristic for global optimization. *Knowledge-Based Systems*, *195*, 105709. https://doi.org/10.1016/j.knosys.2020.105709

Aydogdu, I., Carbas, S., & Akin, A. (2017). Effect of Levy Flight on the discrete optimum design of steel skeletal structures using metaheuristics. *Steel and Composite Structures*, *24*(1), 93–112. https://doi.org/10.12989/scs.2017.24.1.093

Beyer, H.-G., & Schwefel, H.-P. (2002). Evolution strategies – A comprehensive introduction. *Natural Computing*, *1*(1), 3–52. https://doi.org/10.1023/A:1015059928466

Cheng, M.-Y., & Prayogo, D. (2014). Symbiotic Organisms Search: A new metaheuristic optimization algorithm. *Computers & Structures*, *139*, 98–112. https://doi.org/10.1016/j.compstruc.2014.03.007

Cheng, M.-Y., & Cao, M.-T. (2016). Estimating strength of rubberized concrete using evolutionary multivariate adaptive regression splines. *Journal of Civil Engineering and Management*, *22*(5), 711–720. https://doi.org/10.3846/13923730.2014.897989

Chou, J.-S., & Ngo, N.-T. (2017). Modified firefly algorithm for multidimensional optimization in structural design problems. *Structural and Multidisciplinary Optimization*, *55*(6), 2013–2028. https://doi.org/10.1007/s00158-016-1624-x

Chou, J.-S., & Nguyen, N.-M. (2020). FBI inspired meta-optimization. *Applied Soft Computing*, *93*, 106339. https://doi.org/10.1016/j.asoc.2020.106339

Chou, J.-S., & Truong, D.-N.. (2021). A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Applied Mathematics and Computation*, *389*, 125535. https://doi.org/10.1016/j.amc.2020.125535

da Silva, E. K., Barbosa, H. J. C., & Lemonge, A. C. C. (2011). An adaptive constraint handling technique for differential evolution with dynamic use of variants in engineering optimization. *Optimization and Engineering*, *12*(1), 31–54. https://doi.org/10.1007/s11081-010-9114-2

Das, B., Mukherjee, V., & Das, D. (2020). Student psychology based optimization algorithm: A new population based optimization algorithm for solving optimization problems. *Advances in Engineering Software*, *146*, 102804. https://doi.org/10.1016/j.advengsoft.2020.102804

Degertekin, S. O., Lamberti, L., & Ugur, I. B. (2018). Sizing, layout and topology design optimization of truss structures using the Jaya algorithm. *Applied Soft Computing*, *70*, 903–928. https://doi.org/10.1016/j.asoc.2017.10.001

Degertekin, S. O., Lamberti, L., & Ugur, I. B. (2019). Discrete sizing/layout/topology optimization of truss structures with an advanced Jaya algorithm. *Applied Soft Computing*, *79*, 363–390. https://doi.org/10.1016/j.asoc.2019.03.058

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, *1*(1), 3–18. https://doi.org/10.1016/j.swevo.2011.02.002

Do, D. T. T., Lee, S., & Lee, J. (2016). A modified differential evolution algorithm for tensegrity structures. *Composite Structures*, *158*, 11–19. https://doi.org/10.1016/j.compstruct.2016.08.039

Doğan, B., & Ölmez, T. (2015). A new metaheuristic for numerical function optimization: Vortex Search algorithm. *Information Sciences*, *293*, 125–145. https://doi.org/10.1016/j.ins.2014.08.053

Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine*, *1*(4), 28–39. https://doi.org/10.1109/MCI.2006.329691

Erbatur, F., Hasançebi, O., Tütüncü, İ., & Kılıç, H. (2000). Optimal design of planar and space structures with genetic algorithms. *Computers & Structures*, *75*(2), 209–224. https://doi.org/10.1016/S0045-7949(99)00084-X

Erdal, F., Doğan, E., & Saka, M. P. (2011). Optimum design of cellular beams using harmony search and particle swarm optimizers. *Journal of Constructional Steel Research*, *67*(2), 237–247. https://doi.org/10.1016/j.jcsr.2010.07.014

Erol, O. K., & Eksin, I. (2006). A new optimization method: Big Bang–Big Crunch. *Advances in Engineering Software*, *37*(2), 106–111. https://doi.org/10.1016/j.advengsoft.2005.04.005

Es-Haghi, M. S., Salehi, A., & Strauss, A. (2022). Enhanced teacher-learning based algorithm in real size structural optimization. *Journal of Civil Engineering and Management*, *28*(4), 292–304. https://doi.org/10.3846/jcem.2022.16387

Fesanghary, M., Mahdavi, M., Minary-Jolandan, M., & Alizadeh, Y. (2008). Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems. *Computer Methods in Applied Mechanics and Engineering*, *197*(33), 3080–3091. https://doi.org/10.1016/j.cma.2008.02.006

Ficarella, E., Lamberti, L., & Degertekin, S. O. (2021). Comparison of three novel hybrid metaheuristic algorithms for structural optimization problems. *Computers & Structures*, *244*, 106395. https://doi.org/10.1016/j.compstruc.2020.106395

