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

Varun Ojha*[1], Bartolomeo Pantò* [2,3], and Giuseppe Nicosia* [4,5]

[1] *University of Reading, Reading, UK*
[2] *Imperial College London, London, UK*
[3] *Durham University, Durham, UK*
[4] *University of Cambridge, Cambridge, UK*
[5] *University of Catania, Catania, Italy*

Abstract

The paper proposes a multistep incremental procedure based on gradient-free optimization algorithms for nonlinear pre- and post-buckling analyses of space truss structures subjected to large displacements. Space truss structures are widely used in structural engineering to build large steel constructions whose structural typology is particularly vulnerable to progressive collapses due to displacements and buckling mechanisms, potentially leading to structural failure. The procedure proposed in this paper allows the load-equilibrium path analysis of a structure under permanent and variable loadings, explicitly considering geometric nonlinearities, allowing the description of unstable equilibrium stages and potential snap-through mechanisms. The determination of these equilibrium stages requires optimization of Lagrangian kinematics parameters of the system under a load. This paper considers a set of differential evolution algorithms and other heuristic algorithms for this optimization. This optimization problem sets significant challenges due to its undefined parameter domain and a complex search landscape where each parameter is highly sensitive to changes in other parameters because of their physical interactions. Thus, this problem is a class of nonlinear, multimodal, unconstrained, continuous optimization problems, which can be considered a challenging testbench to evaluate optimization algorithms. This paper studies two benchmarks and a large structural system. A *hypersphere search algorithm* is developed to produce accurate solutions for the most difficult benchmark problem. 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 of gradient-free algorithms in analyzing space truss structures.

Keywords: optimization; differential evolution; nonlinear static analysis; structures; instability of equilibrium; buckling analysis.

*Corresponding Author: Varun Ojha, email: v.k.ojha@reading.ac.uk

Preprint submitted to *Engineering Application of Artificial Intelligence*

Contents

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. However, these structures generally show large displacements even under service loadings, and their ultimate response is often characterized by snap-through buckling mechanisms in which the structure passes rapidly from an equilibrium state to a non-adjacent equilibrium configuration [1]. Furthermore, some catastrophic events [2] and numerical studies [3, 4, 5] 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 and allowing for the complete equilibrium path of the structure are needed to assess the structural robustness of truss structures against progressive collapses [6].

In the last decades, many structural optimization procedures, based on linear programming [6, 7, 8], genetic algorithms [9, 10, 11, 12], or iterative finite element procedures [13], have been developed to research optimum design of large truss structures through exploring optimum topology [14] 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.

Many authors proposed mathematical optimization algorithms based on mathematical programming to perform the limit analysis on structural systems such as frictional rigid-block assemblages [15, 16, 17] or elastoplastic Von-Mises steel structures [18, 19]. These approaches provide valuable information about the failure mechanism and the 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 [20, 21, 22] associated with iterative Newton-Raphson techniques [23, 24], 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 [25, 26]. 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 [22] and for approximation of nonlinear behavior of structures [27], covering the drawbacks of Newton-Raphson like procedures. These algorithms are based on quasi-Newton methods [28, 29] 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.

In this paper, a class of global optimization algorithms is proposed for the evaluation of the equilibrium path of space truss structures subjected to large displacements. The free parameters of the optimization problem are the Lagrangian kinematics parameters of the system and the load multiplier. Furthermore, the objective function is defined in terms of global unbalance determined as the difference between the vectors of external and internal forces. To the best of the authors' knowledge, this is the first work performing incremental (multistep) analysis of structures using gradient-free global optimization algorithms.

In this work, two benchmark problems investigated in the literature through Newton-like approaches [20] are considered to assess the accuracy and performances of the selected set of optimization algorithms. Each problem is solved using ad-hoc differential evolution 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.

In summary, this work presents a nonlinear, multimodal, unconstrained, continuous optimization problem concerning structural engineering space truss structure optimization whose search landscape poses significant challenges to existing continuous optimization algorithms in terms of assessing the ability to work on ill-posed problems and produce diverse and accurate solutions with high convergence speed. The implementation of this research work is available on our GitHub page[1].

This paper is organized to first explain the structural model in Section 2.1. Then, Section 3.3.2 defines the optimization problem in Section 2.2, which is followed by a description of optimization algorithms, the methodology, and optimization strategies adopted to solve this problem. The results on two benchmark problems and a test problem are presented and analyzed in Section 4. Lastly, discussions and conclusions are presented in Section 5 and Section 6.

2 Space truss structure problem

2.1 Kinematics

The model's kinematics adopt a Lagrangian description considering the hypotheses of large displacements and small strains. The degrees of freedom are assumed 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 (Figure 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

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

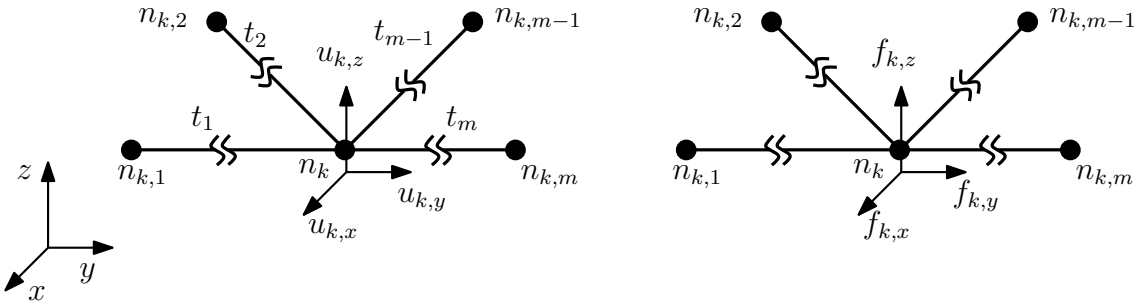

Figure 1: Independent displacements (*Left*) and dual forces (*Right*) of a generic node n_k connecting m trusses $t_1, t_2, \ldots, t_m$ with nodes $n_{k,1}, n_{k,2}, \ldots, n_{k,m}$. The node n_k has its displacements $u_{k,x}, u_{k,y}, u_{k,z}$ and corresponding forces $f_{k,x}, f_{k,y}, f_{k,z}$ along x, y and z dimensions of the space.

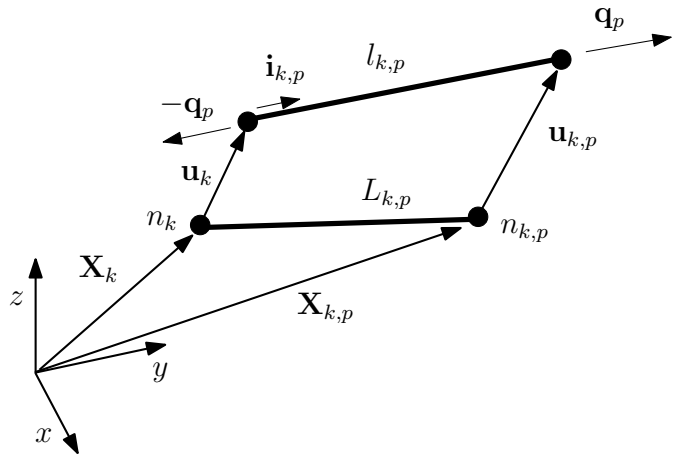

Figure 2: Kinematics and internal forces of a generic p-th truss connecting node n_k with node $n_{k,p}$. The truss has an initial length $L_{k,p}$ and its nodes initial coordinates are $\mathbf{X}_k$ and $\mathbf{X}_{k,p}$. The axial nodal force of the truss is $\mathbf{q}_p$, its current length and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

permanent loads and $\mathbf{f}_k$ represents variable loads, which are amplified by the load multiplier λ (Figure 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 Figure 1. The generic p-th truss $(p = 1, \ldots, m)$ connecting the node n_k with the node $n_{k,p}$ is represented in Figure 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 $\mathbf{q}_p$ the axial nodal forces of the truss.

According to the hypothesis of small strains [20], the truss axial strain (ϵ_p) is evaluated considering the engineering strain as:

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

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 $||\mathbf{v}||$ indicates the Euclidean norm of the vector $\mathbf{v}$.

