Adaptive search space decomposition method for pre- and post- buckling analyses of space truss structures

It is advisable to refer to the publisher's version if you intend to cite from the work. See Guidance on citing.

www.reading.ac.uk/centaur

Adaptive search space decomposition method for pre- and post-buckling analyses of space truss structures

Varun Ojha [a,d,*], Bartolomeo Pantò [b], Giuseppe Nicosia [c]

[a] *University of Reading, Reading, UK*
[b] *Durham University, Durham, UK*
[c] *University of Catania, Catania, Italy*
[d] *Newcastle University, Newcastle, UK*

ARTICLE INFO

ABSTRACT

The paper proposes a novel adaptive search space decomposition method and a novel gradient-free optimization-based formulation for the pre- and post-buckling analyses of space truss structures. Space trusses are often employed in structural engineering to build large steel constructions, such as bridges and domes, whose structural response is characterized by large displacements. Therefore, these structures are vulnerable to progressive collapses due to local or global buckling effects, leading to sudden failures. The method proposed in this paper allows the analysis of the load-equilibrium path of truss structures to permanent and variable loading, including stable and unstable equilibrium stages and explicitly considering geometric nonlinearities. The goal of this work is to determine these equilibrium stages via optimization of the Lagrangian kinematic parameters of the system, determining the global equilibrium. However, this optimization problem is non-trivial due to the undefined parameter domain and the sensitivity and interaction among the Lagrangian parameters. Therefore, we propose to formulate this problem as a nonlinear, multimodal, unconstrained, continuous optimization problem and develop a novel adaptive search space decomposition method, which progressively and adaptively re-defines the search domain (hypersphere) to evaluate the equilibrium of the system using a gradient-free optimization algorithm. We tackle three benchmark problems and evaluate a medium-sized test representing a real structural problem in this paper. The results are compared to those available in the literature regarding displacement–load curves and deformed configurations. The accuracy and robustness of the adopted methodology show a high potential for gradient-free algorithms to analyze space truss structures.

1. Introduction

Space truss structures represent one of the most extensively used structural typologies in civil engineering to build non-ordinary steel structures such as bridges, large span arches, domes, and transmission towers. These structures generally show large displacements even under service loadings, and their ultimate response is often characterized by snap-through post buckling mechanisms in which the structure passes rapidly from an equilibrium state to a non-adjacent equilibrium configuration (Hrinda, 2010). Furthermore, some catastrophic events (Martin and Delatte, 2001) and numerical studies (Smith, 1984; Murtha-Smith, 1988; Blandford, 1996) revealed that truss structures are prone to activate progressive collapses due to equilibrium instability, leading to brittle and sudden failure, which may lead to significant economic and human losses. Hence, complex nonlinear analyses considering the mechanical and geometrical nonlinearities that allow for a complete equilibrium path of the structure are needed to assess the structural

robustness of truss structures against progressive collapses (Gilbert and Tyas, 2003).

In the last decades, many structural optimization procedures, based on linear programming (Gilbert and Tyas, 2003; Tyas et al., 2006; Horst and Tuy, 2013), genetic algorithms (Saka, 2007; Hasançebi et al., 2009; Huang et al., 2019; Lute et al., 2009), or iterative finite element procedures (Hrinda and Nguyen, 2008), have been developed to research optimum design of large truss structures through exploring optimum topology (Yildiz, 2013) or geometry of the system and optimum cross-sectional dimensions for the members. In these methods, the objective function has generally been the total weight of the structure, and the material tensile and compressive strengths have been considered as constraints, eventually considering the elements' local buckling. However, these methods do not provide information on the nonlinear post-buckling response of the structure and its effective safety level. Authors have also used neural networks and optimization algorithms

---

for other structural engineering problems such as structural damage detection (Tran-Ngoc et al., 2021; Khatir et al., 2021), and simulation of fracture mechanics (Khatir and Wahab, 2019), and monitoring of structural health (Khatir et al., 2019).

Many authors proposed mathematical optimization algorithms based on mathematical programming to perform the limit analysis on structural systems such as frictional rigid-block assemblages (Gilbert et al., 2006; Ferris and Tin-Loi, 2001; Baggio and Trovalusci, 2000) or elasto-plastic Von-Mises steel structures (Bisbos and Pardalos, 2007; Bisbos et al., 2005). These approaches provide valuable information about the failure mechanism and corresponding load multiplier. However, they are generally formulated under the hypothesis of small displacements and do not consider the incremental loading process. Thus, they cannot be applied for buckling analyses.

On the other hand, incremental procedures within the finite element framework, based on the Arc-length method (Crisfield, 1991, 1996; Memon and Xiao-zu, 2004) combined with iterative Newton–Raphson techniques (De Borst et al., 2012; Bathe, 2006), represent effective tools for performing nonlinear and buckling analysis considering the material and geometric nonlinearities. According to these strategies, an iterative stepwise linearization of the nonlinear structural behavior is considered. At each iteration, the tangent stiffness matrix and the geometrical stiffness matrix describe the mechanical and geometrical nonlinearities of the system, respectively. Thus, iterative Newton–Raphson procedures are effectively employed to evaluate the nonlinear response of 2D and 3D large structures, even in the presence of damage-plasticity constitutive laws (Chen and Han, 2007; Macorini and Izzuddin, 2011). However, they require a significant computational effort to update the stiffness matrices at each analysis iteration. Moreover, close to the critical points of the structural response, where the equilibrium configuration changes from stable to unstable (or vice-versa), numerical issues may appear, significantly increasing the number of iterations required to get the solution or leading to the divergence of the solution.

A few learning algorithms for neural networks have been developed and applied for the nonlinear modeling of mechanical systems (Memon and Xiao-zu, 2004) and for approximation of nonlinear behavior of structures (Wang and Adeli, 2015), covering the drawbacks of Newton–Raphson like procedures. These algorithms are based on quasi-Newton methods (Li and Yan, 1995; Geradin et al., 1981) that estimate the inverse Hessian of an objective function from the gradient to enable Newton-like optimization algorithms. Thus, not requiring the assembling of global stiffness matrices. Despite their potentialities, these approaches have not yet been exploited for the nonlinear assessment of large structural systems.

The main goal of the paper is to propose a novel formulation of the nonlinear pre- and post-buckling analysis of space truss structure using gradient-free optimization algorithms. In this formulation, we subject a space truss structure (system) to large displacements and optimize the Lagrangian kinematic parameters (displacement) and the load multiplier of the system to guarantee the global equilibrium of the system. In our formulation, we define an objective function (the global equilibrium) in terms of global unbalance determined as the difference between the vectors of external and internal forces on the system. To the best of the authors' knowledge, this is the first work performing incremental (multistep) analysis of structures using gradient-free global optimization algorithms.

The presented optimization problem can be classified as a nonlinear, multimodal, unconstrained, continuous optimization problem. It is a challenging problem to solve as the search space for displacement and load multiplier variables has an undefined upper bound. Additionally, the search landscape poses significant challenges to existing continuous optimization algorithms. Hence, this problem could be considered as a testbench for optimization algorithms. The implementation of this research work is available on our GitHub page.[1]

---

[1] https://github.com/vojha-code/Hypershpere-Search

In this paper, we show experimental approaches considered to decompose and define search space in order to solve this optimization problem. Such experimental approaches allowed us to propose a novel search space decomposition method that we call *adaptive search space decomposition* method, which progressively and adaptively defines new hyperspheres (a bounded search space) for solving the optimization problem. The method is based on iteratively finding new centers for new hypersphere such that the centers follow the global equilibrium path a space truss structure. Thus allowing the study of the equilibrium stages and potential snap-through mechanisms of a space truss structure.

In this work, three benchmark problems investigated in the literature through Newton-like approaches (Crisfield, 1991) are considered to assess the accuracy and performances of the selected set of optimization algorithms and proposed adaptive search space decomposition method. Each problem is solved using ad-hoc differential evolution algorithms (a gradient free optimization algorithms), comparing the results with those available in the literature regarding displacement load capacity curves and failure deformed configurations. Finally, a medium-size structure (a test problem on a 3D reticular beam), designed in the literature following the criteria of minimum weight, is solved to prove the applicability of the proposed procedure and to solve relevant problems in structural engineering.

The rest of the paper is organized as follows: First we introduce the structural model and its formulation as an optimization problem in Sections 2.1 and 2.2. Then the search space decomposition methods are discussed in methodology Section 3. The adaptive search space decomposition method is described in Section 3.2. The results on three benchmark problems and a test problem are presented and analyzed in Section 4. Finally, discussions and conclusions are presented in Sections 5 and 6.

2. Space truss structure problem

2.1. Kinematic and static of space truss structure

The system kinematics described by model adopt a Lagrangian description by considering large displacements and small strain hypotheses. The degrees of freedom are assumed to be coincident with the $3N$ absolute displacements of the N free nodes of the truss system, referred to as the global reference system on x–y–z axes (Fig. 1). The displacements of the generic node n_k with $k = 1, \ldots, N$, are collected in a vector $\mathbf{u}_k = [u_{k,x}, u_{k,y}, u_{k,z}]$ while the corresponding dual nodal forces are collected in a vector $\mathbf{f}_k = \mathbf{f}_{0k} + \lambda \mathbf{f}_k[f_{0k,x}, f_{0k,y}, f_{0k,z}] + \lambda[f_{k,x}, f_{k,y}, f_{k,z}]$ where $\mathbf{f}_{0k}$ represents the permanent loads and $\mathbf{f}_k$ represents variable loads, which are amplified by the load multiplier λ (Fig. 1). Each node can be connected with an arbitrary number (m) of nodes ($n_{k,1}, \ldots, n_{k,m}$) by as many trusses ($t_1, \ldots, t_m$) as shown in Fig. 1. The generic pth truss ($p = 1, \ldots, m$) connecting the node n_k with the node $n_{k,p}$ is represented in Fig. 2, where the vector $\mathbf{X}_k$ and $\mathbf{X}_{k,p}$ represent the initial coordinates of the truss end nodes, $\mathbf{u}_{k,p}$ the displacement vector of the node $n_{k,p}$ and $\mathbf{q}_{k,p}$ the nodal axial forces acting on the truss.

According to the hypothesis of small deformations (Crisfield, 1991), the truss deformation (ϵ_p) is evaluated considering the engineering strain definition as:

$$\delta_{k,p} = \frac{l_{k,p}}{L_{k,p}} - 1 \tag{1}$$

where $L_{k,p} = \|\mathbf{X}_{k,p} - \mathbf{X}_k\|$ and $l_{k,p} = \|\mathbf{X}_{k,p} + \mathbf{u}_{k,p} - \mathbf{X}_k - \mathbf{u}_k\|$ are the initial and current length of the truss, respectively and $\| \cdot \|$ indicates the Euclidean norm.

Given k_p the axial stiffness of the truss and $\mathbf{i}_{k,p}$ the unitary vector identifying its current orientation, the internal force can be written as follows:

$$\mathbf{q}_{k,p} = k_p \delta_{k,p} \mathbf{i}_{k,p} = k_p \left(\frac{\|\mathbf{X}_{k,p} - \mathbf{X}_k\|}{\|\mathbf{X}_{k,p} + \mathbf{u}_{k,p} - \mathbf{X}_k - \mathbf{u}_k\|} - 1 \right) \mathbf{i}_{k,p}, \tag{2}$$
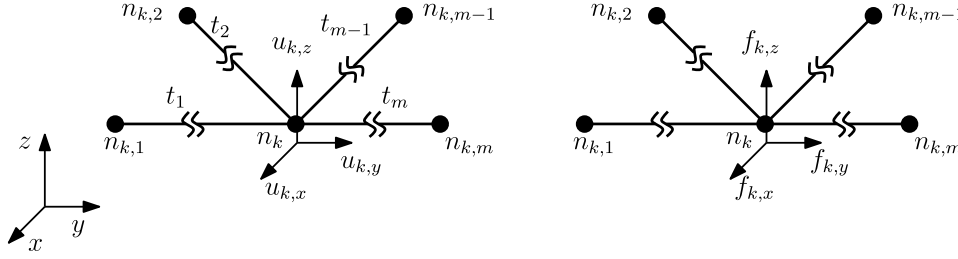
Fig. 1. Independent displacements (*Left*) and dual forces (*Right*) of a generic node n_k connecting m trusses $t_1, t_2, \ldots, t_m$ with nodes $n_{k,1}, n_{k,2}, \ldots, n_{k,m}$. The node n_k has its displacements $u_{k,x}, u_{k,y}, u_{k,z}$ and corresponding forces $f_{k,x}, f_{k,y}, f_{k,z}$ along x, y and z dimensions of the space.
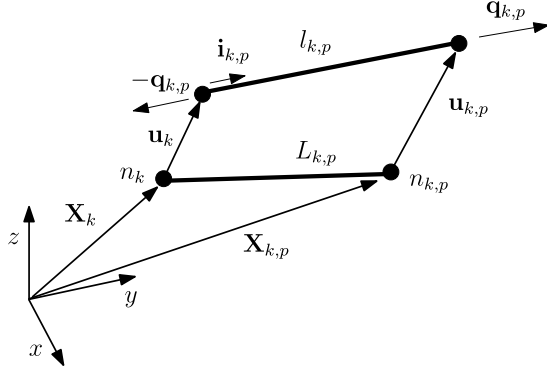

Fig. 2. Kinematic and internal forces of a generic pth truss connecting node n_k with node $n_{k,p}$. The truss has an initial length $L_{k,p}$ and its nodes' initial coordinates are $\mathbf{X}_k$ and $\mathbf{X}_{k,p}$. The axial nodal force of the truss is $\mathbf{q}_p$, its current length and orientation are $l_{k,p}$ and $\mathbf{i}_{k,p}$ and nodal displacements are $\mathbf{u}_k$ and $\mathbf{u}_{k,p}$.

where

$$\mathbf{i}_{k,p} = \frac{(\mathbf{X}_{k,p} + \mathbf{u}_{k,p} - \mathbf{X}_k - \mathbf{u}_k)}{l_{k,p}}. \tag{3}$$

The equilibrium between the external and internal forces at the generic node n_k can be expressed as follows:

$$\mathbf{f}_{0k} + \lambda \mathbf{f}_k + \sum_{p=1}^{m} \mathbf{q}_{k,p} = \mathbf{0} \quad \text{for each } k = 1, \ldots, N. \tag{4}$$

Note that this work does not consider material nonlinearities (i.e., a linear elastic constitutive law characterizes structural elements). Therefore, truss stiffnesses assume constant values during the analysis.