Gandomi, A. H., Yang, X.-S., & Alavi, A. H. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with Computers*, *29*(1), 17–35. https://doi.org/10.1007/s00366-011-0241-y

Gao, Q., Xu, H., & Li, A. (2022). The analysis of commodity demand predication in supply chain network based on particle swarm optimization algorithm. *Journal of Computational and Applied Mathematics*, *400*, 113760. https://doi.org/10.1016/j.cam.2021.113760

Hasançebi, O., Çarbaş, S., Doğan, E., Erdal, F., & Saka, M. P. (2009). Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures. *Computers & Structures*, *87*(5), 284–302. https://doi.org/10.1016/j.compstruc.2009.01.002

Hasançebi, O., Çarbaş, S., Doğan, E., Erdal, F., & Saka, M. P. (2010). Comparison of non-deterministic search techniques in the optimum design of real size steel frames. *Computers & Structures*, *88*(17), 1033–1048. https://doi.org/10.1016/j.compstruc.2010.06.006

Hashim, F. A., & Hussien, A. G. (2022). Snake Optimizer: A novel meta-heuristic optimization algorithm. *Knowledge-Based Systems*, *242*, 108320. https://doi.org/10.1016/j.knosys.2022.108320

Holland, J. H. (1992). Genetic algorithms. *Scientific American*, *267*(1), 66–73. https://doi.org/10.1038/scientificamerican0792-66

Hu, G., Zhong, J., Du, B., & Wei, G. (2022). An enhanced hybrid arithmetic optimization algorithm for engineering applications. *Computer Methods in Applied Mechanics and Engineering*, *394*, 114901. https://doi.org/10.1016/j.cma.2022.114901

Huan, T. T., Kulkarni, A. J., Kanesan, J., Huang, C. J., & Abraham, A. (2017). Ideology algorithm: a socio-inspired optimization methodology. *Neural Computing and Applications*, *28*(1), 845–876. https://doi.org/10.1007/s00521-016-2379-4

Hwang, S.-F., & He, R.-S. (2006). Improving real-parameter genetic algorithm with simulated annealing for engineering problems. *Advances in Engineering Software*, *37*(6), 406–418. https://doi.org/10.1016/j.advengsoft.2005.08.002

Ji, A., Xue, X., Wang, Y., Luo, X., & Zhang, M. (2020). An integrated multi-objectives optimization approach on modelling pavement maintenance strategies for pavement sustainability. *Journal of Civil Engineering and Management*, *26*(8), 717–732. https://doi.org/10.3846/jcem.2020.13751

Joshi, H., & Arora, S. (2017). Enhanced grey wolf optimization algorithm for global optimization. *Fundamenta Informaticae*, *153*, 235–264. https://doi.org/10.3233/FI-2017-1539

Kamoona, A. M., Patra, J. C., & Stojcevski, A. (2018). An enhanced cuckoo search algorithm for solving optimization problems. In *2018 IEEE Congress on Evolutionary Computation* (*CEC*), Rio de Janeiro, Brazil. https://doi.org/10.1109/CEC.2018.8477784

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, *39*(3), 459–471. https://doi.org/10.1007/s10898-007-9149-x

Kaveh, A., & Talatahari, S. (2010). A novel heuristic optimization method: charged system search. *Acta Mechanica*, *213*(3), 267–289. https://doi.org/10.1007/s00707-009-0270-4

Kaveh, A., & Khayatazad, M. (2012). A new meta-heuristic method: Ray Optimization. *Computers* & *Structures*, *112–113*, 283–294. https://doi.org/10.1016/j.compstruc.2012.09.003

Kaveh, A., & Mahdavi, V. R. (2014). Colliding Bodies Optimization method for optimum discrete design of truss structures. *Computers* & *Structures*, *139*, 43–53. https://doi.org/10.1016/j.compstruc.2014.04.006

Kaveh, A., & Bakhshpoori, T. (2016). Water Evaporation Optimization: A novel physically inspired optimization algorithm. *Computers* & *Structures*, *167*, 69–85. https://doi.org/10.1016/j.compstruc.2016.01.008

Kaveh, A., & Ilchi Ghazaan, M. (2018a). *Meta-heuristic algorithms for optimal design of real-size structures*. Springer, Cham. https://doi.org/10.1007/978-3-319-78780-0

Kaveh, A., & Ilchi Ghazaan, M. (2018b). Optimal design of double-layer grids. In *Meta-heuristic algorithms for optimal design of real-size structures* (pp. 65–83). Springer, Cham. https://doi.org/10.1007/978-3-319-78780-0_5

Kaveh, A., & Ilchi Ghazaan, M. (2018c). Optimal design of steel lattice transmission line towers. In *Meta-heuristic algorithms for optimal design of real-size structures* (pp. 123–137). Springer, Cham.https://doi.org/10.1007/978-3-319-78780-0_8

Kaveh, A., & Mahjoubi, S. (2018). Optimum design of double-layer barrel vaults by lion pride optimization algorithm and a comparative study. *Structures*, *13*, 213–229. https://doi.org/10.1016/j.istruc.2018.01.002