Given k_p the axial stiffness of the truss and $\mathbf{i}_{k,p}$ the unitary vector identifying its current orien-

tation, the internal force can be written as follows:

$$\mathbf{q}_{k,p} = \mathbf{i}_{k,p}\ k_p \left(\frac{||\mathbf{X}_{k,p} - \mathbf{X}_k||}{||\mathbf{X}_{k,p} + \mathbf{u}_{k,p} - \mathbf{X}_k - \mathbf{u}_k||} - 1) \right), \tag{2}$$

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 by the following equation:

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

2.2 Global equilibrium

The free variables of the optimization problem are represented by the independent displacements of the structure $\mathbf{u} = [\mathbf{u}_1, \ldots, \mathbf{u}_N]$ and the load multiplier λ. The objective function ($\mathcal{F}$) 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 k-th node, which can be easily obtained from Equation (4):

$$\mathbf{R}_k = \mathbf{f}_{0k} + \lambda \mathbf{f}_k - \sum_{p=1}^{m} \mathbf{q}_{k,p}. \tag{6}$$

The incremental multistep procedure shown in Figure 3 is followed to draw the equilibrium path of the structure in the $d - \lambda$ space, where d is a chosen control point, which can coincide with the physical displacement of a structural node or a function of a number of displacements. For instance, the control point can be assumed coincident with the Euclidean norm of the vector $\mathbf{u}(d = ||\mathbf{u}||)$.

Indicating d_i and λ_i the committed values of the control point and load multiplier at the generic i-th step of the analysis, and $d_T - \lambda_T$ the corresponding trial values, the following constrain is considered:

$$\sqrt{((d_T - d_i)^2 + (\lambda_T - \lambda_i)^2)} \leq \Delta, \tag{7}$$

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:

$$(d_T - d_i)(d_{i-1} - d_i) + (\lambda_T - \lambda_i)(\lambda_{i-1} - \lambda_i) \leq 0, \tag{8}$$

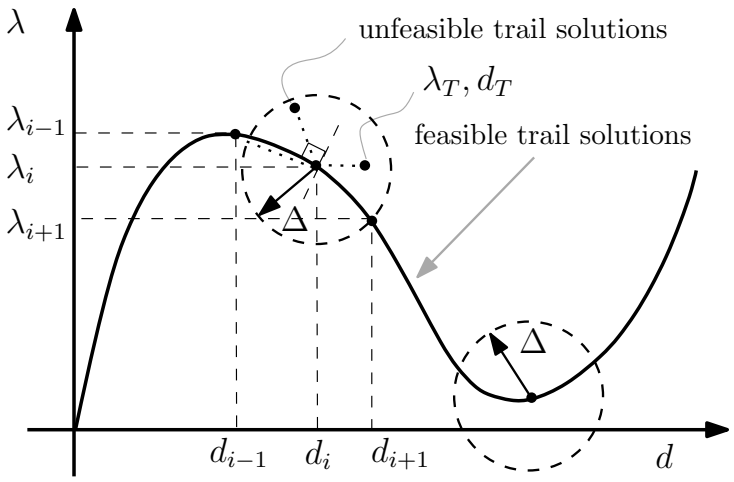

Figure 3: Multistep procedure. Variables d, λ, and Δ respectively are control point, load multiplier, and discrete steps along the control point dimension for producing an equilibrium path (curve in black). Any point on this curve and within a radius Δ that satisfies constraints in Equations (7) and (8) is a feasible solution.

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

The challenging issue with the presented space-truss optimization problem is the lack of displacement and load multiplier domain knowledge, pursued using a set of gradient-free optimization algorithms. That is, the optimization problem associated with the space-truss structural problem is ill-posed, and the free variables' bound values are available only based upon an initial elastic prediction of the structural response, and simple equilibrium considerations and bound are not well defined.

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. Hence, we were able to formulate this space truss structure buckling analysis problem as an optimization problem. The problem was treated as a minimization problem where free variables are the displacements $\mathbf{u} = [\mathbf{u}_1, \ldots, \mathbf{u}_N]$ and the load multiplier λ of a structure. Equation (5) is the objective function that the algorithms aim to minimize, where a value 0 is the global minima of this equation, indicating an equilibrium state of the structure.

The geometric nonlinearities, high physical interaction and sensitivity among variables, and numerous possible imbalance states for a particular load multiplier value make this problem further challenging. The optimization of Equation (5) is therefore classified as a nonlinear, multimodal, unconstrained, continuous, minimization problem whose search landscape is complex containing many local and a few global minima, i.e., its search landscape is rugged.

In order to solve this optimization problem, as said above, an intuitive guess could be made based on an initial elastic prediction of the structural response and simple equilibrium consid-

erations. 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 are used in this research for search domain analysis of this problem.

3 Methodology

3.1 The baseline method

In this research, an exhaustive study was done over the two benchmarks problems consisting in shallow space trusses, which have been numerically investigated in the literature by adopting Newton-Raphson iterative procedures combined with Arc-length methods [1, 20, 23]. The results of these two problems available in the literature enabled 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 amount of external load that can be applied. 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 benchmark problems across the equilibrium path shown in the literature. (This will be discussed later in results Section 4.) Hence, a strategy to decompose the search space was adopted.

3.2 Problem search space decomposition methods

3.2.1 Fixed decomposition of search space

There were two options for decomposing the search space. First, to partition the range of control point displacement dimension into several discrete steps as the literature provided the information on the control point. Second, in addition to partitioning the control point, also incrementally partition space for other displacement variables. Compared to the first option, the second option was much more effective (as will be discussed later in Section 4) in finding solutions across the equilibrium path shown in the literature [20, 21]. The performance of each optimization algorithm is discussed later in the results Section 4.

This search domain analysis for this challenging problem highlighted the strengths and weaknesses of optimization algorithms. 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 strength of algorithms in solving this nonlinear, unconstrained continuous multimodal optimization problem.

3.2.2 Adaptive decomposition of search space

Instead of decomposing the problem search space in a pure discrete form, 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 a initial seed hypersphere. The initial initial seed hypersphere is defined by a very small range of displacement variables and load multiplier variable. 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 form all optimal solutions, it search for a new hypersphere. This process continues until the max displacement of the control point of a problem is reached (see Figure 4).

Algorithmically, the adaptive decomposition of search space method for finding the suitable hyperspheres is as follows:

Step 1. The first step is to initialize the starting hypersphere H_1 with center in $\mathbf{c}_1$ and radius r. The center of H_1 is identified by a 2-dimension vector $[d_1,\ \lambda_1]$, where d_1 is the initial displacement control point (generally zero) and λ_1 is the corresponding load multiplier that guarantees the global equilibrium solution. A gradient-free optimization algorithm then finds several trial solutions within this hypersphere H_1.

Step 2. The second center $\mathbf{c}_2$ is the farthest point in the hypersphere H_1 from its center $\mathbf{c}_1$ among all trial solutions in Step 1 that obey the space truss structure solution feasibility constraints. Then the hypersphere H_2 is formed around the center $\mathbf{c}_2$ with radius r. Similar to step 1, several trial solutions within the hypersphere H_2 are generated for obtaining the hypersphere H_3.

Step 3. The third center $\mathbf{c}_3$ is the farthest point from the center $\mathbf{c}_2$ among all trial solution points $\mathbf{c}_i$ obtained in Step 2 for the hypersphere H_2 and is the point that satisfies the following:

$$\mathbf{b}_i \cdot \mathbf{a} < 0 \quad \text{and} \quad ||\mathbf{b}_i|| > ||\mathbf{b}_j||, \quad i \neq j \text{ for all } i \text{ and } j \text{ in } H_2, \tag{9}$$

where i and j are indexes of trial solutions in the hypersphere H_2, vector $\mathbf{a} = \mathbf{c}_1 - \mathbf{c}_2$, and $\mathbf{b}_i = \mathbf{c}_i - \mathbf{c}_2$. Once the center $\mathbf{c}_3$ is obtained, a hypersphere H_3 is formed with a radius r. This step is shown in Figure 4.