2.2. The global equilibrium as an optimization problem

In this Section, the global equilibrium of the truss system is mathematically characterized and formulated as an optimization problem whose free variables are represented by the Lagrangian parameters of the system, collected in the vector $\mathbf{u} = [\mathbf{u}_1, \ldots, \mathbf{u}_N]$ and the load multiplier λ. The objective function ($\mathcal{F}$) that the optimization algorithms aim to minimize is expressed in terms of global unbalance as follows:

$$\mathcal{F}(\mathbf{u}, \lambda) = \frac{\sqrt{\mathbf{R}_1^2 + \mathbf{R}_2^2 + \cdots \mathbf{R}_N^2}}{\sqrt{\mathbf{f}_1^2 + \mathbf{f}_2^2 + \cdots + \mathbf{f}_N^2}} = 0, \tag{5}$$

where $\mathbf{R}_k, k = 1, \ldots, N$ is the unbalance vector of the kth node, which can be easily obtained from Eq. (4). A value 0 is the global minima of Eq. (5) indicates an equilibrium state of the structure.

The incremental multistep procedure shown in Fig. 3, known in the literature as arc-length method (Crisfield, 1991), can be adapted to draw the equilibrium path of the structure in the $d - \lambda$ space, where d is a chosen contról point, which can coincide with the physical displacement of a structural node or a function of a number of displacements.
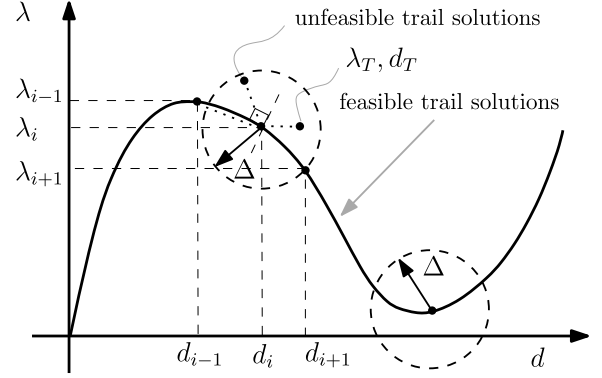

Fig. 3. Multistep procedure. Variables d, λ, and Δ respectively are control point, load multiplier, and discrete steps along the control point dimension for producing an equilibrium path (curve in black). Any optimal point (a solution that satisfy Eq. (5); otherwise sub-optimal solution) on this curve and within a radius Δ that satisfies constraints in Eqs. (6) and (7) is a feasible solution in terms of its characteristics being within a tolerance proximity to the equilibrium path formed by a vertical hypothesis load. Solutions not within this tolerance proximity defined by Eqs. (6) and (7) are unfeasible or belong to a path concerning non-vertical hypothesis load.

For instance, the control point can be assumed coincident with the Euclidean norm of the vector $\mathbf{u}(d = \|\mathbf{u}\|)$. This multistep arc-length procedure is framed within the equilibrium optimization problem, by considering the equality constraint reported in Eqs. (6) and (7), where d_i and λ_i are the control point and load multiplier at the generic ith equilibrated configuration belonging to the equilibrium path, and $d_T - \lambda_T$ the corresponding trial values:

$$\sqrt{((d_T - d_i)^2 + (\lambda_T - \lambda_i)^2)} = \Delta, \tag{6}$$

where Δ is a parameter of the procedure governing the discretization of the equilibrium path; furthermore, if $i > 1$, a second constrain is considered as:

$$(d_T - d_i)(d_{i-1} - d_i) + (\lambda_T - \lambda_i)(\lambda_{i-1} - \lambda_i) \leq 0, \tag{7}$$

where, d_{i-1} and λ_{i-1} are the values of the control point and load multiplier at step $i - 1$.

2.3. Challenges of the global equilibrium optimization

The challenging issue with the optimization problem defined in Section 2.2 is the lack of a properly defined search domain for the free variables displacement and load multiplier, to be pursued using gradient-free optimization algorithms. The search domain for the free variables is only the initial elastic prediction of the structural response and simple equilibrium considerations.

The unavailability of a straightforward search domain definition for the space truss structure makes it an ill-posed problem. However, the gradient-free optimization algorithms allow experimenting with an

intuitive guess of the search domain. Hence, we formulate this space truss structure buckling analysis problem as an optimization problem.

The geometric nonlinearities, high physical interaction and sensitivity among displacement variables, and numerous possible imbalance states for a particular load multiplier make this problem further challenging. This can be seen later in results Section 4 mentioned as sub-optimal and local minima solutions.

We, therefore, classify the optimization of Eq. (5) as a nonlinear, multimodal, unconstrained, continuous, minimization problem whose search landscape is complex containing many local and global minima, i.e., its search landscape is rugged.

In order to solve this optimization problem, first, an intuitive guess was made based on an initial elastic prediction of the structural response and simple equilibrium considerations. However, each displacement variable has its own domain, and fixing one min–max domain range to all variables complicates the search landscape, leading to an impossible optimization (demonstrated later in Section 4). Therefore, a number of strategies for search space decomposition discussed in Section 3 are used in this research to solve this problem as efficiently as possible.

In this paper, we show experimental approaches considered to decompose and define search space in order to solve this optimization problem. Furthermore, such experimental approaches allowed us to propose a novel adaptive search space decomposition method for multistep analyses, denoted as the *hypersphere search* method or *adaptive search space decomposition* method, representing an efficient and robust transposition of the arc-length method into the optimization framework.

3. Methodology

The procedure adopted in this study to solve the optimization problem defined in Section 2, is based on gradient-free algorithms and search space partitioning methods characterized by different levels of complexity. First, an *informed decomposition* of search space was based on the equilibrium-path curve (also referred to as baseline equilibrium path), available for some selected benchmarks, already investigated in the literature by adopting Newton–Raphson iterative procedures combined with arc-length methods (Hrinda, 2010; Crisfield, 1991; De Borst et al., 2012). Second, a more sophisticated *adaptive search space partition* technique is used.

3.1. Informed decomposition of search space

In this study, we consider two alternative options for decomposing the search space for this problem. First, we partition the control point displacement target into several discrete steps as the literature provided the information on the control point.

Second, in addition to partitioning the control point, we incrementally partition the space for all free displacement variables. Compared to the first option, the second option was more effective (as discussed later in Section 4) in finding solutions across the baseline equilibrium path shown in the literature (Crisfield, 1991, 1996). The performance of each optimization algorithm is discussed later in the results reported in Section 4.

These search domain partition strategies highlighted the strengths and weaknesses of the optimization algorithms in solving this challenging class of optimization problems. The goal of the space truss structures optimization was to find solutions across the known equilibrium path of the benchmark problems as dense as possible. Therefore, *denseness*, *coverage of the path*, the *number of optimal solutions*, and *speed of the convergence*, among others, were the main criteria for evaluating the effectiveness of our methodology in solving this nonlinear, unconstrained, continuous and multimodal, optimization problem.

3.2. Adaptive decomposition of search space

Instead of decomposing the problem search space in a pure discrete form mentioned in Section 3.1, a method for *adaptive decomposition of search space* is proposed in this Section. In this method, we automatically search hyperspheres that define suitable search space for the multi-step pre-and post buckling analysis of space truss structures.

This is an iterative procedure where the method starts with an initial seed hypersphere. The initial seed hypersphere is defined by a small range of the free displacement and load multiplier variables. This is to simulate a minute displacement of the structure, which is observed through the control point (node at which vertical load is applied) dimension.

For this seed hypersphere, the method at first evaluates a few trail solutions (e.g. five solutions). Then from all optimal solutions, it searches for a new hypersphere. This process continues until the max displacement of the control point of a problem is reached (see Fig. 4).

The proposed adaptive search space decomposition method, called *hypersphere search algorithm*, finds the most suitable hyperspheres to support gradient-free optimization algorithms to find the global equilibrium effectively. The method comprises the following steps:

Step 1. The first step is to initialize a hypersphere H_k with center in $\mathbf{c}_k$ for $k = 1$ and radius r. The search of the centers is controlled by a 2-dimension vector $[d, \lambda]$, where d is the initial displacement control point (generally zero) and λ is the corresponding load multiplier that guarantees the global equilibrium solution. A gradient-free optimization algorithm then evaluates one or more trial solutions within this hypersphere.

Step 2. The second center $\mathbf{c}_k$ for $k = 2$ is the farthest point in the previous hypersphere H_{k-1} from its center $\mathbf{c}_{k-1}$ among all trial solutions in Step 1 that are optimal (a value 10^{-5}). In other words, the second center $\mathbf{c}_k$ is set to the solution that has max value of d and is optimal. Then the hypersphere H_k for $k = 2$ is formed around the center $\mathbf{c}_k$ with radius r. Similar to Step 1, one or more trial solutions within this new hypersphere H_k are evaluated for obtaining the hypersphere H_{k+1}.

Step 3. The next center $\mathbf{c}_k$ for $k = 3$ is the farthest point from the center $\mathbf{c}_{k-1}$ among all trial solution points $\mathbf{c}_i$ obtained in the hypersphere H_{k-1} and is the point that satisfies the following condition:

$$\mathbf{b}_i \cdot \mathbf{a} < 0 \quad \text{and} \quad \|\mathbf{b}_i\| > \|\mathbf{b}_j\|, \quad i \neq j \text{ for all } i \text{ and } j \text{ in } H_{k-1}, \quad (8)$$

where i and j are indexes of trial solutions in the hypersphere H_{k-1}, vector $\mathbf{a} = \mathbf{c}_{k-2} - \mathbf{c}_{k-1}$, and $\mathbf{b}_i = \mathbf{c}_i - \mathbf{c}_{k-1}$. Once the center $\mathbf{c}_k$ is obtained, a hypersphere H_k is formed with a radius r. This step is shown in Fig. 4.

Step 4. Stop if k is reached a max number of hypersphere trails or a center reached a predefined max displacement $d_{\max}$. Else set $k = k + 1$ go to Step 3 to find other hyperspheres.

The user-defined hyper-parameter of this algorithm is the radius r, the number of trial solutions needed to be produced in a hypersphere, and the stopping criteria. While the radius r depends on the user set value, the number of trial solution evaluation depend on the experimentation. The stopping criteria can be set to either an arbitrary sufficient number of solutions that may approximately cover equilibrium path obtained by the arc-length method, or to a maximum displacement value of the control point dimension obtained by arc-length method (or an intuitive guess that could be made about the buckling of the structure).