Kaveh, A., Hamedani, K. B., & Kamalinejad, M. (2021). An enhanced Forensic-Based Investigation algorithm and its application to optimal design of frequency-constrained dome structures. *Computers* & *Structures*, *256*, 106643. https://doi.org/10.1016/j.compstruc.2021.106643

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 – International Conference on Neural Networks*. https://doi.org/10.1109/ICNN.1995.488968

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*, 671–680. https://doi.org/10.1126/science.220.4598.671

Kumar, M., Kulkarni, A. J., & Satapathy, S. C. (2018). Socio evolution & learning optimization algorithm: A socio-inspired optimization methodology. *Future Generation Computer Systems*, *81*, 252–272. https://doi.org/10.1016/j.future.2017.10.052

Kumar, A., Price, K. V., Mohamed, A. W., Hadi, A. A., & Suganthan, P. N. (2021). *Problem definitions and evaluation criteria for the CEC 2022 special session and competition on single objective bound constrained numerical optimization* (Technical report). https://github.com/P-N-Suganthan

Lam, A. Y. S., & Li, V. O. K. (2010). Chemical-reaction-inspired metaheuristic for optimization. *IEEE Transactions on Evolutionary Computation*, *14*(3), 381–399. https://doi.org/10.1109/TEVC.2009.2033580

Lee, K. S., Geem, Z. W., Lee, S.-h., & Bae, K.-w. (2005). The harmony search heuristic algorithm for discrete structural optimization. *Engineering Optimization*, *37*(7), 663–684. https://doi.org/10.1080/03052150500211895

Lemonge, A. C. C., & Barbosa, H. J. C. (2004). An adaptive penalty scheme for genetic algorithms in structural optimization. *International Journal for Numerical Methods in Engineering*, *59*(5), 703–736. https://doi.org/10.1002/nme.899

Li, M., Zhao, H., Weng, X., & Han, T. (2016). Cognitive behavior optimization algorithm for solving optimization problems. *Applied Soft Computing*, *39*, 199–222. https://doi.org/10.1016/j.asoc.2015.11.015

Li, Y,. & Han, M. (2020). Improved fruit fly algorithm on structural optimization. *Brain Informatics*, *7*(1), 1. https://doi.org/10.1186/s40708-020-0102-9

Logan, D. L. (2016). *A first course in the finite element method*. Cengage Learning.

May, R. (1976). Simple mathematical models with very complicated dynamics. *Nature*, *261*, 459–467. https://doi.org/10.1038/261459a0

Makiabadi, M. H., & Maheri, M. R. (2021). An enhanced symbiotic organism search algorithm (ESOS) for the sizing design of pin connected structures. *Iranian Journal of Science and Technology*, *Transactions of Civil Engineering*, *45*(3), 1371–1396. https://doi.org/10.1007/s40996-020-00471-0

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Mousavirad, S. J., & Ebrahimpour-Komleh, H. (2017). Human mental search: A new population-based metaheuristic optimization algorithm. *Applied Intelligence*, *47*(3), 850–887. https://doi.org/10.1007/s10489-017-0903-6

Omran, M. G. H. (2016). A novel cultural algorithm for real-parameter optimization. *International Journal of Computer Mathematics*, *93*(9), 1541–1563. https://doi.org/10.1080/00207160.2015.1067309

Pholdee, N., & Bureerat, S. (2012). Performance enhancement of multiobjective evolutionary optimisers for truss design using an approximate gradient. *Computers* & *Structures*, *106–107*, 115–124. https://doi.org/10.1016/j.compstruc.2012.04.015

Podolski, M., & Sroka, B. (2019). Cost optimization of multiunit construction projects using linear programming and metaheuristic-based simulated annealing algorithm. *Journal of Civil Engineering and Management*, *25*(8), 848–857. https://doi.org/10.3846/jcem.2019.11308

Ramli, M. R., Abas, Z. A., Desa, M. I., Abidin, Z. Z., & Alazzam, M. B. (2019). Enhanced convergence of Bat Algorithm based on dimensional and inertia weight factor. *Journal of King Saud University – Computer and Information Sciences*, *31*(4), 452–458. https://doi.org/10.1016/j.jksuci.2018.03.010

Rao, R. V. (2016). Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations*, *7*(1), 19–34. https://doi.org/10.5267/j.ijiec.2015.8.004

Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, *43*(3), 303–315. https://doi.org/10.1016/j.cad.2010.12.015

Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences*, *179*(13), 2232–2248. https://doi.org/10.1016/j.ins.2009.03.004

Rocca, P., Oliveri, G., & Massa, A. (2011). Differential evolution as applied to electromagnetics. *IEEE Antennas and Propagation Magazine*, *53*(1), 38–49. https://doi.org/10.1109/MAP.2011.5773566

Sabbah, T. (2020). Enhanced genetic algorithm for optimized classification. In *2020 International Conference on Promising Electronic Technologies* (*ICPET*), Jerusalem, Palestine. https://doi.org/10.1109/ICPET51420.2020.00039