Step 4. At this stage, all three necessary centers $\mathbf{c}_1$, $\mathbf{c}_2$, and $\mathbf{c}_3$ are obtained for the algorithm to keep rolling and incrementally find other hyperspheres until the stopping criteria are met.

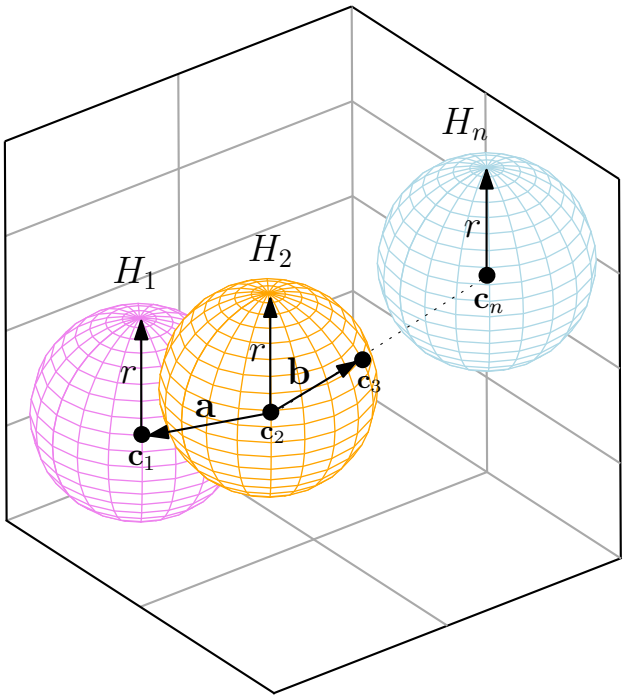

Figure 4: Hyper-sphere search algorithm. An initial hypersphere H_1 is first constructed based on an input radius r and the center $\mathbf{c}_1$. Hyperspheres $H_2, \ldots, H_n$ are discovered by the algorithm automatically by taking a boundary point (i.e., a solution within a hypersphere that has the largest distance from the previous hypersphere's center). For example, center $\mathbf{c}_2$ has a distance $\mathbf{a}$ from center $\mathbf{c}_1$ and center $\mathbf{c}_3$ has a distance $\mathbf{b}$ from center $\mathbf{c}_2$.

The user-defined hyper-parameters of this algorithm are the radius r, the number of trial solutions needed to be produced in a hypersphere, and the stopping criteria. While the radius r, the number of trial solutions depend on experiments, 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 could be made about the buckling of the structure).

3.3 Experiment setup

3.3.1 Algorithms evaluated for the optimization

Before explaining the problem's search domain analysis, this Section briefly explains each optimization algorithm. The first set of algorithms chosen was 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 DIRECT [30]. SA is a heuristic-based gradient-free algorithm [31]. It 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 was 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 [32], while ACO follows ants [33], and PSO follows the behavior of a flock of birds or a school of fish [34, 35]. 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 [34] (PSO-Std) or constriction factor [35] (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 of the family of evolutionary algorithms, which uses mutation, crossover, and selection operations on the solution vectors [36]. There are several versions of the DE algorithm [37]. 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 [38], 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.

Specifically, for each individual vector $\mathbf{v}_i$ in the population, DE/rand/1/bin version randomly selects three unique solutions $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$ for creating a mutation vector $\mathbf{v}_m$, where it takes $\mathbf{v}_1$ as its base vector. While DE/best/2/bin takes the current best solution $\mathbf{v}_{best}$ as its base vector and selects an additional random vector $\mathbf{v}_4$ for the mutation operation. The mutation operation based on a uniform random vector $\mathbf{r}$ of DE/rand/1/bin and DE/best/2/bin are

$$\mathbf{v}_m = \mathbf{v}_1 + \mathbf{r}(\mathbf{v}_2 - \mathbf{v}_3) \quad \text{and} \quad \mathbf{v} = \mathbf{v}_{best} + \mathbf{r}(\mathbf{v}_1 + \mathbf{v}_2 - \mathbf{v}_3 - \mathbf{v}_4). \tag{10}$$

On this mutation vector $\mathbf{v}_m$, a binomial (uniform) crossover operation is performed with a crossover probability Boolean vector $\mathbf{c}$ and its inverted vector $\tilde{\mathbf{c}}$ for creating a new vector $\mathbf{v}_{i,\text{new}}$ as:

$$\mathbf{v}_{i,\text{new}} = \mathbf{c}\mathbf{v}_m + \tilde{\mathbf{c}}\mathbf{v}_i. \tag{11}$$

Lastly, DE replaces $\mathbf{v}_i$ with $\mathbf{v}_{i,\text{new}}$ in the population, if this new solution $\mathbf{v}_{i,\text{new}}$ is better than the older solution $\mathbf{v}_i$.

3.3.2 Hyperparameter setup of 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, and it had a single solution. Similar to DIRECT, SA had a single solution. However, its other hyperparameters temperature and initial temperature reduction rate were 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: ABC's hyperparameter abandonment limit and acceleration coefficient upper bound set to 0.6 and 1; ACO's hyperparameter sample size, selection pressure, and deviation-distance ratio, respectively set to 40, 0.5, and 1; PSO-Std's hyperparameter inertia weight, inertia weight damping ratio, personal learning coefficient, global learning coefficient respectively set to 1, 0.99, 1.5, and 2.0; and PSO-Const's 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 DE algorithm's hyperparameters 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 rest of the experiment settings are described in results and analysis (Section 4) as they are subjective to types of analysis and problems.

4 Results and analysis

4.1 Benchmark 1: Eight-member shallow truss structure

The first benchmark problem was a shallow truss structure comprising eight trusses fully restrained at the base and connected to each other by a central node (Figure 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 mm^2, and Young's modulus of 70 MPa. 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 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 Figure 6.

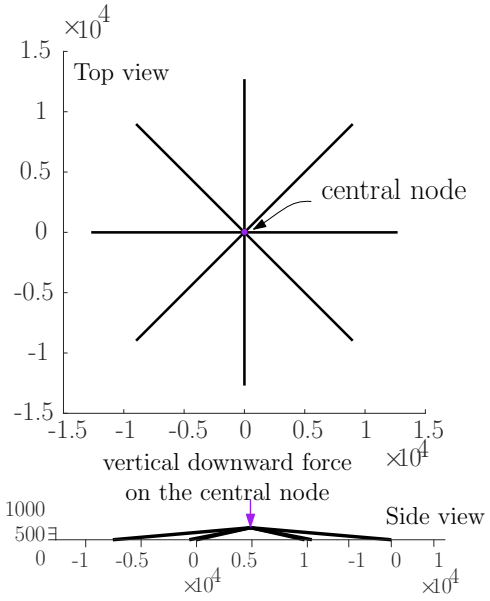

Figure 5: Eight-Member Shallow Truss Structure: Top view and Side view (dimensions in mm). The central node of the structure is indicated by a lighter color dot in Top view and a light color arrow pointer in Side view. The arrow on Side view indicates a vertical downward external force of 4.45 kN applied on the central node.

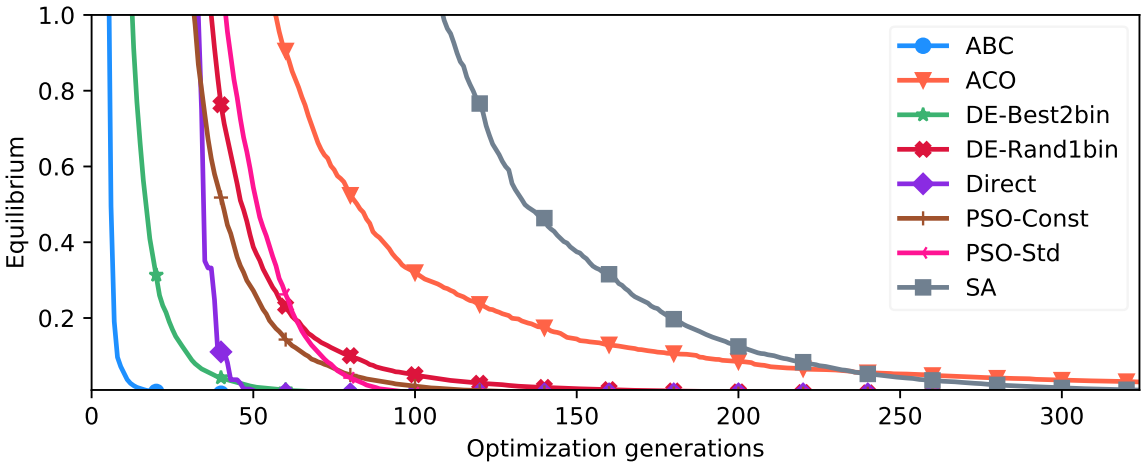

Figure 6: Convergence profile of algorithms averaged over 1000 trail solutions, all with optimum equilibrium value 1×10^{-5}. Algorithms ABC, ACO, DE/best/2/bin, DE/rand/1/bin, Direct, PSO-Const, PSO-Std, and SA, respectively are in blue, orange, green, red, purple, brown, magenta, and gray; and they are respectively marked with symbols $\bullet, \blacktriangledown, \bigstar, \times, \blacklozenge, |, <$, and $\blacksquare$. The shaded width of color around a line represents the standard deviation of the convergence.

