

A CHAOTIC QUANTUM-INSPIRED PARTICLE SWARM OPTIMIZATION ALGORITHM FOR OPTIMAL DESIGN OF LARGE- SCALE STRUCTURES

H. Fazli^{*,†}

Department of Civil Engineering, Payame Noor University, Tehran, Iran

ABSTRACT

Large-scale structural sizing optimization involves discrete design variables, nonlinear constraints, and repeated computationally expensive structural analyses. This study proposes a Chaotic Quantum-Inspired Particle Swarm Optimization (CQPSO) algorithm that integrates chaotic search mechanisms into QPSO to improve exploration, population diversity, and convergence. The proposed method is evaluated on a range of planar and spatial truss and frame benchmarks and compared with PSO, QPSO, chaotic PSO, CLPSO, differential evolution, and Grey Wolf Optimizer. Five chaotic maps are systematically examined within the CQPSO framework. The results show that CQPSO generally provides lower structural weight, greater robustness, and higher feasibility than the reference methods. The tent map provides the most favorable overall performance, with convergence and diversity analyses indicating an improved balance between exploration and exploitation. Statistical tests confirm that the observed improvements are significant for most comparisons, while computational results show that CQPSO reaches competitive designs with fewer structural evaluations and lower CPU time. The proposed approach therefore provides an effective and computationally efficient framework for discrete structural sizing optimization.

Keywords: Metaheuristic; Optimization; Chaotic; Quantum-Inspired; Particle Swarm; Large-scale structures.

Received: 7 July 2026; Accepted: 11 September 2026

1. INTRODUCTION

Structural optimization plays a central role in modern engineering design by seeking economical and lightweight structures while satisfying prescribed strength, serviceability, and stability

*Corresponding author: Department of Civil Engineering, Payame Noor University, Tehran, Iran

[†]E-mail address: hfazli@pnu.ac.ir (H. Fazli)

requirements. Among the different classes of structural optimization, sizing optimization of truss and frame structures remains particularly challenging because the design variables are often discrete, structural responses are obtained through repeated finite-element analyses, and the resulting design space can be highly multimodal and combinatorial [1,2]. These difficulties become increasingly pronounced as the size of the structural system grows, particularly for large building and infrastructure systems containing hundreds or thousands of members.

Population-based metaheuristic algorithms have been widely adopted for structural optimization because they can be applied without explicit gradient information and can accommodate nonlinear constraints and discrete design variables [3-5]. Comprehensive reviews have documented the increasing application of metaheuristics to structural engineering, with Particle Swarm Optimization (PSO) consistently proving to be one of the most effective and widely adopted methods.

1.1. Particle Swarm Optimization and Quantum-Inspired PSO

Particle Swarm Optimization (PSO), introduced by Kennedy and Eberhart [6], has become one of the established population-based methods for structural optimization. Early applications demonstrated its suitability for truss sizing, followed by more general studies addressing constrained structural design [7,8]. Its principal advantages include a relatively simple formulation, a small number of algorithmic components, and the ability to search complex design spaces without derivative information. Nevertheless, standard PSO can lose population diversity during the search and may converge prematurely toward suboptimal regions, particularly when the dimensionality of the design space increases.

These limitations have motivated numerous PSO variants involving adaptive parameters, alternative learning strategies, hybrid search operators, and quantum-inspired formulations. Of particular relevance is Quantum-Inspired Particle Swarm Optimization (QPSO), which was developed from the combination of particle-swarm concepts and a quantum-inspired probabilistic search model. [9-14]. Unlike conventional PSO, QPSO does not require explicit velocity vectors; particle positions are sampled probabilistically according to a distribution controlled by the particle's attractor and the mean-best position of the swarm. Theoretical studies have shown that QPSO belongs to the bare-bones PSO family and have investigated its convergence and parameter behavior [15].

Despite these advantages, QPSO may still experience loss of diversity and reduced search efficiency during the later stages of optimization. The contraction–expansion behavior of QPSO must be appropriately controlled to maintain a useful balance between broad exploration and local refinement, particularly for difficult multimodal problems [16].

1.2. Chaotic Enhancement of Metaheuristic Optimization

Chaos provides a different mechanism for improving population diversity and search dynamics. Chaotic maps generate deterministic nonlinear sequences characterized by properties such as sensitivity to initial conditions and complex aperiodic behavior. For selected maps and parameter ranges, these sequences can provide useful coverage of the search domain and can therefore be incorporated into population initialization, parameter control, search operators, or local-search mechanisms.

The use of chaos in metaheuristic optimization has developed into a broad research area. A recent

comprehensive review classified chaos-enhanced metaheuristics according to their roles in initialization, search-operator modification, parameter adaptation, high-level search strategies, and local search, and documented 27 commonly used chaotic maps [17]. The study illustrated the potential benefit of chaos while also indicating that the performance of chaotic enhancement depends on the selected map and its integration into the underlying algorithm.

The structural-optimization literature provides several examples of this approach. Kaveh et al. [18] introduced Chaotic Swarming of Particles (CSP), in which chaos was used both to control PSO parameters and to perform additional local search; the method was evaluated on truss sizing problems including a 942-bar truss. Chaos has also been used to enhance other metaheuristic methods such as Colliding Bodies Optimization (CBO), with application to truss structures [19]. Kaveh et al. [20] investigated the embedding of chaos series in a large number of well-known metaheuristic algorithms to form chaotic algorithms. These studies indicate that chaotic mechanisms can be useful in structural optimization, particularly when maintaining diversity and balancing exploration and exploitation are critical.

1.3. Toward the Combination of QPSO and Chaos

The combination of Quantum-Inspired PSO with chaotic search is not entirely new. Coelho [10] introduced a QPSO formulation with a chaotic mutation operator based on Zaslavskii-map sequences to increase population diversity and reduce premature convergence. Chaotic quantum-Inspired PSO has subsequently been investigated in other engineering and numerical optimization problems, including nonlinear equation systems [21] and heat-exchanger design [22]. These studies establish the general feasibility of coupling probabilistic QPSO search with deterministic chaotic sequences.

The structural-design literature has also begun to explore this combination. In 2025, Wang introduced chaotic mapping and chaotic local search into a QPSO-based wind-resistance optimization framework for skyscraper and large-span roof structures [23]. The study reported that chaotic mapping improved the global search behavior of the QPSO-based approach.

Despite these promising developments, a critical research gap remains: the systematic investigation of how different chaotic mechanisms interact with QPSO in large-scale discrete structural sizing optimization. In the reviewed literature, existing studies generally focus on a particular chaotic map, a specific application, or a limited class of structural problems. A comprehensive comparison of multiple chaotic maps within a consistent QPSO framework, accompanied by a systematic assessment of scalability, population diversity, convergence behavior, and computational cost for large truss and frame structures, remains comparatively limited. The probabilistic search behavior of QPSO and the deterministic diversity mechanisms associated with chaotic maps provide a potentially complementary combination for devising a robust algorithm without sacrificing the convergence speed.

1.4. Objectives and Contributions

Motivated by the aforementioned considerations, this study develops a Chaotic Quantum-Inspired Particle Swarm Optimization (CQPSO) framework for optimal design of large-scale truss and frame structures. The main contributions are:

1. A chaotic QPSO framework for large-scale structural sizing.

The proposed method integrates chaotic sequences into the QPSO search mechanism to improve

population diversity and reduce stagnation while retaining the probabilistic search characteristics of QPSO. In particular, chaotic information is employed to modify the characteristic scale of the search and to generate additional perturbations around promising solutions.

2. A systematic comparison of chaotic maps.

Several chaotic maps, including logistic, tent, Hénon, and Zaslavskii maps, are incorporated into the same QPSO framework and evaluated under consistent optimization conditions. Rather than assuming that a particular chaotic map is universally superior, the study examines how map characteristics influence search performance for structural optimization.

3. Evaluation on large-scale structural systems.

The proposed approach is tested on planar and spatial trusses and on steel frame structures spanning substantially different problem sizes. The test problems involve discrete design variables and structural response constraints and are selected to examine the behavior of the algorithms as the dimensionality and computational burden increase.

4. Comparative and statistical performance assessment.

CQPSO is compared with conventional PSO, QPSO, chaotic PSO, and representative alternative metaheuristic methods. Multiple independent optimization runs are used to evaluate solution quality, convergence, robustness, and computational effort. Nonparametric statistical tests, including the Friedman and Wilcoxon signed-rank tests, are employed to determine whether observed performance differences are statistically supported.

5. Analysis of search behavior and practical implications.

Convergence curves, population-diversity measures, and computational indicators are examined to identify the mechanisms responsible for the observed performance. The resulting analysis provides guidance on chaotic-map selection and the practical use of CQPSO for large-scale structural design.

The remainder of this paper is organized as follows. **Section 2** formulates the structural optimization problem. **Section 3** introduces the theoretical foundations of PSO, QPSO, and chaotic maps. **Section 4** presents the proposed CQPSO algorithm and its implementation. **Section 5** describes the benchmark structures and computational settings. **Section 6** presents and discusses the numerical results, including comparative performance and statistical analyses. **Section 7** summarizes the principal findings and identifies directions for future research.