Samareh Moosavi, S. H., & Bardsiri, V. K. (2019). Poor and rich optimization algorithm: A new human-based and multi populations algorithm. *Engineering Applications of Artificial Intelligence*, *86*, 165–181. https://doi.org/10.1016/j.engappai.2019.08.025

Shabani, A., Asgarian, B., Salido, M., & Asil Gharebaghi, S. (2020). Search and rescue optimization algorithm: A new optimization method for solving constrained engineering optimization problems. *Expert Systems with Applications*, *161*, 113698. https://doi.org/10.1016/j.eswa.2020.113698

Shareef, H., Ibrahim, A. A., & Mutlag, A. H. (2015). Lightning search algorithm. *Applied Soft Computing*, *36*, 315–333. https://doi.org/10.1016/j.asoc.2015.07.028

Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*, *12*(6), 702–713. https://doi.org/10.1109/TEVC.2008.919004

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

Truong, D.-N., & Chou, J.-S. (2023). Fuzzy adaptive forensic-based investigation algorithm for optimizing frequency-constrained structural dome design. *Mathematics and Computers in Simulation*, *210*, 473–531. https://doi.org/10.1016/j.matcom.2023.03.007

van Laarhoven, P. J. M., & Aarts, E. H. L. (1987). *Simulated annealing*. Dordrecht. Springer Netherlands. https://doi.org/10.1007/978-94-015-7744-1_2

Wu, L., Wu, J., & Wang, T. (2023). Enhancing grasshopper optimization algorithm (GOA) with levy flight for engineering applications. *Scientific Reports*, *13*(1), 124. https://doi.org/10.1038/s41598-022-27144-4

Wu, S.-J., & Chow, P.-T. (1995). Steady-state genetic algorithms for discrete optimization of trusses. *Computers & Structures*, *56*(6), 979–991. https://doi.org/10.1016/0045-7949(94)00551-D

Xue, X., & Chen, X. (2019). Determination of ultimate bearing capacity of shallow foundations using Lssvm algorithm. *Journal of Civil Engineering and Management*, *25*(5), 451–459. https://doi.org/10.3846/jcem.2019.9875

Yang, X.-S., & Deb, S. (2009). Cuckoo search via Lévy flights. In *2009 World Congress on Nature & Biologically Inspired Computing* (*NaBIC*), Coimbatore, India. https://doi.org/10.1109/NABIC.2009.5393690

Yang, X.-S. (2010a). Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation*, *2*(2), 78–84. https://doi.org/10.1504/IJBIC.2010.032124

Yang, X.-S. (2010b). *Nature-inspired metaheuristic algorithms*. Luniver Press.

Yang, X. S., & Hossein Gandomi, A. (2012). Bat algorithm: a novel approach for global engineering optimization. *Engineering Computations*, *29*(5), 464–483. https://doi.org/10.1108/02644401211235834

Yıldızdan, G., & Baykan, Ö. K. (2021). A novel artificial jellyfish search algorithm improved with detailed local search strategy. In *2021 6th International Conference on Computer Science and Engineering* (*UBMK*), Ankara, Turkey. https://doi.org/10.1109/UBMK52708.2021.9559009

Zhao, W., Wang, L., & Zhang, Z. (2019). Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowledge-Based Systems*, *163*, 283–304. https://doi.org/10.1016/j.knosys.2018.08.030

Zhou, Y., Ling, Y., & Luo, Q. (2018). Lévy flight trajectory-based whale optimization algorithm for engineering optimization. *Engineering Computations*, *35*(7), 2406–2428. https://doi.org/10.1108/EC-07-2017-0264

Zong Woo, G., Joong Hoon, K., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: Harmony search. *Simulation*, *76*(2), 60–68. https://doi.org/10.1177/003754970107600201

APPENDIX

A. Supplementary information on mathematical functions

Table A.1. IEEE CEC 2022 benchmark functions (Kumar et al., 2021)

No.	Function	Optimal value	Range	Characteristics
F1	Shifted and fully rotated Zakharov function	300	[–100, 100]	– Unimodal – Nonseparable
F2	Shifted and fully rotated Rosenbrock function	400	[–100, 100]	– Multimodal – Nonseparable – A large number of local optima
F3	Shifted and fully rotated expanded Schaffer F6 function	600	[–100, 100]	– Multimodal – Nonseparable – Asymmetric – A large number of local optima
F4	Shifted and fully rotated noncontinuous Rastrigin function	800	[–100, 100]	– Multimodal – Nonseparable – Asymmetric – A large number of local optima
F5	Shifted and fully rotated Levy function	900	[–100, 100]	– Multimodal – Nonseparable – A large number of local optima
F6	Hybrid function 1 (N = 3)	1800	[–100, 100]	– Unimodal or multimodal, depending on the basic function – Nonseparable subcomponents – Different properties for variable subcomponents
F7	Hybrid function 2 (N = 6)	2000	[–100, 100]	
F8	Hybrid function 3 (N = 5)	2200	[–100, 100]	
F9	Composition function 1 (N = 5)	2300	[–100, 100]	– Multimodal – Nonseparable – Asymmetric – Different properties around local optima
F10	Composition function 2 (N = 4)	2400	[–100, 100]	
F11	Composition function 3 (N = 5)	2600	[–100, 100]	
F12	Composition function 4 (N = 6)	2700	[–100, 100]	

Note: N refers to the number of essential functions.

