A Non-Uniform Cellular Automata Approach for the Design Optimization of Truss Structures

Mohamed El Bouzouiki

A Thesis

In the Department of
Mechanical, Industrial and Aerospace Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy (Mechanical Engineering)
at Concordia University
Montréal, Québec, Canada

September 2023

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Mohamed El Bouzouiki**

Entitled: **A Non-Uniform Cellular Automata Approach for the Optimization of Truss Structures**

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (Mechanical Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________________ Chair
Dr. Pragasen Pillay

______________________________ External Examiner
Dr. Afzal Suleman

______________________________ External to Program
Dr. Anjan Bhowmick

______________________________ Examiner
Dr. Marius Paraschivoiu

______________________________ Examiner
Dr. Javad Daraghi

______________________________ Thesis Supervisor
Dr. Ramin Sdaghati

______________________________ Thesis Co-Supervisor
Dr. Ion Stiharu

Approved by: ______________________________
Dr. Muthukumaran Packirisamy, Graduate Program Director

September 08, 2023

____
Dr. Mourad Debbabi, Dean
Gina Cody School of Engineering and Computer Science

Abstract

A Non-Uniform Cellular Automata Approach for the Optimization of Truss structures

Mohamed EL Bouzouiki, Ph.D.
Concordia University, 2023

Cellular automata (CA) paradigm has been successfully applied to solve the topology and sizing optimization problems of truss structures and continuum bodies. In a conventional uniform Cellular Automata (CA) model, a unit cell's behavior is updated based on information from its eight immediate neighboring nodes (in 2D configuration). However, this neighborhood system may not always be suitable for representing real truss structures. To address this limitation, a non-uniform CA approach has been proposed in this research study for the optimization of truss structures.

The proposed non-uniform CA approach is based on non-identical cells, each of which is defined by a center node and members connecting the center node of the cell to all other nodes in its immediate neighborhood. This differs from the Moore neighborhood concept used in the conventional uniform CA approach, which only considers the eight immediate neighboring nodes. It has been shown that the proposed non-uniform CA approach provides a more realistic representation of truss structures and improves the optimization process.

Moreover, conventional CAs rely on a fixed grid, thus with respect to design optimization of discrete structures, they can only be used for sizing and topology of the structure while the layout optimization cannot be conducted as the coordinates of the nodes remain unchanged throughout the optimization process. In this research study, a novel non-uniform CA design optimization algorithm has been formulated to solve the general problem of topology, sizing and layout optimization of truss structures subject to both stress and displacement constraints. Several benchmark case studies have been provided to demonstrate the efficiency and accuracy of the proposed design optimization methodology.

Dedication

This dissertation work is dedicated to my beloved parents

Acknowlegments

I would like to take this opportunity to greatly and deeply thank my supervisors, Professors Ramin Sedagahati and Ion Stiharu, for their nonstop supports, suggestions and direction. I would like to thank them for all their contributions, guidance, remarkable ideas and encouragement.

I also want to thank administrative, professional and technical staff at Concordia University, particularly Leslie Hossein, Charlene Wald and Sophie Merineau.

My deepest heartfelt gratitude goes to my wife, Safoua. She supported and encouraged me during my PhD study until I finish it and I am really grateful for having her in my life. I would also like to express my profound gratitude to my father Mustapha, my mother Maimouna, and all my all my siblings. Words simply cannot express my love for my family. They have been with me every step of the way and I could not have achieved it without them.

Lastly, I would like to thank Concordia University and whole staff for preparing such an environment and opportunity for me to carry on my studies to the doctoral level.

TABLE OF CONTENTS

Chapter Title **Page**

List of Figures

List of Tables

List of Symbols

Symbol	**Meaning**
W	Total weight of the structure.
n	Total number of nodes in the structure.
m	Total number of members in the structure.
ρ_j	Density of a given member.
A_j	*A*rea of a given member.
L_j	Length of a given member.
δ_j	Elongation of a given truss member.
ε_j	Strain in a given member.
σ_j	Stress in a given member.
P_j	Internal force within a given member.
$\overline{\sigma_j}$	Allowable Stress in a given member.
θ_j	Orientation angle of a given member.
N	Number of elements in a given cell.
S_i	State of a given cell.
u_i	Displacement in the x- direction of a given cell.
v_i	Displacement in the y- direction of a given cell.
$\overline{u_i}$	Allowable displacement in the -x direction of a given cell.
$\overline{v_i}$	Allowable displacement in the -y direction of a given cell.
F_{xi}	External force in the x- direction at a given cell.
F_{yi}	External force in the y- direction at a given cell.
V_i	Total potential energy in a given cell.
U_j	Strain energy density of a given member
U_{cell}	Total strain energy density stored in a given cell.
η_j	Scaling factor of a given member.

List of Abreviations

Abbreviation	**Definition**
CA	Cellular Automata
OC	Optimality Criteria
FSD	Fully Stressed Design
FUD	Fully Utilized Design
NLP	Non-linear Programming
SLP	Sequential Linear Programming
GA	Genetic Algorithms
ACO	Ant Colony Optimization
GSS	Guided Stochastic Search
CSS	Charged System Search
PSO	Particle Swarm Optimization
SA	Simulated Annealing
FA	Firefly Algorithm
HS	Harmony Search
CGA	Cellular Genetic Algorithm
FCD	Fully Constrained Design
SSO	Swallow Swarm Optimization
iPSO	integrated particle swarm optimizer
SCPSO	sequential cellular particle swarm optimization
CSS	C*harged System Search*

CHAPTER 1
Introduction and Scope of Dissertation

1.1 Introduction and motivation

Structural optimization refers to the process of designing structures that meet specific criteria in a manner that is efficient, safe, and cost-effective [1]. This type of design process seeks the most efficient solutions for structures, materials, and components that satisfy all design and functional requirements while ensuring safety.

As a multi-disciplinary field, structural optimization combines engineering analysis, material science knowledge, and advanced optimization techniques to find the most efficient design solutions that meet all design requirements [2]. In this regard, finite element analysis and other numerical techniques are used to evaluate the structural performance of a design under various loading conditions [1]. Additionally, optimization algorithms are incorporated in order to optimize the design parameters of a structure to meet given objectives [2].

Structural optimization has gained significant attention in recent years. The origins of this discipline can be traced back to 1904, when Michell published his ground-breaking work on the topology optimization of truss structural systems. Despite this early start, it took several decades for the technique to become a commonly used design tool. The advent of high-performance computing has revolutionized the field of structural optimization, allowing engineers and researchers to tackle complex optimization problems with ease using two popular methods to solve these problems - mathematical programming methods based on the calculation of gradients of the objective and constraint functions, and optimality criteria methods that derive conditions that characterize an optimal solution. Mathematical programming methods, such as Linear Programming (LP), Non-linear Programming (NLP), Integer Programming (IP), and Dynamic Programming (DP), calculate the gradients of the objective and constraint functions to determine the direction of the optimization. LP and NLP are used to find the optimal solution to linear and non-linear problems. IP is used to find the optimal solution to linear or non-linear problems subjected to linear or non-linear constraints in which some variables restricted to take only integer values. DP is used to find the optimal solution to a problem that can be broken down into a sequence of smaller problems. However, these mathematical programming methods have limitations for solving large structural systems with many design variables and multiple constraints. This has led to the need for more robust methods that have the ability to find global optima when solving large non-convex structural optimization problems. This is where optimality criteria methods come into play. These methods use iterative techniques to change the design until optimal conditions are met, and have proven to be highly effective in solving structural optimization problems.

Heuristic methods have become an increasingly popular tool in the field of structural optimization in recent years, offering a quick and effective approach to solving complex design problems. Heuristic methods are problem-solving strategies that use rules of thumb, intuition, or experience to find a solution that is good enough for a given purpose [3]. In the context of structural optimization, heuristics involve using trial-and-error approaches or metaheuristics to find near-optimal or optimal solutions with relatively little computational effort.

One commonly used heuristic method in structural optimization is the genetic algorithm (GA), which is based on the principles of natural selection and evolution and generates a population of potential solutions to find the best one [9]. The method involves representing the structure using binary strings and applying a crossover and mutation operator to generate new solutions. GAs have been used to optimize various types of structures, including trusses, beams, and plates [3].

Particle Swarm Optimization (PSO) is another popular metaheuristic technique, which is a population-based optimization algorithm that mimics the behavior of bird flocking or fish schooling to identify the best solutions. Each particle represents a potential solution, and, guided by the best solutions found so far, the algorithm iteratively updates a swarm of particles that move in the search space to find the global optimum [4, 10].

Simulated Annealing (SA) is a metaheuristic algorithm that is inspired by the annealing process used in metallurgy. It uses a temperature-controlled process to explore a wide range of solutions while avoiding getting stuck in local optima [11].

The Ant Colony Optimization algorithm is based on the behavior of ants searching for food. As ants move, they lay down pheromone trails that other ants follow. The pheromone trail's strength increases with the amount of food found, and over time, the ants converge on the optimal solution. The algorithm simulates this behavior by constructing solutions by following the pheromone trails and updating the trail strength according to the quality of the solutions found. This way, the algorithm gradually improves the solution until it converges to an optimal solution [12].

Heuristics have been shown to be effective in solving complex structural optimization problems, often yielding results that are comparable to traditional optimization methods [3]. They are capable of handling the nonlinear and nonconvex nature of the problem and exploring a wide search space to find near-global optimal solutions [8]. Furthermore, they offer several advantages, including the ability to handle design problems with multiple objectives and constraints, the ability to find near-global optimal solutions in a relatively short time, and the ease of implementation without any need for gradients of objective and constraint functions [4]. Another advantage of metaheuristic techniques is their ability to handle multi-objective optimization problems, where multiple conflicting objectives need to be considered, and can also be combined with other optimization methodologies like gradient-based optimization to enhance the accuracy of the results and efficiency of the overall optimization process. Additionally, metaheuristic techniques can be parallelized to improve the computational efficiency, which is useful for large-scale truss structures.

However, these techniques still have the drawbacks of slow convergence rate and the need of high number of structural analyses. These shortcomings have led the researchers in the last decade to investigate cellular automata paradigm (CA) to solve structural optimization problems and this trend has opened a wide range of new possibilities that were never before achievable in structural optimization.

The concept of CA dates back to the 1940s when John von Neumann and his colleagues proposed the idea of building self-replicating machines, which laid the foundation for the development of CA [13]. However, it was not until 1970 that mathematician John Conway created the Game of Life, which is one of the most well-known examples of CA [14]. Since the creation of the Game of Life, many other types of CA have been developed and applied to various fields. In physics, CA have been used to simulate the behavior of fluids, gases, and other physical systems [15]. In computer science, CA have been used in the design of algorithms and in parallel computing [16]. In social sciences, CA have been used to model population behavior and the spread of diseases [17].

Cellular automata (CA) is a type of algorithm that simulates the behavior of complex systems through simple rules that govern the behavior of individual cells in a grid or lattice. Each cell's state is updated based on the states of its neighboring cells, following a set of predetermined rules that dictate how the system evolves over time.

In the case of structural optimization, CA can be used to simulate the stress and strain propagation in a structure and determine its equilibrium state under different loads. The goal is to find the optimal distribution of materials or design parameters that minimize the weight or cost of the structure while satisfying the required constraints [18]. CA-based structural optimization involves dividing the structure into a grid of cells and assigning each cell a set of parameters that determine its size, shape, and material properties. The state of each cell represents the local stress and strain at that point, which is subsequently updated based on the conditions of neighboring cells. The rules that govern the behavior of the cells are determined by a fitness function that measures the structural performance, such as the total weight or the maximum stress level [19]. Through iterations of the CA algorithm, the system evolves toward an equilibrium state that minimizes the fitness function. At each iteration, the cells' parameters are adjusted based on the updated stress and strain values, and the fitness function is re-evaluated. This process is repeated until the system converges to a stable state, which represents the optimized design [20].

CA-based structural optimization has been applied to various types of structures, including trusses, frames, and shells [21]. The method has shown promising results in reducing the weight and cost of structures while maintaining their strength and stiffness. However, the computational cost of CA-based optimization can be high, as the number of cells and iterations required to reach a stable solution increase with the complexity of the structure [22].

Cellular automata (CA) are highly versatile in their ability to model complex behavior through the use of simple rules. Despite the significant amount of memory and processing power required for simulating complex systems [23], the parallel computation capabilities of CA make them a valuable tool in various fields of research. As computational power continues to increase, CA are becoming highly suitable for solving large structural optimization problems.

1.2 Optimization of truss structures

Truss structure optimization is a subfield of structural optimization that specifically deals with optimizing truss structures. A truss is a structure made of interconnected members carrying only tension and compression and is widely used in engineering and construction including bridges, towers, and buildings, because of its high strength-to-weight ratio and material efficiency and ease of assembly [5, 6].

The objective of truss structural optimization is to find the optimal connections, member sizes, and truss shape that minimize weight and/or cost while maintaining or improving strength and stability [5]. Cost is often a significant factor in truss design and it is important to minimize costs while maintaining or improving performance. This can be done by using lower-cost materials or simplifying the truss design to reduce weight. The optimization process begins with a truss design, which is then analyzed using numerical methods such as finite element method to determine its strength and stability. The results of this analysis are used to identify areas of improvement, by either changing the truss configuration or using different materials [5].

The aforementioned optimization techniques discussed in section 1.1, including mathematical programming, gradient-based optimization, heuristic optimization, and others can be effectively employed for optimizing truss structures. These techniques allow for the optimization of various structural parameters, such as material properties, geometric dimensions, and load conditions.

The weight minimization problem of 2-d and 3-d elastic truss structures can be achieved by the solution of three different optimization sub-problems: Topology, Sizing and Layout optimization. Topology optimization is a design approach that aims to find the most efficient and structurally sound truss configuration, by optimizing the arrangement of nodes and bars in the structure. This is achieved by iteratively removing unnecessary material or structural members from an initial design, while preserving the overall stiffness and strength of the structure. The optimization process typically starts with an initial design of truss structure, which include a set of truss members and nodes. The optimization algorithm then seeks to minimize a given objective function, such as the weight of the structure or the compliance of the system under certain loads, subject to constraints on the maximum allowable stresses and displacements. The algorithm removes elements that do not contribute significantly to the objective and constraint functions, resulting in a new design iteration that is either stiffer and/or lighter than the previous one.

Another design approach is sizing optimization, which generally aims to optimize the cross-sectional area of each member of a truss structure to minimize its weight, while satisfying certain constraints such as maximum allowable stresses, deflections and frequencies. The process typically starts with an initial design of the truss structure, which includes a set of member cross-sectional areas. The optimization algorithm is then used to iteratively adjust the cross-sectional areas of the members, with the goal of minimizing the weight of the structure while satisfying the required constraints. Finally, the aim of the layout optimization is to find the optimal coordinates of each node within the structure. Simultaneous size, topology and lay optimization is a very challenging task, which will result in an optimal design superior to each sub-problem.

It is important to carefully consider constraints and related boundary conditions during the optimization process, maintain a minimum level of strength or stability, or to meet specific weight or cost requirements [7]. The manufacturing process and assembly of the structure should also be considered, as more efficient techniques, such as additive manufacturing, and cost-effective materials can be used [6].

A general design optimization problem for truss structures subject to stress and displacement constraints can be formally expressed as follows:

$$minimize \quad W(P) = \sum_{j=1}^{m} \rho_j A_j L_j \tag{1a}$$

$$Subject\ to \quad \boldsymbol{Ku} = \boldsymbol{f} \tag{1b}$$

$$|\sigma_j| < \bar{\sigma}_j , \qquad j = 1,2,\cdots,m \tag{1c}$$

$$|u_i| < \bar{u}_i , \qquad i = 1,2,\cdots,k \tag{1d}$$

$$\bar{A}^l < A_j < \bar{A}^u , \qquad j = 1,2,\cdots,m \tag{1e}$$

Equation (1a) defines the objective function. A_j are the design variables, which are considered to be the cross-sectional area of the truss members and m is the total number of bar members in the structure.

Equation (1b) is an equality constraint (stiffness equation) stating the equilibrium condition. $\boldsymbol{K}$ is the stiffness matrix, $\boldsymbol{u}$ is the vector of nodal displacements and $\boldsymbol{f}$ is the vector of nodal forces. Equations (1c) and (1d) are inequality constraints defining the stress and displacement constraints with $\bar{\sigma}_j$ and $\bar{u}_i$ being the allowable stresses and allowable displacements, respectively, and equation (1e) defines the search space for the minimum weight design where $\bar{A}^u$ and, $\bar{A}^l$ are the upper and lower bounds on the cross-sectional areas, respectively.

Member cross-sectional areas are either continuous or, selected from a catalogue of available sections and the lower bound is chosen to ensure that the kinematic stability of the structure is maintained during the optimization process. Solving for topology and sizing consists of minimizing the truss weight by determining the optimal member cross-sectional areas so that the stresses in the truss members do not exceed the allowable stress, the nodal displacements are less than the allowable displacement, and the external loads do not cause a loss of stability of the structure. In this study, topology optimization has been implemented within the sizing optimization problem by removing members having cross-sectional area equal to their lower bound in the final optimal configuration.

When Layout optimization is also considered, nodes' coordinates are added to the design space beside the cross-sectional areas. Sizing and layout optimization problems have been extensively studied and numerous methods have been used to solve this optimization problem. These include mathematical programming methods, Optimality Criteria (OC) methods, Heuristic techniques and Fully Stressed Design (FSD) and Fully Utilized methods (FUD).

1.3 Convexity Considerations in Truss Optimization problems

Convex optimization is the process of minimizing a convex objective function while satisfying a set of convex constraints. In truss weight optimization, the concept of convexity plays a vital role in determining the feasibility and efficiency of the optimization procedure. Convexity ensures that both the objective function and the constraints possess convex properties, which enables the utilization of efficient algorithms to find globally optimal truss designs. However, determining the convexity of a truss optimization problem can be complex and depends on various factors, such as the design variables, objective function, and constraints involved. Therefore, it is important to carefully consider the choice of structural analysis models and design constraints to ensure convexity, and to do so, one has to analyze the convexity conditions specific to each type of truss optimization problem to ascertain its feasibility and efficiency.

It is important to note that while expressing optimization problems as convex optimization problems offers significant advantages in terms of optimization efficiency and global optimality guarantees, it can be sometimes a challenging task due to the nature of the problem at hand. Ongoing advancements in convex optimization techniques and the exploration of nonconvex approaches continue to contribute to the advancement the field of optimization of truss structures.

1.4 Cellular automata for design optimization of truss structures

Cellular Automata (CA) are mathematical models that have been inspired by biological systems and are used to simulate complex systems based on local information and simple rules. These models are discrete, dynamic, and spatially extended, which means they operate on a set of discrete states and update their states over time based on their neighboring states. The use of CA models has been successfully implemented in various fields such as physics, biology, computer science, and social sciences to simulate and predict the behavior of complex systems [23].

In a typical CA model, the physical domain is divided into a lattice of uniform cells. A state vector that describes its local behavior and can represent various characteristics such as the cell energy, density, temperature, or any other variable relevant to the system being modeled characterizes each cell. The state of each cell in the domain is updated based on local transition rules that consider the states of the cell neighbors. These transition rules determine how the state of a cell changes in response to the state of its neighbors [24].

The two major features of conventional CA models are locality and homogeneity. Locality refers to the fact that behavior of each cell is determined by the states of its immediate neighbors, while homogeneity refers to the fact that all cells in the lattice have the same state of transition rules. These features make CA models highly efficient and computationally tractable, enabling the simulation of large-scale complex systems [25].

An example of a conventional uniform CA lattice structure, and the representation of the unit cell, is shown in Figure 1.