Figure 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—Figure 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 Figure 7 and 8 represents the analytical solution from literature [1, 20, 23].

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 towards 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 Figure 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 Figure 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 Sixteen-Member Shallow Truss Structure (Figure 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 mm^2 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 Figure 9.

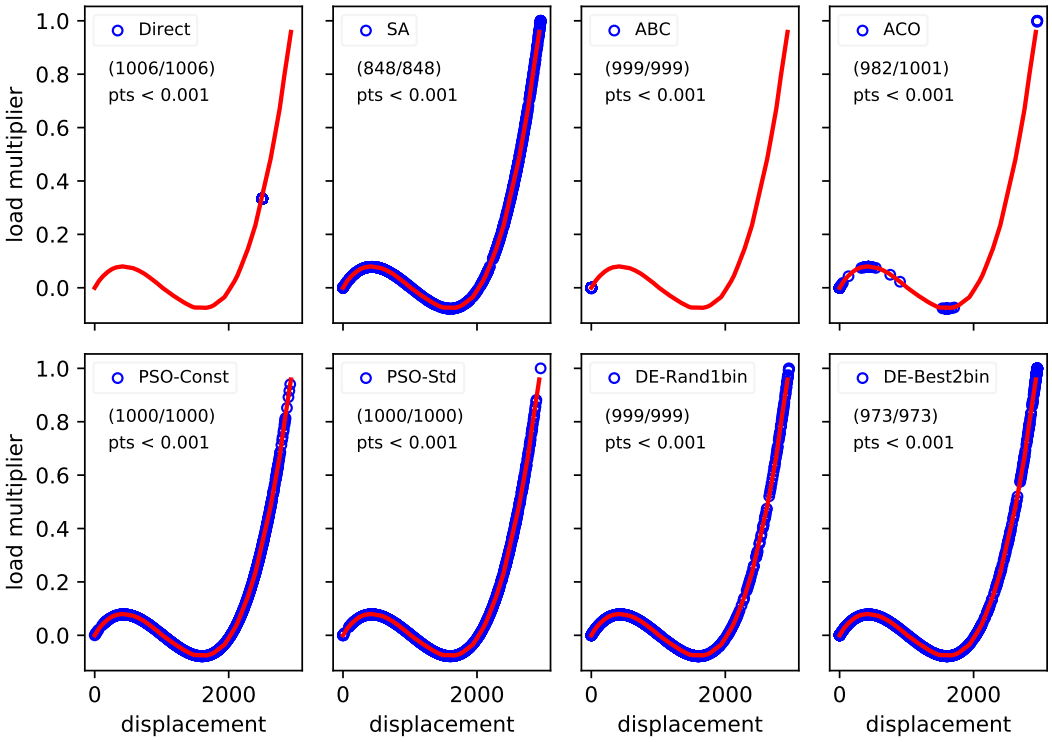

Figure 7: Eight-Member truss structure optimized solutions generated by searching with a domain setting of $[0, 1]$ for load multiplier and $[0\,\text{mm}, 3000\,\text{mm}]$ for displacement variables (DoF), including the control point. Each plot is an x-y plane of control point displacement (x-axis) 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.

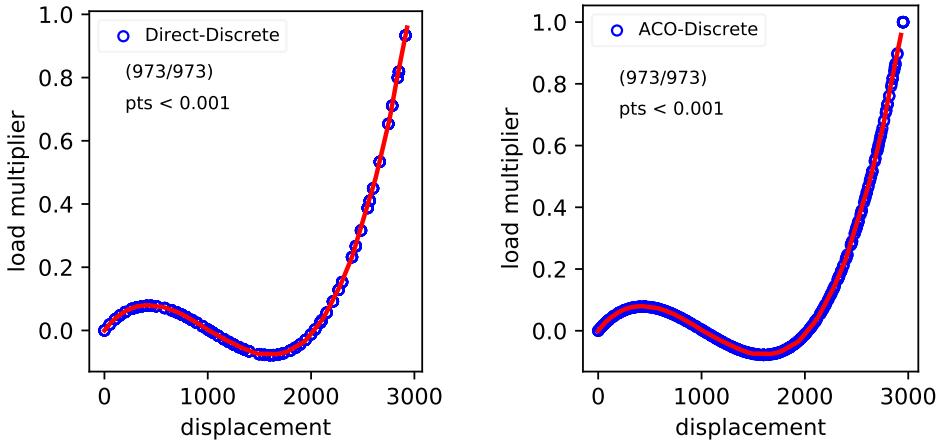

Figure 8: Solutions of Direct (*Left*) and ACO (*Right*) generated by decomposing domain of only control point (variable related to central node), which produced discrete sets like $[0, 250], [250, 500], \cdots, [2750, 3000]$. For each set, approx 100 trail solutions were evaluated.

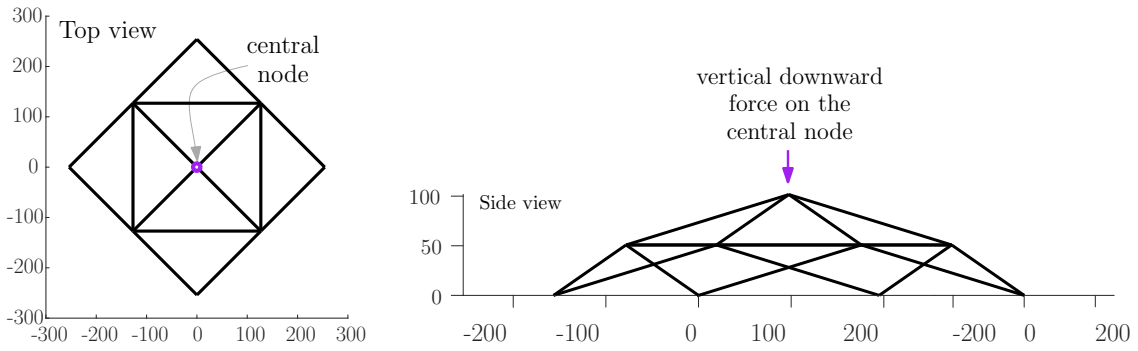

Figure 9: Sixteen-Member Shallow Truss Structure. *Left:* Top View. *Right:* Side View. Th central node of the structure is indicated by a lighter color dot in Top view and a light color arrow pointer in Side view. The arrow on Side view indicates a vertical downward external force of 4450 kN applied on the central node.