B. Supplementary data for statistical values in mathematical tests

Table B1. Statistical optimal values of ABC, CA, DE, ESO[1], ESO[2], FA, GA, and GWO

Fun.	d	Value	ABC	CA	DE	ESO[1]	ESO[2]	FA	GA	GWO
F1	10	Min.	2104.30	300.00	444.11	300.00	300.00	300.00	325.85	307.92
		Max.	4139.89	7629.94	1156.77	300.00	300.00	300.00	2775.97	4267.50
		Mean	3062.74	1139.64	646.65	300.00	300.00	300.00	613.17	1055.06
		Std.	527.69	1441.96	177.84	0.00	0.00	3.25E-04	472.01	1373.04
	20	Min.	18541.42	5962.82	12441.49	300.00	300.00	300.01	310.64	574.69
		Max.	29847.10	45215.69	29567.59	300.00	300.00	300.02	608.17	16658.90
		Mean	23998.37	23871.55	20162.53	300.00	300.00	300.01	416.63	6860.76
		Std.	2774.36	10067.87	3771.11	7.80E-07	6.90E-09	1.98E-03	70.48	3817.55
F2	10	Min.	404.28	400.23	406.63	400.00	400.00	400.00	400.00	400.61
		Max.	408.92	474.11	408.92	408.92	404.77	408.92	470.98	470.86
		Mean	408.47	411.35	407.68	404.38	403.11	406.24	405.85	416.55
		Std.	1.14	14.98	0.62	2.96	2.03	3.78	12.69	16.11
	20	Min.	449.08	417.01	448.42	444.90	405.34	444.90	405.64	445.49
		Max.	449.08	530.81	449.09	449.08	449.08	473.24	474.84	582.53
		Mean	449.08	462.00	449.06	448.94	447.35	452.19	453.22	486.31
		Std.	7.46E-11	37.18	0.12	0.76	8.00	8.35	13.76	36.97

Continue of Table B1

Fun.	d	Value	ABC	CA	DE	ESO[1]	ESO[2]	FA	GA	GWO
F3	10	Min.	600.00	600.00	600.00	600.00	600.00	600.01	600.00	600.01
		Max.	600.00	608.99	600.00	600.00	600.00	600.02	600.00	604.08
		Mean	600.00	601.33	600.00	600.00	600.00	600.02	600.00	600.54
		Std.	7.05E-12	2.21	4.22E-14	0.00	0	1.93E-03	8.34E-05	1.046
	20	Min.	600.00	601.03	600.00	600.00	600.00	600.03	600.00	600.15
		Max.	600.00	648.90	600.00	600.00	600.00	600.04	600.06	606.39
		Mean	600.00	618.88	600.00	600.00	600.00	600.04	600.01	602.50
		Std.	5.54E-11	13.58	0.00	0.00	1.12E-13	3.20E-03	0.01	1.82
F4	10	Min.	814.67	806.96	805.25	800.99	800.00	803.98	803.98	804.99
		Max.	830.85	851.74	820.35	806.96	803.98	823.13	832.01	825.23
		Mean	824.59	822.52	813.45	804.38	802.97	810.38	816.61	811.58
		Std.	3.98	12.07	3.29	1.69	0.89	5.12	7.67	4.97
	20	Min.	889.92	821.89	873.82	805.97	803.98	814.93	817.00	822.92
		Max.	919.53	962.18	914.27	818.90	811.08	849.75	875.76	894.37
		Mean	906.24	879.96	895.08	813.80	809.00	828.91	845.50	845.71
		Std.	6.53	32.76	10.54	3.73	2.12	7.36	14.90	17.54
F5	10	Min.	900.00	900.00	900.00	900.00	900.00	900.00	900.00	900.00
		Max.	900.00	1032.14	900.00	900.18	900.00	900.00	906.31	950.66
		Mean	900.00	927.96	900.00	900.03	900.00	900.00	901.24	904.78
		Std.	0.00	36.95	9.48E-12	0.05	0.00	6.38E-05	1.66	10.39
	20	Min.	900.00	993.92	900.24	900.00	900.00	900.00	900.94	909.31
		Max.	900.00	4130.56	905.91	900.54	900.00	900.00	951.75	1477.08
		Mean	900.00	2391.62	901.17	900.14	900.00	900.00	913.60	1047.36
		Std.	0.00	852.55	1.06	0.18	0.00	3.41E-04	13.34	149.17
F6	10	Min.	2372.54	1925.07	2078.19	1808.61	1808.95	1887.93	1803.84	1921.56
		Max.	8323.28	17760.88	4964.15	5169.89	2208.04	8038.34	6006.86	8135.21
		Mean	4845.30	5537.44	2860.64	2725.45	1934.07	3969.15	3157.90	4916.45
		Std.	1654.19	4939.79	776.71	1010.71	126.34	2161.62	1407.92	2298.56
	20	Min.	333881.33	1993.97	137024.69	1821.74	1837.26	2272.17	1847.36	2023.65
		Max.	1407004.14	26610.04	1935368.20	6812.97	2333.35	20869.14	16868.18	20339900.46
		Mean	939557.05	6510.25	830825.76	3368.99	2025.72	6426.82	4876.37	1729956.41
		Std.	270570.17	6247.68	472479.53	1482.27	141.64	5215.63	3999.99	5205631.17
F7	10	Min.	2011.51	2020.62	2000.00	2000.00	2000.00	2000.71	2000.09	2001.12
		Max.	2023.22	2090.70	2020.00	2020.00	2001.00	2022.06	2020.02	2046.20
		Mean	2018.31	2034.77	2001.88	2010.10	2000.31	2012.53	2014.52	2023.90
		Std.	2.81	16.10	3.67	10.07	0.46	9.99	8.52	8.80
	20	Min.	2057.01	2027.70	2029.70	2011.60	2007.19	2015.29	2021.73	2028.65
		Max.	2090.74	2259.22	2046.15	2033.26	2023.94	2081.54	2145.79	2120.19
		Mean	2078.91	2115.62	2036.35	2024.53	2021.80	2041.00	2054.12	2057.81
		Std.	8.37	61.74	4.42	4.13	2.90	17.45	34.07	20.75
F8	10	Min.	2216.21	2219.87	2203.24	2200.04	2200.02	2200.29	2203.93	2201.64
		Max.	2226.76	2226.74	2221.32	2220.64	2220.01	2222.98	2221.81	2230.00
		Mean	2222.40	2221.77	2213.62	2217.71	2206.14	2210.77	2219.33	2221.03
		Std.	2.73	1.53	5.26	6.51	8.36	10.15	4.32	7.08
	20	Min.	2232.19	2221.02	2224.82	2220.12	2220.03	2220.52	2220.54	2223.63
		Max.	2243.98	2490.71	2228.28	2221.35	2220.88	2237.71	2341.58	2355.58
		Mean	2238.83	2285.34	2226.49	2220.87	2220.63	2221.81	2235.50	2244.19
		Std.	2.67	74.21	0.74	0.30	0.19	3.02	35.83	41.30