3.3. Experiment setup

As described in Section 3, the optimization is performed by adopting gradient-free algorithms and search of the solution was supported by space partitioning methods. In this Section, the setups of these
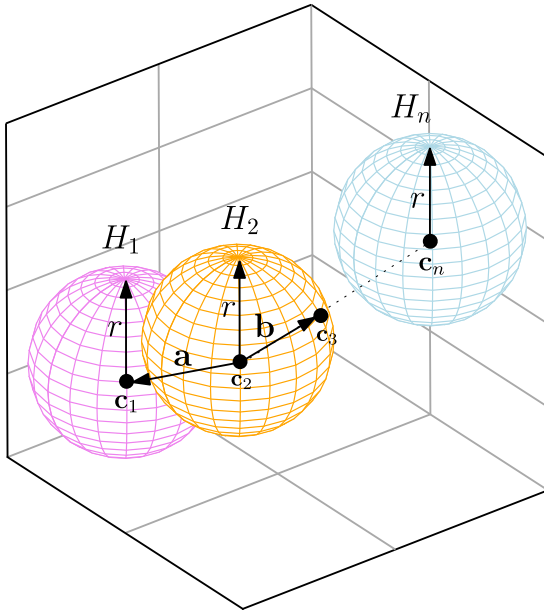
Fig. 4. Adaptive search space decomposition method (i.e., hypersphere search algorithm). An initial hypersphere H_1 is first constructed based on an input radius r and the center c_1. Hyperspheres $H_2, \ldots, H_n$ are discovered by the algorithm automatically by taking a boundary point (i.e., a solution within a hypersphere that has the largest distance from the previous hypersphere's center). For example, center c_2 has a distance **a** from center c_1 and center c_3 has a distance **b** from center c_2.

approaches are described. This methodology is validated on three benchmarks already investigated in the literature by means of iterative Newton–Raphson procedures arc-length methods and tested on a medium-sized test problem. The three benchmarks consisted in 3D shallow truss structures, for which the baseline is available from the literature, enabling the setting of the min–max displacement and load multiplier value. However, these min–max values were an intuitive guess of the range, i.e., the maximum displacement that the control point displacement can take and the range of variability of external load. Therefore, the domain range was initially set to the min and max values of these benchmark problems in the literature. However, optimization algorithms could not find solutions for the first two benchmark problems across the equilibrium path shown in the literature, as discussed later in results Section 4. Hence, a strategy to decompose the search space was adopted. Finally, the results obtained for the three considered benchmarks were compared with the baseline from the literature.

3.3.1. Gradient-free algorithms for the equilibrium path analysis

This section briefly explains each optimization algorithm used for the optimization. The first set of algorithms chosen were single solution optimization algorithms DIRECT and Simulated Annealing (SA). Both algorithms work on a single solution, and they can provide a quick understanding of the problem to be optimized. DIRECT is a pattern search algorithm of a class of Lipschitz optimization methods that relies on a dividing rectangle principle, hence it is called DIRECT (Jones et al., 1993). SA, a heuristic-based gradient-free algorithm (Kirkpatrick et al., 1983), probabilistically accepts solutions while exploring the neighborhood of a search point. The probability of acceptance depends on an energy factor called temperature that goes to zero from a certain initial value in a controlled manner by a factor reducing it in each iteration. This is analogous to the annealing process in metallurgy, hence the name Simulated Annealing.

The second class of algorithms were swarm intelligence-inspired population-based optimization algorithms: Artificial Bee Colony Optimization (ABC), Ant Colony Optimization (ACO), and Particle Swarm

Optimization (PSO). These algorithms emulate the foraging behavior of the swarm. For example, ABC follows the foraging behavior of honeybees (Karaboga and Basturk, 2008), while ACO follows ants (Socha and Dorigo, 2008), and PSO follows the behavior of a flock of birds or a school of fish (Kennedy and Eberhart, 1995; Clerc, 1999). Each of these algorithms has its own framework and convergence properties. For example, ABC maintains three types of honeybees: employed, onlooker, and scout bees. ACO uses Gaussian distribution for its pheromone and solution matrix update. PSO uses an inertia factor (Kennedy and Eberhart, 1995) (PSO-Std) or constriction factor (Clerc, 1999) (PSO-Const) to update the velocity and position of particles. However, common features are updating a solution vector for a defined objective function that governs the solution's quality that needs an update in every generation of the optimization.

The final class of algorithms and the main focus of this research is the differential evolution (DE) algorithm from the family of evolutionary algorithms, which uses mutation, crossover, and selection operations on the solution vectors (Storn and Price, 1997). There are several versions of the DE algorithm (Opara and Arabas, 2019). After an initial performance evaluation (in terms of speed of convergence and diversity of solutions in multiple runs) of a set of 12 versions of DE strategies available in Scipy (Scipy, 2021), this paper selects the two most promising DE versions: DE/rand/1/bin and DE/best/2/bin. These two versions vary in how they select a base vector [randomly (rand) or best] and the number of differences of vectors (one difference vector or two differences of vectors) for the mutation operation in DE. However, both do a binomial (bin) crossover.

3.3.2. Setup of hyperparameters of the algorithms

Initial trials on benchmark problems were performed to fixate hyperparameter values of the algorithms. For DIRECT, except for termination criteria, there was no other hyperparameter to set. Moreover, DIRECT is a single solution based optimization algorithm. Similar to DIRECT, SA is a single solution optimization algorithm. However, SA has its other hyperparameters temperature and initial temperature reduction rate was set to 0.1 and 0.99.

The swarm-based algorithms population size was set to 50, where other hyperparameters specific to each algorithm were as follows: the hyperparameters of ABC was abandonment limit and acceleration coefficient upper bound and they were set to 0.6 and 1; the hyperparameters of ACO sample size, selection pressure, and deviation-distance ratio, respectively set to 40, 0.5, and 1; hyperparameters of PSO-Std inertia weight, inertia weight damping ratio, personal learning coefficient, global learning coefficient respectively set to 1, 0.99, 1.5, and 2.0; and hyperparameters of PSO-Const personal learning coefficient and global learning coefficient were equal to *inertia weight* $\times \phi$, where inertia weight $= 2/(\phi - 2 + \sqrt{\phi^2 - 4 \times \phi})$ for ϕ and inertia weight damping set to 2.05 and 1. Finally, the hyperparameters of DE algorithm scaling factor bound and crossover rate were set to [0.2, 0.9] and 0.9. Two versions of mutation "DE/rand/1/bin" and "DE/best/2/bin were used.

The initial radius for the adaptive search space decomposition method (see Section 3.2) was set to 5 and termination condition was set to max displacement in the baseline of the problems. The rest of the experiment setting about benchmark problems are described in results in Section 4 as when a they are introduced.

4. Results and analysis

4.1. Benchmark 1: Eight-member shallow truss structure

The first benchmark problem was a shallow truss structure comprising eight members fully restrained at the base and connected to each other by a central node (Fig. 5). Each truss has a horizontal length of 12 700 mm and a vertical rise of 1000 mm. All the trusses have the same cross-section area, equal to 6450 mm², and Young's modulus of 70 MPa.
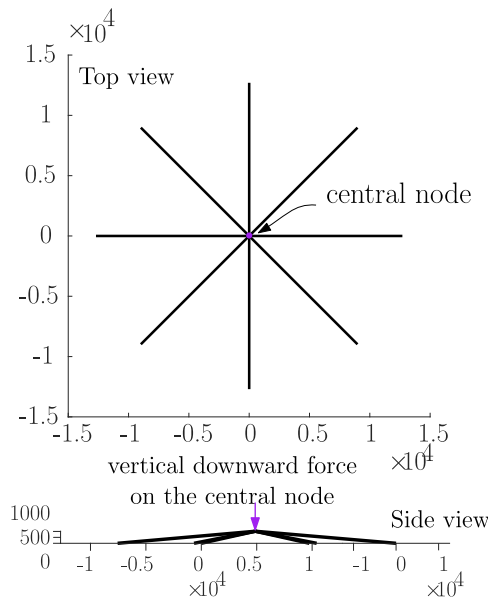
Fig. 5. Eight-Member Shallow Truss Structure: Top view and Side view (dimensions in mm). The central node of the structure is indicated by a lighter color dot in Top view and a light color arrow pointer in Side view. The arrow on Side view indicates a vertical downward external force of 4.45 kN applied on the central node.

The structure is loaded with a concentrated vertical force of 4.45 kN applied at the central node.

The search space for the eight-member truss structure has three degrees of freedom (DoF) corresponding to the three translations of the central point ($N = 1$). Therefore, four variables were subjected to optimization, including the control point of the displacement variable and one load multiplier variable. The domain range for the experiments for the displacement was [0 mm, 3000 mm], and for the load multiplier, it was set to [−0.2, 1]. All select algorithms were applied to optimize the eight-member truss structure on the full domain for initial trials. Initially, 1000 solutions were evaluated. That is, the algorithms were run for 1000 instances. All algorithms for all 1000 trials converged to a global optimum, i.e., the precision of the order of 1×10^{-5}. The convergence speed and variance of the algorithms are shown in Fig. 6.

Fig. 6 shows that ABC converged the fastest, followed by DE/best/2/ bin, DIRECT, PSO-Std, PSO-Constriction (PSO-Const), DE/rand/1/bin, ACO, and SA. However, this performance is significant only if the algorithms were able to obtain solutions across the equilibrium path— Fig. 7 plots solutions obtained by the algorithms on a control-point and load multiplier axis to verify solutions. ABC, DIRECT, and ACO were

not able to discover solutions across the equilibrium path. However, the performances of DIRECT and ACO were better than the performance of ABC. ABC failed to find any solution on the path except for a value of 0.0. The other algorithms, SA, PSO-Const, PSO Std, DE/best/2/bin, DE/rand/1/bin, successfully covered the whole equilibrium path in 1000 trials. The continuous line reported in the graphs of Figs. 7 and 8 represents the analytical solution from the literature (Hrinda, 2010; Crisfield, 1991; De Borst et al., 2012).

The properties and the framework of the algorithms played a role in such characteristics of the solutions. For example, the DIRECT algorithm divides the rectangle and starts from the initial range, and partitions are based on the first solution obtained for the objective function for a given range. Moreover, since it has a deterministic approach toward optimization, it seemingly finds a solution at the center of the value of the first variable (load multiplier). Hence, in all 1000 trials, it found the exact same solution every time.

The other best-performing algorithms SA, PSO-Const, PSO Std, DE/best/2/bin, and DE/rand/1/bin, use uniform distribution to initialize the solutions. This has enabled them to cover the full range of the domain uniformly. On the other hand, ACO's new solutions are sampled from a Gaussian distribution, leading to the solutions following the central tendency (see Fig. 7).

Since DIRECT and ACO were unable to cover the whole of the equilibrium path, the discrete control-point search space trials were performed to evaluate if these algorithms have the potential to find all other solutions for this problem. Both algorithms were able to cover the equilibrium path (see Fig. 8). However, ACO could densely cover the equilibrium path, whereas DIRECT shows sparsity for the higher displacement and load multiplier values. These initial results helped select fewer best-performing algorithms to optimize the second benchmark's problem of the space truss structures.

4.2. Benchmark 2: Sixteen-member shallow truss structure

The second benchmark problem was the Sixteen-Member Shallow Truss Structure (Fig. 9). The structure has four constraint nodes and five free nodes ($N = 5$). Therefore, it is characterized by 15 DoF. The global dimensions in the plane are 254 mm in each direction, and the global vertical rise is 100 mm. All the trusses have the same cross-section area of 645 mm² and Young's modulus of 68 950 MPa. The structure is loaded with a concentrated vertical force of 4450 kN applied at the central node of the structure in Fig. 9.