4.2.1 Results of the fixed decomposition of search space

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 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 Figure 10). Therefore, a decomposed domain analysis procedure (see Section 3.2) was launched to find a suitable domain for respective variables. Again, the continuous red line in Figure 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 Figure 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 Figure 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

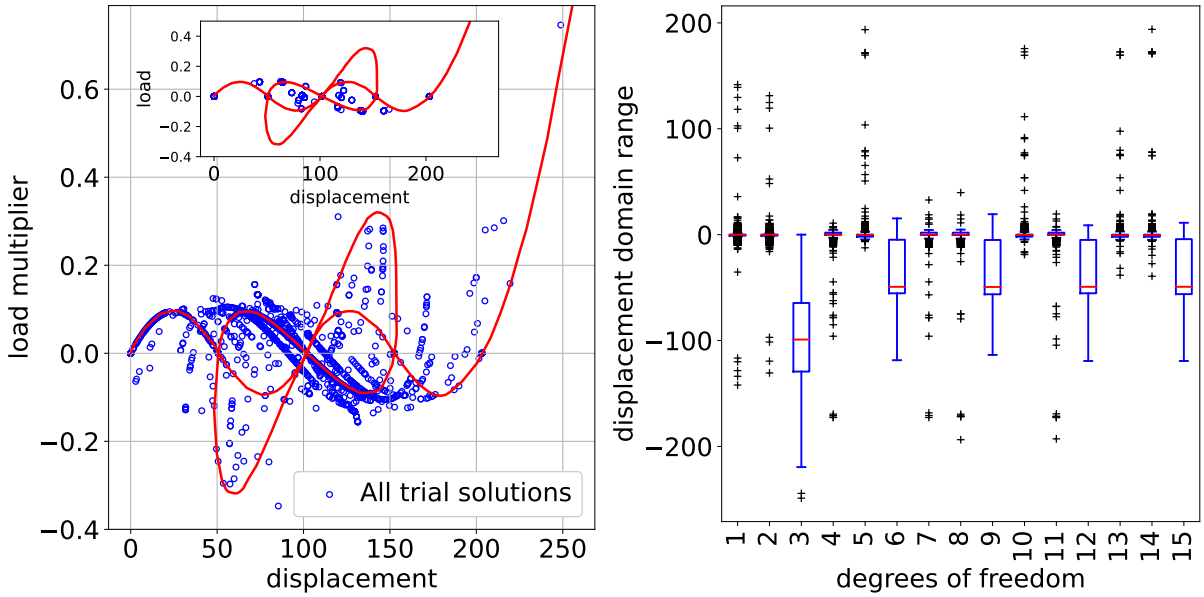

Figure 10: Sixteen Members initial trial solutions. *Left:* initial trials for all discretized domain (inner-plot shows solution obtained for the experiment on full non-discretized domain [0 mm, 250 mm]). *Right:* Displacement variables (DoF) domain analysis and approximation.

optimization algorithm as well. However, these solutions correspond to alternative equilibrium paths characterized by a non-vertical displacement of the central node.

Figure 10(right) shows the displacement value distribution of each variable. The plot in Figure 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 the learned search domain

On the obtained knowledge of the domain shown in Figure 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. Figure 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 Figure 11(left).

Figure 11(right) offers a microscopic view of the convergence of these algorithms. In Fig-

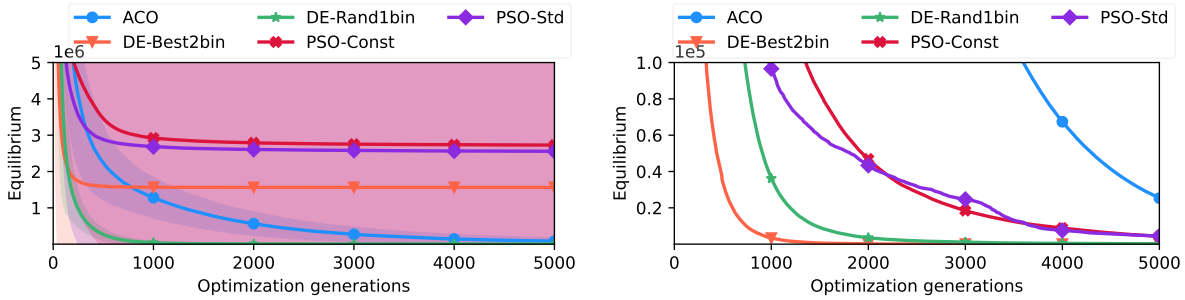

Figure 11: Optimization of Sixteen-Member shallow truss structure using five algorithms: (*Left*) Convergence profile of algorithms averaged over solutions with equilibrium value of 500 and (*Right*) a closer snapshot of convergence profile with non-converging solutions filtered out. Algorithms ACO, 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 Equation (5).

ure 11(right), all solutions converged to a suboptimal equilibrium function [Equation (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, DE/rand/1/bin algorithm's all solutions also converged to global minima and 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 [1, 20, 23] 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 Figure 11(right) were plotted on an x-y plane of control point displacement and the load multiplier in Figure 12.

Solutions accuracy analysis against the analytical curve. Figure 12 plots suboptimal solutions in cyan and optimal solutions in blue. As shown in Figure 12(row 1), the ACO algorithm,

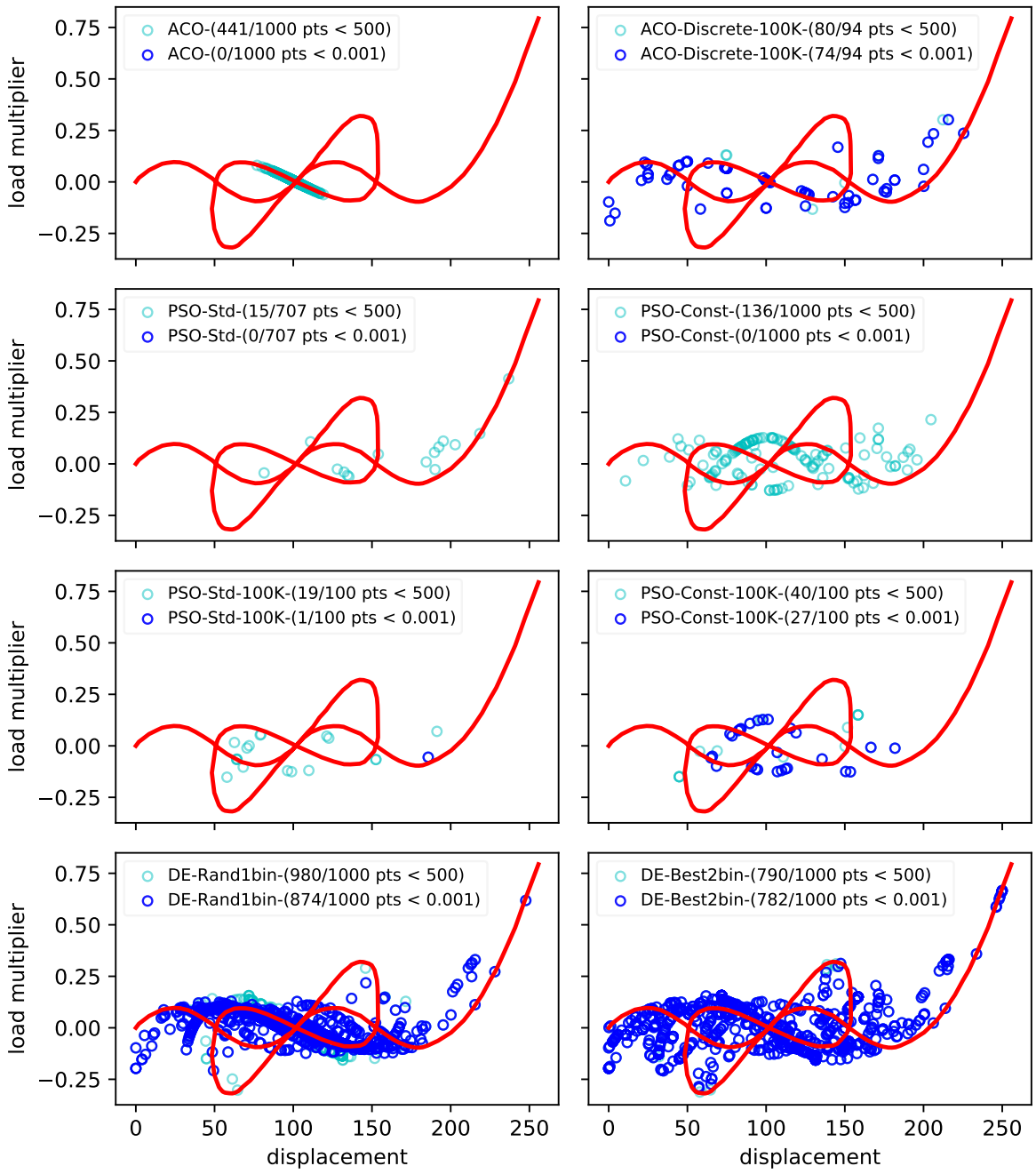

Figure 12: Quality of solutions (in blue solutions with a precision of 0.001 as their equilibrium value and solution in cyan are with precision 500) produced by optimization algorithms over 10 000 generations and 100 000 generations for ACO-Discrete and PSO-100K versions. Arc-length method based equilibrium 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.