End of Table B1

Fun.	d	Value	ABC	CA	DE	ESO[1]	ESO[2]	FA	GA	GWO
F9	10	Min.	2529.28	2487.34	2529.28	2529.28	2529.28	2529.28	2529.28	2529.28
		Max.	2529.28	2592.53	2529.28	2529.28	2529.28	2529.28	2529.29	2635.30
		Mean	2529.28	2502.78	2529.28	2529.28	2529.28	2529.28	2529.28	2548.95
		Std.	0.00	20.78	0.00	0.00	0.00	0.00	0.00	29.31
	20	Min.	2480.78	2465.71	2480.78	2480.78	2480.78	2480.78	2480.78	2481.33
		Max.	2480.78	2493.66	2480.78	2480.78	2480.78	2480.78	2480.96	2569.35
		Mean	2480.78	2474.59	2480.78	2480.78	2480.78	2480.78	2480.79	2495.97
		Std.	7.81E-08	7.65	1.60E-07	0.00	3.87E-13	1.12E-04	3.24E-02	19.71
F10	10	Min.	2500.36	2500.34	2491.64	2500.16	2428.35	2500.12	2500.24	2500.24
		Max.	2500.64	3282.12	2500.77	2500.26	2500.21	2500.29	2632.13	2616.27
		Mean	2500.47	2576.42	2500.12	2500.21	2497.79	2500.20	2531.93	2544.94
		Std.	0.08	146.82	1.60	0.03	13.12	0.03	53.25	55.67
	20	Min.	2501.13	2500.78	2438.49	2421.04	2409.36	2500.25	2500.37	2500.29
		Max.	2524.14	4846.82	2510.31	2551.22	2500.29	2633.79	3138.87	4269.80
		Mean	2503.46	3548.22	2497.72	2492.38	2460.02	2509.17	2658.05	3033.94
		Std.	4.85	650.37	13.08	38.48	30.57	33.59	230.71	592.48
F11	10	Min.	2861.66	2845.76	2858.62	2861.40	2858.62	2858.62	2863.92	2859.68
		Max.	2864.22	2900.00	2863.31	2865.21	2863.88	2865.21	2872.93	2867.27
		Mean	2863.04	2852.38	2860.88	2864.04	2862.99	2863.51	2868.02	2864.11
		Std.	0.68	13.33	1.28	1.00	1.22	1.54	2.52	1.36
	20	Min.	2932.01	2891.73	2935.11	2938.26	2932.50	2931.38	2941.91	2942.66
		Max.	2939.92	2900.98	2944.40	2953.74	2941.25	2968.88	3023.69	2989.95
		Mean	2934.87	2899.57	2939.74	2945.31	2938.41	2944.19	2962.65	2957.19
		Std.	1.46	1.73	1.98	4.14	2.63	7.52	15.98	10.43
F12	10	Min.	2860.73	2846.30	2858.66	2860.68	2859.37	2858.62	2865.34	2859.24
		Max.	2864.11	2900.00	2863.88	2865.21	2863.88	2866.66	2878.69	2872.95
		Mean	2862.94	2855.92	2860.81	2864.00	2863.13	2863.38	2868.70	2863.72
		Std.	0.76	15.84	1.43	1.18	1.00	1.71	2.73	2.61
	20	Min.	2929.18	2899.51	2935.64	2937.23	2932.86	2932.20	2941.98	2940.33
		Max.	2937.12	2903.49	2942.89	2952.90	2942.46	2959.63	2975.91	3033.86
		Mean	2934.72	2900.10	2939.10	2945.14	2939.37	2943.01	2956.82	2959.60
		Std.	1.56	0.65	1.75	5.31	2.76	6.09	10.63	20.10