4.2.1. Results of the informed search space decomposition methods

The search space for the sixteen-member truss structure comprises 16 variables to be optimized, including the 15 free displacements of the structure and the load multiplier variable. The domain range of
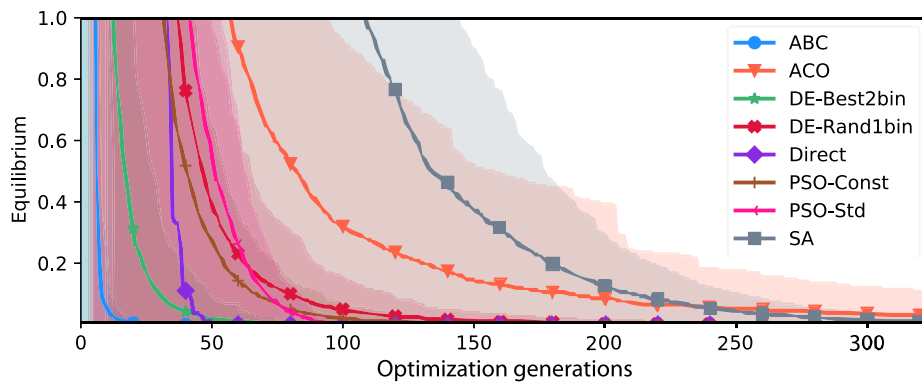

Fig. 6. Convergence profile of algorithms averaged over 1000 trail solutions, all with optimum equilibrium value 1×10^{-5}. Algorithms ABC, ACO, DE/best/2/bin, DE/rand/1/bin, Direct, PSO-Const, PSO-Std, and SA, respectively are in blue, orange, green, red, purple, brown, magenta, and gray; and they are respectively marked with symbols •, ▼, ★, ×, ♦, |, <, and ■. The shaded width of color around a line represents the standard deviation of the convergence.
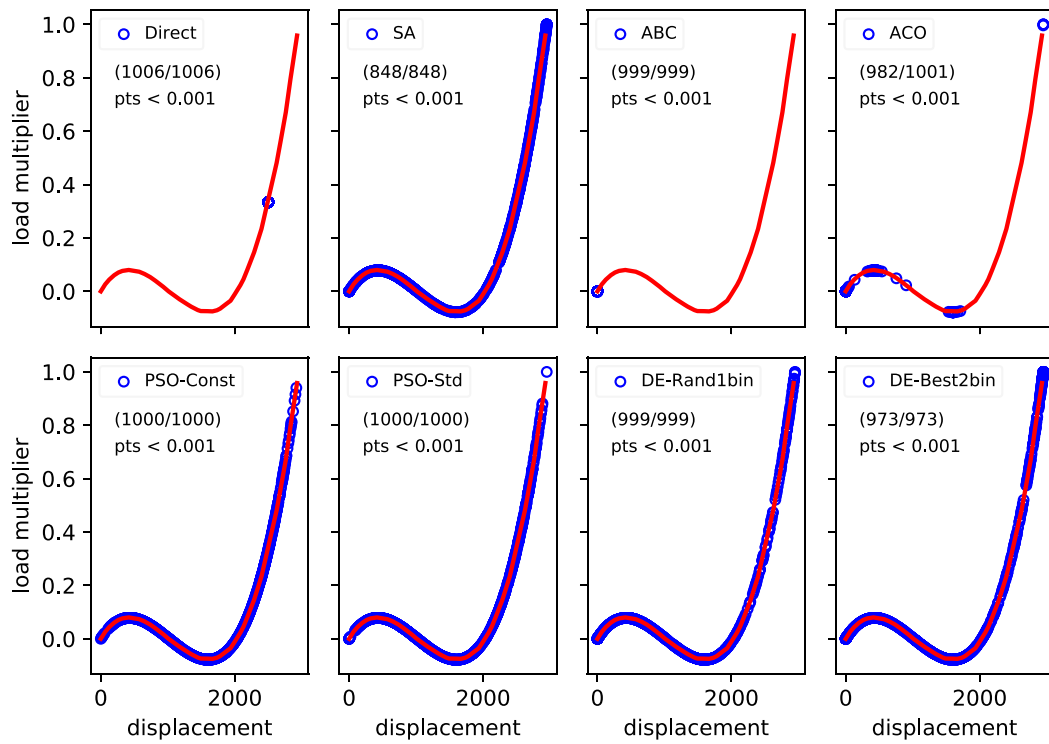
Fig. 7. Eight-Member truss structure optimized solutions generated by searching with a domain setting of $[0, 1]$ for load multiplier and $[0\,mm, 3000\,mm]$ for displacement variables (DoF), including the control point. Each plot is an x–y plane of control point displacement (x-axis) against load multiplier (y-axis). The solutions are shown by blue circles, and the red line represents the equilibrium path obtained by the standard Arc-length method mentioned in Section 2.2. From top left to bottom right, the algorithms are Direct (*top left*), SA, ABC, ACO, PSO-Std, PSO-Const, DE/rand/1/bin, and DE/best/2/bin (*bottom-right*). For each algorithm, the percentage of successful solutions (pts < 0.001) generated are indicated in the plots, e.g., DIRECT produced 100% successful points. i.e., 1006 solutions have a precision of 0.001 out of a total of 1006 trials. Similarly, SA produced 848/848 solutions with a precision of 0.001.
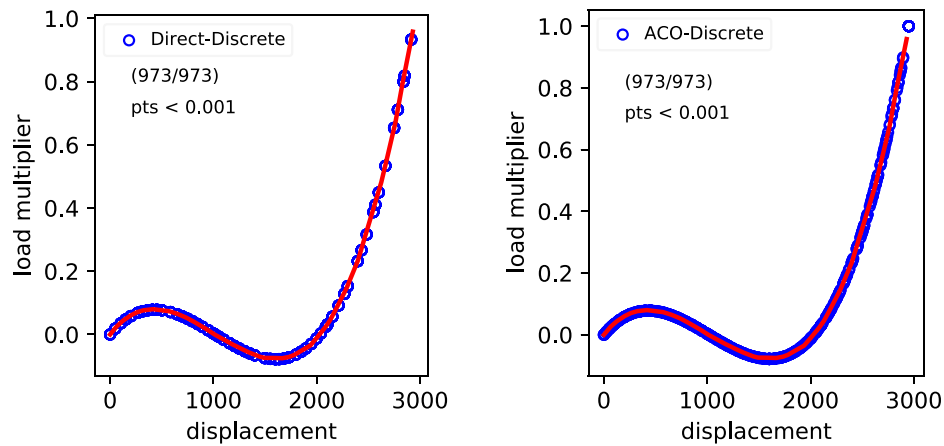

Fig. 8. Solutions of Direct (*Left*) and ACO (*Right*) generated by decomposing domain of only control point (variable related to central node), which produced discrete sets like $[0, 250], [250, 500], \dots, [2750, 3000]$. For each set, approx 100 trail solutions were evaluated.
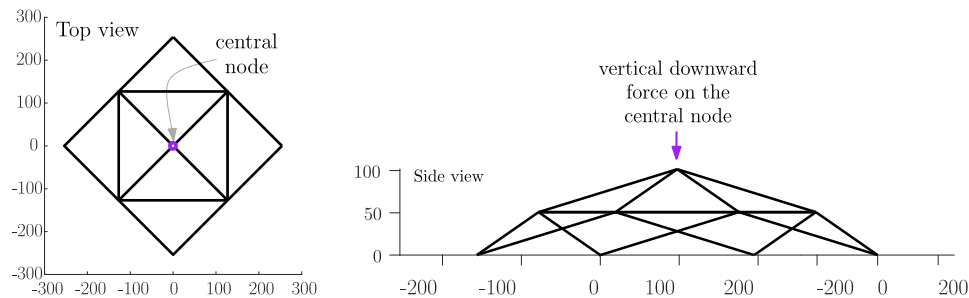

Fig. 9. Sixteen-Member Shallow Truss Structure. *Left:* Top View. *Right:* Side View. Th central node of the structure is indicated by a lighter color dot in Top view and a light color arrow pointer in Side view. The arrow on Side view indicates a vertical downward external force of $4450\,kN$ applied on the central node.
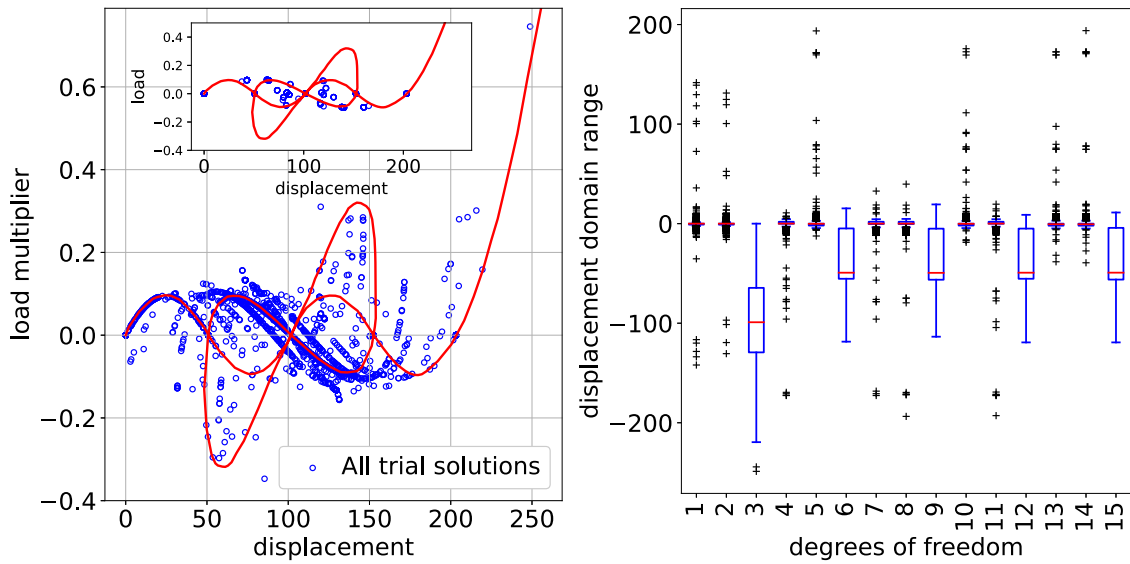
Fig. 10. Sixteen Members initial trial solutions. *Left:* initial trials for all discretized domain (inner-plot shows solution obtained for the experiment on full non-discretized domain [0 mm, 250 mm]). *Right:* Displacement variables (DoF) domain analysis and approximation.
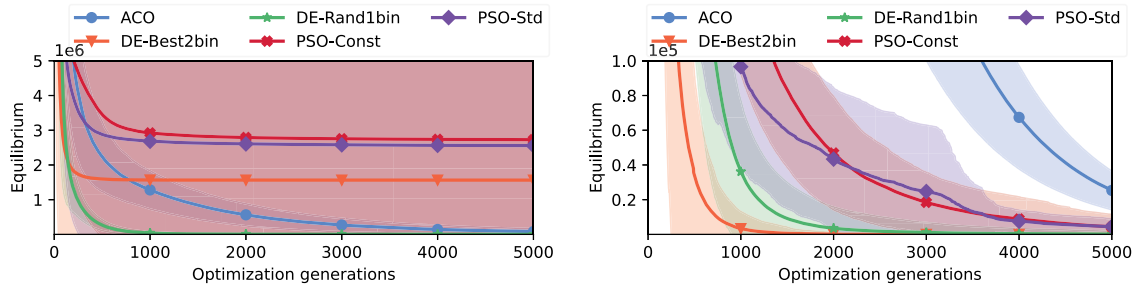

Fig. 11. Optimization of Sixteen-Member shallow truss structure using five algorithms: (*Left*) Convergence profile of algorithms averaged over solutions with equilibrium value of 500 and (*Right*) a closer snapshot of convergence profile with non-converging solutions filtered out. Algorithms ACO, DE/best/2/bin, DE/rand/1/bin, PSO-Const, and PSO-Std are in blue, orange, green, red, and purple; and they are marked with symbols •,▼,★,×,♦. The shaded width of color around a line represents the standard deviation of the convergence. The *x*-axis indicate number of generations and *y*-axis indicate objective as per Eq. (5).

the displacement variables was [0 mm, 250 mm], and the load multiplier domain was set to [−0.4, 0.85].

Full domain analysis. After setting an intuitive domain of the variables of the sixteen-member truss structure problem, the best performing algorithm of the first benchmark, i.e., DE/rand/1/bin was first applied to solve this sixteen-member truss structure. However, contrary to the results of the first benchmark, only a few solutions were obtained in all 1000 trials, and a tiny number of solutions were obtained on the equilibrium path (see the inner plot in Fig. 10). Therefore, a decomposed domain analysis procedure (see Section 3.1) was launched to find a suitable domain for respective variables. Again, the continuous red line in Fig. 10 indicates the analytical result from the literature, obtained under the hypothesis of vertical displacement of the central point.

Domain decomposition analysis. The decomposed domain analysis procedure partitioned the control point domain and the domain of other displacement variables. Based on the knowledge from the literature, the discretization of the control point dimension was straightforward. However, the discretization of the other variables could not be performed individually. Hence, an intuitive incremental guess was applied.

The grid space in Fig. 10(left) shows the procedure adopted to analyze suitable domains for these variables. For each grid space, 50 trial solutions were evaluated. All trial solutions that provided global-optimum value are plotted on an x–y plane of control point displacement and load multiplier in Fig. 10(left). The DE/rand/1/bin provided several solutions within the range [0 mm, 225 mm] of the

control point in good agreement with the analytical solution available in the literature. Other solutions, far from the analytical curve, are provided by this optimization algorithm as well. However, these solutions correspond to alternative equilibrium paths characterized by a non-vertical displacement of the central node.

Fig. 10(right) shows the displacement value distribution of each variable. The plot in Fig. 10(right) provides the min–max range of each displacement variable for which global optimum solutions could be obtained. The knowledge of this min–max range (i.e., displacement domain knowledge) was then be used by all optimization algorithms.

4.2.2. Results of the optimization on a learned search space

On the obtained knowledge of the domain shown in Fig. 10(right), ACO, DE/best/2/bin, DE/rand/1/bin, PSO-Const, and PSO-Std algorithms were applied (with the domain setting of each displacement variable) for stopping criteria of solution accuracy 1×10^{-5} and maximum iterations of 50 000. Fig. 11(left) shows each algorithm's average (of all solutions, including optimal and suboptimal) convergence profile. The average convergence profile rejects PSO-Const and PSO-Std algorithms. However, this only means that PSO obtained many poor (suboptimal) solutions (or even non-converging solutions) that affected the average convergence profile. A similar result was obtained for DE/best/2/bin, where poor solutions affected its average convergence profile. ACO and DE/rand/1/bin show more robustness in this scenario. They show the ability to escape local minima in most cases, which is observed from these two algorithms' smooth average convergence profile in Fig. 11(left).