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 Figure 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 Figure 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 (Figure 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 Figure 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 [39] was performed to analyze whether a few sets of solutions follow

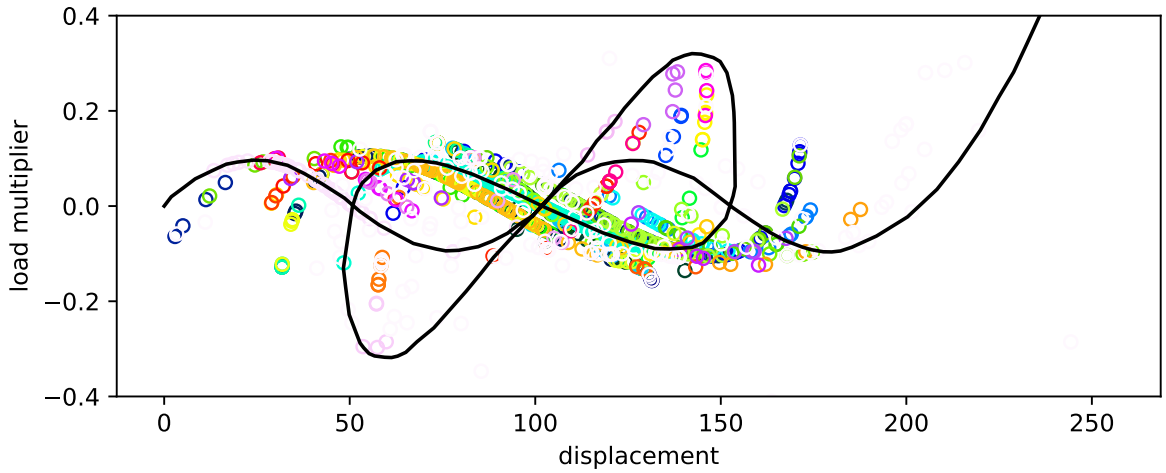

Figure 13: Clustering of solutions (obtained by DE/rand/1/bin algorithm) for identifying different feasible characteristics of equilibrium path (in different color circles).

similar properties and whether they can be grouped together that identify particular equilibrium path characteristics. The DBSCAN algorithm was chosen here because of this equilibrium path characteristics identification required to scan through the neighborhood of the point cloud formed by solutions, and the nearest connected points were expected to cluster together to represent particular characteristics. Figure 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 decomposition of search space

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 $[-10\,\text{mm}, 10\,\text{mm}]$ for the displacement variables and $[-0.2, 0.2]$ for the load multiplier variable. First, a set of 5 trial solutions was evaluated in step 1 pertaining to hypersphere initialization and first hypersphere center identification of this algorithm (cf. Section 3.2.2). Then, a radius $r = 5\,\text{mm}$

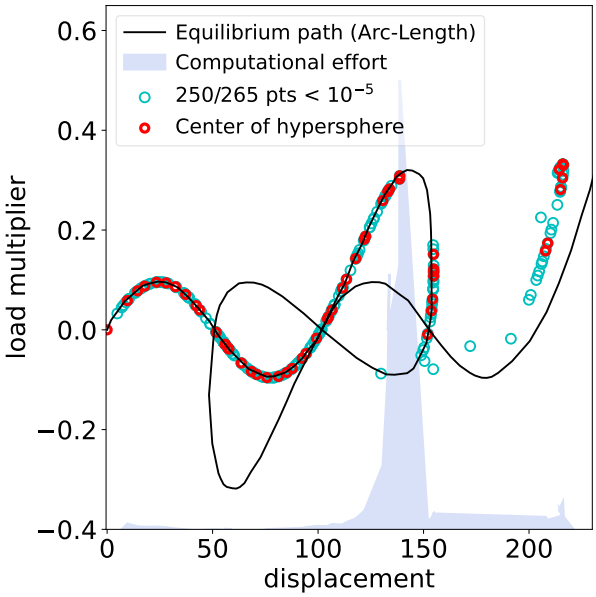

Figure 14: Quality of solutions produced by the hypersphere search algorithm with input radius $r = 5\,\text{mm}$. Solutions in cyan are trial solutions of respective hypersphere whose centers are indicated in red.

was used for the construction of the next hypersphere around the obtained center, and another set of 5 trial solutions was 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.

Figure 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 $r = 5\,\text{mm}$ 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 point). This was done by making radius r adaptive to the iteration of hypersphere construction as per this expression: $r_{\text{new}} = r_{\text{prev}} \times 0.5$. The analysis in Figure 15(left) shows that the algorithm can follow the sharp breaking path; however, it is computationally challenging and slow. Eventually, the hyperspheres become small and infeasible for the algorithm to continue.

Another trial was an additive approach where radius r was gradually increased by a value of 5mm, i.e., $r_{\text{new}} = r_{\text{prev}} + 5$. Figure 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 like the many alternatives found in clustering analysis in Section 4.2.1.

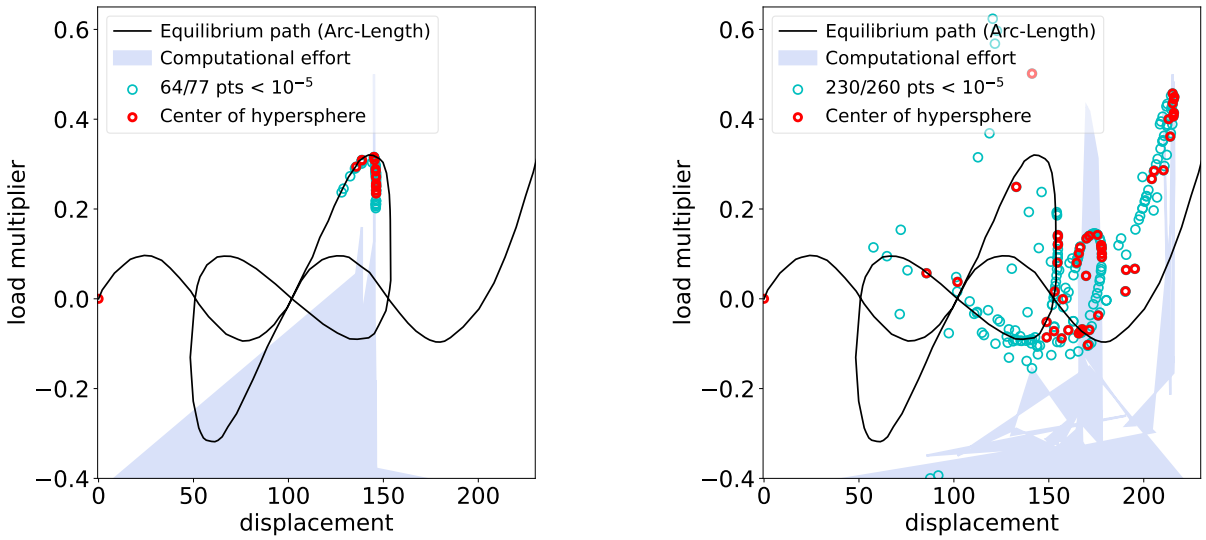

Figure 15: Sixteen-Member solutions produced by hypersphere search algorithm with input radius $r_{\text{new}} = r_{\text{prev}} \times 0.5$ (*Left*) and with input radius $r_{\text{new}} = r_{\text{prev}} + 5\,\text{mm}$ (*Right*). Solutions in cyan are trial solutions of respective hypersphere whose 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 Figure 14.

4.3 Test Problem: 3D 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 mm, a width of 2000 mm, and a height of 750 mm, arranged with the geometrical layout shown in Figure 16(row 1). The beam is composed of 33 trusses, 10 free nodes, and 30 DoF. Each truss has 2500 mm^2 of cross-section area and 200 000 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 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 feasible 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 mm, 2500 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 Figure 16. In Figure 16, rows 2 to 5 show deformed shapes of four equilibrated and feasible solutions.