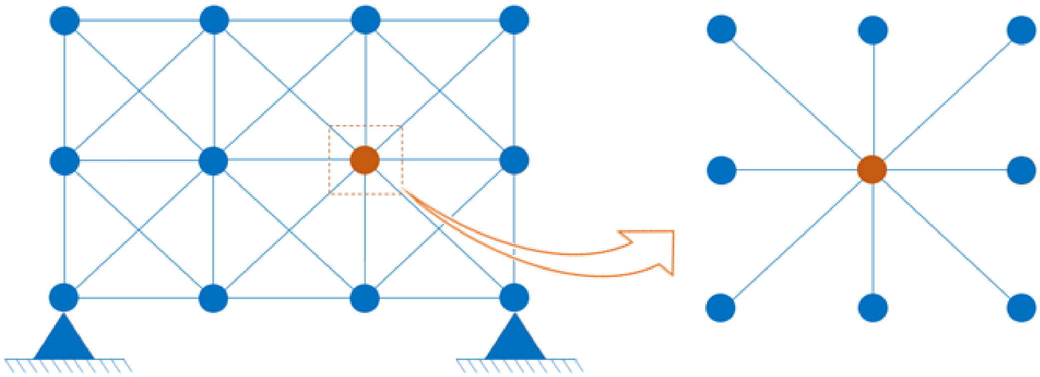


Figure 1. A uniform CA representation of a two-dimensional truss structure and the representation of a unit cell

Considering Figure 1, in a conventional uniform Cellular Automata (CA) model, the behavior of a unit cell is updated based on the information from its eight immediate neighboring nodes, also known as the "Moore neighborhood" [132]. However, this neighborhood system may not always be suitable for representing real truss structures. To address this limitation, a non-uniform CA approach has been proposed in this research study for the optimization of truss structures.

1.5 Research scope and objectives

In the past, Cellular Automata (CA) has been used successfully to optimize the topology and size of truss structures [117, 132]. However, the conventional CA algorithm is limited to uniform truss structures using identical cells and cannot handle displacement constraints, moreover they are not applicable to layout optimization problems. To overcome these limitations, this research study proposes a non-uniform CA algorithm that matches the boundary of the computational domain to that of the real structure, making it capable of solving topology, size, and layout optimization problems with stress, displacement, and local stability constraints.

The main objective of the proposed research is to develop a non-uniform CA algorithm for finding the minimum weight design of truss structures. The proposed algorithm does not require selective assignment of cell properties, as the computational domain's boundary is the same as that of the real structure. The algorithm also uses an improved version of the FSD/FUD at the cell level, avoiding the computation of all sensitivities of the design parameters at each design iteration. The analyses are embedded in the design optimization cycles. The proposed CA does not rely on a fixed grid network, and thus by incorporating nodal coordinates into the vector of design variables, the proposed approach has been extended to solve the layout optimization in conjunction with topology and size optimization using a combination the FSD approach and a strain energy criterion. The developed design optimization methodology has two phases. The first phase involves an improved version of the FSD/FUD at the cell level to find the minimum weight design of 2-D and 3-D elastic truss structures subjected to displacement and stress constraints. In the second phase, a bi-level non-uniform CA algorithm optimizes cross-sectional areas and node coordinates separately to achieve a minimum weight while satisfying stress and displacement constraints. An alternating procedure couples the two types of design variables during the optimization process until the optimum solution is achieved.

1.6 Organization of the Thesis

The thesis is structured into six chapters, each with its own purpose and focus. Chapter 1 serves as an introduction to the subject of structural optimization of truss structures, providing an overview of the various techniques that have been used to solve such problems and introducing the algorithm that has been implemented in this study.

Chapter 2 is dedicated to reviewing the literature relevant to the field of structural design optimization, discussing recent works related to the subject matter. In Chapter 3, proposed design optimization methodology using non-uniform CA paradigm and its implementation are presented.

Chapter 4 presents the work done in the first phase of this research, which was published in the article entitled "A Non-Uniform Cellular Automata Framework for Topology and Size Optimization of Truss Structures Subjected to Stress and Displacement Constraints", Computers and structures, Vol. 242, 2021. This chapter elaborates on the implementation of the proposed non-uniform cellular automata algorithm and its application in solving topology and size optimization problems for truss structures that are subjected to stress and displacement constraints.

Chapter 5 presents the research work presented in the second article entitled "Development of a Non-Uniform Cellular Automata Framework for Sizing, Topology and Layout Optimization of Truss Structures", currently under final review in the Journal of Engineering Optimization. This chapter discusses the development of a bi-level non-uniform cellular automata algorithm that simultaneously optimizes the sizing, topology, and layout of truss structures. The algorithm is demonstrated to effectively minimize the weight of truss structures, while satisfying stress, displacement, and local stability constraints.

Finally, Chapter 6 concludes the thesis by summarizing the key findings and contributions of the study. The chapter also offers recommendations for future research directions in the field of structural optimization of truss structures.

Chapter 2
A State-of-the-Art Review

2.1 Introduction

Modern structural optimization has a rich history that dates back to the 1960s when mathematical programming techniques were first used to solve basic structural design problems. Over time, new approximation techniques have been developed and optimization approaches have been refined to solve more complex structural design problems with minimal computational cost while achieving optimal results. Today, the field boasts an impressive body of research that covers a broad range of structural optimization problems, making it a practical tool in structural design.

One of the significant developments in the field of modern structural optimization has been the use of optimization methods for truss structures. Trusses are widely used in the construction of bridges, roofs, towers, and other structures, and optimizing their design has been a critical area of research. The history of the optimization of truss structures has been marked by continuous improvement in approximation techniques and optimization approaches, leading to practical applications in structural design. The classification of optimization methods for truss structures has provided a framework for solving complex design problems and achieving optimal results. The different methods for the optimization of truss structures can be classified into:

- Mathematical programming (numerical search) methods [28-44, 102,104].
- Optimality Criteria (OC) methods [45-55].
- Convex approximation methods [56-64].
- Metaheuristic techniques such as Genetic Algorithms (GA) [65-71, 102,104], Ant Colony Optimization (ACO) [72-74], Guided Stochastic Search (GSS) [75-77], Charged System Search (CSS) [78-80], Particle Swarm Optimization (PSO) [81-86], Simulated Annealing (SA) [87-89], Firefly Algorithm (FA) [90-92], Big Bang-Big Crunch optimization [93-95], Harmony search optimization [96-98] and differential evolution (DE) [99-101].

- Fully Stressed Design (FSD) and Fully Utilized Design methods (FUD) [108-114].

Each of these optimization techniques have some limitations, which other methods have attempted to alleviate. Patnaik et al. [52] provided a thorough comparison between some of these methods, analyzing their performance in terms of computational efficiency and design quality. The results show that some algorithms are more suitable for certain types of problems and that no single algorithm is universally superior.

The use of mathematical programming techniques in structural optimization started with Schmit's landmark paper in 1960. Schmit [28] studied a simple three-bar truss and showed that the minimum weight structure is not always the one in which each member is fully stressed. This contradicted the traditional assumption in the minimum weight design in which each member of the truss must be fully utilized in at least one load condition. This discovery led to extensive research in structural optimization. Dorn et al. [29] also used mathematical programming techniques in the field by applying a linear programming (LP) method to find the optimal layout of a truss structure. An assessment of the applications of mathematical programming methods to structural optimization during the decade from 1959 to 1969 was provided by Schmit [30].

In many structural design problems, the objective function and/or constraints are nonlinear in nature. Nonlinear programming (NLP) extends the capabilities of linear programming (LP) by allowing for the inclusion of nonlinear objective functions and constraints [31]. Pedersen [32] used a nonlinear programming method that employed gradients to solve the minimum weight design problem of plane truss structures.

Vanderplaats and Moses [33] employed the method of feasible directions to optimize truss designs. This method starts with an initial feasible solution and iteratively explores the design space by moving in directions that satisfy the constraints and potentially enhance the objective function. The algorithm selects the direction that yields the greatest improvement and repeats the process until an optimal solution is found. The method of feasible directions relies on the availability of the analytic gradient of the objective function and the constraint functions associated with active constraints.

Adeli and Kamal [34] proposed an algorithm for optimizing the minimum weight design of space trusses with fixed geometry. The algorithm formulates the problem as a nonlinear programming (NLP) problem based on the virtual work method. The objective function is linear and subject to linear size and stress constraints, while the displacement constraints are nonlinear.

Nonlinear Programming (NLP) problems can often be approximated as linear problems, allowing the use of Linear Programming (LP) methods for their solution. This approximation technique, called Sequential Linear Programming (SLP) [31, 35], involves linearizing the nonlinear objective function and constraints of an NLP problem at each iteration and thus, transforming the NLP problem into a series of linear subproblems, which can be optimized using efficient LP techniques.

Hansen and Vanderplaats [36] proposed a truss optimization method that uses Taylor series expansions to approximate member forces instead of stresses and displacements. Pedersen and Nielsen [37] later used a sequential linear programming (SLP) method to minimize weight of truss structures subjected to multiple load cases and constraints on displacement, stress, and frequency. References [38] and [39] highlight recent articles that investigate the utilization of Sequential Linear Programming (SLP) in truss design problems. To maintain the accuracy of the approximate analysis during the utilization of Sequential Linear Programming (SLP), it is necessary to set bounds, referred to as move limits, on the allowable magnitudes of design changes, thus constraining the extent of modifications that can be made while relying on the linear approximations.

Schmit and Farshi [40] introduced a cost-effective nonlinear approximation method called sequential approximate optimization, which combines the Fully Stressed Design (FSD) approach (explained later) for obtaining a gross material distribution and employs mathematical programming techniques for the detailed design.

In a recent study, Jie and Tabatabaei [41] explore the optimization of truss structures under uncertain loading using non-linear programming methods. Additionally, the same authors, along with Akbari [42], apply non-linear programming techniques for the optimization of truss structures with discrete variables. A comprehensive examination of diverse mathematical programming methods for structural optimization is provided in reference [43].

Mathematical programming techniques have the advantage of being general and can potentially be used to solve various structural optimization problems in a consistent manner. However, mathematical programming methods can be inefficient and converge to global optimum solutions with difficulty (they are generally trapped in local optimum point close to starting initial point), particularly for problems with a large number of design variables, and require the calculation of gradients of both objective and constraint functions which may be very difficult to obtain for most practical problems [44,69,70]. This limitation has led to efforts to not only looking for ways to improve the efficiency of these methods, but also to the development of the optimality criteria (OC) methods as an alternative approach to structural optimization. While optimality criteria methods emerged, mathematical programming techniques continued to improve and expand their use to a wider range of optimization problems such as layout optimization problems, leading to competition between the two methods. Unlike Mathematical Programming methods that primarily rely on information around the current point in the design space to find the search direction and how far to go in order to best reduce the weight of the structure directly. Optimality Criteria methods on the other hand, first establish specific conditions that characterize the optimal structure and subsequently, through an iterative procedure, adjust or modify the design to fulfill those conditions while indirectly optimizing the structure. These optimality criteria can be either intuitive like the Fully Stressed Design (FSD) approach, or derived from governing equations.

Since their introduction in the late 1960s, Optimality Criteria methods have become a fundamental approach employed by designers for structural optimization. Prager's pioneering work [45] marked the first application of Optimality Criteria (OC) methods in structural optimization. He introduced an optimality criterion based on strain energy distribution within the optimal structure, and showed that, under stress constraints, if the work of the applied loads is limited as an equality constraint, the resulting optimum structure exhibits a uniform energy density distribution. In the same year, Prager and Shield [46] introduced derivable optimality criteria methods that utilized variational approaches for optimizing specific continuum design problems. These methods presented optimality conditions expressed as differential equations, with their solutions describing the shape of the optimal structure.

Based on work of Prager et al. [45,46], Venkayya [47] presented an energy criteria method for designing structures subjected to static loading and proved the method to be extremely efficient at obtaining minimum weight structures and much faster compared to linear and nonlinear programming methods. Subsequently, many other studies have explored this efficient approach [48-50]. An assessment of the merits and limitations of the optimality criteria methods was provided by Patnaik [52]. The main drawback of OC methods is that they also require evaluation of gradients of the objective and constraint functions and maybe computationally inefficient even sometimes for modest small size problems [52]. Moreover, gradient-based methods are not capable of handling discrete design variables. Despite this limitation, OC methods continued to be popular in the field of structural optimization [53-55].

Convex approximation methods [56-64] is another class of gradient-based methods, which, despite some limitations, can offer significant advantages in terms of optimization efficiency and global optimality is generally guaranteed. Convex optimization refers to the general class of optimization problems where the objective function and constraints are convex. Linear programming is a specific form of convex optimization problem where both the objective function and constraints are linear. Numerous studies have been conducted on the convexity of truss weight optimization problems. Bendsoe and Sigmund's influential work [56] provided valuable insights into the convexity of such problems. They established conditions under which the topology optimization problem can be formulated as a convex optimization problem. They showed that, by assuming linear elastic material behavior and incorporating volume constraints, the optimization problem can be formulated as a convex problem. This allows for the use of efficient algorithms to find the global optimum. However, certain nonlinearities in the constraints or objective function can result in non-convex problems, requiring specialized optimization techniques. While Bendsoe and Sigmund [56] primarily focused on topology optimization for structures, their principles are applicable to other methods of truss optimization. For sizing and shape optimization problems, the conditions for convexity can vary depending on the specific formulation. Nonlinear behaviors, additional constraints or objectives, and discrete design variables can all contribute to the non-convexity of the optimization problem. Kaveh and Ilchi Ghazaan [57] investigated the impact of design variables, objective functions, and constraints (e.g., stress, displacement, and buckling constraints) on the convexity of truss sizing optimization problems. They provided mathematical formulations and conditions for determining when a truss sizing problem can be treated as a convex program. A comprehensive review of the convexity of structural optimization problems, including truss sizing optimization can be found in [58]. The authors discussed the fundamental concepts of convexity, convex optimization techniques, and various conditions for convexity. They analyzed different types of optimization problems, including single-objective and multi-objective problems, and explored their convexity properties. The review also covers convexity analysis for various structural systems, including truss structures, emphasizing the importance of convexity in improving the efficiency and reliability of optimization algorithms. It is important to note that most real-world truss structures may exhibit nonlinear behaviors, such as yielding, buckling, or other complex phenomena.

In such cases, the optimization problem may become non-convex, and specialized techniques are typically employed to handle non-convexities in truss optimization problems. They aim to enable efficient and effective approximation and find the global optimum solution of the problems, either by linearizing constraints, constructing convex subproblems, or using quadratic approximations. One of the most popular techniques in handling non-convex optimization problems is the Convex Linearization method, commonly known as CONLIN [59]. Introduced by Fleury in 1989, CONLIN is specifically designed for problems characterized by nonlinear expressions in both the objective function and constraints. The key idea behind CONLIN is to divide the feasible region into multiple convex subdomains and approximate the nonlinear functions within each subdomain using linear functions. By performing convex approximations locally, the overall optimization problem can be formulated as a linear programming problem or a convex quadratic programming problem, which can be solved efficiently. CONLIN provides a trade-off between computational efficiency and accuracy, as the linear approximations may introduce some degree of error compared to the original nonlinear functions. However, it can be a valuable tool when dealing with complex optimization problems involving nonlinearities.

The Method of Moving Asymptotes (MMA) [60] is an optimization technique that also addresses non-convex optimization problems by employing separable and convex approximations. MMA utilizes an iterative approach to update a set of "asymptotes" in order to approximate the original non-convex problem. It constructs a series of convex subproblems that progressively approximate the underlying problem. By updating the asymptotes and solving these subproblems, the algorithm directs the search towards the global optimum. MMA is particularly effective in solving topology optimization problems. While it may be slower than OC methods for simpler compliance optimization problems, it demonstrates excellent convergence properties for more complex problems involving multiple constraints [56]. Both MMA and CONLIN rely on the utilization of separable and convex approximations, which constitute a fundamental aspect of these techniques. The separability property implies that the necessary optimality conditions of the subproblems do not interrelate or couple with each other. This characteristic has a profound impact on minimizing the computational effort required to solve the subproblems, particularly when dealing with optimization problems that involve only a small number of constraints [56].

Zhang and Domaszewski [61] proposed an approximation method called DQA-GMMA to solve non-convex truss optimization problems. it is a combination of the Diagonal Quadratic Approximation (DQA), which is a technique commonly used in the context of nonlinear programming, and the Generalized Method of the Moving Asymptotes (GMMA). The convexity and the separable form of this approximation makes it effective in handling sizing and shape variables concurrently and finding the globally optimum truss design.

Non-convex optimization problems can sometimes be transformed into convex optimization problems through the introduction of supplementary variables and constraints. Some examples of the application of convex relaxation techniques in truss optimization can be found in [62-64].

It is important to note that the while convexity offers significant advantages in terms of optimization efficiency and finding the global optimum, it may impose limitations on the design space exploration. Some non-convex truss weight optimization problems with complex design constraints or nonlinear behaviors may require the use of more advanced optimization techniques, such as random or stochastic based search algorithms, also called heuristic algorithms [65-101], which have proven to be effective in addressing various optimization problems, especially highly nonlinear structural optimization problems. These methods require only function evaluations (no gradient requirements) and generally converge to near global optimum solution. However, random search algorithms often have a slow convergence rate and require a high number of objective function evaluations without guaranteeing convergence towards global optima, especially for problems with many design variables [102-104]. This can make them less efficient than gradient-based methods for certain optimization problems. Therefore, researchers have proposed hybrid algorithms that combine gradient and stochastic optimization techniques to improve search accuracy and efficiency to catch the global optimum solution.

Zuo et al. [105] proposed a hybrid optimization approach combining the Optimality Criteria (OC) and Genetic Algorithm (GA) methods to solve truss design optimization problems subjected to frequency constraints. The proposed approach (OC-GA) aims to improve the efficiency and accuracy of the optimization process by combining the advantages of both methods. The approach was able to catch the global optimum solution more efficiently compared with traditional optimization methods.

Kaveh and Javadi [106] introduced a hybrid algorithm (HRPSO) to optimize the shape and size of trusses with multiple frequency constraints using a combination of Harmony Search (HS) and Ray Optimizer (RO) algorithms to enhance the performance of the Particle Swarm Optimization (PSO) algorithm. The also showed that their hybridized approach outperforms the others in terms of convergence speed and accuracy.

Another hybrid algorithm used for optimizing truss layout has been proposed by Gholizadeh [107]. This method utilized the strengths of both the Particle Swarm Optimization (PSO) algorithm and the Cellular Automata (CA) approach to improve the efficiency and effectiveness of the optimization process. Overall, the combination of multiple optimization techniques and their integration through has showed to improve the optimization process, leading to better designs with improved performance.

The last category of optimization techniques, which have been effectively utilized in structural optimization, are the FSD (Fully Stress Design) and FUD (Fully Utilized Design) methods [108-114]. The FSD method is effective in solving truss optimization problems under stress constraints; however, this methodology cannot handle structural optimization problems under displacement constraints. This limitation was actually the main reason behind the development of the NLP (Nonlinear Programming) and OC (Optimal Control) methods in the 1960's. The limitation of FSD to address optimization problems under displacement constraints has also led to the development FUD approach, which is a simple method that scales the design obtained from FSD uniformly to satisfy the most infeasible displacement constraint. While FUD can tackle structural optimization problems under both stress and displacement constraints, they have the drawback of producing overdesigned solutions.