Fig. 11(right) offers a microscopic view of the convergence of these algorithms. In Fig. 11(right), all solutions converged to a suboptimal equilibrium function [Eq. (5)] value of 500, selected after a few analyses of suboptimal points to see how best the algorithm's profile could be studied. The convergence profile is the average of all converging solutions to this suboptimal value, and all non-converging (all solutions stuck to local minima or did not converge at all) were filtered out. In this plot, DE/best/2/bin is the fastest converging algorithm, with all its select solutions being converged to global minima, i.e., 1×10^{-5} in less than 5000 iterations.

Similarly, all solutions of the DE/rand/1/bin algorithm converged to global minima and this algorithm was the second-fastest. PSO-Const and PSO-Std were the subsequent fastest converging algorithms, where PSO-Const showed smoother convergence than the PSO-Std. However, the average convergence profiles of both reached near a suboptimal equilibrium value of 500 in 5000 iterations. Finally, the ACO algorithm's convergence was the last among this set of algorithms, and it is the slowest to reach the suboptimal value of 500.

The convergence profiles analysis compared these algorithms in terms of their ability to escape local minima, speed of convergence, and the ability to find global optimum solutions. However, the accuracy of solutions of this class of space truss structure problem is also about how the solution compared to the analytical solution from literature like arc length (Hrinda, 2010; Crisfield, 1991; De Borst et al., 2012) for achieving the equilibrium path characterized by a vertical displacement of the central node. Therefore, the filtered solution obtained by ACO, DE/best/2/bin, DE/rand/1/bin, PSO-Const, and PSO-Std in Fig. 11(right) were plotted on an x–y plane of control point displacement and the load multiplier in Fig. 12.

Solutions accuracy analysis against the analytical curve. Fig. 12 plots suboptimal solutions in cyan and optimal solutions in blue. As shown in Fig. 12 (row 1), the ACO algorithm, although accurate in finding solutions on the equilibrium path, was only able to find suboptimal solutions. However, the discrete domain analysis along the control point displacement dimension was performed for generating 100 more trial solutions of ACO with 100 000 iterations as its stopping criteria. This analysis improved the results, and there were several optimal solutions on the equilibrium path; however, they were mainly scattered across the path.

Similar to the ACO algorithm, PSO-Std and PSO-Const were unable to find any near-optimal solution in 50 000 iterations. Additionally, PSO-Std and PSO-Const solutions were scattered (mostly belong to alternative equilibrium paths characterized by a non-vertical displacement), and rarely solutions were on the equilibrium path characterized by vertical displacement (see cyan points in row 2 of Fig. 12). Since PSO version solutions were spread across the full domain of the problem, unlike ACO, it was not required to discretize the domain for further analysis. Instead, an analysis involving generating additional 100 new trial solutions for a 100 000 iteration as stopping criteria was performed. Increasing iteration was for testing PSO robustness was evident from the fact that the solutions spread across, and the solutions were closer to suboptimal value than ACO (see Fig. 11(right)). However, even in this analysis, PSO-Std could not find any near-optimal solutions, and although PSO-Const found many near-optimal solutions, they were mainly inaccurate when compared with the required equilibrium path characterized by vertical displacement.

In contrast to ACO and PSO algorithms, DE versions were able to find a high percentage of accurate, near-optimal, and optimal solutions (Fig. 12, last row). Interestingly, the DE version DE/rand/1/bin was more accurate than DE/best/2/bin. A possible explanation is that premature convergence to local minima is more frequent in DE/best/2/bin than in DE/rand/1/bin. This is because solutions of DE/best/2/bin start following the local best solution, and if the best solution finds a trajectory leading to local minima on the hypersurface of the solution search space, then the DE/best/2/bin will not converge to global minima. This fact is evident in Fig. 11(left), where DE/best/2/bin average

convergence profile is poorly affected by solutions that are stuck into local minima.

Since both DE/rand/1/bin and DE/best/2/bin were able to provide a high percentage of near-optimal and accurate solutions, it was not necessary to do either a discrete or an extensive iteration analysis. However, many solutions were not on the equilibrium path characterized by vertical displacement. Rather, many solutions belong to the equilibrium path of varied characterizations. This is a disadvantage as it is only possible to pick a solution and analyze the equilibrium, but the space truss structure's equilibrium path (or a buckling direction) could not be observed clearly for the hypothesis's vertical downward force. Hence, an equilibrium path identification analysis was performed.

Equilibrium path characteristics identification analysis. A clustering analysis using the DBSCAN algorithm (Ester et al., 1996) was performed to analyze whether a few sets of solutions follow similar properties and whether they can be grouped together that identify particular equilibrium path characteristics. The DBSCAN algorithm was chosen because its characteristics to scan through point cloud. The scanning of the neighborhood of the point cloud formed by solutions were also required for the equilibrium path characteristics identification, where the nearest connected points were expected to cluster together to represent particular path characteristic. Fig. 13 shows the clustering analysis results on DE/rand/1/bin solutions, and it does find a set of solutions that appear to follow particular equilibrium path characteristics. Hence, one can choose a cluster (a color) representing particular equilibrium path characteristics.

One advantage of clustering analysis is that it offers several equilibrium paths compared to the standard civil engineering Arc-length method. However, solutions are not in order, and this approach produces too many equilibrium paths of varied unknown characteristics, especially many belonging to non-vertical displacement. Moreover, this analysis leaves with another issue, i.e., a cluster selection problem needed to be solved if one would like to choose only a single equilibrium path. Therefore, a more efficient method was required. In this research, a new algorithm called hypersphere search algorithms is presented for solving these class problems as accurately as possible.

4.2.3. Results of the adaptive search space decomposition method

Further analysis on the sixteen members truss structure problem was performed through the proposed hypersphere search algorithm. The algorithm started with a small domain of [−10 mm, 10 mm] for the displacement variables and [−0.2, 0.2] for the load multiplier variable. First, a set of 5 trial solutions were evaluated in step 1 pertaining to hypersphere initialization and first hypersphere center identification of this algorithm (cf. Section 3.2). Then, a radius $r = 5$ mm was used for the construction of the next hypersphere around the obtained center, and another set of 5 trial solutions were evaluated to choose the next center for the next hypersphere, and this process continued until the control point reached the fixed maximum value of 250 mm or the maximum number of 1000 trials was reached.

Fig. 14 shows the results of the hypersphere search algorithm. The obtained centers are shown in red, and all other optimal solutions are shown in cyan. This analysis has produced accurate solutions and a unique equilibrium path characterization for the vertical displacement. The only limitation of this method is the difficulty of this ill-posed nonlinear post-buckling analysis, where this algorithm finds it computationally challenging to search the next hypersphere where the structure has reached its breaking point (buckling point). That is, the radius $r = 5$ mm for the hypersphere seems unable to find the next appropriate hypersphere due to sharp and sudden change in the search space (see a blue-shaded spike of the computational needs). This poses a highly challenging task for optimization algorithms.

Another set of experiments was done to fine-tune user-defined hyperparameter radius r of the hypersphere search algorithm. The first trial was to reduce to radius r by a factor of half each time it was unable to find the next hypersphere (i.e., when it reached the breaking
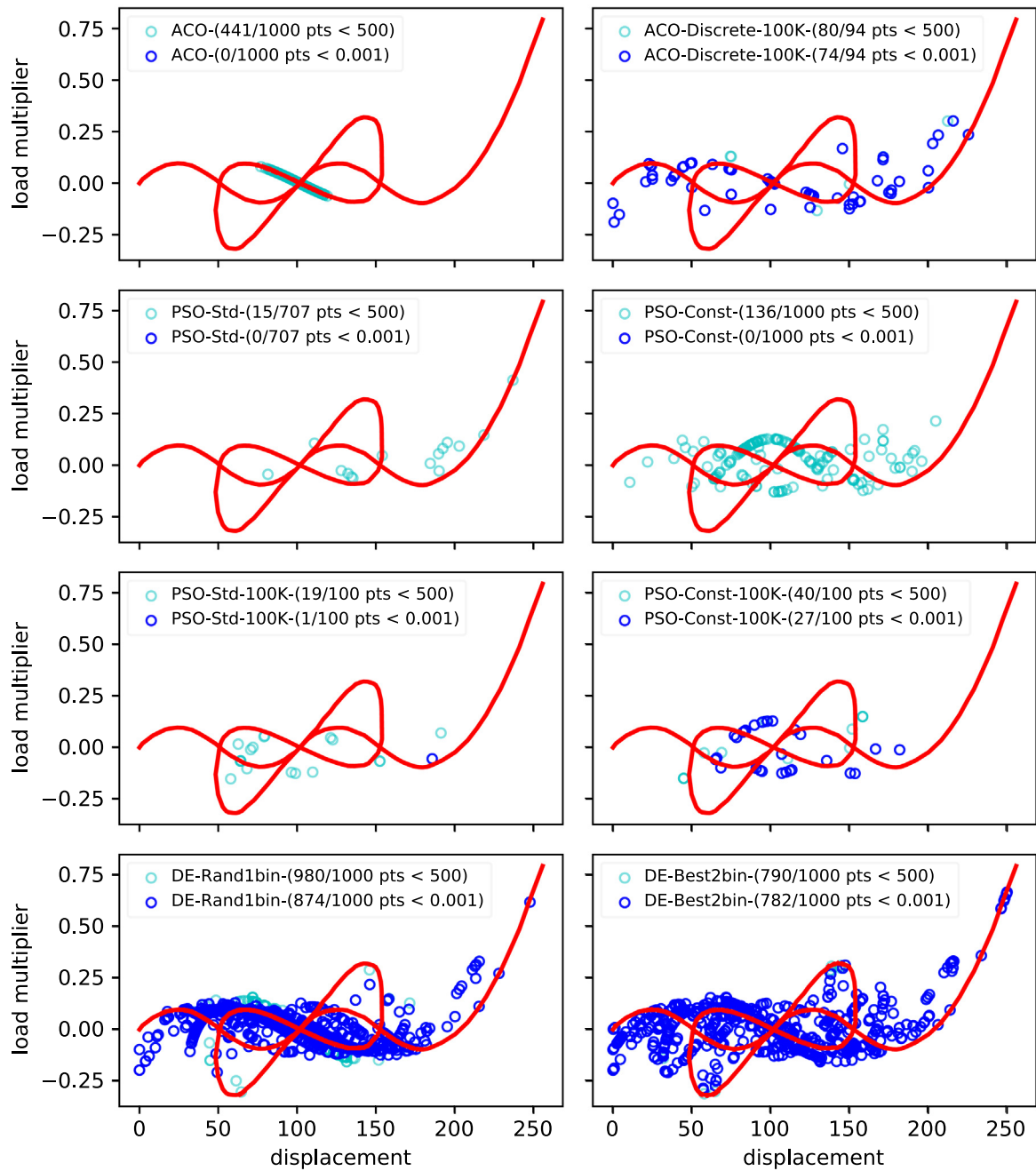
Fig. 12. Quality of solutions (in blue solutions with a precision of 0.001 as their equilibrium value and solution in cyan are with precision 500) produced by optimization algorithms over 10 000 generations and 100 000 generations for ACO-Discrete and PSO-100K versions. Arc-length method based equilibrium path is shown in red. From top-left to bottom-right algorithms are ACO, ACO (with a discrete domain of control points over 100 000 generations), PSO-Std, PSO-Const, PSO-Std with 100 000 generations, PSO-Const with 100 000 generations, DE/rand/1/bin, and DE/best/2/bin.

point). This was done by making radius r adaptive to the iteration of hypersphere construction as per this expression: $r_{new} = r_{prev} \times 0.5$. The analysis in Fig. 15(left) shows that the algorithm can follow the sharp breaking path; however, it is computationally challenging and slow. This is because eventually, the hyperspheres become small and infeasible for the algorithm to continue. The computational effort of this stage is shown by the spikes (peak values) of computational effort in Figs. 14 and 15.

Another trial was an additive approach where radius r was gradually increased by a value of 5 mm, i.e., $r_{new} = r_{prev} + 5$. Fig. 15(right) shows the algorithm was able to escape the breaking point, but it deviated from finding an exact unique path, and it jumped to other feasible solutions (belonging to non-vertical load) like the many alternatives found in clustering analysis in Section 4.2.1.