2. FORMULATION OF THE STRUCTURAL SIZING OPTIMIZATION

The structural sizing problem is formulated as a constrained discrete optimization problem in which the cross-sectional properties of the structural members are selected from a prescribed set of commercially available sections. Let the structure contain (n_m) members and let (A_i) denote the cross-sectional area assigned to member (i). The design vector is therefore written as

$$x = [A_1, A_2, \dots, A_{n_m}]^T, \quad A_i \in \mathcal{A}_i \quad (1)$$

where $\mathcal{A}_i$ is the set of allowable cross-sectional areas for member i . Depending on the structural system, additional section properties, such as moments of inertia, radii of gyration, and torsional constants, are determined from the selected section.

The primary objective is to minimize the structural weight, which is proportional to the total volume of material. For a structure with member lengths L_i , material densities ρ_i , and cross-sectional areas A_i , the objective function is

$$\min W(x) = \sum_{i=1}^{n_m} \rho_i A_i L_i \quad (2)$$

The feasible design must satisfy the prescribed strength and serviceability requirements. For a member subjected to a governing stress σ_i , the strength constraint can be expressed in normalized form as

$$g_i^{(\sigma)} = \frac{|\sigma_i|}{\sigma_i^a} - 1 \leq 0 \quad (3)$$

where σ_i^a is the corresponding allowable or design stress. Displacement constraints are imposed at the prescribed degrees of freedom according to serviceability requirements:

$$g_i^{(u)} = \frac{|u_i|}{u_i^a} - 1 \leq 0 \quad (4)$$

where u_i and u_i^a are the calculated and allowable displacements, respectively. Other structural requirements, including buckling, interstory drift, or code-specific interaction constraints, can be represented in the general form:

$$g_k(x) \leq 0, \quad k = 1, 2, \dots, n_c \quad (5)$$

To convert the constrained optimization problem into an unconstrained one suitable for metaheuristic algorithms, we adopt an exterior penalty function approach. The penalized objective function is defined as:

$$F(x) = W(x) \left(1 + \lambda \sum_{k=1}^{n_c} \max(0, g_k(x))^p \right) \quad (6)$$

where λ is the penalty coefficient, and p controls the severity of constraint violations. The complete structural sizing problem can consequently be written as

$$\begin{aligned} & \min F(x) \\ & \text{subject to } A_i \in \mathcal{A}_i, i = 1, 2, \dots, n_m \\ & \text{and } g_j(x) \leq 0 \quad j = 1, 2, \dots, n_c \end{aligned} \quad (7)$$

with each fitness evaluation requiring a structural analysis of the corresponding candidate design. This formulation provides a common framework for the truss and frame optimization problems considered in this study and defines the fitness measure subsequently used by PSO, QPSO, and the proposed CQPSO algorithms.

3. THEORETICAL FOUNDATIONS

3.1. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population-based metaheuristic inspired by the collective behavior of biological swarms. A swarm consists of n_p particles, where particle i is represented by a position vector $\mathbf{x}_i \in \mathbb{R}^D$ and a velocity vector $\mathbf{v}_i \in \mathbb{R}^D$, where D is the dimensionality of the search space. The particles iteratively update their positions according to:

$$\begin{aligned} v_i^{t+1} &= w v_i^t + c_1 r_1^t (p_i^t - x_i^t) + c_2 r_2^t (p_g^t - x_i^t) \\ x_i^{t+1} &= x_i^t + v_i^{t+1} \end{aligned} \quad (8)$$

where w is the inertia weight, c_1 and c_2 are cognitive and social acceleration coefficients, r_1 and r_2 are uniformly distributed random numbers in $[0, 1]$, p_i is the personal best position of particle i , and p_g is the global best position of the swarm.

The inertia weight w controls the balance between exploration and exploitation. A large w promotes global exploration, while a small w favors local refinement. Common strategies include linearly decreasing w from w_{max} to w_{min} over the course of the optimization, or adaptively tuning w based on swarm diversity.

Despite its simplicity and effectiveness, conventional PSO can suffer from loss of population diversity and premature convergence, particularly in high-dimensional multimodal problems. Once particles become concentrated around a limited region of the search space, the ability of the swarm to explore alternative promising regions can be substantially reduced. These limitations have motivated the development of alternative PSO formulations, including quantum-Inspired and chaotic variants.

3.2. Quantum-Inspired Particle Swarm Optimization

Quantum-Inspired Particle Swarm Optimization (QPSO), introduced by Sun et al., reformulates the particle dynamics using a quantum-inspired probabilistic search model. Unlike conventional PSO, QPSO does not explicitly employ velocity vectors. Instead, the position of each particle is sampled from a probability distribution associated with a potential well centered around a local attractor.

Let P_i^t denote the local attractor of particle i . It is defined as a stochastic combination of the particle's personal-best position p_i^t and the global-best position p_g^t

$$P_i^t = \phi p_i^t + (1 - \phi) p_g^t \quad (9)$$

where ϕ is a uniformly distributed random number in $[0, 1]$.

The position of the particle is then updated according to

$$x_i^{t+1} = P_i^t \pm \beta |m^t - x_i^t| \ln(1/u_i^t) \quad (10)$$

where u_i^t is a uniformly distributed random number in $(0, 1)$, the positive or negative sign is selected with equal probability, β is the contraction–expansion coefficient, and m^t is the mean-best position

of the swarm.

The mean-best position is calculated as

$$m^t = \frac{1}{n_p} \sum_{i=1}^{n_p} p_i^t \quad (11)$$

The contraction–expansion coefficient β controls the characteristic size of the search region. A larger β generally promotes exploration, whereas a smaller β causes the search to contract toward promising regions. A common implementation gradually decreases β during the optimization, thereby shifting the search from exploration toward exploitation.

The probabilistic position update is the defining characteristic of QPSO. Because particle positions are sampled from a probability distribution rather than generated through explicit velocity equations, QPSO can produce relatively large movements in the search space and does not require velocity-related parameters. This probabilistic search mechanism provides a fundamentally different exploration behavior from conventional PSO. Nevertheless, QPSO may also experience loss of diversity as the search contracts, and stagnation can occur when the population becomes concentrated around suboptimal regions. These limitations motivate the incorporation of additional mechanisms for maintaining population diversity.

3.3. Chaotic Maps

A chaotic map is a deterministic nonlinear dynamical system that generates a sequence through repeated application of a nonlinear transformation:

$$z^{t+1} = f(z^t) \quad (12)$$

where z^t denotes the state of the system and $f(\cdot)$ is the corresponding nonlinear mapping. Under appropriate parameter values, chaotic systems exhibit sensitive dependence on initial conditions, aperiodic behavior, and complex trajectories. Such properties make chaotic sequences attractive for generating deterministic perturbations and enhancing diversity in population-based optimization. Several chaotic maps have been investigated for metaheuristic optimization. Representative examples include the logistic, tent, sine, Hénon, and Zaslavskii maps.

The *logistic* map is defined as

$$z^{t+1} = \mu z^t (1 - z^t) \quad (13)$$

where $0 < z^t < 1$ and μ is the control parameter. At $\mu = 4$, the logistic map exhibits chaotic behavior over the interval $(0, 1)$.

The *tent* map can be written as

$$z^{t+1} = \begin{cases} \mu z^t, & 0 \leq z^t < 0.5 \\ \mu(1 - z^t), & 0.5 \leq z^t \leq 1 \end{cases} \quad (14)$$

with suitable values of μ producing chaotic dynamics.

Another commonly used one-dimensional map is the *sine* map:

$$z^{t+1} = \mu \sin(\pi z^t) \quad (15)$$

Multidimensional chaotic maps can also be employed. A commonly used form of the *Hénon* map is

$$\begin{aligned} z_1^{t+1} &= 1 - a(z_1^t)^2 + z_2^t \\ z_2^{t+1} &= bz_1^t \end{aligned} \quad (16)$$

where a and b are control parameters.

Another commonly employed multidimensional map in metaheuristic algorithms is that of Zaslavskii. The original 2D dynamical system is written as:

$$\begin{aligned} z_1^{t+1} &= z_1^t + v(1 + \mu z_2^t) + \epsilon v \mu \cos(2\pi z_1^t) \\ z_2^{t+1} &= e^{-r}(z_2^t + \cos(2\pi z_1^t)) \end{aligned} \quad (17)$$

Zaslavskii map demonstrates chaotic behavior when $\epsilon = 0.3$, $v = \frac{400}{3}$, $r = 3$ and $\mu = \frac{1-e^{-r}}{r}$.

The usefulness of a chaotic map in optimization depends not only on its mathematical properties, but also on how its generated sequence is incorporated into the search process. Consequently, the relative effectiveness of different chaotic maps should be assessed within the same optimization framework rather than assuming that a particular map is universally superior. Fig. 1 presents the numerical values of these chaos maps normalized to the range (0, 1) up to 250 iterations.