The shortcomings of aforementioned methods, have initiated the development of other optimization techniques. In recent decades, the exploration of artificial life simulation algorithms for optimization problems has been a focus of researchers. These algorithms, inspired by natural processes such as evolution, swarm behavior, and foraging, aim to understand and emulate these phenomena. Among these techniques, we find the commonly used metaheuristic techniques like Genetic Algorithms (GA) [65-71], Particle Swarm Optimization (PSO) [81-86], Ant Colony Optimization (ACO) [72-74], and Artificial Bee Colony (ABC) [103], which have been extensively studied and successfully applied in solving truss optimization problems. Another noteworthy approach is Cellular Automata (CA) algorithms. CA algorithms have attracted attention due to their effectiveness in solving not only structural optimization problems but also a wide range of other optimization problems. While the metaheuristic, artificial life simulation algorithms focus on optimization and search-based approaches, Cellular Automata (CA) algorithms possess unique characteristics that set them apart described below:

- Local Interactions: Cellular automata algorithms are based on the concept of local interactions, where the behavior of an individual cell is determined by its neighboring cells. Each cell updates its state based on predefined rules that consider the states of nearby cells. This local interaction allows for the emergence of complex global behavior from simple local rules.
- Discrete Space and Time: Cellular automata operate on a discrete grid-like space, where each cell can be in a finite number of states. Time in cellular automata is also discretized into discrete steps. At each time step, the state of each cell in the CA grid is updated based on the rules of the CA algorithm.
- Emergent Behavior: One of the key strengths of cellular automata algorithms is their ability to generate emergent behavior. Simple local rules can lead to the emergence of complex, global patterns or phenomena that were not explicitly programmed. This property makes cellular automata well suited for studying self-organization and emergent properties in complex systems.

- Simplicity and Parallelism: Cellular automata algorithms are often conceptually simple and computationally efficient. The parallel nature of cellular automata allows for efficient simulations on parallel architectures or distributed computing systems.
- Versatility: Cellular automata algorithms can model a wide range of systems and phenomena. They have been used to simulate physical processes, ecological systems, social dynamics, and various other domains. The flexibility and adaptability of cellular automata make them suitable for exploring and understanding complex systems in different fields.

2.2 Cellular automata for the optimization of truss structures

Hajela [115] was one of the pioneers who first used a cellular automata approach in structural design optimization problems. He incorporated CA update rules in the design process for the solution of two-dimensional elasticity problems. It was concluded that the CA and artificial life simulations techniques in general have significant potential for optimizing the design of complex structures and thus can become a valuable tool for engineers and designers; however, more research is needed to fully understand their capabilities and limitations.

Kita and Toyoda [116] also used the cellular automata paradigm for solving topology optimization problems for discrete and continuum structures. They first discussed the limitations of traditional structural design methods, which often rely on simple assumptions and may not fully capture the behavior and mechanics of complex systems. They then introduced the concept of CA and describe how they can be used to model the behavior of structural systems. They used CA rules for updating the topology and generating new designs while the analysis of the structure was performed using the finite element method. They provided several examples of the application of CA in structural design, such as optimizing the shape of a truss structure and minimizing the weight of a plate structure while maintaining the structural integrity.

Tatting and Gürdal [117] were pioneers in using cellular automata (CA) for simultaneous analysis and design (SAND) to solve topology optimization problems for truss structures that exhibit linear and nonlinear responses, as well as buckling and plastic deformation. SAND approach involves simultaneously optimizing the design of a structure while analyzing its performance. They further extended their research to the design of two-dimensional continuum elastic structures, such as plates and shells, by modeling the properties of the elastic medium with an equivalent truss-cell [118]. The apparent thickness of the continuum cell was determined by computing the equivalent cross-sectional area of the bars in the truss-cell, which maintained the equivalence of strain energy between the continuous and truss structures. CA was successfully used to optimize the shape of a plate subjected to uniform and non-uniform loadings, as well as to minimize the weight of a shell structure.

Another application of the CA paradigm with a SAND approach was presented by Abdalla and Gürdal [119] for the solution of eigenvalue problems, which can involve complex systems with nonlinear behavior. They provided several examples such as the design optimization of a column under buckling constraints and identifying critical members in a truss structure with nonlinear behavior. Canyurt and Hajela [120] introduced the concept of Cellular Computation Models (CCM) and describe how they can be used in the design of structural systems. They applied a hybrid approach combining Cellular Automata (CA) and Cellular Genetic Algorithm (CGA) for optimizing truss structures using the SAND approach. Faramarzi and Afshar [121] also utilized CA in a hybrid approach with Linear Programming (LP) to solve truss optimization problems with displacement constraints. Tajs and Bochenek [122] used CA to solve the column buckling design problem by locally optimizing the topology of the structure based on compliance minimization, rather than globally maximizing the buckling load. They showed how the CA could be used to design columns with optimal topologies that maximize the buckling load and concluded by discussing potential future applications of their approach in the design of other types of structures.

Cellular Automata (CA) techniques are a type of optimization method that use local rules to guide the optimization process. They have the advantage of being simple to implement and inherently parallel, but they also have some limitations. One disadvantage of CA techniques is that they only consider a subset of the design space. Another limitation is that a uniform CA requires an identical cell structure, which means that the computational domain has to be rectangular. This poses challenges when designing truss structures with irregular boundaries, as cells outside the domain must be set to zero values for member cross-sectional areas, displacements, and nodal forces in order to match the shape of the actual domain. Thus, conventional CA cannot be applied for lay out structural design optimizations in which the nodal coordinates are design variables rendering non-uniform cells. Sipper [123] discusses the concept of co-evolution in non-uniform cellular automata (CA) for performing computations, where each cell may contain different rule. They proposed a new method for solving complex problems by using a fitness function to evaluate the performance of the CA and then using a genetic algorithm to evolve the rule set of the CA.

The article by Vichniac et al. [124] introduced the concept of inhomogeneous cellular automata (INCA). INCA is a type of CA where the cells have different update rules based on their location within the grid. Two types of INCA were proposed: annealed and quenched. Annealed INCA has a random distribution of update rules, while quenched INCA has a fixed distribution of update rules. They demonstrated the effectiveness of INCA by using it to simulate the behavior of a binary fluid. The results show that INCA can accurately model the behavior of complex systems and is more efficient than traditional methods of simulation.

2.3 Layout optimization of truss structures

The goal of layout optimization in truss structures is mainly to determine the best arrangement of structural nodes to minimize the weight of structures while satisfying the required constraints. When it is performed in conjunction with sizing optimization, and due to the inherent coupling between sizing and layout variables, the most efficient way to achieve the optimal design is through one-level approaches which involve solving for both sizing and layout simultaneously. This comes with a disadvantage though, since it requires combining variables of different types into a single large design vector. Because of the large size of the search space, this technique is usually numerically unstable, lengthy and has a slow rate of convergence [125]. An alternative solution to this issue is the use of bi-level methods, which consist of dividing joint coordinates and cross-sectional areas of members into two sets of design variables and then applying a different algorithm for each. All the categories of optimization techniques that have been presented previously, i.e. Mathematical programming methods, Convex approximation methods, Fully Stressed Design (FSD) approaches, and metaheuristic techniques, have been used in either single or dual-level schemes to solve the general problem of topology, sizing and layout optimization.

Dorn et al. [29] was the first to use a linear programming (LP) method in order to find the optimal layout from a given ground-structure composed of permissible joints and bars. They successfully identified the optimal configuration by eliminating members with zero cross-sectional areas in the final solution.

Pedersen [32] discusses the use of a nonlinear programming method to solve the minimum weight design problem of plane truss structures. He specifically focused on the layout of the truss structure and how it can be optimized to meet multiple load cases using a nonlinear programming (LP) method employing the gradients of the objective and constraint functions to find the optimal design. The optimization problem was treated with and without consideration of stability and displacement constraints. Pedersen and Nielsen [37] later used a sequential linear programming (SLP) optimization approach for weight minimization of general 3D truss structures subjected to multiple load cases and involving displacement, stress and eigen-frequency constraints, and with the design variables being simultaneously the cross-sectional area and the positions of the joints.

Hansen and Vanderplaats [36] developed an optimization technique that uses an approximation approach to reduce the computational effort required to find the optimal configuration of a truss. They reduced the degree of coupling between sizing and layout variables by using Taylor series expansions to approximate member forces instead of stresses and displacements and showed that the proposed approach provides solutions that are very close to the optimal solutions.

A thorough review of mathematical programming techniques used in the layout optimization of skeletal structures has been provided by Topping [44]. Applications of convex approximation methods to solve the layout optimization problem can also be found in [57-59]. However, as mentioned previously, these gradient-based methods do not have the mechanism to capture the true global optimum solution [64].

Gil and Andreu [109] proposed a single level technique to solve the optimization problem of plane truss structures involving constraints on stresses, displacements, and stability. Their technique combines the fully stressed design (FSD) approach with a conjugate gradient method to overcome the difficulties arising from merging sizing and layout design variables. The authors demonstrated the efficiency of their optimization approach through a case study of a truss structure used in a bridge and were able to improve the performance of the truss structure while satisfying all the constraints.

In their work, Vanderplaats and Moses [110] developed a bi-level optimization method for optimum design of truss structures. The method involves splitting the design space into two distinct sets of variables: sizing variables, which are optimized using the FSD approach, and layout variables, which are optimized using the constrained steepest descent method. These two sets of variables are dependent on each other, but are optimized separately. To ensure that the optimization process is effective, an alternating procedure is employed to couple the two types of design variables. This means that the sizing and layout variables are updated iteratively, with each iteration improving the overall design until an optimal solution is achieved. It was shown that this bi-level approach could significantly improve the performance of truss structures while satisfying various constraints.

Wang el al. [111] used a similar approach to Vanderplaats and Moses [110] in their work, which involves an alternating procedure for optimizing the sizing and layout of truss structures subjected to constraints on stress, buckling, and displacement. The Fully Stressed Design (FSD) approach was initially used to optimize the sizing variables, while an evolutionary node shifting method was applied to manipulate layout variables based on sensitivity analyses. While the approach proved effective in yielding accurate optimal results when the initial design is near the optimal solution, it comes at a high computational cost. Flager et al. [126] used a new bi-level hierarchical method to optimize the layout and member sizing of both determinate and indeterminate truss structures. The upper-level of the proposed hybrid optimization algorithm focuses on finding the optimum joint positions using a gradient-based optimization method that operates on continuous layout variables while the lower level uses a Fully Constrained Design (FCD) method to find members cross-sectional areas.

Bi-level optimization methods are useful because they can separate the sizing and layout optimization problems, which can substantially reduce the complexity of the overall optimization process. However, it is important to note that these methods may not always result in the global optimum design because the problems are not linearly separable. This means that the coupling between the sizing and layout variables can affect the overall optimization process, making it difficult to guarantee a global optimum design. Metaheuristic algorithms have also been used to optimize truss structures in both single-level and bi-level optimization schemes. These algorithms are designed to search for optimal solutions by exploring a large number of potential design solutions and evaluating their fitness and can be used to optimize for topology, sizing, and layout to minimize the weight of the structure. Deb and Gulati [65], for instance, proposed the use of genetic algorithms (GA) for simultaneous optimization of topology, sizing, and layout of truss structures to minimize their weight under stress, displacement, and kinematic stability constraints. They used a fixed-length vector of design variables representing member cross-sectional areas and joint coordinates. They demonstrated the effectiveness of genetic algorithms for the optimization of truss structures by solving several truss structures with different load cases and constraints, and showed that the GA approach was able to find near-global optimal solutions for these problems.

Wu and Chow [68] also utilized Genetic Algorithm (GA) in an integrated optimization approach that combines the discrete sizing variables mixed with continuous layout joint variables. The objective function of the optimization problem was to minimize the weight of the truss structures subjected to stress and displacement constraints and the approach was shown to be effective in finding optimal solutions with a low computational cost. Rahami et al. [66] also utilized a genetic algorithm (GA) in combination with energy and force methods to simultaneously optimize the topology, sizing, and layout of truss structures to minimize their weight. The approach included formulating the optimization problem in terms of energy concepts and introducing a new methodology to reduce the number of input variables, which enhanced the efficiency of the GA algorithm in the structural optimization process.

Fourie and Groenwold [82] proposed the use of a particle swarm optimization (PSO) algorithm for size and layout optimization of truss structures. The PSO algorithm was found to be efficient and effective for size and layout optimization of truss structures and the optimization results suggest that the PSO algorithm outperformed GA in terms of solution quality, and that it is comparable to the gradient-based recursive quadratic programming algorithm in terms of convergence rate. Kaveh et al. [83] employed a Hybrid PSO and Swallow Swarm Optimization (SSO) algorithm to solve for layout and sizing in the weight minimization problem of truss structures involving multiple natural frequency constraints.

Swallow Swarm Optimization (SSO) is another metaheuristic optimization algorithm inspired by the behavior of swallows in nature. The algorithm is based on the observation that swallows fly in a coordinated manner, adjusting their flight paths to avoid obstacles and optimize their navigation. The algorithm was applied to several truss structures, and the results show that it is effective in finding optimal solutions that satisfy the dynamic constraints. Mortazavi et al. [84] employed an improved version of PSO called integrated particle swarm optimizer (iPSO) for simultaneous sizing, layout, and topology optimization of truss structures. The approach combines particle swarm optimization (PSO) and finite element analysis to optimize the truss structures for minimum weight while satisfying the design constraints. The PSO algorithm is used to search the design space, and the finite element analysis is used to evaluate the fitness of the solutions. The iPSO method was applied to optimize the weight of various truss structures under different loading conditions and showed to be effective in finding optimal solutions that satisfy the design constraints. However, for some problems, the iPSO method showed to have a slow convergence rate compared to other metaheuristic techniques.

Ahrari et al. [114] proposed a novel approach for the simultaneous optimization of topology, sizing, and layout of truss structures. The approach combines the Fully Stressed Design (FSD) method with an Evolution Strategy (ES) technique, resulting in a hybrid optimization algorithm called FSD-ES. The results of their study showed that FSD-ES was highly efficient and competitive when compared to other optimization techniques and led to optimal designs with reduced weight while satisfying all of design constraints. Luh and Lin [72] proposed a bi-level ant colony optimization (ACO) algorithm to simultaneously optimize the topology, sizing, and layout of 2-D and 3-D truss structures subjected to stress, deflection, and kinematic stability constraints. The problem was formulated as a mixed-integer nonlinear programming (MINLP) problem, and the ACO algorithm was adapted to handle discrete variables and nonlinear constraints. The performance of the ACO algorithm was compared with other optimization techniques, such as the Genetic Algorithm (GA) and the Sequential Quadratic Programming (SQP) method. The results showed that the two-stage ACO is capable to find a better solution in terms of weight and convergence speed. In addition, a sensitivity analysis was conducted to investigate the effect of various parameters on the optimization results. The results showed that the performance of the ACO algorithm was sensitive to the parameter settings, and selecting the appropriate parameters was crucial for achieving accurate optimal results.

Gholizadeh [85] proposed a new hybrid optimization algorithm called Sequential Cellular Particle Swarm Optimization (SCPSO) for layout optimization of truss structures. The algorithm combines the Cellular Automata (CA) and Particle Swarm Optimization (PSO) algorithms to optimize the layout of truss structures. The CA algorithm is used to create an initial population of solutions, and the PSO algorithm is used to refine these solutions to find the optimal layout of truss structures. The SCPSO algorithm was tested on several benchmark problems and was able to find better optimum solutions at higher convergence rates compared with other optimization methods. The study demonstrated the effectiveness of the SCPSO algorithm for truss structure optimization and highlighted the potential of combining different optimization techniques for better results.

Other metaheuristic techniques that have also been used to solve the sizing and layout optimization problem include among others: The C*harged System Search* (CSS) [78], the *Ray Optimization* (RO) *algorithm* [127,128] and the recently developed Jaya algorithm (JAYA) [129,130].

2.4 Conclusion

In this chapter, a state-of-the-art review on optimization methods for truss structures has been provided. The concept of cellular automata is explained, highlighting its potential for achieving efficient and robust truss designs. The application of cellular automata-based techniques and their advantages in optimization of truss structures have been discussed. Furthermore, the chapter explores layout optimization methods for truss structures including various approaches and algorithms used to determine the optimal layout configuration of truss elements. The significance of layout optimization in enhancing structural performance and minimizing material usage is emphasized. In summary, Chapter 2 highlights the potential of cellular automata and layout optimization methods, providing a solid foundation for the subsequent chapters of the thesis.

Chapter 3
Design Optimization Methodology Based on Non-Uniform Cellular Automata Paradigm

3.1. Introduction

In this chapter first, a general description of the design optimization methodology using non-uniform cellular automata approach together with brief discussion on analysis and design update rules are provided. Finally, the computer implementation of the proposed non-uniform CA methodology for both analysis and design optimization of truss structures is discussed.

3.2 Non-uniform Cellular Automata for Design Optimization of Truss Structures

As discussed in Chapte 1, in conventional uniform Cellular Automata (CA) model, the behavior of a unit cell in 2D-Truss strcutures is updated based on the information from its eight immediate neighboring nodes (also known as the "Moore neighborhood"). However, this neighborhood system may not always be suitable for representing real truss structures as all the cells must have exactly the same number of neighboring nodes and connecting members. Thus, the missing nodes have to be added to those cells located on the boundary. Moreover the uniform CA cannot be used for lay optimization in which node coordinates are considered as design variables. To address this limitation, a non-uniform CA approach has been proposed in this research study for the optimization of truss structures

The non-uniform CA approach is based on non-identical cells, each of which is defined by a center node and members connecting the center node of the cell to all other nodes in its immediate neighborhood. This differs from the Moore neighborhood concept used in the uniform CA approach, which only considers the eight immediate neighboring nodes. This non-uniform CA approach provides a more realistic representation of truss structures and improves the optimization process.

Figure 2 illustrates the concept of the proposed non-uniform Cellular Automata (CA) model for a two-dimensional truss structure. Each structural node is connected to other nodes, and a representative unit cell is shown on the boundary. The number of cell centers in the non-uniform CA model is equal to the number of nodes in the truss structure. In other words, there is a one-to-one relationship between nodes and cells. Thus 2D truss shown in Figure 12 contains 12 cells reprsented by each node. The non-homogeneity feature of the model, as opposed to the conventional homogenous CA model shown in Figure 2, permits exploration of a larger design space by allowing additional bars to be added to the initial design. Moreover, in contrast to uniform CA, the proposed non-uniform CA model also allows to solve layout optimization by adding the positions of the nodes to the set of design variables along with the cross-sectional areas of the members. This permits for a more comprehensive approach to truss optimization, as the position of the nodes is a critical factor in determining the structure's overall stability and strength.

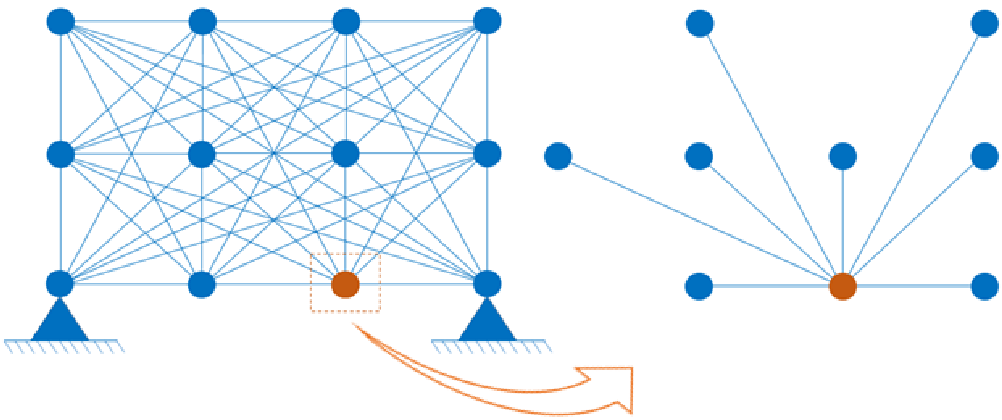


Figure 2. A non-uniform CA representation of a two-dimensional truss structure and the representation of a non-uniform unit cell

The proposed non-uniform CA approach for structural optimization departs from conventional CA algorithms in that the cells are not necessarily identical and each cell has a number of members equal to the number of surrounding nodes, as opposed to a fixed number in a conventional uniform CA. The local behavior of each lattice cell is defined by a state vector that consists of field and design variables. The state vector of each cell is updated, with the analysis rules considering the structural behavior and the design rules optimizing the structural configuration by changing the cross-sectional areas of the members or the positions of the nodes.