4.3. Benchmark 3: Twenty four member shallow truss structure

The third benchmark, represented in Fig. 16(left side), is a space truss structure composed of 24 trusses, whose axial rigidity is 960.5 kN, seven free nodes, and 21 free displacements. The permanent load comprises a vertical force of 50 N applied on the central node and six vertical forces applied on the other free nodes of the structure. The structure has been analyzed in Rezaiee-Pajand and Alamatian (2011) using a dynamic relaxation method. Therefore, the load path curve of the structure, referring to the vertical displacement of the central node, is known in the range from 0 mm to 50 mm. Fig. 16 (right side) shows the load–deflection curve of the central node and the solutions provided by the adopted optimization procedure. The two critical points are located
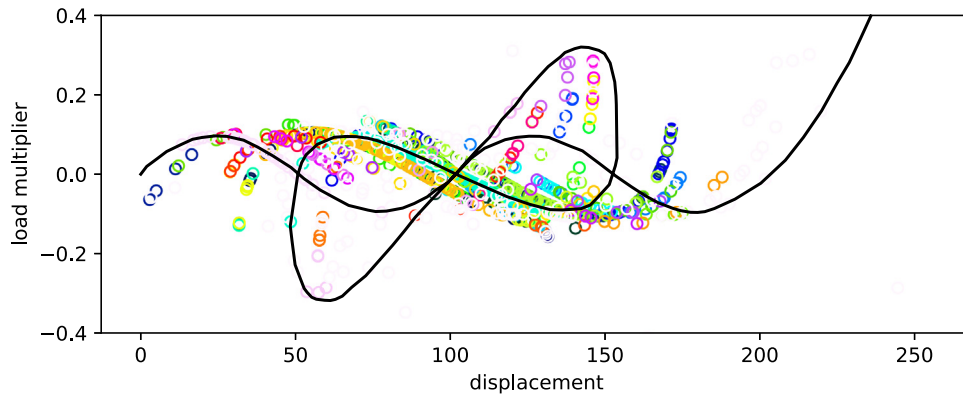
Fig. 13. Clustering of solutions (obtained by DE/rand/1/bin algorithm) for identifying different feasible characteristics of equilibrium path (in different color circles).
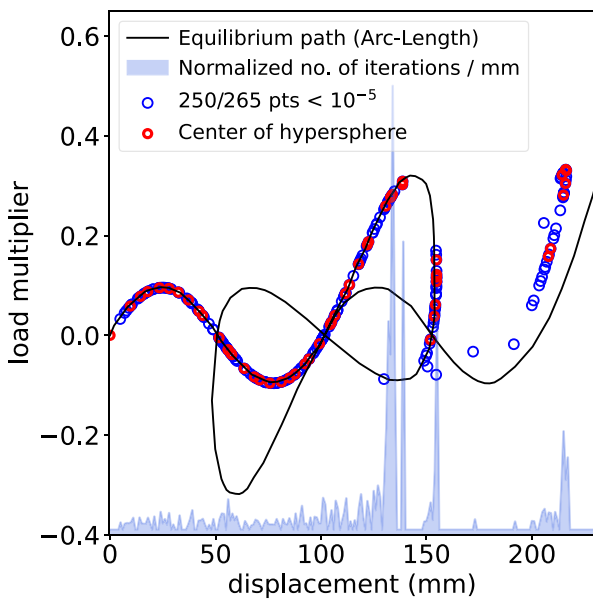

Fig. 14. Quality of solutions produced by the hypersphere search algorithm with input radius $r = 5\,\mathrm{mm}$. Solutions in blue are trial solutions of respective hypersphere whose centers are indicated in red. The computational effort in terms of normalized (between -0.4 and 0.5) total number of DE generations spent per mm displacement is shown in shaded blue area. The average computation effort is 8125 generations and max (i.e the peak) computation effort is $274\,168$ generations.

approximately at $8\,\mathrm{mm}$ and $30\,\mathrm{mm}$. Between these displacement values, the structural response is characterized by an evident snap-through mechanism.

The hypersphere method for the adaptive search space decomposition together with DE algorithms provided a good-dense distribution of solutions in most of the investigated range of displacements, including the two critical zones where a limited number of iterations were required for the convergence. The proposed methodology was also capable of describing the snap-through path with a significant number of solutions. However, a significantly higher number of iterations were required in this response stage. In addition, from the comparison, it is possible to observe that the solutions provided by the proposed methodology follow the baseline with a good level of accuracy within the entire investigated range of displacements. Finally, Fig. 17 shows different deformed shapes of the structure at different displacement magnitudes.

4.4. Test problem: spatial reticular beam structure

This Section presents the results of the analysis on a medium space truss structure representative of a 3D steel beam supporting a roof system. The structure has a span of $8000\,\mathrm{mm}$, a width of $2000\,\mathrm{mm}$, and a height of $750\,\mathrm{mm}$, arranged with the geometrical layout shown in Fig. 18(row 1). The beam is composed of 33 trusses, 10 free nodes, and 30 DoF. Each truss has $2500\,\mathrm{mm}^2$ of cross-section area and $200\,000\,\mathrm{MPa}$ of Young's modulus. Four simple supports are considered at the two ends of the beam, and the external load is represented by four vertical forces, equal to $100\,\mathrm{kN}$, applied at the beam top nodes.

On this problem, DE/rand/1/bin and DE/best/2/bin algorithms were applied as they were the most suitable algorithms for this class of structure analysis problem in the benchmark problem optimization. The goal was to find a few feasible solutions to approximate an equilibrium path containing pre-and post-buckling stages of a 3D steel beam. Hence, the variable domain setup was $[0\,\mathrm{mm}, 2500\,\mathrm{mm}]$ for the displacement and $[1, 300]$ for the load multiplier. In this setting, both DE algorithms versions were applied with stopping criteria of 1 million iterations or convergence accuracy of 1×10^{-5}. Several instances of both versions of DE reached an accuracy of 1×10^{-5}. However, when filtering solutions, DE/rand/1/bin was found to be producing a greater number of solutions that were accurate compared to DE/best/2/bin. Hence, solutions obtained by DE/rand/1/bin are shown in Fig. 18. In Fig. 18, rows 2 to 5 show deformed shapes of four equilibrated and feasible solutions.

A rough representation of the equilibrium path is represented in Fig. 19 in terms of maximum vertical displacement of the structure and the load multiplier to characterize these solutions. The dashed lines connecting the solutions represent qualitatively the equilibrium path between these solutions which can be drawn by considering a refined set of solutions. However, despite the limited number of solutions considered, it was possible to highlight the main characteristics of the pre- and post-buckling behavior of the structure.

More in detail, it is possible to observe that solutions A and B belong to the stable equilibrium (pre-buckling) region, where the increase of deformations corresponds to the increase of the external loads and the deformed shape is symmetric (cf. Figs. 18 and 19). On the other hand, solution C shows a buckling behavior (cf. Figs. 18 and 19) involving the truss elements of the right part of the beam, leading to the decrease of the load multiplier and the loss of the symmetry of the deformed shape. Finally, Solution B is characterized by the global buckling with the overturning of the structure (cf. Figs. 18 and 19). After this point, the system behaves as a catenary rather than a beam, and the equilibrium is again stable.
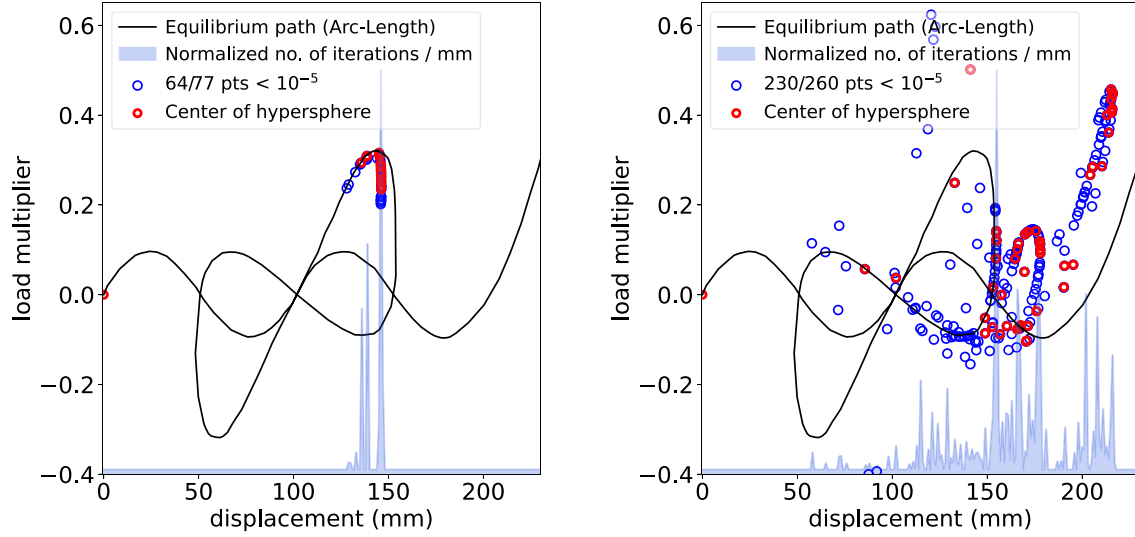
Fig. 15. Sixteen-Member solutions produced by hypersphere search algorithm with input radius $r_{new} = r_{prev} \times 0.5$ (*Left*) and with input radius $r_{new} = r_{prev} + 5\,mm$ (*Right*). Reduction of the radius make algorithm stay at the buckling point and increase of the radius lead to non-vertical hypothesis equilibrium stages/path. Solutions in blue are trial solutions of respective hypersphere whose centers are indicated in red. Both experiments in *Left* and *Right* were started from a "center" immediate behind the breaking points (see computational spike) obtained by the experiment in Fig. 14.
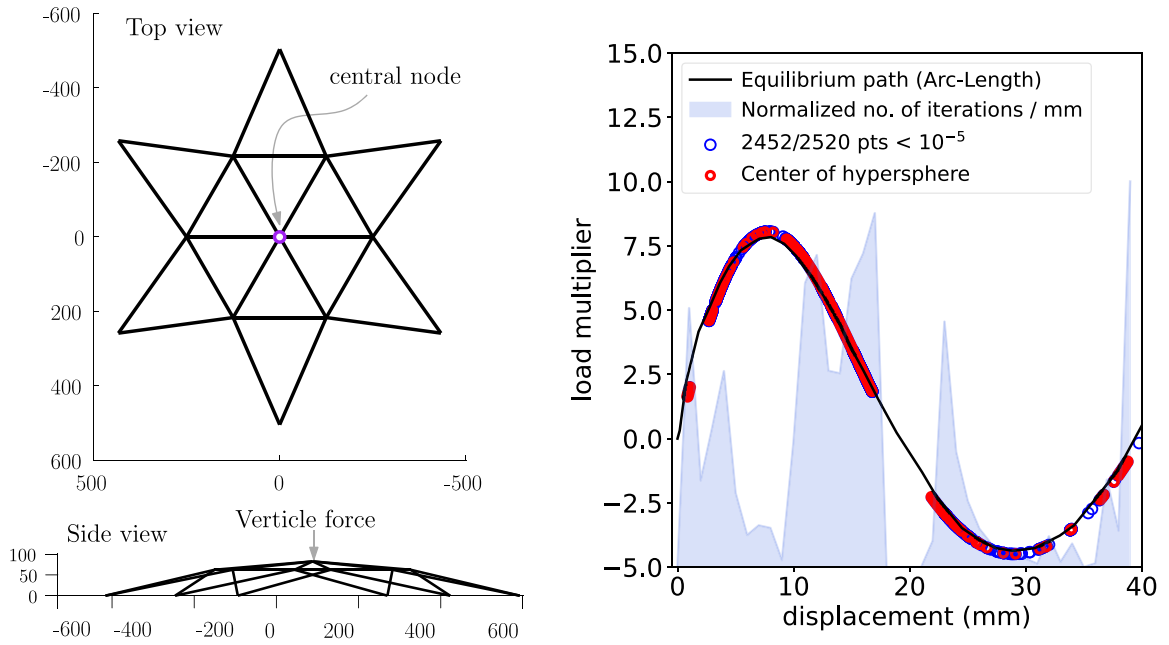

Fig. 16. Twenty Four Member Shallow Truss Structure. *Left:* Top View and Side View of the structure. The central node of the structure is indicated by a lighter color dot in Top view and a light color arrow pointer in Side view. The arrow on Side view indicates a vertical downward external force of 960.5 kN applied on the central node. *Right:* The results of adaptive search space decomposition method applied on this benchmark. Blue circles are optimal solution, red circles are centers of the hyperspheres of the adaptive search space decomposition method. Shaded curve area is the computational effort in terms of normalized (between -5 and 10) sum of the number of generations spent of searching solutions at a hypersphere center per mm displacement. The average computation effort is 24 468 generations and max (i.e the pick) computation effort is 37 369.