A rough representation of the equilibrium path is represented in Figure 17 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. Figure 16 and 17). On the other hand, solution C shows a buckling behavior (cf. Figure 16 and 17) 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. Figure 16 and 17). After this point, the system behaves as a catenary rather than a beam, and the equilibrium is again stable.

5 Discussions

This paper presents an analysis and a methodology based on gradient-free optimization algorithms for assessing the structural response of space truss structures subjected to large displacements, characterized by a high nonlinearity and unstable post-buckling behavior. Compared to the standard Newton-Raphson procedures, the proposed methodology does not require the assemblage and update of the global stiffness and geometric stiffnesses (Section 2.2). Moreover, it is not exposed to the potential numerical issues affecting the classic Newton-Raphson and Arc-length methods in correspondence of 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, and PSO), and an evolutionary algorithm (DE) (Section 3.3.2). It also evaluated other popular algorithms: multilevel coordinate search [40] and covariance matrix adaptation evolution strategy [41]. However, these two algorithms were not converging for these problems as this problem's search domain was ill-informed. A separate study may require for performing a more comprehensive sensitivity analysis of these algorithms for an appropriate formulation of these problems. Hence, the scope of this work mainly relied on DE algorithm

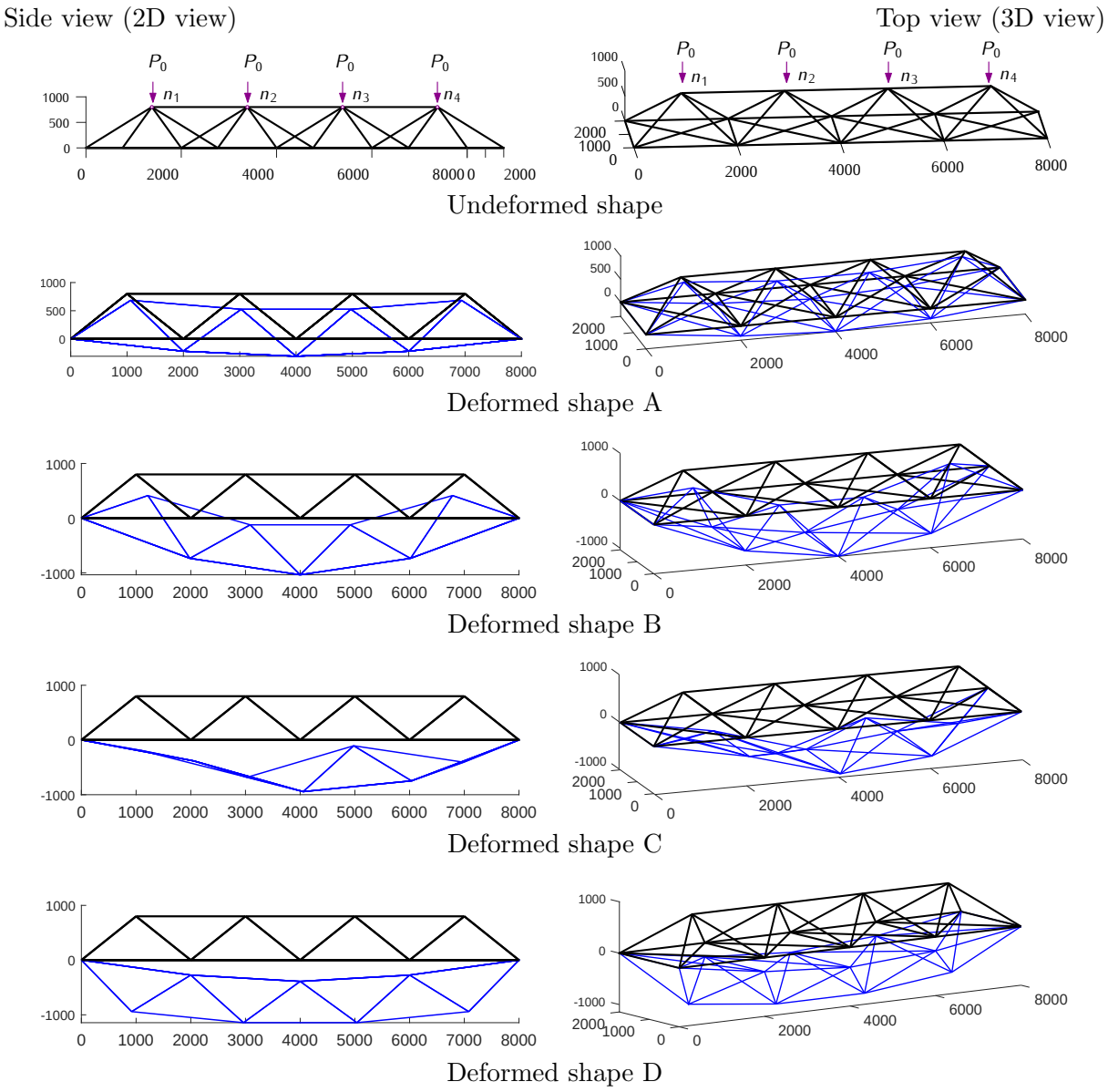

Figure 16: Equilibrium stages of 3D reticular beam structure optimized using DE algorithm version DE/rand/1/bin. Lighter color arrows (on Side view and Top View) indicates the application of the vertical downward force ($P_0 = 100\,\text{kN}$) applied on the nodes n_1, n_2, n_3, and n_4 of 3D reticular 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 Equation (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.

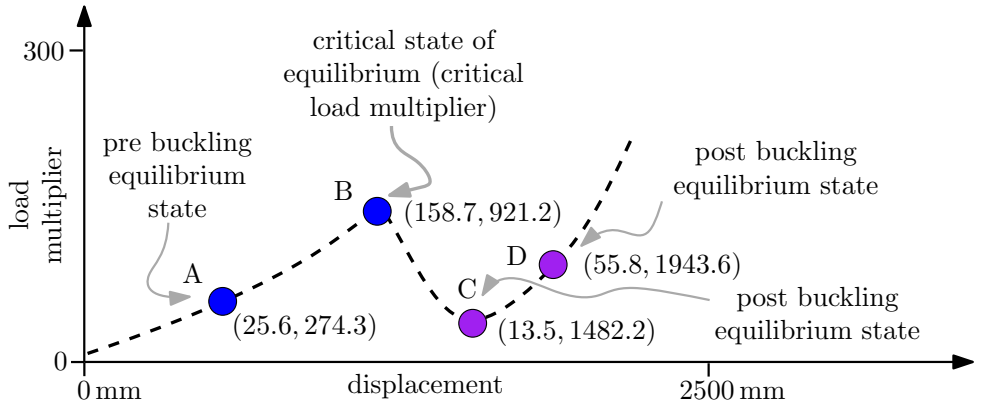

Figure 17: Pre- and post buckling equilibrium states of the 3D reticular beam. These states are obtained for a multistep analysis mentioned in Figure 3. The dotted line is the hypothesis equilibrium path connecting the obtained states for this test problem.

versions as they were competitively the most successful on this problem.

Two benchmark structures and one test case structure were presented as an optimization problem to these optimization algorithms (Section 4). The optimization of space truss structures was found highly challenging as variables were highly sensitive to each other due to their physical interactions. 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). This makes this problem a class of nonlinear, multimodal, unconstrained optimization.

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. Figure 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 complex for DIRECT to work properly. A similar observation is applied to the Simulated annealing algorithm, which was able to solve the first more minor benchmark problem of four variables but 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. Figure 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. Figure 7 and 12). The DE versions were the best performing algorithms

for this class of ill-posed problems. Despite being best performing, DE versions were unable to find the expected number of accurate solutions on the equilibrium path (Figure 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 (Figure 10 and Figure 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 (Figure 11).

The proposed hypersphere algorithm incrementally constructed informed hyperspheres using the knowledge from the previous hyperspheres (Section 3.2.2). This algorithm provided solutions accurately, and its performance was comparable to the classical Arc-length method used in civil engineering for the equilibrium path analysis (Figure 14). However, the main challenges 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 (Figure 15).