In Cellular Automata (CA), the local behavior of each lattice cell is described by a state vector containing field and design variables. The field variables of a truss structure include nodal forces and nodal displacements, while the design variables are the cross-sectional areas of the cell members for the size and topology optimization and the cross-sectional areas together with nodal coordinates for simultaneous size, topology and layout optimization. For the sake of simplicity, the state vector for a cell in a two-dimensional truss structure on a plane may be mathematically represented by:

$$S_i = \left(\{u_i, v_i\}, \{F_{xi}, F_{yi}\}, \{A_1, A_2, \dots, A_N\}, \{x_i, y_i\}\right) \tag{2}$$

where S_i represents the collection of field and design variables within a particular cell i. The field variables refer to the nodal displacements, $\{u_i, v_i\}$, and nodal forces, $\{F_{xi}, F_{yi}\}$ in the x and y directions at the center node, while the design variables represent the cross-sectional areas $\{A_1, A_2, \dots, A_N\}$ of the N members in the cell i, and $\{x_i, y_i\}$ as the coordinates of the cell's center node.

As mentioned above, the calculation of the new state of each cell in the lattice in Cellular Automata (CA) optimization of truss structures is achieved through update rules that take into account the states of neighboring cells. These update rules usually rely on mathematical models that establish a relationship between the stiffness and stress of the members in a truss structure and their corresponding forces and displacements [26]. Moreover, the update rules may integrate extra constraints to guarantee that the updated state of each cell conforms to the design rules and physical principles that govern the behavior of the truss structure.

In this research study, both analysis and design have been concurrently conducted using the developed non-uniform CA algorithm incorporating analysis and design updating rules. Updating rules are usually performed using either the Jacobi iterative method or Gauss-Seidel iterative method [132]. Jacobi iterative method is an explicit approach in which all new values of state variables are computed from the old ones. This means that the updated values are not used to calculate other new values. On the other hand, the Gauss-Seidel iterative method is an implicit technique, which uses updated values to compute new ones. In other words, the algorithm iteratively computes new values of the unknown variables by using the most recently updated values for the other unknowns. The Gauss-Seidel iterative method is known for its fast convergence and it is widely used in solving linear equations in numerical analysis and scientific computing. In this study, the Gauss-Seidel iterative method was selected in conjunction with the CA algorithm for efficient updating of state variables in structural optimization problems.

3.3 Updating Rules

3.3.1 Analysis Update

The analysis update rules are implemented following each design solution to ensure that the equilibrium is met at the cellular level. For instance for 2D truss structures in Figure 2, the field variables $\{u_i, v_i\}$ and $\{F_{xi}, F_{yi}\}$ are obtained by minimizing the potential energy of the cell. The analysis rule can be either linear or nonlinear, depending on how the strains are calculated [117]. This research is mainly focused on linear analysis. The elongation of a truss member *j* in a representative non-uniform cell shown in Figure 3 can be expressed as follows:

$$\delta_j = (u_j - u_i)\cos\theta_j + (v_j - v_i)\sin\theta_j \tag{3}$$

in which direction cosines can be obtained as:

$$\cos\theta_j = \frac{x_j - x_i}{L_j} \quad \text{and} \quad \sin\theta_j = \frac{y_j - y_i}{L_j} \tag{4}$$

where (u_i, v_i) and (x_i, y_i) denote the nodal displacement and nodal coordinate vector of the i^{th} cell center node, respectively. Similarly, (u_j, v_j) and (x_j, y_j) denote the displacement and coordinates of the far end node of the j^{th} member ($j = 1,\ldots, N$). It should be noted that the j^{th} member has the first node *i* and end node *j*, defining the local axis from node *i* to node *j* as illustrated in Figure 3. The orientation angle of the member with respect to its unreformed configuration is denoted by θ_j, and N represents the number of member elements within the cell.

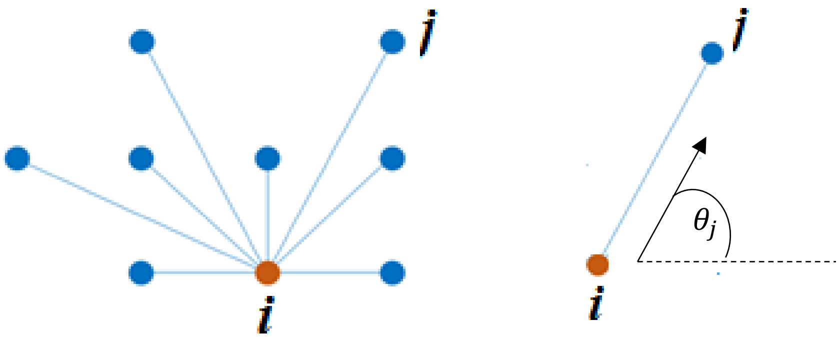


Figure 3. A non-uniform cell with center node i, and a typical j[th] member in the cell with end node j

The strain ε_j, stress σ_j and force, F_j, in each member j of the cell can then be written as:

$$\varepsilon_j = \frac{\delta_j}{L_j} \tag{5}$$

$$\sigma_j = E_j \varepsilon_j \tag{6}$$

$$F_j = \sigma_j A_j \tag{7}$$

where E_j is the material modulus of elasticity for the member j. The potential energy of a cell i can now be expressed as:

$$V_i = \sum_{j=1}^{N} \frac{E_j A_j L_j {\varepsilon_j}^2}{2} - F_{xi} u_i - F_{yi} v_i \tag{8}$$

where F_{xi} and F_{yi} denote the external loads applied at the center node of the cell i. To obtain the governing equilibrium equations at the cellular level, we substitute equation (5) for strain and equation (3) for elongation into equation (8), and then minimize the resulting potential energy expression. The resulting equations can be expressed as follows:

$$\frac{\partial V_i}{\partial u_i} = 0, \quad \frac{\partial V_i}{\partial v_i} = 0 \tag{9}$$

Differentiating equation (8) with respect to the nodal displacements at node i gives:

$$\begin{cases} \frac{\partial}{\partial u_i}\left(\sum_{j=1}^{N} \frac{E_j A_j L_j {\varepsilon_j}^2}{2} - F_{xi} u_i - F_{yi} v_i\right) = 0 \\ \frac{\partial}{\partial v_i}\left(\sum_{j=1}^{N} \frac{E_j A_j L_j {\varepsilon_j}^2}{2} - F_{xi} u_i - F_{yi} v_i\right) = 0 \end{cases} \tag{10}$$

Now, substituting for strain $\varepsilon_j = \frac{\delta_j}{L_j}$ into Eq. (10) yields:

$$\begin{cases} \frac{\partial}{\partial u_i}\left(\sum_{j=1}^{N} \frac{E_j A_j {\delta_j}^2}{2L_j} - F_{xi} u_i - F_{yi} v_i\right) = 0 \\ \frac{\partial}{\partial v_i}\left(\sum_{j=1}^{N} \frac{E_j A_j {\delta_j}^2}{2L_j} - F_{xi} u_i - F_{yi} v_i\right) = 0 \end{cases} \tag{11}$$

which becomes:

$$\begin{cases} \sum_{j=1}^{N} \frac{E_j A_j \delta_j}{L_j}\left(\frac{\partial \delta_j}{\partial u_i}\right) - F_{xi} = 0 \\ \sum_{j=1}^{N} \frac{E_j A_j \delta_j}{L_j}\left(\frac{\partial \delta_j}{\partial v_i}\right) - F_{yi} = 0 \end{cases} \tag{12}$$

Finally considering $\delta_j = (u_j - u_i)\cos\theta_j + (v_j - v_i)\sin\theta_j$, after differentiation, the Eq. (12), representing equilibrium equations for the cell i may be expressed as:

$$\begin{gathered} \sum_{j=1}^{N} \frac{E_j A_j}{L_j}\cos^2\theta_j\, u_i + \sum_{j=1}^{N} \frac{E_j A_j}{L_j}\sin\theta_j \cos\theta_j\, v_i = F_{xi} + \sum_{j=1}^{N} \frac{E_j A_j}{L_j}\left(\cos^2\theta_j\, u_j + \sin\theta_j\cos\theta_j\, v_j\right) \\ \sum_{j=1}^{N} \frac{E_j A_j}{L_j}\sin\theta_j \cos\theta_j\, u_i + \sum_{j=1}^{N} \frac{E_j A_j}{L_j}\sin^2\theta_j\, v_i = F_{yi} + \sum_{j=1}^{N} \frac{E_j A_j}{L_j}\left(\sin\theta_j\cos\theta_j\, u_j + \sin^2\theta_j\, v_j\right) \end{gathered} \tag{13}$$

During the optimization process, at each iteration of the analysis loop, the cell displacements are evaluated based on the updated values of the displacements of the neighboring cells using the Gauss-Seidel iterative method.

3.3.2. Design Update

The rules for updating a design depend heavily on the specific constraints of the problem. When dealing with only stress-constrained problems, the Fully Stressed Design (FSD) approach [112, 113] is often utilized to determine the updating rule for the cross-sectional area. This rule involves adjusting the cross-sectional area based on the level of stress experienced in the structure, ensuring that the material is being efficiently used while maintaining the required level of structural strength. The FSD rule can be written as:

$$ {A_j}^{t+1} = {A_j}^{t} \frac{|{\sigma_j}^t|}{\bar{\sigma}_j} \tag{14}$$

For sizing and topology optimization, which involve both stress and displacement constraints, FSD cannot be used as it, will generally yield a design with violated displacement constraints. In this research study, an effective cellular design update rule has been developed in such a way that only the influence of the contributing cell members to the displacement of a given cell with a violated displacement is considered. This proposed design update rule for the cross-sectional areas, will be discussed in detail in chapter 4. The proposed design update rules for the case of Layout, Sizing and topology optimization, which involve, in addition to cross-sectional areas, the coordinates of the nodes will also be presented in chapter 5.

3.4 Computer Implelemntation

A computer code has been developed in Fortran using the object-oriented paradigm [131] to solve truss optimization problems based on the proposed analysis and design optimization strategy based on non-uniform cellular automata. Object-oriented programming (OOP) can improve the efficiency of a cellular automata (CA) algorithm by providing a structured and modular approach to the implementation of the algorithm.

One of the main benefits of OOP is the ability to encapsulate data and methods within objects. This allows for the separation of the implementation of the CA algorithm from the data used by the algorithm, making it easier to understand, maintain, and modify the code. OOP also allows for the creation of reusable classes, which can be used to implement different types of CA algorithms. This can improve the overall efficiency of the implementation, as well as make it easier to test and debug the code. Additionally, OOP allows for the use of inheritance and polymorphism, which can be used to create a hierarchy of classes that share common properties and methods. This can make the implementation of the algorithm more efficient and easier to understand. Finally, OOP allows for the implementation of parallelism and concurrency through the use of thread-safe classes and methods. This can improve the performance of the algorithm when parallelized, and makes it easier to implement parallelism and concurrency in the algorithm, thus significantly increasing the efficiency of the design optimization algorithm.

The proposed non-uniform CA algorithm benefits from the framework provided by Object-Oriented Programming (OOP). A "cell class" is created within the OOP system to define the properties and behavior of a cell object. Such properties include the number of cell members and neighboring nodes, as well as an object array encompassing the cell members, and another object array containing the cell neighboring nodes. Additionally, each cell is assigned a "state object" to describe its evolution throughout the optimization process. To ensure the code is reusable, reliable, and universal in solving truss optimization problems, class libraries are employed to provide well-defined and pretested reusable components. The resulting object-oriented CA-based optimization algorithm is represented in a simplified UML diagram in Figure 4, where classes, attributes, methods, and relationships are depicted to illustrate its structure.

In this diagram, we have four main classes: Node, Element, Cell and State. Each class has its own set of attributes, which are shown as the variables listed under the class name. The "Node" class has the attributes of the node's Id, node's coordinates, and node's elements which is an object array encompassing all the bars meeting at the node. The "Element" class has the attributes of the element's type, element's Id, both nodes of that element, and other properties of the element. The "Cell" class has the attributes of the cell type, cell center node, the object array containing of all the bars in the cell, number of elements in the cell, the object array containing all the neighboring nodes of the cell, the number of the neighboring nodes, and the attribute "state" which links the "Cell" class to the "State" class which for 2D truss for instance has the attributes of displacements $\{u_i, v_i\}$, external nodal forces $\{F_{xi}, F_{yi}\}$, cross-sectional areas of the members within the cell $\{A_1, A_2, \ldots, A_N\}$, and the coordinates of the cell's center node. In addition to the attributes, the classes also have methods, which are shown as functions listed under the class name. However, in this diagram, the methods are not included for the sake of simplicity. The lines between the classes show the relationships between them. In this diagram, there is a relationship between "Cell" and "State", where to each "Cell" is associated a "State". This relationship is shown with an arrow pointing to the "State" class side. In conclusion, the class diagram above provides a visual representation of the structure of the CA optimization program and the relationships between its classes.

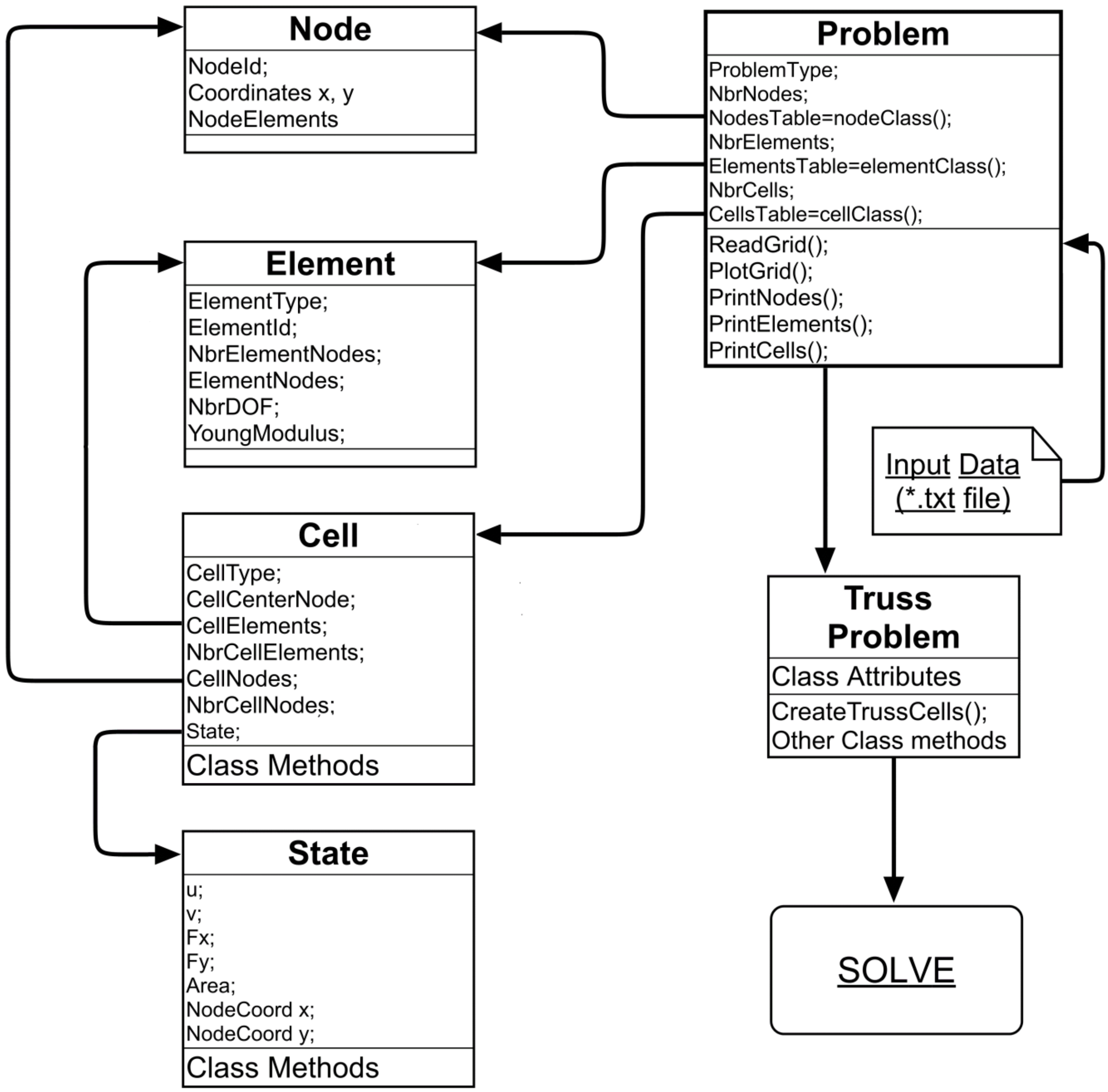


Figure 4. A simplified UML diagram of the object-oriented CA Algorithm

3.5 Conclusion

Chapter 3 presents a design optimization methodology based on the non-uniform cellular automata paradigm for truss structures. The main focus of the chapter is on the utilization of non-uniform cellular automata for the design optimization of truss structures. The concept of non-uniform cellular automata is explained, emphasizing its potential for achieving efficient and effective truss designs. The section provides insights into the application of non-uniform cellular automata-based techniques and their advantages in optimizing truss structures.

The chapter further delves into the updating rules employed in the methodology. It discusses the analysis update, which involves updating the structural analysis information within the cellular automata framework. Additionally, the design update is explored, addressing the process of updating the design variables based on the information obtained from the cellular automata analysis. Furthermore, the importance of the Object-Oriented approach in implementing the proposed non-uniform CA algorithm is discussed.

In conclusion, Chapter 3 presents a comprehensive methodology for design optimization of truss structures based on the non-uniform cellular automata paradigm. The chapter discusses the theoretical foundations of the methodology, including the non-uniform cellular automata concept, updating rules, and computer implementation considerations.

Chapter 4
A Non-Uniform Cellular Automata Framework for Topology and Size Optimization of Truss Structures Subjected to Stress and Displacement Constraints

4.1 Introduction

This chapter presents a novel approach for Sizing and Topology optimization using a non-uniform Cellular Automata method that is based on non-identical cells and a modified FSD/FUD approach. The proposed algorithm is implemented using Fortran programming language and the Object-Oriented paradigm, and it aims to minimize the weight of truss structures under both stress and displacement constraints. To demonstrate its effectiveness and accuracy, the proposed methodology will be applied to various benchmark 2D and 3-D truss design problems.

4.2 Problem Formulation

The main objective of the design optimization is to find the best way to distribute material within a structure by solving topology and size optimization problems. The ultimate aim is to minimize the overall weight of the structure while ensuring that a set of stress and displacement constraints are met. The optimization problem can be mathematically formulated as:

$$\textit{minimize} \quad W(P) = \sum_{j=1}^{m} \rho_j A_j L_j \tag{15a}$$

$$\textit{Subject to} \quad Ku = f \tag{15b}$$

$$|\,\sigma_j| < \bar{\sigma}_j\,, \qquad j = 1,2,\cdots,m \tag{15c}$$

$$|u_i| < \bar{u}_i\,, \qquad i = 1,2,\cdots,k \tag{15d}$$

$$\bar{A}^l \;<\; A_j < \bar{A}^u\,, \qquad j = 1,2,\cdots,m \tag{15e}$$

Equation (15a) represents the objective function of the weight minimization problem where ρ_j , A_j , L_j and m denote the density, the cross-sectional areas, L_j and the total number of bar members in the structure. The constraints include the stiffness equation (15b) which must be satisfied to maintain the equilibrium condition, stress constraints (15c), displacement constraints (15d), and side constraints (15e). In these equations, $\boldsymbol{K}$ represents the stiffness matrix, $\boldsymbol{u}$ represents the vector of nodal displacements, $\boldsymbol{f}$ represents the vector of nodal forces, and A_j represents the design variables, which are the cross-sectional area of the truss members. The variables $\bar{\sigma}_j$ and $\bar{u}_i$ represent allowable stresses and allowable displacements, respectively, while $\bar{A}^u$ and, $\bar{A}^l$ are the upper and lower bounds on the cross-sectional areas, respectively. The variable k indicates the number of displacement constraints. The lower bound of the cross-sectional area is set to a small positive value to ensure that the structure maintains kinematic stability during the optimization process. Additionally, topology optimization is integrated into the size optimization problem by eliminating members with a cross-sectional area equal to their lower bound in the final optimal configuration.