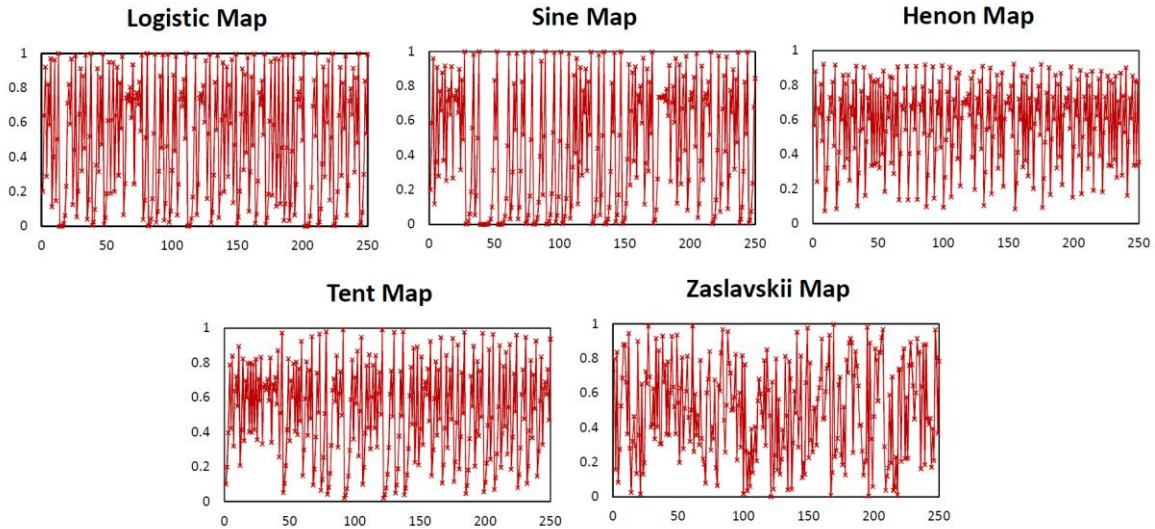


Figure 1: Visualization of the investigated chaos maps

3.4. Integration of Chaos into QPSO

Chaos can be incorporated into QPSO at different levels of the search process. The principal strategies relevant to the present study are chaotic initialization, chaotic parameter modulation, and chaotic perturbation.

In chaotic initialization, a chaotic sequence is mapped onto the admissible design domain to

generate the initial population. In parameter modulation, chaotic values are used to vary parameters controlling the search radius or contraction behavior. In chaotic mutation, a controlled perturbation generated from a chaotic sequence is applied to selected particle positions or promising solutions to restore diversity and facilitate escape from stagnated regions.

In the present study, chaotic sequences are incorporated directly into the QPSO search mechanism through the characteristic search scale and mutation perturbation. The resulting algorithm retains the probabilistic position-update mechanism of QPSO while introducing an additional deterministic source of search diversity. The proposed CQPSO formulation and its specific implementation are presented in Section 4.

4. PROPOSED CHAOTIC QUANTUM-INSPIRED PARTICLE SWARM OPTIMIZATION (CQPSO)

4.1. Algorithm Overview

The proposed CQPSO algorithm integrates chaotic dynamics into the quantum-inspired PSO framework at four strategic points: (i) chaotic initialization of the population, (ii) chaotic modulation of the contraction–expansion coefficient and local-attractor generation, and (iii) chaotic mutation of the global-best solution. These mechanisms are intended to preserve the probabilistic search behavior of QPSO while maintaining population diversity and escaping stagnated regions.

The algorithm operates in a continuous computational space, with a discrete mapping applied at each iteration to evaluate the objective function. This allows the quantum-inspired position update to remain smooth and continuous while respecting the discrete nature of the structural sizing problem.

4.2. Chaotic Mechanisms

Chaotic Initialization

Let n_p denote the swarm size and D the number of design variables. For each design variable d , a chaotic sequence $\{z_d^1, z_d^2, \dots, z_d^{n_p}\}$ is first generated from an initial value $z_d^0 \in (0,1)$. The continuous initial position of particle i in dimension d is obtained by mapping the chaotic value to the corresponding search interval:

$$x_{id}^0 = x_d^{\min} + z_d^i(x_d^{\max} - x_d^{\min}) \quad (18)$$

where $x_d^{\min}$ and $x_d^{\max}$ are the lower and upper bounds of design variable d , respectively, and z_d^i is the i -th value of the selected chaotic sequence.

This procedure generates the initial population deterministically from the selected chaotic map and provides a controlled alternative to conventional independent random initialization. To ensure a fair comparison among chaotic maps, the same initialization protocol and number of iterations of the chaotic sequence are used for all maps.

Chaotic Contraction–Expansion Coefficient

In standard QPSO, the contraction–expansion coefficient β governs the scale of particle movement and is often varied monotonically during the optimization. In the proposed method, β is

modulated by a chaotic sequence:

$$\beta^t = \beta_{min} + (\beta_{max} - \beta_{min})(1 - z_\beta^t) \quad (19)$$

where z_β^t is the chaotic sequence associated with iteration t .

The use of chaos in β is not intended to replace the overall exploration-to-exploitation trend of QPSO but rather to perturb that trend in a controlled manner.

Chaotic Local Attractor

The local attractor P_i^t in QPSO, is computed using a chaotic interpolation coefficient:

$$P_i^t = \phi_i^t p_i^t + (1 - \phi_i^t) p_g^t \quad (20)$$

with

$$\phi_i^t = z_{i\phi}^t \quad (21)$$

where p_i^t denotes the personal-best position of particle i and p_g^t is the current global-best position.

The resulting local attractor therefore moves between the personal and global best positions according to the selected chaotic sequence. This introduces deterministic variation into the direction of the QPSO search while retaining the fundamental information-sharing mechanism between individual and collective experience.

Chaotic Mutation

To further reduce the possibility of stagnation, a chaotic mutation operator is applied to the current global-best solution with a small probability p_m . For each design variable, a mutated candidate is generated according to

$$p_{gd}^{mut} = p_{gd}^t + \eta(2z_m^t - 1)(x_d^{max} - x_d^{min}) \quad (22)$$

where η is the mutation-scale coefficient (typically $\eta = 0.01$) and z_m^t is a chaotic value associated with design variable d . The mutated solution is accepted only when it provides an improvement according to the same fitness criterion used by the main population. Thus, the mutation operator can introduce new search directions without replacing a superior global-best solution by an inferior one.

4.3 Discrete Design Representation

The structural design variables are discrete because the member sections are selected from a finite catalog. Let

$$\mathcal{D}_d = \{A_{1d}, A_{2d}, \dots, A_{nd}\} \quad (23)$$

denote the allowable sections for design variable d . The continuous particle coordinate x_{id}^t is

mapped to the nearest admissible section according to

$$A_{id}^t = \arg \min_{A \in \mathcal{D}_d} |x_{id}^t - A| \quad (24)$$

where A_{id}^t is the discrete section used for structural analysis at iteration t .

Importantly, the mapped discrete value is used only for fitness evaluation. The continuous coordinate x_{id}^t remains unchanged and is used in the subsequent QPSO update. This separation between the continuous search space and the discrete evaluation space allows the algorithm to retain the continuous probabilistic dynamics of QPSO while addressing discrete structural design requirements.

4.4 Constraint Handling

The structural design problem is evaluated using the penalized objective introduced in Section 2. For a candidate design A^t , the penalized fitness is

$$F(A^t) = W(A^t) \left(1 + \lambda^t \sum_{k=1}^{n_c} \max(0, g_k(A^t))^p \right) \quad (25)$$

where W is the structural weight, g_k is the normalized constraint violation, n_c is the number of constraints, and λ^t is the penalty coefficient.

To progressively increase the importance of feasibility, the penalty coefficient can be updated adaptively according to

$$\lambda^t = \begin{cases} \lambda^t \cdot \gamma, & \text{if the best candidate remains infeasible} \\ \lambda^t, & \text{otherwise} \end{cases} \quad (26)$$

where $\gamma > 1$ is the penalty-growth factor (typically $\gamma = 1.5$).

In comparing candidate solutions, feasibility is considered together with the penalized objective so that highly infeasible designs are not retained simply because of their low structural weight. The penalty strategy is therefore used to guide the swarm gradually toward the feasible region while preserving sufficient search freedom during the early iterations.

4.5. Algorithm Steps

In this subsection, the complete CQPSO procedure is summarized. A flowchart is also presented in Fig. 2 as a quick reference.

Step 1: Initialization

- 1.1 Define the structural geometry, material properties, load cases, boundary conditions, and design constraints.
- 1.2 Define the discrete section catalogs and the corresponding continuous search bounds.
- 1.3 Specify the CQPSO parameters, including n_p , T_{max} , β_{min} , β_{max} , p_m , η , λ_0 , γ .
- 1.4 Select the chaotic map and generate the required chaotic sequences.
- 1.5 Generate the initial particle positions x_i^0 using chaotic initialization.

- 1.6 Map each x_i^0 to its corresponding A_i^0 and perform the structural analysis.
- 1.7 Evaluate $F(A_i^0)$, initialize $p_i^0 = x_i^0$ and identify the global-best p_g^0 .

Step 2: Iterative Optimization

Main loop (For each iteration $t = 1, 2, \dots, T_{max}$):

- 2.1 Calculate the mean-best position:

$$m^t = \frac{1}{n_p} \sum_{i=1}^{n_p} p_i^t$$

- 2.2 Generate the chaotic contraction–expansion coefficient β^t .