Table B2. Statistical optimal values of HS, JAYA, PSO, SA, SO, SOS, and TLBO

Fun.	d	Value	HS	JAYA	PSO	SA	SO	SOS	TLBO
F1	10	Min.	300.88	12160.86	300.00	303.26	300.00	300.00	300.00
		Max.	591.65	50336.74	300.00	321.70	300.00	300.00	300.00
		Mean	342.42	25199.78	300.00	312.53	300.00	300.00	300.00
		Std.	60.83	8254.36	0.00	4.56	0.00	0.00	0.00
	20	Min.	510.16	31882.40	300.00	345.57	300.00	300.00	300.00
		Max.	1333.19	118486.64	300.00	535.21	300.00	300.02	300.00
		Mean	821.11	76387.62	300.00	421.57	300.00	300.00	300.00
		Std.	215.81	21362.39	0.00	35.08	5.24E-4	4.17E-3	1.03E-06
F2	10	Min.	400.00	412.20	400.01	405.36	400.01	400.00	400.00
		Max.	408.92	500.56	470.78	409.24	408.92	470.78	470.78
		Mean	406.38	439.23	406.02	408.58	402.61	409.63	405.53
		Std.	3.28	19.83	12.80	1.24	3.14	15.57	12.84

Continue of Table B2

Fun.	d	Value	HS	JAYA	PSO	SA	SO	SOS	TLBO
F2	20	Min.	449.08	811.87	400.00	446.49	402.96	400.00	402.48
		Max.	474.98	1847.27	471.08	450.40	471.88	473.31	474.77
		Mean	452.41	1150.57	446.19	447.60	444.19	443.44	448.39
		Std.	8.51	253.12	19.41	0.78	14.11	21.19	18.82
F3	10	Min.	600.00	612.42	600.00	601.05	600.00	600	600.00
		Max.	600.00	624.74	619.70	601.90	600.15	600.00	600.19
		Mean	600.00	618.67	601.77	601.57	600.01	600.00	600.01
		Std.	2.60E-07	3.49	4.06	0.21	0.03	9.06E-06	0.034
	20	Min.	600.00	629.65	600.45	603.15	600.03	600.00	600.04
		Max.	600.00	667.74	639.55	604.38	600.36	600.06	608.05
		Mean	600.00	645.69	613.02	603.80	600.17	600.01	602.06
		Std.	0.00	9.69	10.08	0.36	0.10	0.01	1.74
F4	10	Min.	800.99	831.61	804.97	802.87	803.98	801.99	800.99
		Max.	809.95	859.21	831.84	810.53	813.93	820.78	812.94
		Mean	805.45	846.26	817.58	807.27	808.86	810.11	806.52
		Std.	2.26	7.07	7.35	1.85	2.91	4.77	2.87
	20	Min.	860.50	918.75	825.87	826.90	813.93	809.15	812.93
		Max.	902.27	989.68	876.61	860.70	829.85	898.27	855.56
		Mean	886.88	955.76	851.11	843.13	821.86	844.86	828.19
		Std.	9.75	15.43	13.24	7.53	4.80	21.26	8.91
F5	10	Min.	900.00	928.89	900.00	900.84	900.00	900.00	900.00
		Max.	902.40	1022.25	901.91	901.99	900.18	900.09	902.77
		Mean	900.27	968.58	900.40	901.36	900.01	900.01	900.27
		Std.	0.51	26.51	0.56	0.33	0.04	0.02	0.56
	20	Min.	900.00	1889.22	904.22	908.68	900.00	900.18	900.09
		Max.	910.04	5663.92	4778.71	917.33	1049.28	910.41	959.38
		Mean	901.92	3339.69	1338.20	913.76	916.82	901.93	914.99
		Std.	2.72	938.90	763.54	2.15	40.48	2.13	13.16
F6	10	Min.	1805.89	301947.73	1806.90	1982.46	1843.99	1814.38	1893.49
		Max.	6313.36	13003715.97	7308.82	8566.82	6813.60	6918.72	6654.43
		Mean	2953.77	3601011.96	3218.84	4502.37	2685.61	2858.97	2718.62
		Std.	1190.57	2896745.52	1520.67	2283.85	1081.28	1215.09	1065.62
	20	Min.	1836.22	27528483.61	1867.36	44818.82	1907.11	1886.04	1920.95
		Max.	12155.82	367229216.12	19886.07	310663.83	18983.09	20285.99	23427.98
		Mean	4214.63	162628824.16	6107.94	143998.01	7225.03	6662.28	5817.89
		Std.	2849.44	95195636.06	5476.22	77510.57	5103.51	5058.09	5341.92
F7	10	Min.	2000.00	2033.35	2004.60	2008.18	2000.99	2000.00	2000.99
		Max.	2020.01	2079.20	2051.84	2141.38	2033.50	2020.34	2025.54
		Mean	2009.62	2051.38	2029.88	2024.13	2016.40	2003.20	2010.96
		Std.	9.88	9.61	12.00	22.90	9.54	6.81	9.46
	20	Min.	2016.15	2125.04	2034.99	2037.33	2024.54	2006.21	2022.73
		Max.	2065.90	2259.39	2219.81	2185.30	2098.26	2065.15	2061.81
		Mean	2033.93	2183.71	2107.44	2078.46	2045.83	2034.31	2040.09
		Std.	12.49	30.43	61.65	51.22	21.83	14.23	9.20
F8	10	Min.	2200.72	2226.70	2220.00	2207.65	2200.51	2200.40	2200.26
		Max.	2221.10	2239.50	2318.85	2345.97	2223.55	2222.00	2226.82
		Mean	2216.84	2230.60	2223.89	2243.80	2220.13	2216.72	2217.25
		Std.	6.06	3.21	17.94	46.42	3.84	7.75	9.98