4.3 Design Update

As mentioned in Chapter 3, the rules for updating the design depend on the specific constraints of the problem. The updating rule for the cross-sectional area based on FSD can be expressed in Eq. (14) which is again stated below for the sake of clarity as:

$$A_j{}^{t+1} = A_j{}^{t} \frac{|\sigma_j{}^{t}|}{\bar{\sigma}_j} \tag{16}$$

When a unit cell comprises collinear bars as depicted in Figure 5, the bar with the lower internal force value is removed during the FSD process. This is because the bar with the higher force value offers a more natural pathway for the force to flow through compared with other collinear bars.

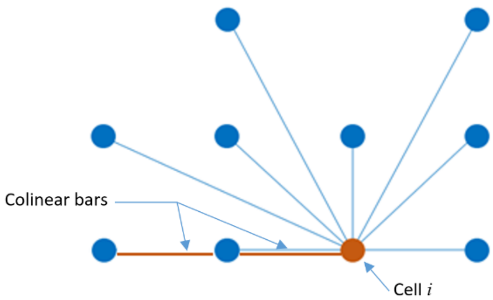


Figure 5. A non-uniform CA cell containing collinear elements

Although the Fully Stressed Design (FSD) is an efficient method for solving truss optimization problems with stress constraints, it cannot handle problems that involve displacement constraints. In such cases, the Fully Utilized Design (FUD) [52] approach is commonly used. The FUD approach is based on a simple concept, which consists of scaling the FSD solution uniformly to satisfy the most infeasible displacement constraint. Although simple, the FUD approach has the limitation of usually producing overdesign solutions. Patnaik et al. [52] presented an improved version of the FUD approach called the Modified Fully Utilized Design approach (MFUD). Despite being more efficient than FUD, MFUD can still generate overdesign solutions. Makris and Panagiotis [133] proposed a more efficient energy-based method to handle displacement-constrained truss optimization problems. This method involves applying a virtual unit load at the cell where the displacement constraint is violated, calculating the virtual strain energy in each member, and determining the contribution of each member to the nodal displacement based on its virtual strain energy density. The virtual strain energy in each member is calculated by multiplying the actual displacement in the member by the internal force induced due to the virtual unit load. The contribution of each member in the structure to the nodal displacement is proportional to the virtual strain energy density (virtual strain energy per unit volume) of the member U_j normalized by the mean value $\overline{U}$, which is averaged over all members of the structure. For a truss structure, U_j can be described by the following relation:

$$U_j = \frac{1}{A_j L_j}(\delta P_j \times \delta_j) = \frac{1}{A_j L_j}\left(\delta P_j \times \frac{F_j L_j}{E_j A_j}\right) = \frac{F_j\, \delta P_j}{E_j {A_j}^2} \tag{17}$$

where δP_j is the axial force in the j member of the cell due to the virtual unit force in direction of the desired contained displacement and F_j represents the force in the same member due to the real load.

This study tackles displacement-constrained truss optimization problems by utilizing a similar approach, which involves determining the contribution of each member to the nodal displacement of a specific cell. The first step in this approach is to obtain the FSD solution. When a truss member is fully stressed, the maximum strain energy is stored within that member, which allows us to eliminate its contribution to the displacement of a given cell by fixing it at its maximum stress level. As a result, there is no need to calculate the contribution of that member to the displacement of a cell with a violated displacement constraint, as is conducted in the conventional FUD method. Instead, we can use a cellular design update rule that only considers the influence of the contributing cell members to the displacement of a given cell with a violated displacement constraint. Based on this idea, the proposed design update rule for the cross-sectional area of the j^{th} member within the i^{th} cell can be expressed using the following equation:

$$A_{j,i}{}^{t+1} = A_{j,i}{}^{t} \left(1 + \eta_j \frac{|u_k| - \bar{u}_k}{\bar{u}_k}\right) \tag{18}$$

where u_k is the violated displacement associated with the cell k exceeding the allowable limit $\bar{u}_k$. η_j is a scaling factor, which is given by:

$$\eta_j = \begin{cases} 1 & \text{for } FUD \\ 0 & \text{for } FSD \end{cases} \tag{19}$$

In summary, the proposed approach consists of two phases. The first phase involves the application of the FSD method without any consideration of the displacement constraints. In the second phase, the correction for active displacement constraints is performed by altering the cross-sectional areas of only those bars that contribute to the respective active displacement constraint while keeping the non-contributing members at their maximum stress level. The overall procedure can be summarized in the following steps:

1- Obtain the FSD solution for the entire structure.
2- Sort the violated displacement constraints in descending order. For the sake of simplicity, let us assume that there are violated displacement constraints associated with cells k and k+1, with $\frac{|u_k| - \bar{u}_k}{\bar{u}_k} > \frac{|u_{k+1}| - \bar{u}_{k+1}}{\bar{u}_{k+1}}$
3- Determine the uniform proration factor based on the most violated displacement constraint as $1 + \frac{|u_k| - \bar{u}_k}{\bar{u}_k}$.
4- Loop through the cells of the structure and for each cell; uniformly scale ($\eta_j = 1$) using the proration factor obtained in step 3. If the scaling of the members of the cell i does not change the value of u_k, then, these members should remain unaltered, i.e., kept at their FSD condition with maximum stress level, without any scaling ($\eta_j = 0$). Otherwise, change the cross-sectional area of the contributing members using the uniform proration factor.
5- Repeat step 4 until the displacement constraint is satisfied within a very small tolerance value ϵ, set at 10^{-6}, i.e., $\frac{|u_k|}{\bar{u}_k} - 1 < \epsilon$.

6- Apply steps 2-5 for the remaining violated displacement constraints until all of them are satisfied.

From Equation (19), it is clear that to consider the impact of members on the displacement of the active cell k, uniform scaling of those members is required. Alternatively, this contribution could be removed by keeping the total strain energy within the i^{th} cell constant, which can be done by maintaining those members at their maximum stress level.

4.4. Computer Implementation of the Proposed Analysis and Design Optimization Strategy

The optimization algorithm for topology and sizing optimization of truss structures is based on a two-level approach where the design loop is nested inside the analysis loop. The first level is the analysis loop, which aims to determine the nodal displacements of the structure given the external loads and a given set of cross-sectional areas.

The algorithm begins with a set of initial nodal displacement values and iterates through all the cells of the structure, using the equilibrium relations in Eq. (13) to find new values of the displacements for each cell. Once all nodal displacement values have converged, the design loop calculates new values for the design variables Aj using Eq. (18) for the given displacement field.

The proposed non-uniform CA algorithm for topology and sizing optimization of a truss structure subject to stress and displacement constraints is shown in a simplified flowchart in Figure 6.

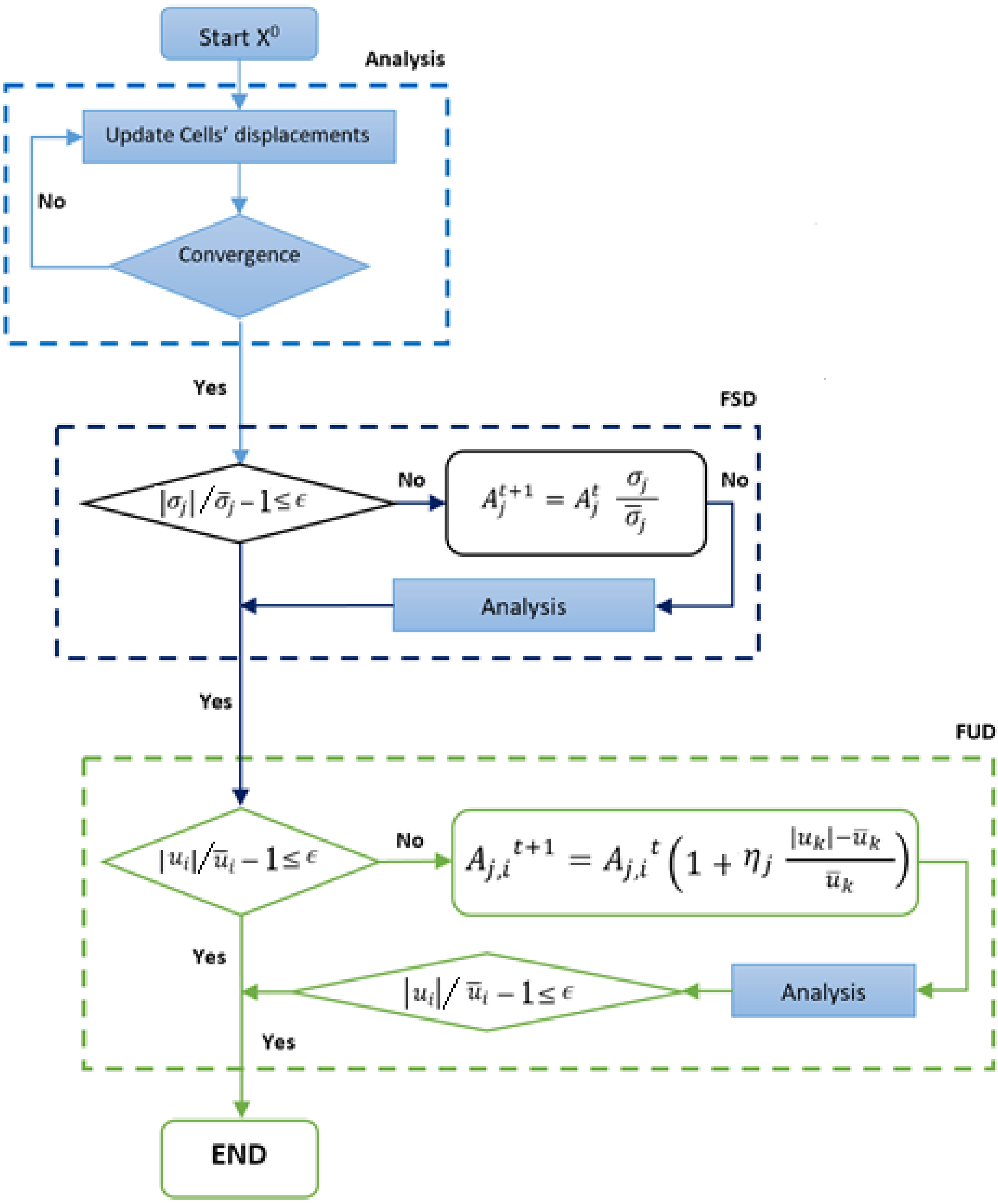


Figure 6. Flowchart of the non-uniform CA algorithm

4.5. Benchmark Numerical Examples

The proposed non-uniform CA approach has been tested on various benchmark problems. The results are presented and compared with those obtained using other optimization techniques published in literature. The main goal in all of the optimization problems discussed in the following sections is to find the optimal size and topology configuration to minimize the weight of truss structures, while ensuring that the stress and displacement constraints are met. It is noted that topology optimization is implemented within the size optimization problem, where members whose cross-sectional areas hit the lower bounds (close to zero) are removed.

The proposed algorithm guarantees equilibrium at each optimization iteration, and allows for a lower bound on the cross-sectional area to be set extremely low, resulting in negligible contribution of members whose cross-sectional areas hit the lower bounds to the final stiffness of the structure. In all following case studies, structures with optimized topology (after removing the members with lower bound values) have been analyzed using the optimal cross-sectional areas provided in tables to assure the stability and also satisfaction of displacement and stress constraints. Furthermore, re-optimization of the final topology layouts to evaluate optimal cross-sectional areas was conducted, and the refined cross-sectional areas were found to be very close to the initial values, indicating that the removal of members with lower bound values had a negligible effect on the final optimum solutions.

4.5.1. Problem 1: 11-Member, 6-Node Truss Structure

The proposed CA-based analysis and optimization approach was used to solve a truss structure with 11 members and 6 nodes, as shown in Figure 7. The aim was to minimize the weight of the structure while adhering to stress and displacement constraints. Table 1 provides the material properties and problem parameters. The material density and modulus of elasticity were set at ρ = 0.1 lb/in3 and E = 104 ksi, respectively, for all members. Two external loads of 100,000 lb each were applied at the two end nodes, as depicted in Figure 7. The stress in all members should not exceed the specified allowable value of ±25 ksi, and the vertical displacements at the two end nodes were limited to a maximum of 2 inches. The value for all the components of the initial design vector X_o is set at $\bar{A}^u = 35 \text{ in}^2$.

Table 1. Problem parameters-11 member, 6-node truss structure

Young's modulus (E)	10^4 ksi
Density (ρ)	0.1 lb/in
Allowable stress ($\bar{\sigma}$)	±25 ksi
Allowable displacement ($\bar{u}$)	2.00 in
$\bar{A}^u$	35 in^2
$\bar{A}^l$	10^{-10} in^2

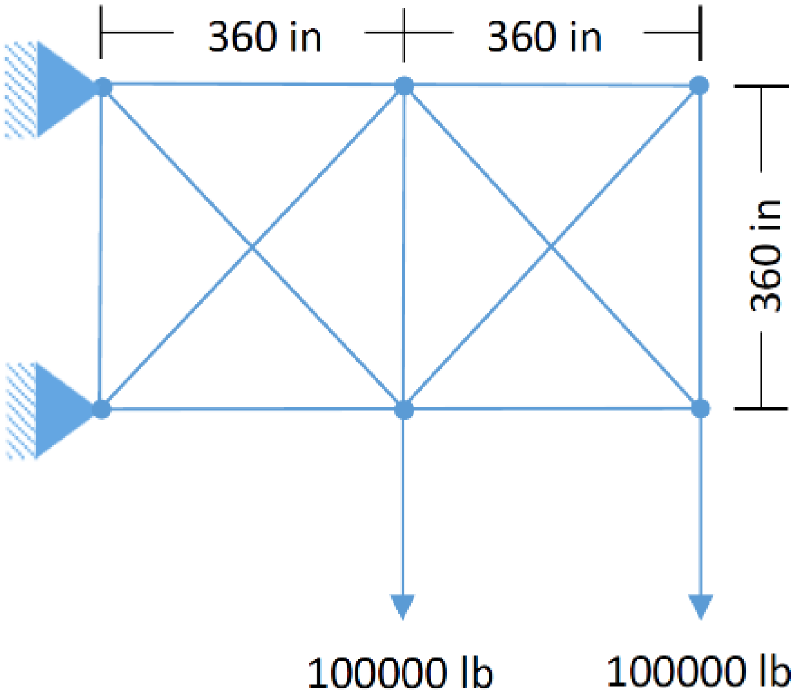


Figure 7. 11-member, 6-node truss structure

Various optimization methods have been used by researchers to solve this benchmark optimization problem. For instance, Deb and Gulati [65] used a Genetic Algorithm (GA) and achieved an optimal configuration shown in Figure 8 with a total weight of 4899.15 lb. Their optimal solution required 49500 objective function evaluations. Luh and Lin [72] utilized an ant colony optimization (ACO) algorithm and achieved the same truss topology with a total weight of 4899.11 lb, but after 41000 objective function evaluations. Faramarzi and Afshar [121] solved this problem using a hybrid CA and LP algorithm and obtained the same topology with an optimal weight of 4898.31 lb, with only 240 objective function evaluations. Our proposed methodology, based on a non-uniform CA, resulted in the same optimal topology configuration but with a slightly lower total weight of 4898.22 lb and only 169 objective function evaluations.

Table 2 compares the optimal size cross-sectional areas obtained using the proposed method with those reported in [65], [72], and [121]. The vertical displacements at the constrained nodes (under loads) in the optimal configuration were found to be active (2.00 in). Additionally, Table 3 provides the stress distribution in the optimal truss configuration. As it can be observed, the stress magnitude in all elements is lower than the allowable value of 25 ksi, indicating that there are no active stress constraints.

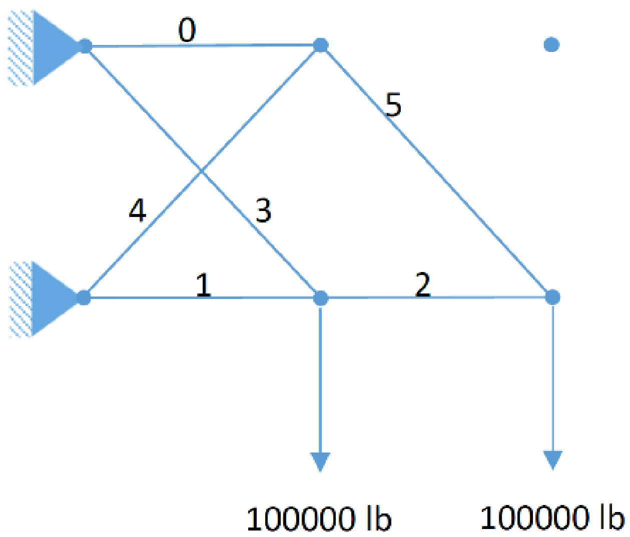


Figure 8. Optimized topology for the 11-member Truss structure

Table 2. Optimal cross-sectional areas (in^2) for the 11-member Truss structure

Member no.	Deb and Gulati [65]	Luh and Lin [72]	Faramarzi and Afshar [121]	Proposed Method
0	29.6800	29.8100	30.0953	30.0600
1	22.0700	22.2400	22.1321	22.263
2	15.3000	15.1500	15.0476	15.028
3	6.0900	6.0800	6.0802	6.0732
4	21.4400	21.3900	21.2806	21.256
5	21.2900	21.2400	21.2806	21.257
Weight (lb)	4899.15	4899.11	4898.31	4898.22
# of Objective function Evaluations	49500	41000	240	180

Table 3. Member stresses in the optimal configuration for the 11-member truss

Member	Stress (psi)
0	6653.4
1	-8983.4
2	-6654.1
3	23286.0
4	-6653.4
5	6653.1

Interestingly, the effectiveness of the proposed algorithm was found to be unaffected by the choice of initial design vector, as demonstrated in convergence history shown in

Figure 9. Despite starting from widely separated initial points, with all components of the design vector set at 35 in², 1 in², and 10 in², the algorithm was able to converge to the same optimal solution.