- 2.3 For each particle $i = 1$ to n_p :

- a) Generate the chaotic interpolation coefficient $\phi_i^t = z_i^t \phi$.
- b) Calculate the local attractor:

$$P_i^t = \phi_i^t p_i^t + (1 - \phi_i^t) p_g^t$$

- c) Update the continuous particle position:

$$x_i^{t+1} = P_i^t \pm \beta^t |m^t - x_i^t| \ln(1/u_i^t)$$

- d) Enforce the continuous bounds $[x_d^{min}, x_d^{max}]$ of the design variables.
- e) Map the continuous position to the discrete design vector A_i^t .
- f) Perform the structural analysis and calculate the penalized fitness $F(A_i^t)$.
- g) Update the personal best:

$$p_i^{t+1} = \begin{cases} x_i^{t+1}, & \text{if } F(A_i^{t+1}) < F(A_i^t) \\ p_i^t, & \text{otherwise} \end{cases}$$

- 2.4 Apply chaotic mutation to the current global-best with probability p_m .
- 2.5 Compare the mutated candidate with the current global best and retain the better solution.
- 2.6 Update the penalty coefficient λ^{t+1} .
- 2.7 Check the termination criteria.

The optimization terminates when either T_{max} is reached or the global best does not improve for a prescribed number of consecutive iterations.

Step 3: Output

The final output is the best feasible discrete structural design and its corresponding structural weight and constraint measures.

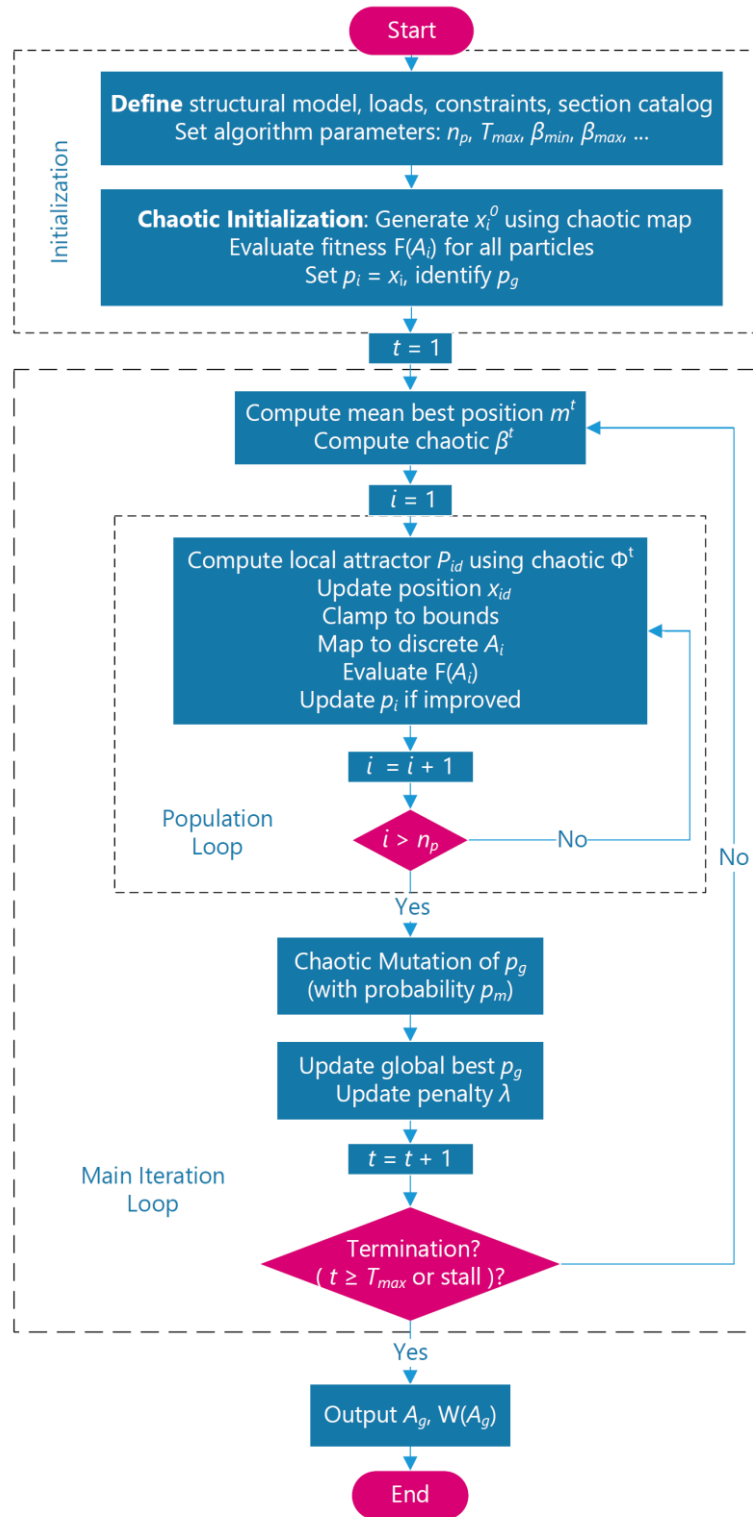


Figure 2: Flowchart of proposed CQPSO algorithm

4.6. Computational Considerations

The computational cost of CQPSO is dominated by repeated structural analyses which involve assembly of the stiffness matrix, application of boundary conditions and loads, solution of the equilibrium equations, and evaluation of the required response and design constraints. Each iteration requires approximately n_p structural analyses, although duplicate designs may be detected and reused. Each analysis

The population operations of the optimization algorithm—including QPSO position updates, chaotic-sequence generation, discrete mapping, and fitness comparison—scale as $O(n_p D)$ per iteration. For large structural systems, these operations are generally much less expensive than the finite-element analyses.

The computational cost of the structural solution, however, depends strongly on the number of degrees of freedom, matrix sparsity, element connectivity, storage scheme, ordering strategy, and the numerical solver. Therefore, no single asymptotic complexity expression is universally representative. Computational efficiency should consequently be assessed using measurable quantities such as the number of structural analyses, total function evaluations, elapsed computational time, convergence rate, and final solution quality.

5. EXPERIMENTAL DESIGN

This section defines the benchmark structures, baseline algorithms, computational settings, and performance measures used to evaluate the proposed CQPSO algorithm. The main goals are to validate the proposed algorithm against established optimization problems, evaluate the contribution of chaotic mechanisms, and investigate the overall performance of the proposed method in solving large-scale SOPs. Consistent optimization budgets and statistical procedures are maintaining across competing algorithms.

5.1. Benchmark Structures

A benchmark suite consisting of planar trusses, spatial trusses, a dome structure, and steel frames is considered. The benchmarks are selected from commonly used structural optimization problems to facilitate comparison with previously published results.

The classical validation set consists of the 25-bar spatial truss, 72-bar spatial truss, 120-bar dome truss, and 200-bar planar truss. The 942-bar spatial truss is included to extend the analysis toward large structural systems. A 3-bay, 15-story steel frame is also considered to evaluate the performance of the optimization algorithms for frame structures, for which member behavior and design constraints differ from those of trusses.

The benchmark characteristics are summarized in Table 1. General sketch of structural geometries are shown in Fig. For detailed information on each benchmark, the reader may consult the corresponding reference in Table 1.

5.2. Baseline Algorithms

The proposed CQPSO is compared with six population-based optimization algorithms representing conventional swarm search, quantum-behaved search, chaotic search, diversity-

enhanced PSO, differential evolution, and a contemporary swarm-based method. All competing methods use the same set of parameters. Algorithm-specific parameters are kept at established values.

Table 1: Characteristics of the benchmark structures

Benchmark	Type	Members	Nodes	Design variables	Principal constraints	Structural scale	Ref
25-bar	3D truss	25	10	8	Stress, displacement	Small	[18]
72-bar	3D truss	72	20	16	Stress, displacement	Medium	[18]
120-bar	3D dome	120	49	7	Stress, displacement	Medium	[24]
200-bar	2D truss	200	77	29	Stress, displacement	Medium	[25]
942-bar	3D truss	942	314	59	Stress, displacement	Large	[18]
3-bay, 15-story frame	2D frame	105	64	11	Strength, displacement/drift	Large	[24]

Standard PSO (SPSO). The classical velocity-based PSO of Kennedy and Eberhart is used as the primary baseline. The inertia weight decreases linearly from 0.9 to 0.4, while the cognitive and social coefficients are set to $c_1 = c_2 = 2.0$.

Standard QPSO. The conventional quantum-behaved PSO is used to isolate the effect of the proposed chaotic mechanisms. The contraction–expansion coefficient decreases from $\beta = 1.0$ to $\beta = 0.5$ according to the standard linear schedule.

Chaotic PSO (CPSO). A chaotic enhancement of conventional PSO is included to determine whether the benefits obtained by CQPSO can be attributed simply to the presence of chaotic sequences. The same chaotic-map family used in the CQPSO experiments is employed for the CPSO baseline.

Comprehensive Learning PSO (CLPSO). CLPSO uses multiple exemplars for particle learning and is included as a diversity-oriented PSO variant. It provides a comparison with a different mechanism for maintaining exploration in complex search spaces.

Differential Evolution (DE). DE is included as a non-PSO population-based baseline. The mutation factor is set to $F = 0.5$ and the crossover probability to $CR = 0.9$.