5. Discussions

This paper presents a novel methodology based on gradient-free optimization algorithms for assessing the nonlinear structural response of space truss structures subjected to large displacements, accounting for geometrical nonlinearities. This methodology allows for evaluating the load path of the structure, including stable and unstable equilibrium stages, thus characterizing its post-buckling behavior and assessing the vulnerability to progressive collapses. Compared to the standard

Newton–Raphson procedures, the proposed methodology does not require the assemblage and update of the global stiffness and geometric matrices (Section 2.2). Moreover, it is not exposed to the potential numerical issues affecting the Newton–Raphson and arc-length methods in correspondence to the critical points, where the equilibrium changes from the stable to the unstable.

This research applied four classes of optimization algorithms: deterministic optimization algorithm (DIRECT), single solution-based algorithm (Simulated Annealing), swarm inspired algorithms (ABC, ACO,
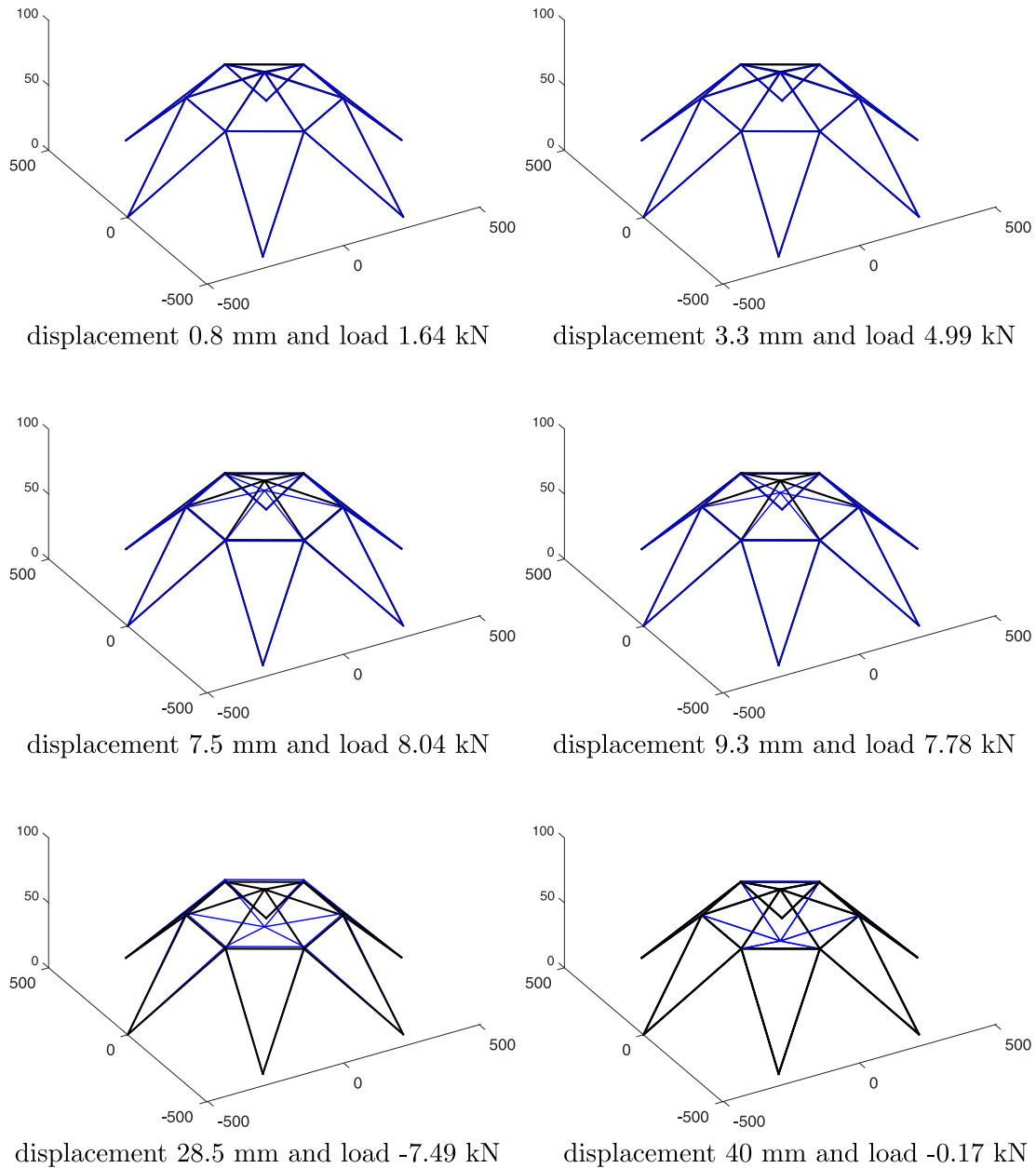
displacement 0.8 mm and load 1.64 kN

displacement 3.3 mm and load 4.99 kN

displacement 7.5 mm and load 8.04 kN

displacement 9.3 mm and load 7.78 kN

displacement 28.5 mm and load -7.49 kN

displacement 40 mm and load -0.17 kN

Fig. 17. Deformed shapes and equilibrium stages of twenty four member shallow truss structure member space truss structure.

and PSO), and an evolutionary algorithm (DE) (Section 3.3.1). The DE algorithm was found to be better at converging among all algorithms from our initial trial of three benchmark problems. Hence, the scope of this work mainly relied on the DE algorithm as it was competitively the most successful in solving this class of problems. Finally, this study developed a multistep analysis algorithm, the hypersphere algorithm for the adaptive decomposition search space to support gradient free algorithms effectively describe the structure load path.

Three benchmarks of 3D truss structures and one test case representative of a medium-sized structural problem were tackled as optimization problems. The optimization problem associated with the global structural equilibrium (Section 2.2) can be classified as a nonlinear, multimodel, unconstrained minimization optimization problem, which presents significant challenges to optimization algorithms. It was highly challenging to solve as variables were highly sensitive to each other

due to their physical interactions (see Section 4). Moreover, this optimization problem has a significantly large number of local minima. Additionally, this problem has multiple global minima, but the global minima were strongly linked to the load multiplier variable and the direction of loadings (downward forces on a structure), making this problem a nonlinear, multimodal, unconstrained optimization problem. Based on the results of numerical experiments conducted in Section 3, it was possible to draw the following conclusions:

- The DIRECT algorithm was able to find solutions for the first benchmark but could not explore any other solutions except when the search space was manually partitioned to help it find other solutions and when the search space was highly restricted after domain analyses (cf. Figs. 7 and 8). Apart from that, DIRECT struggled to find any other solutions to benchmark problems. This is because the hyperspace became ill-defined or search space too
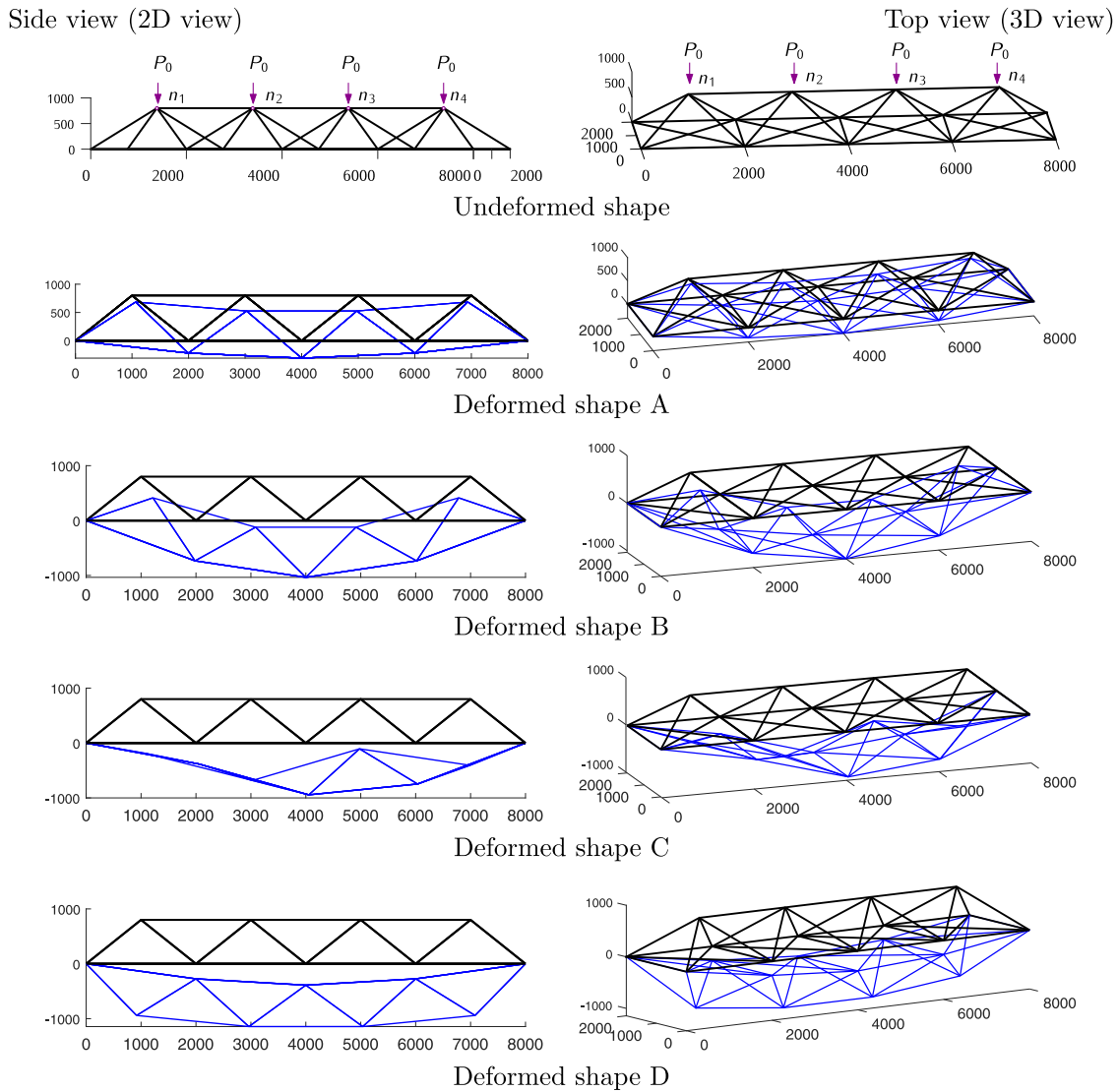
Fig. 18. Equilibrium stages of 3D reticular beam structure optimized using DE algorithm version DE/rand/1/bin. Lighter color arrows (on Side view and Top View) indicates the application of the vertical downward force (P_0 = 100 kN) applied on the nodes n_1, n_2, n_3, and n_4 of 3D reticular beam structure. The undeformed (original) structure shapes are drawn with black lines and deformed structure shapes are drawn with blue lines. The structure shapes 2D view is in the left column and their 3D view is in the right column. All deformed shapes presented achieved an accuracy of 1×10^{-5} on the objective function Eq. (5). Row 1 is original structure and rows 2 (Shape A), 3 (Shape B), 4 (Shape C), 5 (Shape D) are deformed shapes for respective applied external lead multipliers values 25.6302, 158.6989, 13.5174, 55.8157. The deformed shapes A, B, C, and D respectively has their control point displacements 274.2566, 921.2034, 1482.2, and 1943.6.

complex for DIRECT to work properly. A similar observation is applied to the Simulated annealing algorithm, which was able to solve the first problem, a small benchmark problem consisting of four variables, but SA could not effectively solve the 16 variables sixteen-member problem.

- ACO demonstrated a tendency to find a solution only at the center of the domain. A close examination suggests that the ACO uses Gaussian distribution to generate new solutions in its iterations, and that may lead the solutions following the denser region of the search space (cf. Figs. 7 and 12). However, this observation is merely a hypothesis given the nature of this algorithm and this problem as observed during the experimentation.
- Swarm-inspired PSO algorithms were able to solve the first benchmark, but they had relatively poor performance on the second benchmark (cf. Figs. 7 and 12). The DE versions were the best performing algorithms for this class of ill-posed problems.