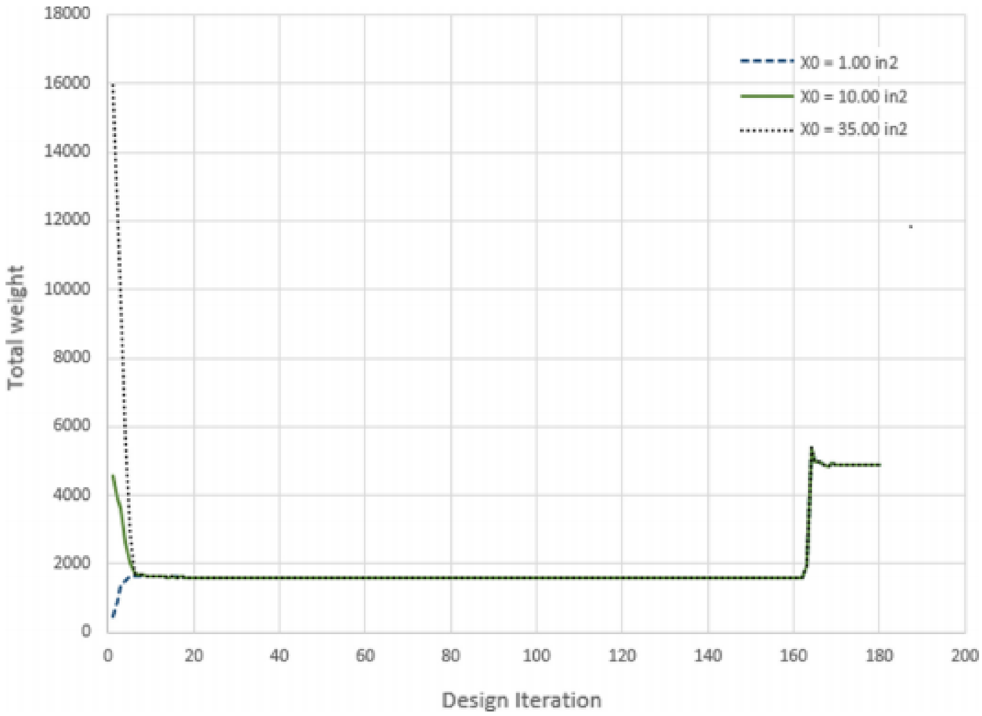


Figure 9. Convergence history for the 11-member, 6-node truss structure

4.5.2. Problem 2: 15-Member, 6-Node Truss Structure

This benchmark problem is similar to the Problem 1 with the exception that the initial design is a fully connected truss structure, which is illustrated in Figure 10. The material properties and problem parameters are similar to those outlined in Table 1. It is noted that the value for all the components of the initial design vector is also set at $\bar{A}^u = 35 \text{ in}^2$.

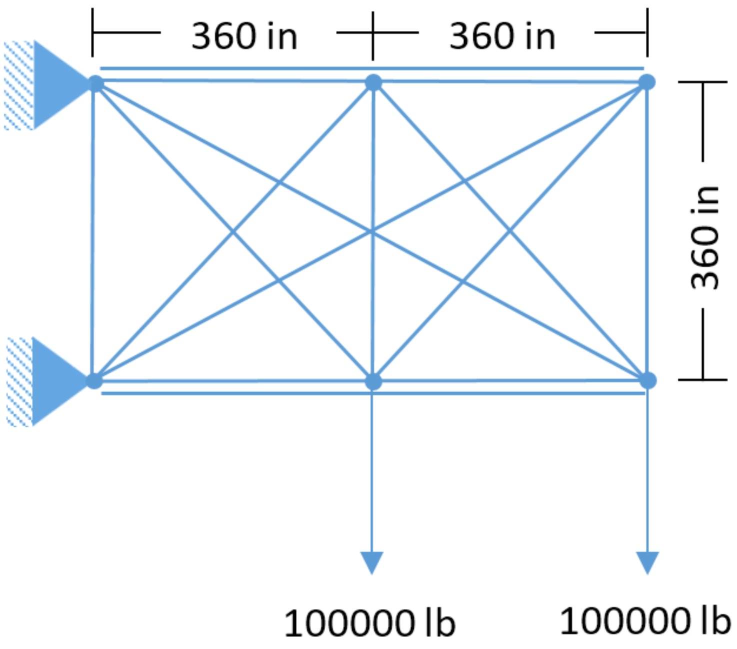


Figure 10. 15-member, 6-node truss structure

The proposed algorithm generated an optimal topology configuration as depicted in Figure 11. The optimized topology found using the proposed non-uniform CA approach is similar to those found using other methods. Deb and Gulati [65] s utilized a Genetic Algorithm with a population size of 450 and achieved an optimal weight of 4731.65 lb after 85050 objective function evaluations. Luh and Lin [72] used an ACO algorithm and obtained an optimal weight of 4730.824 lb with 41000 objective function evaluations. Faramarzi and Afshar [121] employed a hybrid CA and LP algorithm and reported an optimal structure with a total weight of 4730.42 lb after 310 evaluations. Using the proposed method, the total optimal weight is found to be 4730.40 lb, which is slightly lower than those previously reported, and it required only 267 objective function evaluations. Table 4 compares the optimal size cross-sectional areas obtained using the proposed method with those obtained using other optimization techniques. As in the previous problem, the vertical displacements at the constrained nodes (under loads) in the optimal configuration were found to be active (2.00 in), and the stress magnitude in all members is lower than the allowable value of 25 ksi, with no active stress constraints. Table 5 provides the stresses in the optimal truss configuration for the 15-member truss.

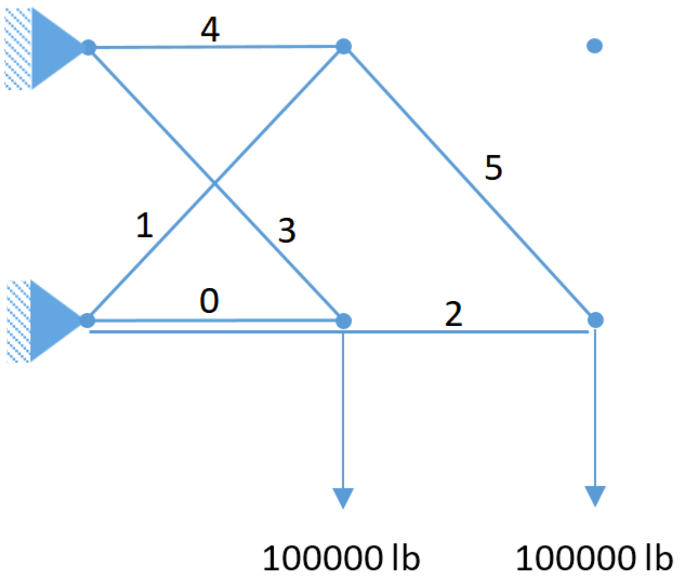


Figure 11. Optimized topology for the 15-member truss structure

Table 4. Comparison of optimal cross-sections (in^2) for the 15-member truss

Member no.	Deb and Gulati [65]	Luh and Lin [72]	Faramarzi and Afshar [121]	Proposed Method
0	5.219	5.428	5.4000	5.4000
1	20.310	20.549	20.3647	20.365
2	14.593	14.308	14.4000	14.400
3	7.772	7.617	7.6367	7.6368
4	28.187	28.876	28.8001	28.800
5	20.650	20.265	20.3647	20.365
Weight (lb)	4731.650	4730.824	4730.4237	4730.400
# of Objective function Evaluations	85050	41000	310	267

Table 5. Member stresses in the optimal configuration for the 15-member truss

Member no.	Stress (Psi)
0	-18519.00
1	-6944.40
2	-6944.40
3	18519.00
4	6944.40
5	6944.40

The convergence history for different initial designs is shown in Figure 12 to demonstrate the insensitivity of the proposed algorithm with respect to the selection of the initial design point. As it can be realized, while there are slight differences in the number of structural analysis, optimized results are similar starting from different initial points. For instance, setting all components of the initial design vector to lower value of 1 in^2 results in a computational cost of 276 structural analyses, practically the same as the 267 analyses required for the initial uniform design of 35 in^2.

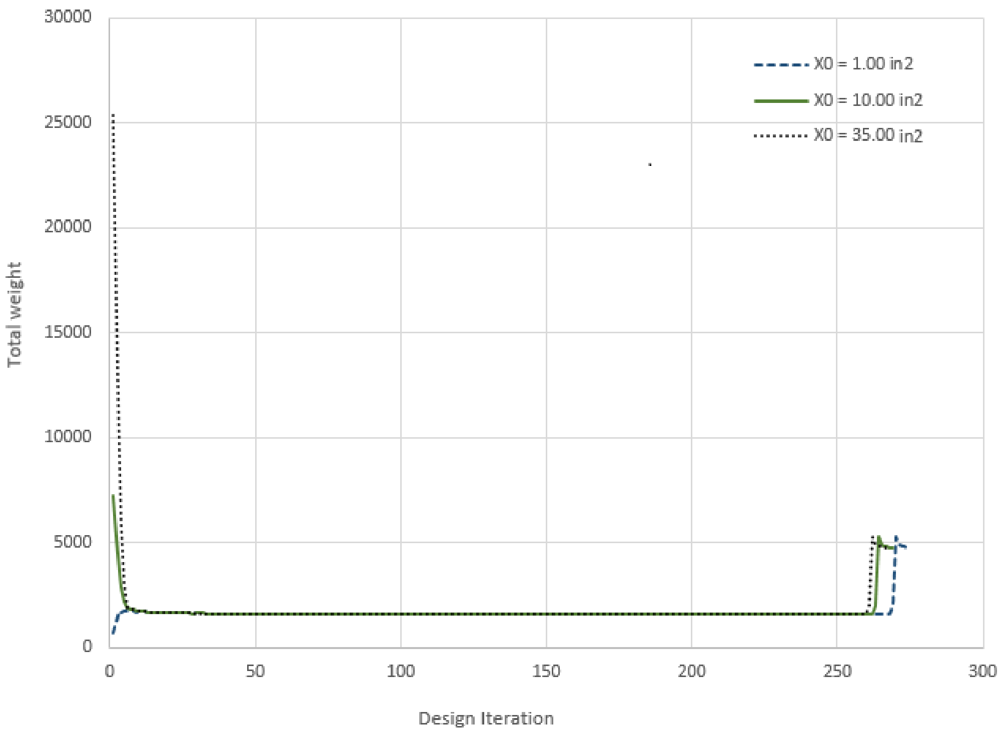


Figure 12. Convergence history for the 15-member, 6-node truss structure.

4.5.3. Problem 3: 45-Member, 10-Node Truss Structure

The proposed technique has also been employed to solve the 45-member, 10-node truss structure presented shown in Figure 13. The material properties and problem parameters are comparable to those of the 11-member truss example outlined in Table 1 (Problem 1), except that the lower and upper bounds of the cross-sectional area are set to 10^{-6} in^2 and 1.0 in^2, respectively. In addition, the stress in each member must not exceed 25 ksi, and the vertical displacement at the nodes of external loads is limited to a maximum of 2 inches. The value for the components of the initial design vector is also set at $\bar{A}^u = 1 \text{ in}^2$. This problem is also a well-known benchmark problem that has been solved using various optimization techniques.

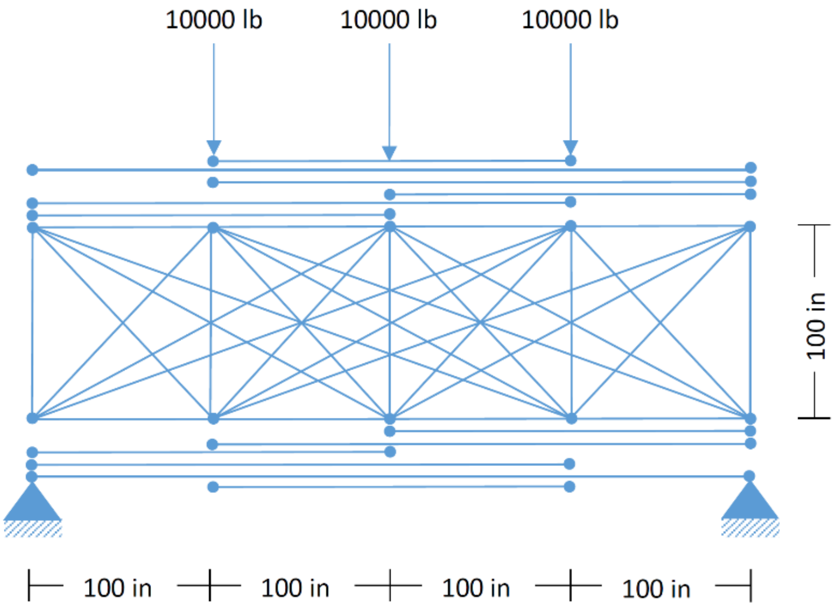


Figure 13. 45-member, 10-node ground structure

The optimal topology configuration obtained using the proposed method is presented in Figure 14. The topology is comparable to the one obtained by Deb and Gulati [65] using GA and the one obtained by Faramarzi and Afshar [121] using a hybrid CA and LP method. Deb and Gulati [65] reported a total weight of 44.033 lb, while Faramarzi and Afshar [121] achieved a slightly smaller weight value of 44.000 lb with 840 objective function evaluations. The optimal configuration obtained using the proposed method has the same overall weight of 44.000 lb as reported in [121], but only required 169 objective function evaluations. The stress in the optimal truss configuration is found to be uniformly distributed in each member, reaching its allowable value of

25 ksi. Additionally, all displacement constraints are satisfied, with vertical displacement under loads found to be -1.2500 in at the middle node and -0.7500 in at the two other nodes. Table 6 presents the optimal cross-sectional areas of the members for the various optimum design algorithms.

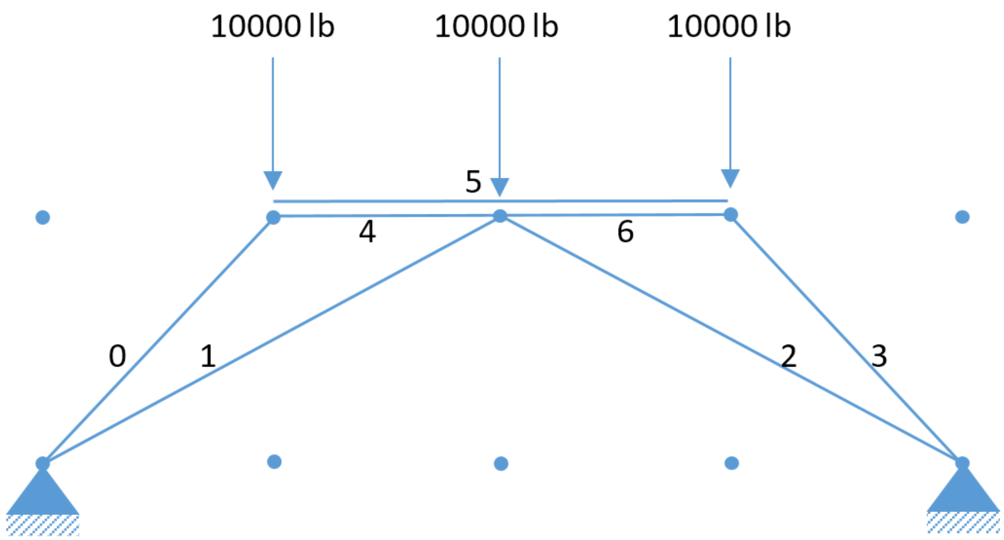


Figure 14. Optimized topology for the 45-member truss structure

Table 6. Comparison of optimal cross-sections (in^2) for the 45-member truss

Member no.	Deb and Gulati [65]	Faramarzi and Afshar [121]	Proposed Method
0	0.566	0.5656	0.5656
1	0.477	0.4472	0.4472
2	0.477	0.4472	0.4472
3	0.566	0.5656	0.5656
4	0.082	0.4000	0.3999
5	0.321	-	-
6	0.080	0.4000	0.4000
Weight (lb)	44.033	44.000	44.000
# of Objective Function Evaluations	-	840	169

The convergence history for this particular problem is shown in Figure 15, which further demonstrates that the algorithm is not affected by the initial design points, as they all lead to the same optimal solution with only slight differences in the number of structural analyses required.

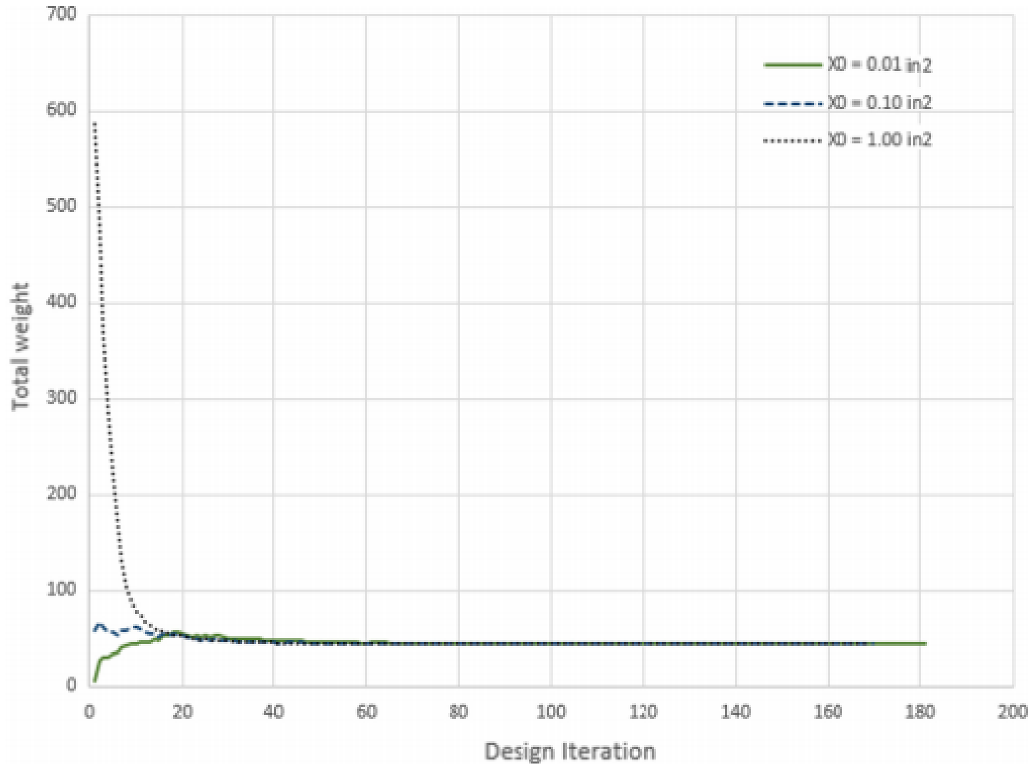


Figure 15. Convergence history for the 45-member truss structure

4.5.4. Problem 4: 39-Member, 12-Node Truss Structure

This example involves a ground structure consisting of 39 members, as depicted in Figure 16. The constraints and material properties for this problem are the same as those of the 11-member truss problem in Problem 1, except for the bounds of the cross-sectional area, which range from 10^{-6} in^2 to 2.25 in^2. The displacement at nodes under external loads should be less than or equal to 2 inches, and the stress in each member should not exceed the maximum allowable value of 20 ksi. The value for the components of the initial design vector is set at $\bar{A}^u = 2.25 \text{ in}^2$

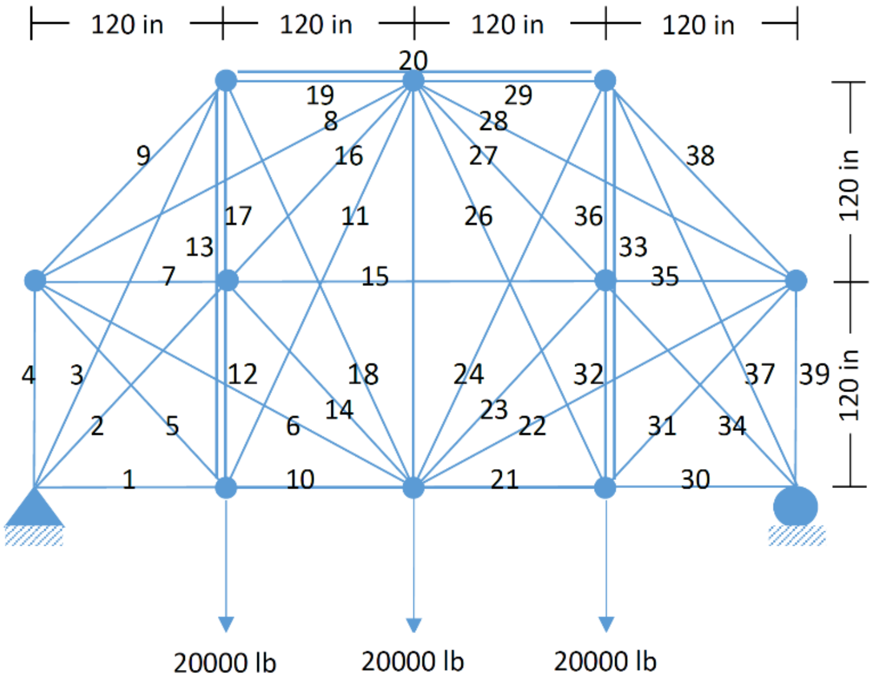


Figure 16. 39-member, 12-node ground structure

The optimal topology obtained using the proposed CA approach is shown in Figure 17. The optimal topology configurations obtained by Deb and Gulati [65] using GA with different population sizes are also presented in Figures 18 and 19 for the sake of comparison. It can be observed that the optimal topology obtained by the proposed non-uniform CA approach is different from those found using GA reported by Deb and Gulati [65]. Table 7 also provides a comparison of the optimal cross-sectional areas. The best optimum design found by Deb and Gulati (Figure 19) had a total weight of 196.546 lb and required a population size of 840. On the other hand, the total weight of the optimal truss configuration found using the proposed CA approach was found to be 192.00 lb and required only 399 objective function evaluations (it is noted that the total number of generations or number of structural analyses was not reported in Ref.[65]). The stress in each member of the optimal truss configuration was found to be equal to the allowable stress of 20 ksi while satisfying all displacement constraints.