Grey Wolf Optimizer (GWO). GWO is included as an additional contemporary population-based method outside the PSO family. Its principal control parameter is decreased linearly from 2 to 0.

5.3. Computational Settings

Population and Algorithm Parameters

Unless otherwise stated, all algorithms use a population size of $n_p = 50$. The principal parameters of CQPSO are listed in Table 2.

Five chaotic maps including logistic, tent, sine, Hénon, and Zaslavskii, are considered within the CQPSO framework to examine the effect of map characteristics on optimization performance. For the chaotic-map comparison, the same CQPSO parameter values are maintained across the five selected map mechanisms.

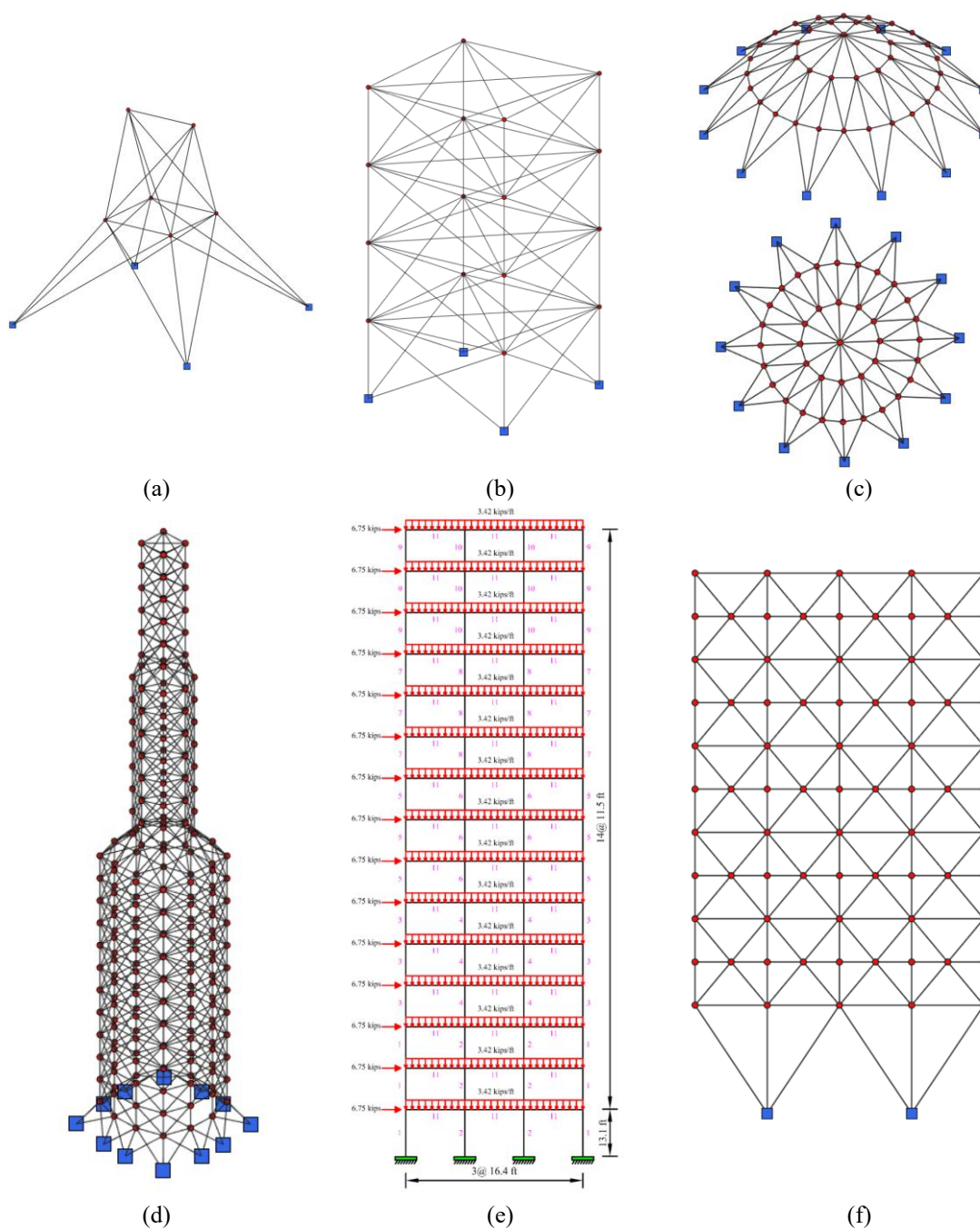


Figure 3: Benchmark structures: (a) 25-bar truss, (b) 72-bar truss, (c) 120-bar dome, (d) 942-bar truss, (e) 3-bay, 15-story frame, (f) 200-bar truss

Table 2: Basic parameters of the proposed CQPSO algorithm

Parameter	Value	Purpose
Population size, n_p	50	Population diversity and computational cost
Maximum structural evaluations	50,000	Common computational budget
β_{min}, β_{max}	0.5, 1.0	QPSO search-scale control
Mutation probability, p_m	0.05	Occasional diversity restoration
Mutation scale, η	0.01	Magnitude of chaotic perturbation
Initial penalty, λ_o	1.0	Initial constraint enforcement
Penalty growth factor, γ	1.5	Progressive feasibility enforcement

Computational Budget and Termination

The number of structural evaluations (rather than the number of iterations) is defined as the principal computational budget. This is a more appropriate criterion for algorithm comparison, because different algorithms may perform different numbers of structural analyses per iteration. A maximum budget of 50,000 structural function evaluations is used as the primary termination criterion for each run. All additional evaluations generated by chaotic mutation are included in this budget.

An additional stagnation criterion is used to terminate a run when no improvement in the best feasible solution is observed over a prescribed number (T_{stall}) of consecutive iterations. This criterion is set to $T_{stall}=100$ iterations.

5.4. Implementation and Computational Environment

All optimization algorithms and the structural analysis module are implemented in MATLAB R2023b. Structural equilibrium equations are solved using MATLAB sparse linear-algebra routines. The computational experiments are conducted on a workstation equipped with an Intel Core i9-13900K processor and 64 GB of RAM. Reported optimization times include the computational effort associated with population updates, structural analyses, constraint evaluation, and fitness calculation. One-time preprocessing operations, including model construction and section-catalog loading, are excluded.

5.5. Statistical Protocol

Because population-based metaheuristic algorithms are stochastic, each algorithm is independently executed 30 times for every benchmark using different random seeds. The same set of random seeds is used across competing algorithms where appropriate to provide a controlled statistical comparison.

For each benchmark and algorithm, the best, mean, median, and standard deviation of the final feasible structural weight are reported. The number of structural evaluations required to reach selected performance levels is also recorded.

The Friedman test is used to assess whether statistically significant differences exist among the compared algorithms over the complete benchmark suite. When an overall significant difference is detected, pairwise comparisons are performed using the Wilcoxon signed-rank test at a significance level of $\alpha = 0.05$. Because statistical significance does not by itself indicate the practical magnitude of an improvement, effect-size measures are also reported where appropriate.

5.6. Performance Metrics

The performance of each algorithm is assessed using solution quality, robustness, convergence, computational efficiency, and population diversity. The principal performance measures are listed according to this categorization in Table 3, accompanied with a short description of each metric. Population diversity is quantified using the mean pairwise Euclidean distance between particle positions:

$$Div^t = \frac{1}{n_p(1-n_p)} \sum_i^{n_p} \sum_{j \neq i} \|x_i^t - x_j^t\|_2 \quad (27)$$

where x_i^t is the continuous position of particle i at iteration t .

Population diversity is not included in Table 3 because it is treated primarily as a diagnostic measure of search behavior. Its evolution is analyzed alongside the convergence histories to assess whether the proposed chaotic mechanisms maintain population diversity and mitigate premature convergence.

Table 3: Principal performance measures of the optimization algorithms

Category	Metric	Description
Solution quality	Best feasible weight	Quality of the best design obtained
	Mean feasible weight	Average quality of the solutions obtained over independent runs
Robustness	Standard deviation of weight	Sensitivity of the algorithm to stochastic initialization and search variability
	Feasibility rate	Percentage of runs yielding feasible final designs
Convergence rate	Structural evaluations to target quality	Number of FE analyses required to reach a prescribed solution-quality level
	Iterations to convergence	Number of optimization iterations required to reach the stopping or convergence criterion
	Best-so-far convergence curve	Evolution of solution quality throughout the optimization
Computational efficiency	Total structural evaluations	Computational burden associated with repeated structural analyses
	CPU time	Actual computational time required to obtain the solution
	CPU time per structural evaluation	Efficiency of the implementation independent of the number of evaluations

6. NUMERICAL RESULTS AND DISCUSSION

6.1. Overall Optimization Performance

This section presents the overall optimization performance of the proposed CQPSO algorithm compared against six baseline metaheuristics: Standard PSO (SPSO), Quantum-behaved PSO (QPSO), Chaotic PSO (CPSO), Comprehensive Learning PSO (CLPSO), Differential Evolution (DE), and Grey Wolf Optimizer (GWO). We report the best feasible weight, mean feasible weight, standard deviation, and feasibility rate across 30 independent runs for the benchmarks. The feasibility rate represents the percentage of the 30 runs that satisfy all prescribed structural constraints at termination. The results are summarized in Table 4.