- Despite being the best performing, DE versions were unable to find the expected number of accurate solutions on the equilibrium path (Fig. 10). Only after a domain decomposition analysis to better inform the algorithms about the domain range of variables, DE versions were able to find a high percentage of accurate solutions (Figs. 10 and 12).
- Among DE/best/2/bin and DE/rand/1/bin, the DE/rand/1/bin was found to be best performing for such challenging optimization problems; the convergence profiles of both show that DE/best/2/bin tended to fall into local minim more often than the DE/rand/1/bin. This was attributed to solutions following the local best solution leading to local minima (Fig. 11).
- On the test problem, the 3D reticular beam, which is a real medium-size structure, DE/rand/1/bin, like in the case of the second benchmark, outperformed DE/best/2/bin. It was able to find more optimal solutions for the test problem 3D reticular beam. The results of DE/rand/1/bin presented in Section 4.4
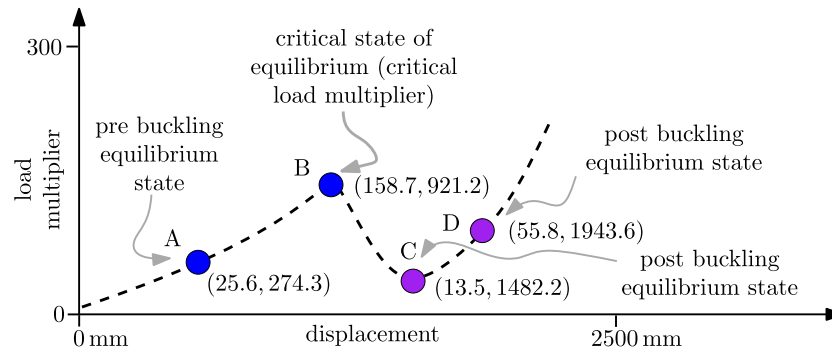
Fig. 19. Pre- and post buckling equilibrium states of the 3D reticular beam. These states are obtained for a multistep analysis mentioned in Fig. 3. The dotted line is the hypothesis equilibrium path connecting the obtained states for this test problem.

have high precision and produce solutions for both pre- and post buckling stages (Fig. 18). These solutions allowed producing hypothesizing the equilibrium path (Fig. 19).

- The proposed hypersphere algorithm, i.e. adaptive search space decomposition method, incrementally constructed hyperspheres using the knowledge from the previous hyperspheres (Section 3.2). This method helped DE find solutions accurately, and its performance was comparable to the arc-length method used in civil engineering for the equilibrium path analysis (Figs. 14 and 16).
- The main challenges with the proposed adaptive search space decomposition method were to tune the value of the user-defined hyperparameter radius of the hyperspheres, which included whether to fix it a single value for all iterations or make it adaptive to iterations (Section 4.2.3). Additionally, this algorithm suffers from an obvious issue concerning sudden and sharp change in the hypersphere's domain at the breaking/buckling point of the structure. Since the buckling point changed the domain sharply and suddenly, this algorithm struggled to find the next hypersphere (Fig. 15).

In summary, the main outcome of this research is to demonstrate that the use of heuristic-based optimization algorithms can be adopted for solving nonlinear structural problems, in particular for assessing the geometrically-nonlinear response of space truss structures, which are particularly prone to show complex, unstable equilibrium stages and snap-through mechanisms. This outcome was achieved by successfully solving three benchmark problems to a high degree of accuracy and provided several optimal solutions for the test problem that helped hypothesize an equilibrium path connecting four pre- and post buckling equilibrium states (solutions). These presented results demonstrate that the proposed procedure can be adopted to describe complex post-buckling behaviors of large structural systems. Therefore, this methodology can represent an effective alternative approach to Newton–Raphson strategies for buckling analyses of truss structures or combined with them to improve the robustness and accuracy of structural analyses.

This research provided solutions to this challenging nonlinear, multimodel, unconstrained minimization optimization problem, which the optimization research community can consider as a testbench to test and evaluate new optimization algorithms. This research was able to solve this multimodel optimization problem by running various instances. However, one would ideally run a single instance to find as many solutions as possible. For example, the population diversity of DE (the best performing algorithm for this set of problems) was extremely low, i.e., almost all individuals in the population produce extremely similar solutions. Hence, only one point on the equilibrium path could be considered in one instance of a run. Therefore, this problem can

be presented as a test problem to assess the quality of an algorithm's diverse solutions.

Additionally, the number of iterations required for solving problems with increasing DoF (free variables) was exponentially increasing. For example, space truss structure optimization of 4 variables took on average between 150–500 iterations, 16 variables took on average between 5000–10 000 iterations (for some solutions, it took larger than 10 000 iterations depending on the position of solutions on the equilibrium path), 21 variable took 20 000–25 000 iterations, and 30 variables took about 100 000–150 000 iterations. Hence, the dimension of this problem is also presenting significant challenges to optimization algorithms. In summary, this test problem presents challenges to optimization algorithms to produce accurate and diverse solutions with high convergence speed.

6. Conclusions

This work presents a novel analysis methodology based on gradient-free optimization algorithms for the nonlinear structural analysis of space trusses. The proposed methodology formulated the optimization of the global equilibrium of the system as a nonlinear, multimodal, unconstrained, continuous optimization problem. This problem is solved within a new effective multistep analysis procedure providing the nonlinear load path of the system. The proposed methodology can represent an effective method, alternative to Newton–Raphson procedures or combined with them for nonlinear post-buckling analysis of real structural systems.

The search landscape of this problem and the interaction between the free (displacement) variables pose significant challenges to existing continuous optimization algorithms to produce diverse and accurate solutions with high convergence speed. In this research, a number of strategies for search domain decomposition are presented. Consequently, a novel adaptive research space partition method, called hypersphere search algorithm, is proposed and applied to solve this problem. This algorithm iteratively moves through the search landscape of the problems to find as many accurate solutions as possible to provide the complete load-path curve of the structure.

Among the different investigated algorithms, the Differential Evolution (DE) algorithm is identified as the most competitive algorithm to solve this class of problems. Therefore, DE algorithms were applied to solve a test problem concerning a 3D medium-sized reticular beam. As a result, the algorithm optimally produced deformed equilibrium shapes and hypothesis equilibrium path, proving the capability to the procedure to be applied for assessing real structural systems.

CRediT authorship contribution statement

Varun Ojha: Conception of the research, Implementation, Experiments, Writing and reviewing the paper. **Bartolomeo Pantò:** Conception of the research, Implementation, Experiments, Writing and reviewing the paper. **Giuseppe Nicosia:** Conception of the research, Implementation, Experiments, Writing and reviewing the paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Baggio, C., Trovalusci, P., 2000. Collapse behaviour of three-dimensional brick-block systems using non-linear programming. Struct. Eng. Mech. 10 (2), 181.

Bathe, K.-J., 2006. Finite Element Procedures. Klaus-Jurgen Bathe.

Bisbos, C., Makrodimopoulos, A., Pardalos, P.M., 2005. Second-order cone programming approaches to static shakedown analysis in steel plasticity. Optim. Methods Softw. 20 (1), 25–52.

Bisbos, C.D., Pardalos, P.M., 2007. Second-order cone and semidefinite representations of material failure criteria. J. Optim. Theory Appl. 134 (2), 275–301.

Blandford, G., 1996. Progressive failure analysis of inelastic space truss structures. Comput. Struct. 58 (5), 981–990.

Chen, W.-F., Han, D.-J., 2007. Plasticity for Structural Engineers. J. Ross Publishing.

Clerc, M., 1999. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99, vol. 3. IEEE, pp. 1951–1957.

Crisfield, M.A., 1991. Nonlinear Finite Element Analysis of Solids and Structures. Volume 1: Essentials. Wiley & Sons.

Crisfield, M.A., 1996. Nonlinear Finite Element Analysis of Solids and Structures. Volume 2: Advanced Topics. Wiley & Sons.

De Borst, R., Crisfield, M.A., Remmers, J.J., Verhoosel, C.V., 2012. Nonlinear finite element analysis of solids and structures. John Wiley & Sons.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In: KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, vol. 96, no. 34. pp. 226–231.

Ferris, M., Tin-Loi, F., 2001. Limit analysis of frictional block assemblies as a mathematical program with complementarity constraints. Int. J. Mech. Sci. 43 (1), 209–224.

Geradin, M., Idelsohn, S., Hogge, M., 1981. Computational strategies for the solution of large nonlinear problems via quasi-Newton methods. Comput. Struct. 13 (1–3), 73–81.

Gilbert, M., Casapulla, C., Ahmed, H., 2006. Limit analysis of masonry block structures with non-associative frictional joints using linear programming. Comput. Struct. 84 (13–14), 873–887.

Gilbert, M., Tyas, A., 2003. Layout optimization of large-scale pin-jointed frames. Eng. Comput..

Hasançebi, O., Çarbaş, S., Doğan, E., Erdal, F., Saka, M., 2009. Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures. Comput. Struct. 87 (5–6), 284–302.

Horst, R., Tuy, H., 2013. Global Optimization: Deterministic Approaches. Springer.

Hrinda, G., 2010. Snap-through instability patterns in truss structures. In: 51st AIAA/ASME/ ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference. p. 2611.

Hrinda, G.A., Nguyen, D.T., 2008. Optimization of stability-constrained geometrically nonlinear shallow trusses using an arc length sparse method with a strain energy density approach. Finite Elem. Anal. Des. 44 (15), 933–950.

Huang, H., An, H., Ma, H., Chen, S., 2019. An engineering method for complex structural optimization involving both size and topology design variables. Internat. J. Numer. Methods Engrg. 117 (3), 291–315.

Jones, D.R., Perttunen, C.D., Stuckman, B.E., 1993. Lipschitzian optimization without the Lipschitz constant. J. Optim. Theory Appl. 79 (1), 157–181.

Karaboga, D., Basturk, B., 2008. On the performance of artificial bee colony (ABC) algorithm. Appl. Soft Comput. 8 (1), 687–697.

Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. In: Proceedings of ICNN'95-International Conference on Neural Networks, vol. 4. IEEE, pp. 1942–1948.

Khatir, S., Tiachacht, S., Le Thanh, C., Ghandourah, E., Mirjalili, S., Wahab, M.A., 2021. An improved artificial neural network using arithmetic optimization algorithm for damage assessment in FGM composite plates. Compos. Struct. 273, 114287.

Khatir, S., Wahab, M.A., 2019. Fast simulations for solving fracture mechanics inverse problems using POD-RBF XIGA and Jaya algorithm. Eng. Fract. Mech. 205, 285–300.

Khatir, S., Wahab, M.A., Boutchicha, D., Khatir, T., 2019. Structural health monitoring using modal strain energy damage indicator coupled with teaching-learning-based optimization algorithm and isogeometric analysis. J. Sound Vib. 448, 230–246.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science 220 (4598), 671–680.

Li, C.J., Yan, L., 1995. Mechanical system modelling using recurrent neural networks via quasi-Newton learning methods. Appl. Math. Model. 19 (7), 421–428.

Lute, V., Upadhyay, A., Singh, K.K., 2009. Computationally efficient analysis of cable-stayed bridge for GA-based optimization. Eng. Appl. Artif. Intell. 22 (4–5), 750–758.

Macorini, L., Izzuddin, B., 2011. A non-linear interface element for 3D mesoscale analysis of brick-masonry structures. Internat. J. Numer. Methods Engrg. 85 (12), 1584–1608.

Martin, R., Delatte, N.J., 2001. Another look at hartford civic center coliseum collapse. J. Perf. Construct. Facilities 15 (1), 31–36.

Memon, B.-A., Xiao-zu, S., 2004. Arc-length technique for nonlinear finite element analysis. J. Zhejiang Univ.-Sci. A 5 (5), 618–628.

Murtha-Smith, E., 1988. Alternate path analysis of space trusses for progressive collapse. J. Struct. Eng. 114 (9), 1978–1999.

Opara, K.R., Arabas, J., 2019. Differential evolution: A survey of theoretical analyses. Swarm Evol. Comput. 44, 546–558.

Rezaiee-Pajand, M., Alamatian, J., 2011. Automatic DR structural analysis of snap-through and snap-back using optimized load increments. J. Struct. Eng. 137 (1), 109–116.

Saka, M., 2007. Optimum design of steel frames using stochastic search techniques based on natural phenomena: a review. Civ. Eng. Comput. Tools Tech. 105–147.

Scipy, 2021. Scipy.optimize.differential_evolution. https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.differential_evolution.html.

Smith, E.A., 1984. Space truss nonlinear analysis. J. Struct. Eng. 110 (4), 688–705.

Socha, K., Dorigo, M., 2008. Ant colony optimization for continuous domains. European J. Oper. Res. 185 (3), 1155–1173.

Storn, R., Price, K., 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. 11 (4), 341–359.

Tran-Ngoc, H., Khatir, S., Ho-Khac, H., De Roeck, G., Bui-Tien, T., Wahab, M.A., 2021. Efficient artificial neural networks based on a hybrid metaheuristic optimization algorithm for damage detection in laminated composite structures. Compos. Struct. 262, 113339.

Tyas, A., Gilbert, M., Pritchard, T., 2006. Practical plastic layout optimization of trusses incorporating stability considerations. Comput. Struct. 84 (3–4), 115–126.

Wang, N., Adeli, H., 2015. Self-constructing wavelet neural network algorithm for nonlinear control of large structures. Eng. Appl. Artif. Intell. 41, 249–258.

Yildiz, A.R., 2013. Comparison of evolutionary-based optimization algorithms for structural design optimization. Eng. Appl. Artif. Intell. 26 (1), 327–333.