End of Table B2

Fun.	d	Value	HS	JAYA	PSO	SA	SO	SOS	TLBO
F8	20	Min.	2220.55	2236.87	2220.73	2228.54	2220.58	2220.95	2224.57
		Max.	2223.23	2268.60	2341.64	2246.12	2238.86	2228.90	2347.36
		Mean	2221.45	2251.38	2254.18	2231.26	2222.99	2224.24	2234.83
		Std.	0.57	8.28	53.01	3.09	3.30	1.65	21.69
F9	10	Min.	2529.28	2534.24	2529.28	2529.32	2529.28	2529.28	2529.28
		Max.	2529.37	2573.32	2529.28	2529.42	2529.28	2529.28	2529.28
		Mean	2529.30	2548.79	2529.28	2529.37	2529.28	2529.28	2529.28
		Std.	0.02	9.30	0.00	0.03	0.00	0.00	0.00
	20	Min.	2481.00	2607.57	2480.78	2481.38	2480.78	2480.78	2480.78
		Max.	2483.84	2997.30	2480.78	2481.87	2480.78	2480.78	2480.78
		Mean	2482.08	2754.43	2480.78	2481.65	2480.78	2480.78	2480.78
		Std.	8.04E-01	94.09	1.23E-12	0.12	6.02E-05	2.67E-13	1.18E-12
F10	10	Min.	2400.00	2473.26	2500.23	2476.39	2400.44	2500.15	2500.20
		Max.	2614.32	2666.96	2624.07	2611.78	2632.95	2500.44	2607.56
		Mean	2546.63	2536.88	2581.01	2537.45	2512.93	2500.30	2507.42
		Std.	57.17	56.42	53.86	52.12	70.57	0.06	27.00
	20	Min.	2400.03	2534.61	2500.35	2500.46	2420.27	2500.40	2500.43
		Max.	2500.49	5993.31	4645.80	3630.92	3099.05	2641.63	3220.73
		Mean	2403.46	3688.43	3653.37	2983.71	2677.98	2514.00	2562.46
		Std.	18.33	1421.76	642.84	295.73	186.44	41.18	142.51
F11	10	Min.	2861.44	2869.13	2862.70	2859.11	2863.90	2858.73	2861.11
		Max.	2867.82	2912.20	2923.38	2863.55	2887.48	2865.21	2867.53
		Mean	2865.35	2877.61	2867.92	2861.58	2869.13	2862.54	2863.95
		Std.	1.88	8.71	10.65	1.63	5.36	1.71	1.46
	20	Min.	2936.21	3015.41	2951.34	2934.95	2941.77	2936.95	2941.51
		Max.	2961.62	3163.37	3098.25	2946.27	3021.22	2998.54	3007.13
		Mean	2946.38	3075.85	2985.95	2939.05	2976.25	2955.65	2956.88
		Std.	6.13	39.05	31.11	3.09	18.76	12.30	16.42
F12	10	Min.	2861.41	2870.08	2864.03	2859.42	2864.64	2858.62	2858.62
		Max.	2869.89	2928.41	2919.97	2863.22	2889.26	2865.17	2867.58
		Mean	2866.21	2878.28	2868.05	2861.61	2869.05	2862.05	2864.12
		Std.	2.23	10.99	9.98	1.21	4.75	1.52	1.91
	20	Min.	2931.39	2988.10	2945.44	2935.21	2946.63	2937.72	2934.74
		Max.	2961.51	3149.75	3092.88	2943.32	3061.97	3123.09	3044.73
		Mean	2947.03	3075.70	2998.05	2938.48	2977.01	2957.14	2958.22
		Std.	7.38	35.03	38.34	2.20	20.61	32.95	22.07