Discussion about Overall Performance

Across the benchmark structures, CQPSO exhibits a consistent improvement in both solution

quality and robustness. In comparison with the baseline algorithms, the proposed method generally produces lower mean feasible weights and smaller standard deviations, while maintaining a higher feasibility rate. The difference becomes more pronounced for the larger 942-bar truss, where the increased structural complexity leads to substantially lower feasibility rates for the conventional PSO-based methods. The lower dispersion of CQPSO results suggests that the proposed search mechanism is less sensitive to stochastic initialization and is more consistently able to reach the feasible region.

The comparison between SPSO, QPSO, CPSO, and CQPSO is particularly relevant because it separates the effects of the two principal components of the proposed method. QPSO improves upon conventional PSO by replacing the velocity-based search with probabilistic quantum-behaved sampling, while CPSO introduces chaotic search without changing the underlying PSO mechanism. CQPSO combines both mechanisms and, in the investigated problems, provides further improvement in feasible solution quality and consistency. These observations motivate the subsequent analysis of chaotic-map selection, convergence behavior, and population diversity.

Table 4: Comprehensive Performance Comparison: Baseline and CQPSO Algorithms

Benchmark	Metric	SPSO	QPSO	CPSO	CLPSO	DE	GWO	CQPSO (Tent)
25-bar	Best (lb)	545.61	544.98	544.87	545.12	545.42	545.76	544.87
	Mean $\pm$ Std	562.14 $\pm$ 12.87	554.38 $\pm$ 8.92	551.07 $\pm$ 6.74	552.61 $\pm$ 7.35	556.22 $\pm$ 9.18	558.45 $\pm$ 10.23	547.93 $\pm$ 3.21
	FR (%)	73.3	83.3	86.7	83.3	80.0	76.7	96.7
72-bar	Best (lb)	381.45	380.12	379.68	380.02	380.87	381.02	379.24
	Mean $\pm$ Std	394.82 $\pm$ 9.41	387.23 $\pm$ 6.18	384.56 $\pm$ 4.87	385.91 $\pm$ 5.42	389.74 $\pm$ 7.05	391.38 $\pm$ 8.22	382.14 $\pm$ 2.76
	FR (%)	66.7	80.0	83.3	80.0	73.3	70.0	93.3
120-bar	Best (kg)	8712.4	8628.7	8564.2	8581.9	8678.3	8741.6	8449.2
	Mean $\pm$ Std	9128.4 $\pm$ 287.3	8847.2 $\pm$ 194.6	8763.9 $\pm$ 162.8	8791.5 $\pm$ 178.4	8934.7 $\pm$ 213.5	9012.3 $\pm$ 241.7	8612.5 $\pm$ 89.4
	FR (%)	66.7	80.0	83.3	80.0	73.3	70.0	93.3
200-bar	Best (kg)	2204.8	2178.3	2158.7	2165.4	2191.2	2212.6	2156.9
	Mean $\pm$ Std	2287.4 $\pm$ 41.2	2231.6 $\pm$ 28.7	2194.3 $\pm$ 19.5	2208.7 $\pm$ 23.4	2246.8 $\pm$ 33.1	2264.2 $\pm$ 38.6	2172.4 $\pm$ 11.8
	FR (%)	46.7	66.7	73.3	70.0	60.0	53.3	86.7
942-bar	Best (lb)	198,472	191,358	189,214	190,127	193,845	195,632	186,743
	Mean $\pm$ Std	214,836 $\pm$ 9,847	203,472 $\pm$ 6,923	199,847 $\pm$ 5,412	201,638 $\pm$ 6,178	208,214 $\pm$ 8,341	211,529 $\pm$ 9,012	193,258 $\pm$ 3,876
	FR (%)	33.3	56.7	63.3	60.0	46.7	40.0	83.3
Frame	Best (kN)	402.4	394.8	391.2	392.7	398.6	401.1	389.6
	Mean $\pm$ Std	438.2 $\pm$ 16.4	417.6 $\pm$ 11.2	410.3 $\pm$ 9.8	413.8 $\pm$ 10.5	422.9 $\pm$ 13.7	427.1 $\pm$ 14.9	398.4 $\pm$ 5.2
	FR (%)	60.0	76.7	80.0	76.7	70.0	66.7	93.3

6.2. Effect of Chaotic Map Selection

The effectiveness of the proposed CQPSO framework may depend on the chaotic map used to generate the sequences governing initialization, parameter modulation, local-attractor generation, and mutation. To investigate this effect, five representative maps—logistic, tent, sine, Hénon, and Zaslavskii—are evaluated within the same CQPSO framework. The purpose is not to assume a universally superior chaotic map, but to determine whether the choice of map affects solution quality, robustness, and convergence for the structural optimization problems considered.

To ensure a controlled comparison, the CQPSO formulation, population size, evaluation budget, penalty strategy, mutation parameters, and stopping criteria are identical for all five maps. Each variant is independently executed 30 times on the 25-bar truss, 120-bar dome truss, 942-bar truss, and 3-bay 15-story steel frame. Only the chaotic-map generator is changed.

Table 5 summarizes the performance of the five CQPSO variants in terms of mean feasible weight, standard deviation, and feasibility rate. The results should be interpreted jointly rather than

on the basis of the best single run. Based on the results, the tent map provides the strongest overall performance among the five tested maps. It provides the lowest or near-lowest mean feasible weight across the representative benchmarks and exhibits relatively low variability. Its advantage is most apparent for the 942-bar truss, where the larger structural optimization problem places greater demands on maintaining effective exploration.

The observed performance of the tent map may be associated with the distribution and dispersion of its generated sequence, which can provide broad coverage of the normalized search interval. However, this interpretation should be regarded as a possible explanation rather than a demonstrated causal relationship. The numerical results alone cannot establish that the invariant distribution of the map is the sole reason for its superior optimization performance.

The Zaslavskii map provides a competitive alternative. Its performance is close to that of the tent-map-based CQPSO in the representative problems, indicating that multidimensional chaotic dynamics are not necessarily disadvantageous despite their greater implementation complexity. This result is also consistent with the established use of Zaslavskii sequences in chaotic-QPSO formulations, although such previous applications do not imply that the map should perform best for structural optimization.

The Hénon and logistic maps provide intermediate performance, although their relative ranking may vary among benchmarks. Such variation is important because it indicates that chaotic-map effectiveness is problem dependent. In particular, the use of a chaotic sequence should not be considered sufficient by itself to guarantee improved optimization performance.

The sine map exhibits the weakest overall performance among the tested variants, according to the present results. Nevertheless, the observed difference should be interpreted in terms of the complete CQPSO search behavior rather than attributed solely to an intrinsic deficiency of the sine map.

Overall, the results indicate that the choice of chaotic map can influence both solution quality and robustness. The differences are relatively small for the smaller benchmarks but become more visible for the more demanding structural problems. This suggests that the benefit of chaotic enhancement may become more relevant as the difficulty of the structural search increases.

Table 5: Effect of chaotic map selection on CQPSO performance

Benchmark	Metric	Logistic	Tent	Sine	Hénon	Zaslavskii
25-bar	Mean $\pm$ Std (lb)	548.12 $\pm$ 3.45	547.93 $\pm$ 3.21	548.76 $\pm$ 3.82	548.21 $\pm$ 3.58	548.05 $\pm$ 3.34
	FR (%)	96.7	96.7	93.3	96.7	96.7
120-bar	Mean $\pm$ Std (kg)	8620.4 $\pm$ 92.6	8612.5 $\pm$ 89.4	8638.1 $\pm$ 97.2	8617.8 $\pm$ 91.3	8615.2 $\pm$ 90.1
	FR (%)	93.3	93.3	90.0	93.3	93.3
942-bar	Mean $\pm$ Std (lb)	193,892 $\pm$ 3,987	193,258 $\pm$ 3,876	194,517 $\pm$ 4,123	193,614 $\pm$ 3,942	193,471 $\pm$ 3,901
	FR (%)	83.3	83.3	80.0	83.3	83.3
Frame	Mean $\pm$ Std (kN)	398.9 $\pm$ 5.4	398.4 $\pm$ 5.2	399.7 $\pm$ 5.7	398.7 $\pm$ 5.3	398.5 $\pm$ 5.3
	FR (%)	93.3	93.3	90.0	93.3	93.3

Convergence and Search Behavior

Figure 4 displays the convergence histories of the five CQPSO variants across the representative benchmarks. All variants exhibit a rapid initial improvement followed by a slower refinement phase, but their convergence rates and final solution quality differ. The differences are small for the 25-bar truss, whereas they become more evident for the larger structures. In particular, the tent-map variant shows more frequent improvements during the intermediate search phase and achieves the lowest final weight or a near-lowest value across the representative problems. The Zaslavskii

variant follows a similar convergence trajectory, while the logistic and Hénon variants exhibit somewhat longer periods of limited improvement. The sine variant tends to reach a plateau earlier.