On the test problem, the 3D reticular beam, which is a medium-size real 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 4.3 have high precision and produce solutions for both pre- and post buckling stages (Figure 16). These solutions allowed producing hypothesis equilibrium path (Figure 17).

This research demonstrated the use of optimization algorithms for space truss structures optimization. In addition, it successfully solved 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 results demonstrate that the proposed procedure can describe complex post-buckling behaviors of structures.

More importantly, this research demonstrated that this heuristic-based optimization of space truss structures, unlike standard Newton-Raphson procedures single analytical solution (equilibrium path), can produce solutions (equilibrium path) characterized by vertical hypothesis displacement as well other possible characterization of non-vertical hypothesis displacement. Therefore, this presents an effective alternative to Newton-Raphson procedures or combined with them to analyze real structural systems subjected to extreme loads or progressive col-

lapses.

At the same time, this nonlinear, multimodel, unconstrained minimization optimization problem presents significant challenges to optimization algorithms. This research presents this problem as a testbench problem to the optimization research community to test algorithms on this difficult and challenging problem. 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), 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 nonlinear, multimodal, unconstrained, continuous optimization problem concerning structural engineering: space truss structure optimization. The search landscape of this problem poses significant challenges to existing continuous optimization algorithms in terms of assessing the ability to work on ill-posed problems and produce diverse and accurate solutions with high convergence speed. In this research, a number of strategies for search domain analyses are presented. Consequently, a hypersphere search algorithm is proposed that iteratively moves through the problem's search landscape to find as many accurate solutions as possible for a benchmark space truss structure. Differential evolution is identified as the most competitive algorithm to solve this class of problems which is eventually applied to solve a test problem concerning 3D reticular beam, where the algorithm optimally produced deformed equilibrium shapes and hypothesis equilibrium path. This research presents an effective alternative to Newton-Raphson procedures to analyze real structural systems subjected to extreme loads or progressive collapses.

References

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

[2] R. Martin and N. J. Delatte, "Another look at hartford civic center coliseum collapse," *Journal of Performance of Constructed Facilities*, vol. 15, no. 1, pp. 31–36, 2001.

[3] E. A. Smith, "Space truss nonlinear analysis," *Journal of Structural Engineering*, vol. 110, no. 4, pp. 688–705, 1984.

[4] E. Murtha-Smith, "Alternate path analysis of space trusses for progressive collapse," *Journal of Structural Engineering*, vol. 114, no. 9, pp. 1978–1999, 1988.

[5] G. Blandford, "Progressive failure analysis of inelastic space truss structures," *Computers & structures*, vol. 58, no. 5, pp. 981–990, 1996.

[6] M. Gilbert and A. Tyas, "Layout optimization of large-scale pin-jointed frames," *Engineering Computations*, 2003.

[7] A. Tyas, M. Gilbert, and T. Pritchard, "Practical plastic layout optimization of trusses incorporating stability considerations," *Computers & Structures*, vol. 84, no. 3-4, pp. 115–126, 2006.

[8] R. Horst and H. Tuy, *Global optimization: Deterministic approaches*. Springer, 2013.

[9] M. Saka, "Optimum design of steel frames using stochastic search techniques based on natural phenomena: a review," *Civil Engineering Computations: Tools and Techniques*, pp. 105–147, 2007.

[10] O. Hasançebi, S. Çarbaş, E. Doğan, F. Erdal, and M. Saka, "Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures," *Computers & Structures*, vol. 87, no. 5-6, pp. 284–302, 2009.

[11] H. Huang, H. An, H. Ma, and S. Chen, "An engineering method for complex structural optimization involving both size and topology design variables," *International Journal for Numerical Methods in Engineering*, vol. 117, no. 3, pp. 291–315, 2019.

[12] V. Lute, A. Upadhyay, and K. K. Singh, "Computationally efficient analysis of cable-stayed bridge for ga-based optimization," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 4-5, pp. 750–758, 2009.

[13] G. A. Hrinda and D. T. Nguyen, "Optimization of stability-constrained geometrically nonlinear shallow trusses using an arc length sparse method with a strain energy density approach," *Finite Elements in Analysis and Design*, vol. 44, no. 15, pp. 933–950, 2008.

[14] A. R. Yildiz, "Comparison of evolutionary-based optimization algorithms for structural design optimization," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 327–333, 2013.

[15] M. Gilbert, C. Casapulla, and H. Ahmed, "Limit analysis of masonry block structures with non-associative frictional joints using linear programming," *Computers & Structures*, vol. 84, no. 13-14, pp. 873–887, 2006.

[16] M. Ferris and F. Tin-Loi, "Limit analysis of frictional block assemblies as a mathematical program with complementarity constraints," *International Journal of Mechanical Sciences*, vol. 43, no. 1, pp. 209–224, 2001.

[17] C. Baggio and P. Trovalusci, "Collapse behaviour of three-dimensional brick-block systems using non-linear programming," *Structural Engineering and Mechanics*, vol. 10, no. 2, p. 181, 2000.

[18] C. D. Bisbos and P. M. Pardalos, "Second-order cone and semidefinite representations of material failure criteria," *Journal of Optimization Theory and Applications*, vol. 134, no. 2, pp. 275–301, 2007.

[19] C. Bisbos, A. Makrodimopoulos, and P. M. Pardalos, "Second-order cone programming approaches to static shakedown analysis in steel plasticity," *Optimization Methods and Software*, vol. 20, no. 1, pp. 25–52, 2005.

[20] M. A. Crisfield, *Nonlinear finite element analysis of solids and structures. Volume 1: Essentials.* Wiley & Sons, 1991.

[21] M. A. Crisfield, *Nonlinear finite element analysis of solids and structures. Volume 2: Advanced Topics.* Wiley & Sons, 1996.

[22] B.-A. Memon and S. Xiao-zu, "Arc-length technique for nonlinear finite element analysis," *Journal of Zhejiang University-Science A*, vol. 5, no. 5, pp. 618–628, 2004.

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

[24] K.-J. Bathe, *Finite element procedures.* Klaus-Jurgen Bathe, 2006.

[25] W.-F. Chen and D.-J. Han, *Plasticity for structural engineers.* J. Ross Publishing, 2007.

[26] L. Macorini and B. Izzuddin, "A non-linear interface element for 3D mesoscale analysis of brick-masonry structures," *International Journal for Numerical Methods in Engineering*, vol. 85, no. 12, pp. 1584–1608, 2011.

[27] N. Wang and H. Adeli, "Self-constructing wavelet neural network algorithm for nonlinear control of large structures," *Engineering Applications of Artificial Intelligence*, vol. 41, pp. 249–258, 2015.

[28] C. J. Li and L. Yan, "Mechanical system modelling using recurrent neural networks via quasi-newton learning methods," *Applied Mathematical Modelling*, vol. 19, no. 7, pp. 421–428, 1995.

[29] M. Geradin, S. Idelsohn, and M. Hogge, "Computational strategies for the solution of large nonlinear problems via quasi-newton methods," *Computers & Structures*, vol. 13, no. 1-3, pp. 73–81, 1981.

[30] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, "Lipschitzian optimization without the lipschitz constant," *Journal of Optimization Theory and Applications*, vol. 79, no. 1, pp. 157–181, 1993.

[31] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[32] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing*, vol. 8, no. 1, pp. 687–697, 2008.

[33] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1155–1173, 2008.

[34] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[35] M. Clerc, "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, 1999, pp. 1951–1957.

[36] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[37] K. R. Opara and J. Arabas, "Differential evolution: A survey of theoretical analyses," *Swarm and Evolutionary Computation*, vol. 44, pp. 546–558, 2019.

[38] Scipy, "scipy.optimize.differential_evolution," https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.differential_evolution.html, Dec 2021.

[39] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "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, 1996, pp. 226–231.

[40] W. Huyer and A. Neumaier, "Global optimization by multilevel coordinate search," *Journal of Global Optimization*, vol. 14, no. 4, pp. 331–355, 1999.

[41] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *Proceedings of IEEE International Conference on Evolutionary Computation*. IEEE, 1996, pp. 312–317.