Figure 20 displays the convergence history of the proposed optimizer for the chosen initial design points, highlighting the rapid convergence of the algorithm to the optimal solution. It is noteworthy that the proposed algorithm is found to be insensitive to the initial design points, as all initial designs lead to the same optimal solution with only minor variations in the number of structural analyses required to reach convergence.

It can be observed from case study problems 1-4 that the proposed non-uniform CA based design optimization approach proposed yields designs that are either the same or lighter compared to those obtained by other optimization techniques while requiring significantly less structural analyses.

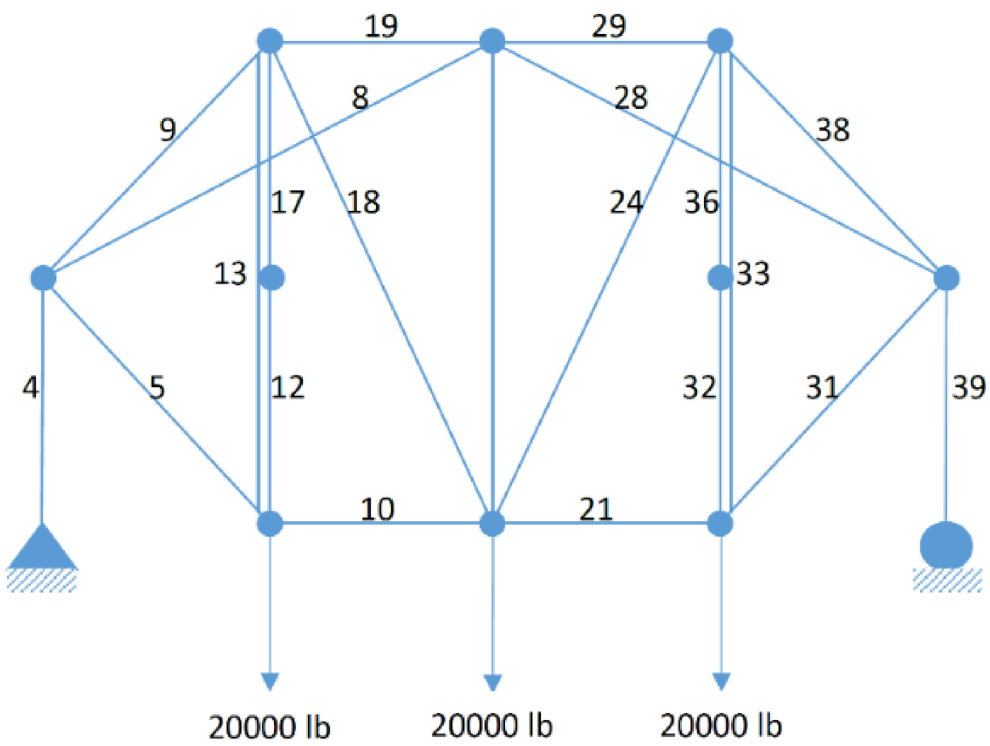


Figure 17. Optimized topology for the 39-member truss structure obtained using the proposed CA approach

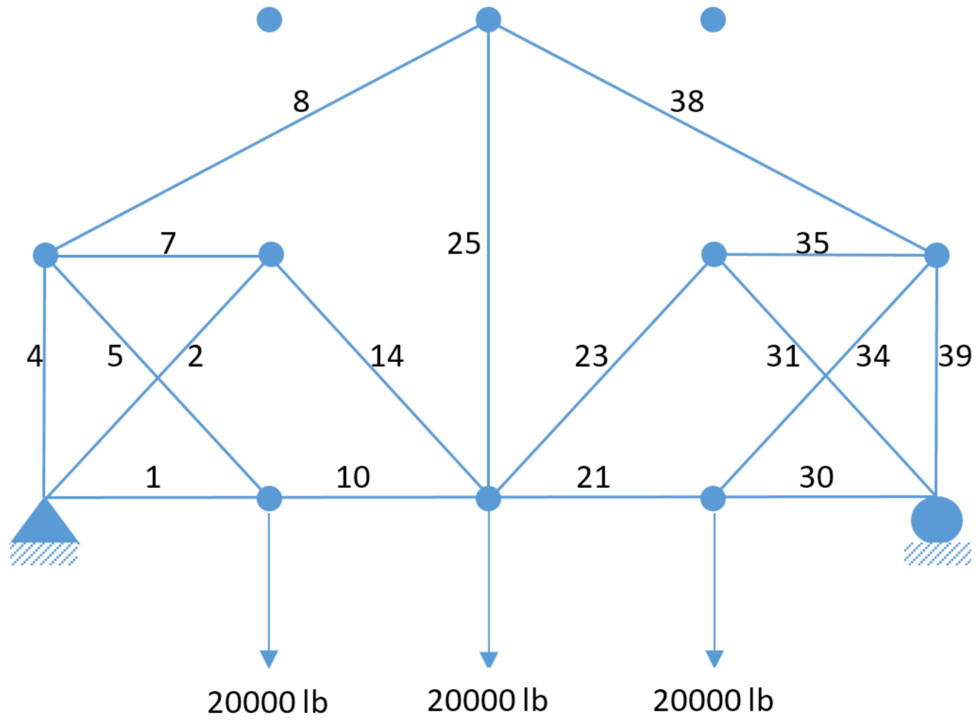


Figure 18. Optimized topology for the 39-member truss structure reported by Deb and Gulati [65] using GA, with a population size of 630

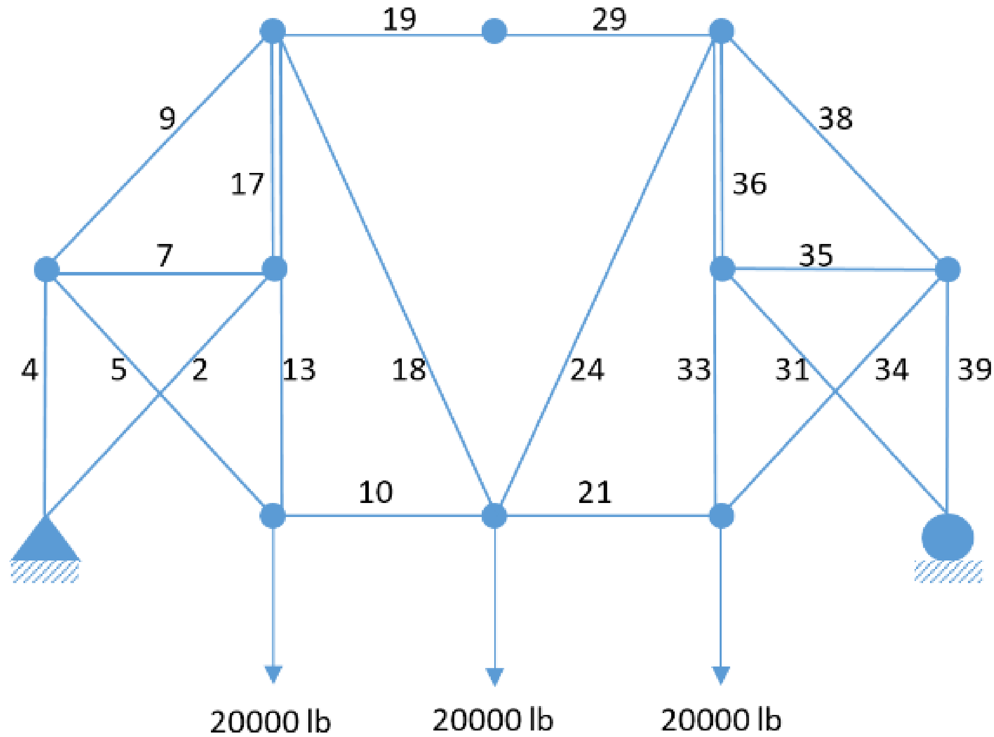


Figure 19. Optimized topology for the 39-member truss structure reported by Deb and Gulati [65] using GA, with a population size of 840

Table 7: Comparison of optimal cross-sections (in^2) for the 39-member truss

Member no.	Deb and Gulati [13] Population size = 630	Deb and Gulati [13] Population size = 840	Proposed Method
1,30	0.05	-	-
2,34	0.052	0.051	-
3,37	-	-	-
4,39	1.501	1.502	1.5000
5,31	1.416	1.061	1.1702
6,22	-	-	-
7,35	0.050	0.051	-
8,28	1.118	-	0.3465
9,38	-	0.052	0.7319
10,21	1.001	0.751	0.8277
11,26	-	-	-
12,32	-	-	0.0862
13,33	-	0.251	0.0862
14,23	0.050	-	-
15	-	-	-
16,27	-	-	-
17,36	0.052	-	0.0862
18,24	-	0.559	0.3857
19,29	-	-	0.3450
20	-	1.005	0.3450
25	1.002	-	0.309
Weight (lb)	198.00	196.546	192.000
No. structural analyses	-	-	399

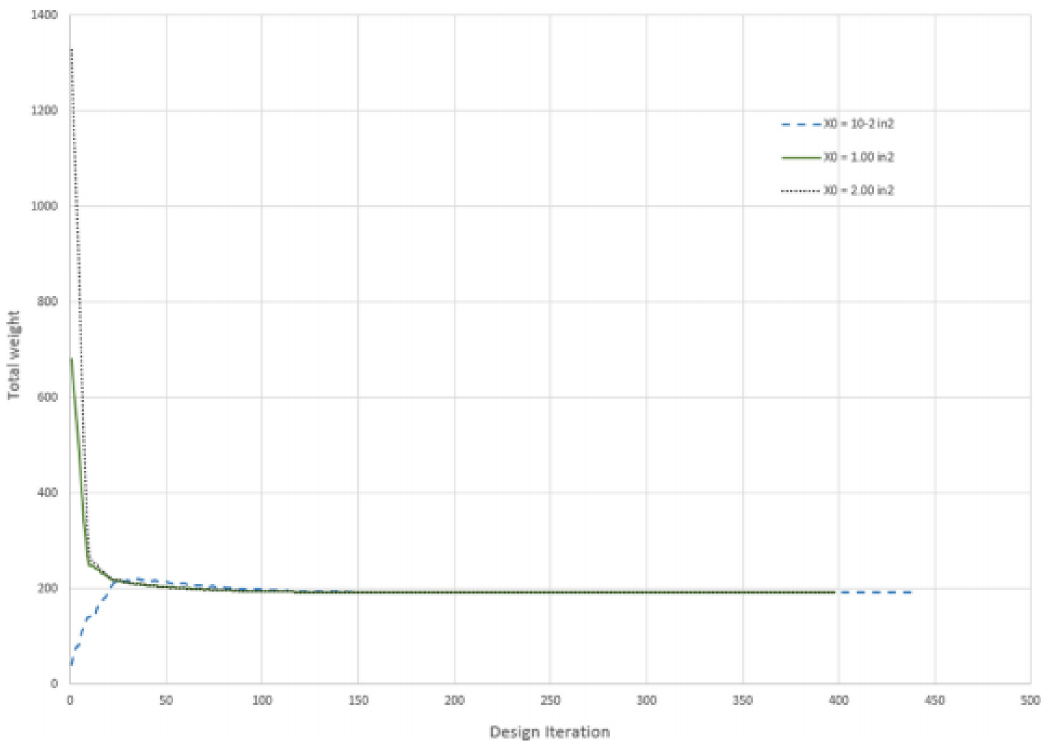


Figure 20. Convergence history for the 39-member, 12-node truss structure

4.5.5. Problem 5: Geodesic Dome

The geodesic dome structure shown in Figure 21, consisting of 132 members and 61 nodes, is a well-known benchmark space truss structure. This is a relatively large size structure, which has been previously studied by Patnaik et al. [108] who solved this space truss design optimization problem using different optimization techniques, including MFUD and gradient-based methods based on SUMT (Sequential Unconstrained Minimization Technique). In this problem, the structure is subjected to a single downward vertical load of 2000 lb at the apex of the dome (node 1). The maximum allowable stress for each member is set at 25 ksi. Additionally, the displacement at node 1 must not exceed 0.5 inches, and the displacements in all coordinate directions of the boundary nodes (nodes 38 through 61) are set to zero. The material properties and other parameters are listed in Table 8. The value for the components of the initial design vector for this problem is also set at $\bar{A}^u = 1\ \text{in}^2$.

Table 8. Problem parameters for the Geodesic dome

Young's modulus (E)	10^4 ksi
Density (ρ)	0.1 lb/in
Allowable stress ($\bar{\sigma}$)	±25 ksi
Allowable displacement	0.5 in
$\bar{A}^u$	1 in^2
$\bar{A}^l$	10^{-2} in^2

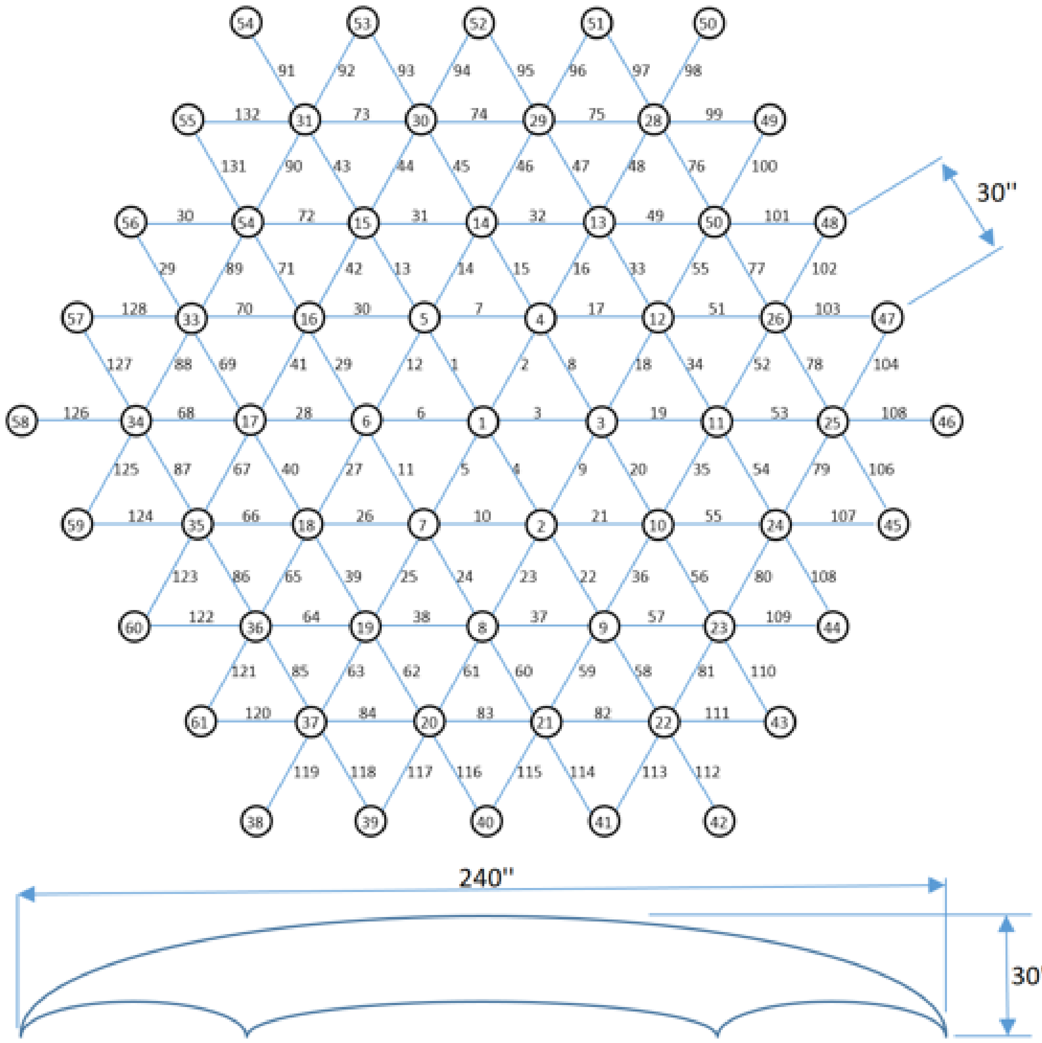


Figure 21. Geodesic dome

Table 9 presents the optimal results obtained using the proposed non-uniform CA approach for the 132-member geodesic dome structure. The proposed method successfully converged to a total weight of 95.67 lb after only 373 objective function evaluations. The displacement constraint at node 1 under the load was found to be active while stress in elements are all lower than the allowable value. The weight of the obtained design is significantly lighter than those found by MFUD (119.44 lb) and SUMT (118.65 lb) as reported in Ref. [108]. The convergence of the proposed algorithm to the optimal solution is shown in Figure 22, which also again shows the insensitivity of the proposed algorithm to the initial design points.