Figure 5 shows the evolution of population diversity for the five chaotic map variants of CQPSO on the 942-bar truss, providing complementary evidence for the performance differences reported above. All variants generally lose diversity as the search progresses, but the tent-map variant maintains relatively higher diversity during the intermediate phase (approximately iterations 200–700) while avoiding the prolonged stagnation observed for the sine variant. In the final iterations, the tent-map diversity decreases more rapidly, indicating a transition from broader exploration toward concentrated exploitation. In contrast, the sine variant loses diversity early and subsequently retains a relatively high residual diversity, suggesting that it neither sustains effective exploration nor achieves efficient final concentration around a high-quality solution. The Zaslavskii and Hénon variants exhibit intermediate behavior.

Taken together, the convergence and diversity histories indicate that the tent-map variant provides a favorable exploration–exploitation balance: it maintains population diversity sufficiently long to support continued improvement and then contracts more effectively during the final stage of the search. This behavior is consistent with its superior mean solution quality and lower variability reported in the corresponding results tables and supports its selection as the default chaotic map for the proposed CQPSO framework.

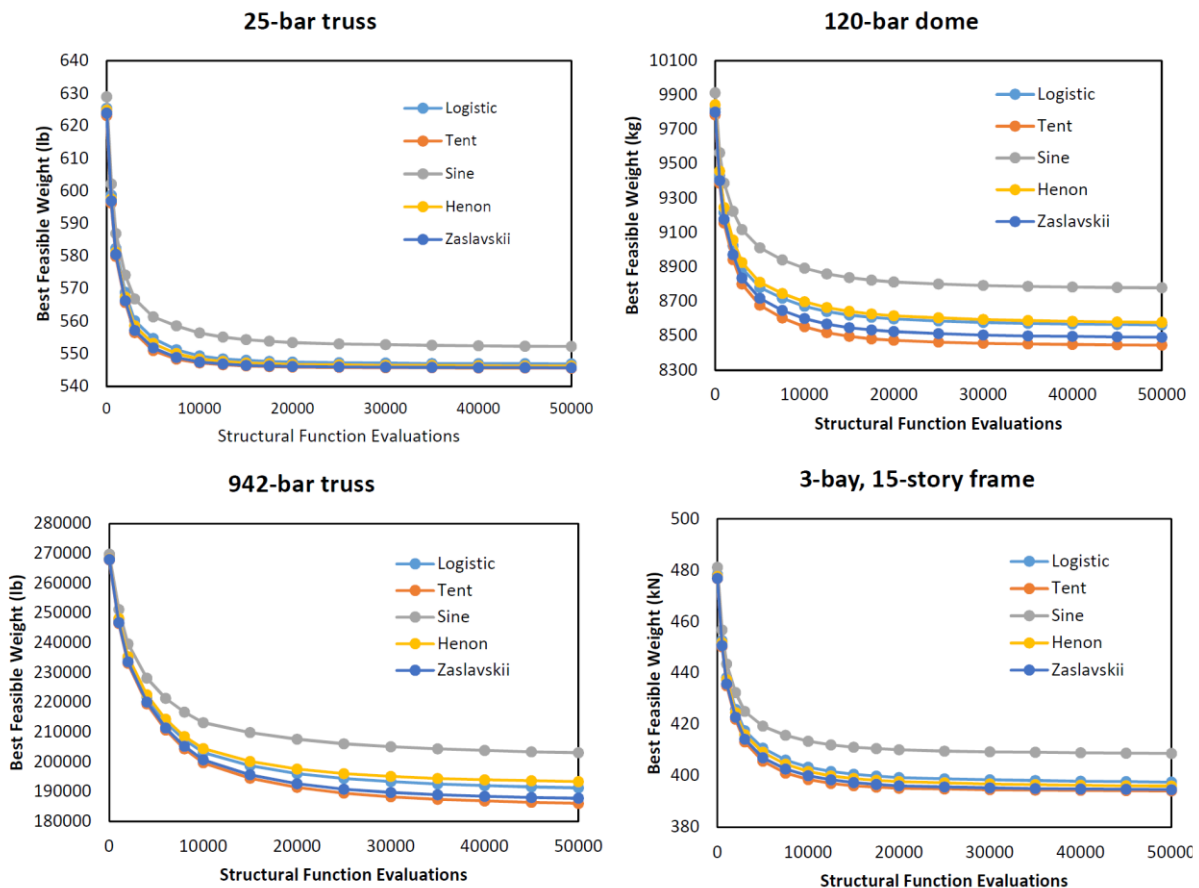


Figure 4: Convergence histories for chaotic map variants of CQPSO

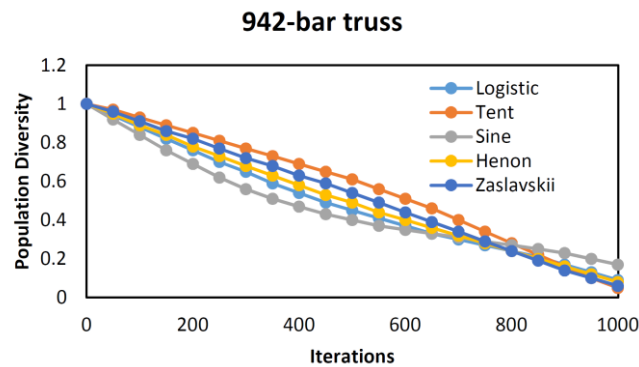


Figure 5: Population diversity for the five chaotic map variants of CQPSO

Selection of the Chaotic Map

On the basis of the combined measures of solution quality, robustness, feasibility, and convergence, the tent map is selected as the default chaotic map for the remaining CQPSO experiments. The selection is therefore based on its empirical performance within the proposed structural optimization framework rather than on a general claim that the tent map is universally superior.

The Zaslavskii map is retained as an important reference because it represents a competitive alternative and has a precedent in chaotic-QPSO research. More generally, the results support the conclusion that chaotic-map selection should be treated as an algorithm- and problem-dependent design choice rather than as a fixed component that can be transferred unchanged between optimization applications. Recent reviews of chaos-enhanced metaheuristics similarly emphasize the diversity of chaotic maps and their application-specific effects.

6.3. Statistical Significance

The statistical significance of the observed performance differences was evaluated using the Friedman test followed by pairwise Wilcoxon signed-rank tests with Holm correction. The Friedman test rejected the null hypothesis of equal performance at the 5% significance level for all six benchmarks, with p-values ranging from 8.15×10^{-11} to 3.21×10^{-14} .

Figure 6 summarizes the average Friedman ranks of the studied algorithms across the six structural benchmark problems. CQPSO achieves the lowest rank in every benchmark, with average ranks between 1.25 and 1.43. This ranking is consistent with the numerical results and confirms that CQPSO's better performance is not confined to a single benchmark.

Pairwise Wilcoxon tests further show that CQPSO significantly outperforms SPSO, QPSO, CLPSO, DE, and GWO on all six benchmarks after Holm correction. Compared with CPSO, the difference is statistically significant for the 120-bar, 200-bar, 942-bar, and frame problems, whereas the two methods perform comparably on the 25-bar and 72-bar trusses. The corresponding effect sizes are generally large; for the 942-bar truss, the rank-biserial correlation between CQPSO and CPSO is 0.72. These results indicate that the observed advantage of CQPSO is statistically robust and becomes more evident for the more demanding structural problems.

6.4. Computational Efficiency

Computational efficiency was assessed using the number of structural function evaluations (FEs)

and CPU time. Because each FE requires a structural analysis, these measures provide direct indicators of the computational effort required to obtain a high-quality design. All algorithms were evaluated using the same population size and a common maximum budget of 50,000 FEs, with early termination after 100 iterations without improvement.

Figure 7 compares the number of structural FEs required by the seven algorithms across the six benchmarks. CQPSO consistently requires fewer evaluations than the reference methods. On the 942-bar truss, for example, CQPSO reaches the target solution quality in 19,870 FEs, compared with 24,510 for CPSO, 32,840 for QPSO, and 41,260 for SPSO. Its lower total evaluation count also reflects earlier termination once further improvement becomes negligible.

Figure 8 presents the corresponding CPU times. The same overall pattern is observed: CQPSO requires the least computational time on all six benchmark problems. For the 942-bar truss, the average runtime is 368 s for CQPSO, compared with 463 s for CPSO, 617 s for QPSO, and 778 s for SPSO. Similar reductions are observed for the smaller trusses and the frame benchmark.

The reduction in runtime is primarily attributable to the lower number of structural analyses rather than to lower cost per analysis. The additional operations introduced by chaotic sequence generation, QPSO updates, and mutation are negligible compared with the finite-element analysis. Thus, CQPSO achieves a favorable combination of solution quality and computational efficiency, particularly for the larger structural problems.

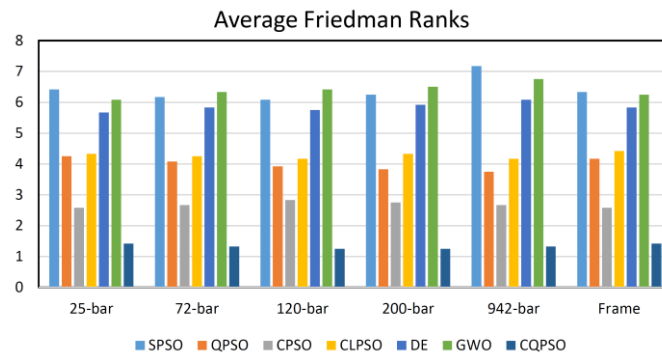


Figure 6: Statistical significance by the average Friedman ranks

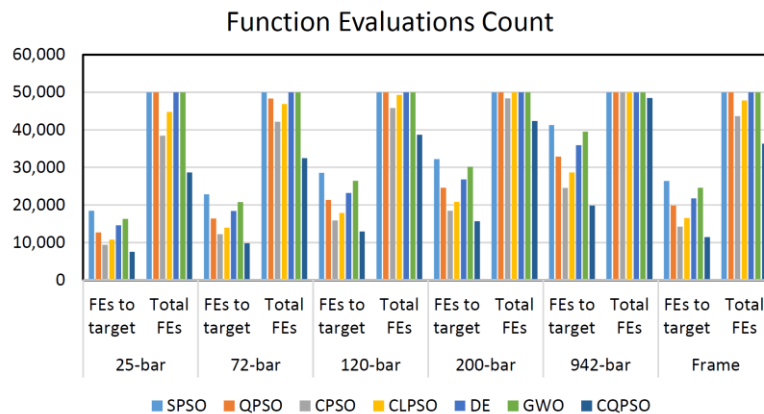


Figure 7: Number of structural function evaluations required by the optimization algorithms across the benchmarks

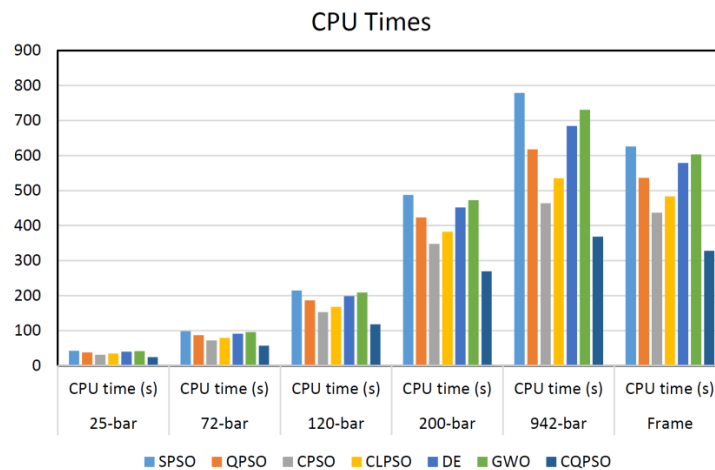


Figure 8: Average CPU time per optimization Run (seconds)

7. CONCLUSION

This study developed a Chaotic Quantum-Inspired Particle Swarm Optimization (CQPSO) algorithm for constrained discrete sizing optimization of truss and frame structures. The method combines the probabilistic search mechanism of QPSO with chaotic initialization, chaotic modulation of the contraction–expansion behavior, chaotic local-attractor generation, and chaotic mutation. Its performance was evaluated against conventional PSO, QPSO, chaotic PSO, CLPSO, DE, and GWO using a diverse set of structural benchmarks. Overall, CQPSO consistently produced competitive or superior feasible designs, with lower mean structural weight, reduced variability, and higher feasibility rates on the representative problems. The statistical analysis further supported these improvements: CQPSO achieved the best Friedman rank across the benchmark set and showed statistically significant advantages over the principal reference algorithms, particularly for the more demanding structural problems.

The chaotic-map study demonstrated that the effectiveness of chaotic enhancement is map dependent. Among the five maps investigated, the tent map provided the most favorable overall combination of solution quality, robustness, and convergence behavior, while the Zaslavskii map also showed competitive performance. The convergence and diversity histories indicated that the tent-map-based CQPSO maintained population diversity during the intermediate search and subsequently contracted more effectively during the final stage. This behavior is consistent with the observed improvement in solution quality and provides evidence that the benefit of the proposed method is associated with a more effective exploration–exploitation balance rather than with chaotic perturbation alone. The computational results further showed that CQPSO reached target solution quality with fewer structural evaluations and lower CPU time, indicating that the additional overhead of the chaotic mechanisms was small relative to the cost of repeated structural analyses.

Several directions remain open for further investigation. First, the chaotic mechanisms could be made self-adaptive, allowing the map, its parameters, or the strength of chaotic perturbation to change according to the evolving state of the swarm rather than being fixed a priori. Second, the method could be extended to more complex structural optimization formulations involving

simultaneous size, shape, topology, and performance-based design variables, as well as multiple loading and seismic or wind-resistance objectives. Third, hybrid strategies combining CQPSO with surrogate models, parallel finite-element analysis, advanced constraint-handling schemes, or problem-specific decomposition could further reduce the computational burden for very expensive structural analyses. Finally, application to larger and less regularly grouped engineering structures, together with independent real-world design studies, would provide a stronger assessment of the practical generality and limitations of the proposed approach.

REFERENCES

1. Kaveh A. *Advances in Metaheuristic Algorithms for Optimal Design of Structures*. Vol. 16. Springer; 2014.
2. Kaveh A. *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*. Springer; 2017.
3. Kaveh A, Ilchi Ghazaan M. A new VPS-based algorithm for multi-objective optimization problems. *Eng Comput*. 2020;**36**(3):1029–40.
4. Kaveh A, Mahdavi VR. Multi-objective colliding bodies optimization algorithm for design of trusses. *J Comput Des Eng*. 2019;**6**(1):49–59.
5. Kaveh A, Zolghadr A. Shape and size optimization of truss structures with frequency constraints using enhanced charged system search algorithm. 2011.
6. Kennedy J, Eberhart R. Particle swarm optimization. In: *Proc ICNN'95 Int Conf Neural Netw*. 1995:1942–48.
7. Kaveh A, Ghazaan MI. Optimum design of skeletal structures using PSO-based algorithms. *Period Polytech Civ Eng*. 2017;**61**(2):184–95.
8. Kaveh A, Mahdavi VR. A hybrid CBO–PSO algorithm for optimal design of truss structures with dynamic constraints. *Appl Soft Comput*. 2015;**34**:260–73.
9. Cheng W, Qin Y, Gu J, et al. Improved quantum-behaved particle swarm method for optimizing complex thin plate structure. *KSCE J Civ Eng*. 2023;**27**(4):1672–81.
10. dos Santos Coelho L. A quantum particle swarm optimizer with chaotic mutation operator. *Chaos Solitons Fractals*. 2008;**37**(5):1409–18.
11. Gholizadeh S, Moghadas RK. Performance-based optimum design of steel frames by an improved quantum particle swarm optimization. *Adv Struct Eng*. 2014;**17**(2):143–56.
12. Talatahari S, Azizi M. Optimal design of real-size building structures using quantum-behaved developed swarm optimizer. *Struct Des Tall Spec Build*. 2020;**29**(11):e1747.
13. Talatahari S, Azizi M, Toloo M, Baghalzadeh Shishehgarkhaneh M. Optimization of large-scale frame structures using fuzzy adaptive quantum inspired charged system search. *Int J Steel Struct*. 2022;**22**(3):686–707.
14. Xu Z, Cui Y, Li B. A quantum-inspired particle swarm optimization for sizing optimization of truss structures. In: *J Phys Conf Ser*. 2021:042127.

15. Sun J, Wu X, Palade V, Fang W, Lai CH, Xu W. Convergence analysis and improvements of quantum-behaved particle swarm optimization. *Inf Sci.* 2012;**193**:81–103.
16. Fang W, Sun J, Ding Y, Wu X, Xu W. A review of quantum-behaved particle swarm optimization. *IETE Tech Rev.* 2010;**27**(4):336–48.
17. Limane A, Zitouni F, Harous S, et al. Chaos-enhanced metaheuristics: classification, comparison, and convergence analysis. *Complex Intell Syst.* 2025;**11**(3):177.
18. Kaveh A, Sheikholeslami R, Talatahari S, Keshvari-Ilkhichi M. Chaotic swarming of particles: a new method for size optimization of truss structures. *Adv Eng Softw.* 2014;**67**:136–47.
19. Kaveh A, Dadras A, Montazeran A. Chaotic enhanced colliding bodies algorithms for size optimization of truss structures. *Acta Mech.* 2018;**229**(7):2883–907.
20. Kaveh A, Yousefpoor H. *Chaotic Meta-Heuristic Algorithms for Optimal Design of Structures*. Springer; 2024.
21. Turgut OE, Turgut MS, Coban MT. Chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations. *Comput Math Appl.* 2014;**68**(4):508–30.
22. Mariani VC, Duck ARK, Guerra FA, dos Santos Coelho L, Rao RV. A chaotic quantum-behaved particle swarm approach applied to optimization of heat exchangers. *Appl Therm Eng.* 2012;**42**:119–28.
23. Wang S. Wind resistance performance optimization of PSO algorithm in skyscrapers design. *Sustain Build.* 2025;**8**:2.
24. Kaveh A, Biabani Hamedani K. Set theoretical variants of optimization algorithms for optimal design of skeletal structures. *Int J Optim Civ Eng.* 2020;**10**(4):669–700.
25. Kaveh A, Yousefpoor H. Chaotically enhanced meta-heuristic algorithms for optimal design of truss structures with frequency constraints. *Period Polytech Civ Eng.* 2022;**66**(3):900–21.