Table 9. Optimal cross-sections (in^2) for the 132-member geodesic dome

Member no.	**Member area, in.²**	**Stress (Psi)**	
A_1	1.71390	-3178.1	A_1 : Members 1-6
A_2	1.18860	3170.4	A_2 : Members 7-12
A_3	0.40182	-3185.6	A_3 : Members 13, 16,19, 22, 25 & 28
A_4	0.13116	-3193.6	A_4 : Members 14, 15, 17, 18, 20, 21, 23, 24, 26, 27, 29, 30
A_5	0.25970	3148.9	A_5 : Members 31-42
A_6	0.15376	-3283.2	A_6 : Members 43, 48, 53, 58, 63, 68
A_7	0.13454	-3196.2	A_7 : Members 45, 46, 50, 51, 55, 56, 60, 61, 65, 66, 70 & 71
A_8	0.10199	3248.9	A_8 : Members 73, 75, 76, 78, 79, 81, 82, 84, 85, 87, 88 & 90
A_9	0.14479	2959.2	A_9 : Members 74,77,80,83,86 & 89
A_{10}	0.03707	-3769.4	A_{10}: Members 91,98,105,112,119 & 126
A_{11}	0.09212	-3651.3	A_{11}: Members 93,96,100,103,107,110,114,117,121,124, 128 & 131
A_{12}	0.01000	-	A_{12}:Members 44,47,49,52,54,57,59,62,64,67,69,72,92,94,95,97, 99,101,102,104,106,108,109,113,115,116,118,120,122,123,125,127, 129,130,132
Weight (lb)	**95.67**		

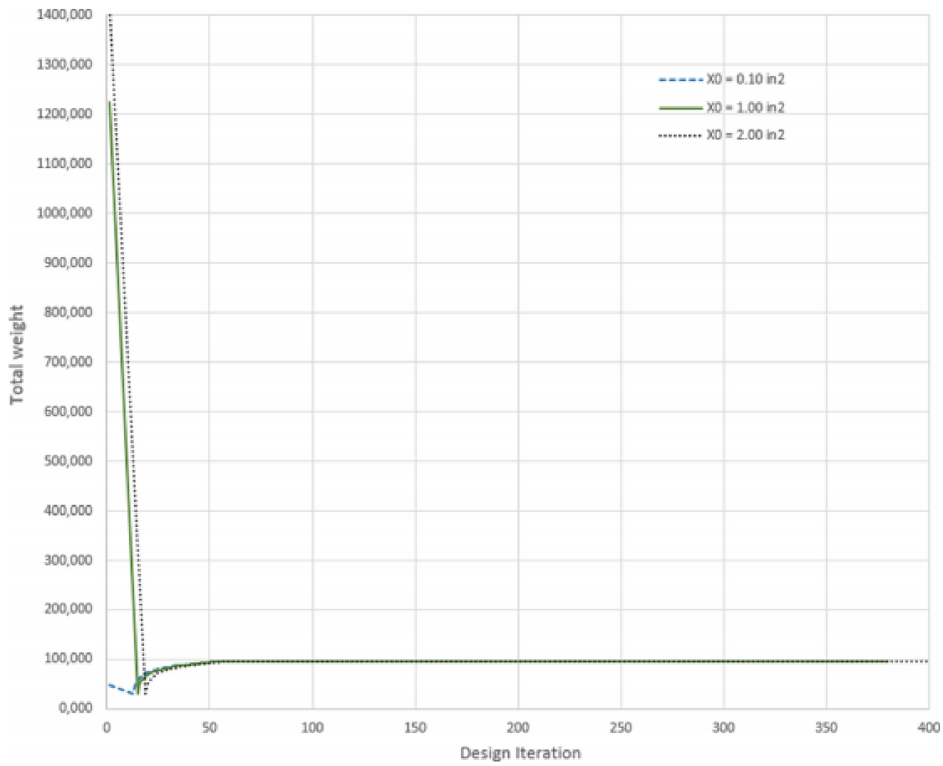


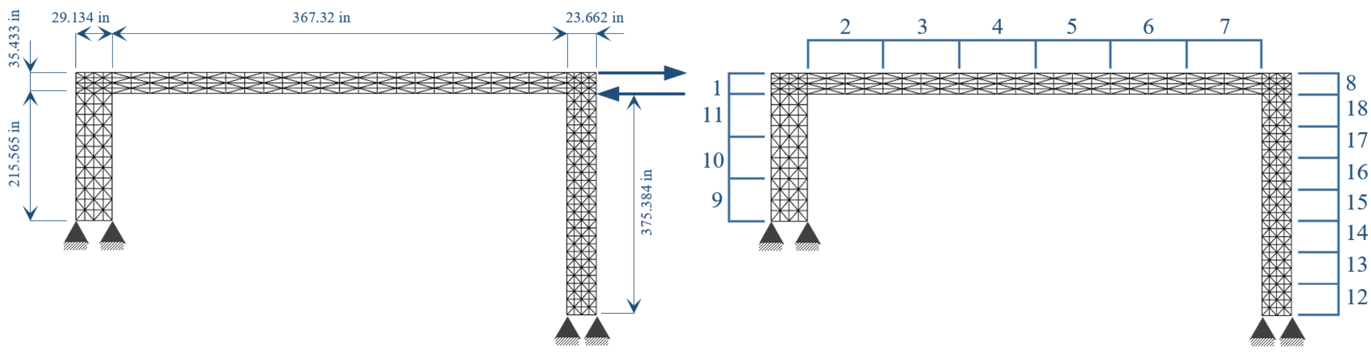
Figure 22. Convergence history for the geodesic dome truss structure

4.5.6. Problem 6: 940-member, 365-node truss structure

Finally, a large-scale plane truss structure with 940 members and 365 nodes shown in Figure 23 is studied. This problem has been previously addressed by Gashemi [69] and Dede et al. [71] using genetic algorithms. The material properties and problem constraints are provided in Table 10. The allowable stress limit for each member is set at 29 ksi, and the maximum allowable displacement at all nodes in both x and y directions is considered to be 1.5748 in. The value for the components of the design vector is initially set at $\bar{A}^u$= 5 in^2.

Table 10. Problem parameters for the 940-member, 365-node truss structure

Young's modulus (E)	100 Msi
Density (ρ)	0.1 lb/in
Allowable stress ($\bar{\sigma}$)	±29 ksi
Allowable displacement ($\bar{u}$)	1.5748 in
$\bar{A}^u$	5 in^2
$\bar{A}^l$	10^{-3} in^2



(a) configuration of the 940-member structure.

(b) member grouping of the 940-member

Figure 23. 940-member, 365-node truss structure, (a) configuration of the 940-member truss structure, (b) member grouping of the 940-member truss structure.

The optimal results obtained using the proposed non-uniform CA approach are presented in Table 11. The method was successful in reaching an optimal design with a total weight of 360.67 lb, after only 775 structural analyses, without violating any displacement or stress constraints. The optimal design obtained using the proposed method was found to be lighter than those found by Gashemi et al. [69] (370.96 lb) and Dede et al. [71] (368.56 lb). However, it should be noted that the total number of generations or structural analyses required by both GA algorithms were not reported in [69, 71]. The convergence history of the proposed algorithm is shown in Figure 24, which again indicates rapid convergence to the optimal solution for this large-scale optimization problem. The graph also demonstrates the insensitivity of the proposed algorithm to initial design points.

Table 11. Comparison of optimal cross-sections (in^2) for the 940-member truss structure

Group no.	Gashemi et al. [69] Ps = 200	Dede et al. [71] Ps = 80	Proposed Method
A_1	0.0918	0.0765	0.0870
A_2	0.1677	0.1515	0.1187
A_3	0.2329	0.2356	0.2362
A_4	0.4045	0.4036	0.4026
A_5	0.5748	0.5694	0.5690
A_6	0.7552	0.7444	0.7355
A_7	0.8927	0.8954	0.8987
A_8	1.5510	1.5348	1.5210
A_9	0.0921	0.0903	0.0865
A_{10}	0.0685	0.0604	0.0571
A_{11}	0.1157	0.1193	0.1122
A_{12}	0.1395	0.1377	0.1323
A_{13}	0.0643	0.0643	0.0627
A_{14}	0.0516	0.0516	0.0491
A_{15}	0.0948	0.0966	0.0929
A_{16}	0.1440	0.1485	0.1400
A_{17}	0.1916	0.1934	0.1871
A_{18}	0.2627	0.2510	0.2436
Weight (lb)	**370.96**	**368.56**	**360.67**
No. structural analyses	**-**	**-**	**1443**

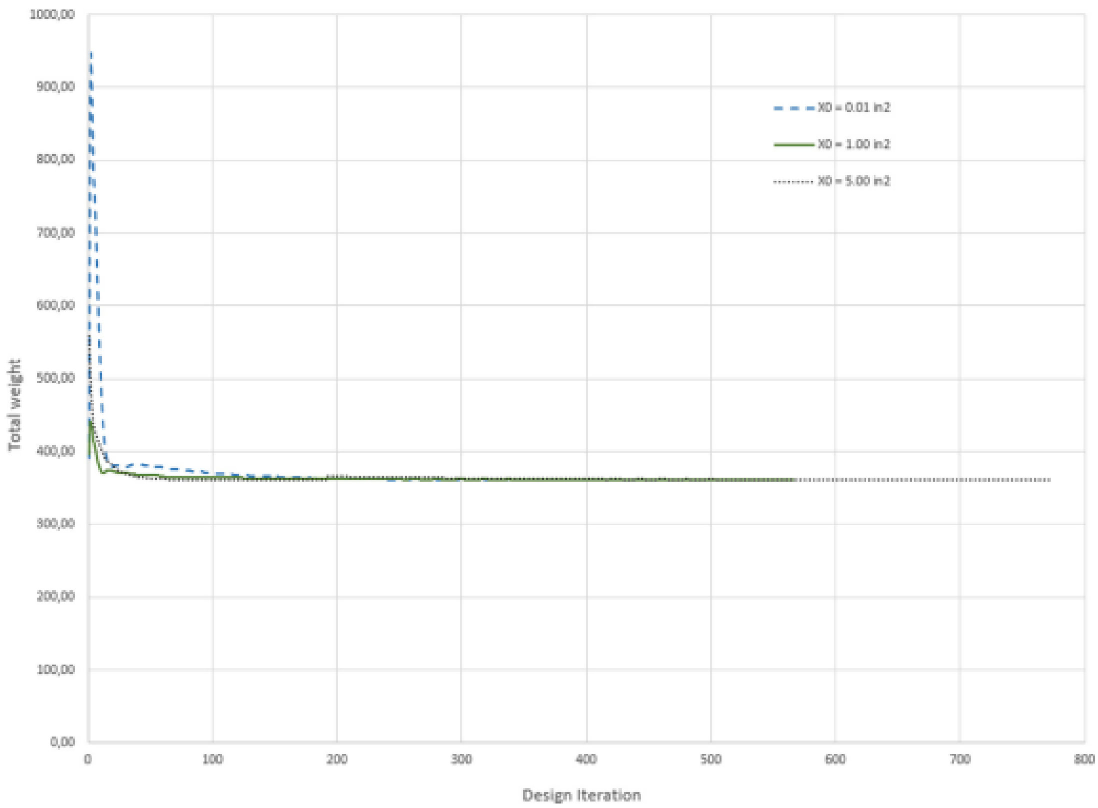


Figure 24. Convergence history for the 940-member truss structure

4.6. Conclusion

A novel CA based approach for the optimization of truss type structures has been proposed. The method has been tested on various benchmark problems. The obtained results show that the method can achieve the same or better level of efficiency and effectiveness compared to other methods with a reduced computational cost. A further research goal is to exploit the non-homogeneity feature of the presented method to solve the general problem of Topology, Sizing and Layout optimization by adding the nodes' coordinates to the set of state variables, and thus, allowing the nodes of the structure to move through the optimization process. Moreover, the performance of the presented algorithm can be enhanced by exploiting the parallelism feature of the CA techniques, which makes them highly suitable for solving large size structural optimization problems.

Chapter 5
A Non-Uniform Cellular Automata Framework for Sizing, Topology and Layout Optimization of Truss

5.1. Introduction

This chapter introduces a bi-level non-uniform Cellular Automata (CA) algorithm for optimizing truss structures in terms of sizing, topology, and layout. In the previous phase of this research, the propsoed non-uniform CA algorithm was successfully implemented to solve weight optimization problems for sizing and topology optimization of plane and space truss structures subjected to stress and displacement constraints. In this chapter, the aim is to extend the developed non-unform CA based design optimization methodology by including the position of the cell node (joints) coordinates into the vector of design variables, which permits simultanous sizing, topology and layout optimization. Several benchmark problems were provided to demonstrate the efficiency and accuracy of the proposed methodology.

5.2. Problem Statement

The problem involves identifying the optimal cross-sectional areas, material distribution, and joint positions for truss structures that minimize their total weight while satisfying stress and displacement constraints. This optimization problem can be expressed mathematically as:

$$minimize \quad W_T = \rho \sum_{j=1}^{m} A_j L_j \tag{20a}$$

$$subject\ to \quad \boldsymbol{Ku} = \boldsymbol{f} \tag{20b}$$

$$\frac{|\sigma_j|}{\bar{\sigma}_j} - 1 \leq 0\,, \qquad j = 1,2,\cdots,m \tag{20c}$$

$$\frac{|u_i|}{\bar{u}_i} - 1 \leq 0\,, \qquad i = 1,2,\cdots,p \tag{20d}$$

$$\bar{A}^l \leq A_j \leq \bar{A}^u\,, \qquad j = 1,2,\cdots,m \tag{20e}$$

$$|\Delta x_i| \leq \Delta\bar{x}_i\,, \qquad i = 1,2,\cdots,p \tag{20f}$$

where ρ is the weight density (assumed to be uniform for all members), m is the total number of bar members in the structure. $\boldsymbol{K}$ is the stiffness matrix, $\boldsymbol{f}$ is the vector of nodal forces and $\boldsymbol{u}$ is the vector of nodal displacements. The equilibrium is satisfied through the stiffness equation in constraint (20b) while constraints (20c) and (20d) are the stress and displacement constraints in which $\bar{\sigma}_j$ and $\bar{u}_i$ represent the allowable values on stress and displacement, respectively. It is noted that to guard against buckling the allowable stress $\bar{\sigma}_j$ in compression members is taken to be the minimum value between the allowable compressive stress and the stress at which Euler buckling occurs. Constraints (20e) and (20f) are presented to limit cross-sectional areas and nodal coordinates, respectively where $\bar{A}^u$ and, $\bar{A}^l$ are the upper and lower bounds on the cross-sectional areas, while $\Delta\bar{x}_i$ are the maximum allowable move on nodal coordinates. A very small positive value is used as the lower bound for cross-sectional area to maintain the kinematic stability of the structure during optimization. The vector of design variables consist of member cross-sectional areas and position of nodal coordinates. It should be noted that topology optimization is implemented within the sizing optimization problem by removing members with a cross-sectional area equal to their lower bound in the final optimal configuration.

5.3. Design Update

The design update rule in this approach involves two stages. The first stage involves optimization of sizing and topology through design update rules for the cross-sectional areas, while the second stage involves optimization of layout through design update rules for nodal coordinates. Both of these stages are implemented using the proposed non-uniform CA approach.

5.3.1. Design update rules for the cross-sectional area

The Fully Stressed Design approach (FSD) [44, 112] has been again adopted in this study for the optimization of sizing variables. Although FSD may not always lead to the most optimal results for truss structures that are indeterminate or have multiple load cases [44], it has been deemed as the most appropriate approach in this study because of its simplicity and compatibility with a strain energy criterion that assumes uniform strain energy distribution in optimally designed structures. As discussed in Chapter 4, the FSD method suggests that every member in an optimally sized structure should attain its stress limit. Therefore, the update rule for the cross-sectional areas of the members based on FSD as mentioned in Chapter 4 can be again expressed as:

$$A_j^{k+1} = A_j^{k} \frac{|\sigma_j{}^k|}{\bar{\sigma}_j} \tag{21}$$

where k denotes design iteration number for the cross-sectional areas. Moreover, the strain energy in each member of the truss structure can be expressed as:

$$U_j = \frac{\sigma_j{}^2 L_j A_j}{2E_j} \tag{22}$$

The total strain energy of the structure and the contribution of each cell to the total strain energy of the structure can thus be presented as follows:

$$U_T = \sum_{j=1}^{m} \frac{\sigma_j{}^2 L_j A_j}{2E_j} \tag{23}$$

and

$$U_c^i = \frac{1}{2} \sum_{j=1}^{N} \frac{\sigma_j{}^2 L_j A_j}{2E_j} \tag{24}$$

where m denotes the number of members within the structure, and N is the total number of members within a given cell. To ensure that the sum of all strain energy associated with cells is equal to U_T, a ½ factor is introduced for consistency. If all members are under their maximum stress, then shifting the nodes to minimize the total strain energy becomes equivalent to the minimization of the total weight. Thus, if the coordinate x_i of a given node is shifted, the resulting changes in total weight and total strain energy can be expressed as:

$$\frac{dW_T}{dx_i} = \rho \frac{d}{dx_i} \left(\sum_{j=1}^{m} L_j A_j \right) \tag{25}$$

$$\frac{dU_T}{dx_i} = \frac{\bar{\sigma}^2}{2E} \frac{d}{dx_i} \left(\sum_{j=1}^{m} L_j A_j \right) = \frac{\bar{\sigma}^2}{2\rho E} \frac{dW_T}{dx_i} \tag{26}$$

To minimize the total weight of a structure, we need to shift the positions of the nodes in a way that minimizes the total strain energy. Thus, applying the FSD after every change in joint coordinates ensures that the optimization process moves in the direction that minimize the total weight of the structure.

However, determining the shifting step for each node that represents a given cell can be challenging, particularly if there are buckling constraints or cross-section linking involved. In such cases, the allowable stress is not constant for all members, making the problem ill conditioned. Nevertheless, this does not significantly affect the efficiency of the algorithm as it will be shown later in the benchmark examples.

5.3.2. Design update rule for the nodes' coordinates

The rules for updating a cell coordinates are determined by analyzing the distribution of strain energy within the structure. As mentioned above, this energy-based approach involves moving the cells coordinates in a way that minimizes the total strain energy of the structure, while keeping each member fully stressed. Essentially, we can achieve the optimal geometry by keeping each member at its maximum stress level while minimizing the overall strain energy of the structure.

The change in the total strain energy resulting from shifting a particular node can be evaluated using Taylor expansion and linear approximation around the current nodal coordinates as:

$$\Delta U_T \cong \left(\frac{dU_T}{dx_i}\right)\Delta x_i + \left(\frac{dU_T}{dy_i}\right)\Delta y_i + \left(\frac{dU_T}{dz_i}\right)\Delta z_i \tag{27}$$

In this study, the optimization process entails shifting the node position in a single direction during each iteration. Assuming that the direction is along the x_i, in order to ensure that the shift leads to descent direction (i.e., $\Delta U_T \leq 0$), we must have:

$$sign(\Delta x_i) = -sign\left(\frac{dU_T}{dx_i}\right) \tag{28}$$

If we assume that the shifting steps have equal values, the most effective direction for optimization would be the one that leads to the smallest change ΔU_T . As a result, the search direction for shifting is selected based on the component of the gradient vector with the smallest value. The gradient vector is given by:

$$\nabla U_T = \left[\frac{dU_T}{dx_i}, \frac{dU_T}{dy_i}, \frac{dU_T}{dz_i}\right]. \tag{29}$$

On the other hand, the shifting step, Δx_i in the search direction is calculated at the cell level by enforcing the cell strain energy towards its average value. This average value is determined by computing the average strain energy for the cell, which is given by:

$$\overline{U}_c = \frac{U_T}{N} \tag{30}$$

where $\overline{U}_c$ is the average cell strain energy U_T is the total strain energy of the structure and N is the number of members within a cell. It is worth mentioning that the average value of strain energy $\overline{U}_c$ is computed by averaging the values of all cells within the structure. The shifting step Δx_i for a specific node (cell) $\boldsymbol{i}$ is then determined using the general equation of the proposed gradient-based descent algorithm as:

$$\Delta x_i = -\alpha_t * sign\left(\frac{dU_T}{dx_i}\right) sign\left(\frac{\left(\Delta U_c^i\right)^t}{\left(\Delta U_c^i\right)^0}\right)\left[0.1 * L_{min} + \beta\left|\frac{\left(\Delta U_c^i\right)^t}{\left(\frac{dU_c}{dx_i}\right)^t}\right|\right] \quad (31)$$

where

$$\left(\Delta U_c^i\right)^t = \left(U_c^i\right)^t - \overline{U}_c^{\,t} \quad (32)$$

To ensure that the shifting direction is reversed when necessary, the equation (31) includes the term $sign\left(\left(\Delta U_c^i\right)^t / \left(\Delta U_c^i\right)^0\right)$. This term helps to monitor the change in the sign of ΔU_c^i, and if the value becomes negative (i.e., $sign\left(\left(\Delta U_c^i\right)^t / \left(\Delta U_c^i\right)^0\right) < 0$), the shifting direction must be reversed. L_{min} is the minimum member length within the structure, and α and β are control parameters determining the distance of travel in the design space. During each iteration (t), the shifting step of $0.1 * L_{min}$ operates at the global level and remains the same for all cells. However, the term $\beta\left|\left(\Delta U_c^i\right)^t / \left(\frac{dU_c}{dx_i}\right)^t\right|$ is a local term used to direct the cell strain energy towards its average value. The control parameters α and β are incorporated in the algorithm to enhance its convergence. These parameters are crucial for improving the performance of the algorithm and ensuring its successful convergence. The choice of step size significantly affects the performance and convergence of the algorithm. While a smaller step size typically results in more precise solutions, it may slow down the convergence rate and, in certain instances, cause the algorithm to become trapped in a local optimum. To address this, the optimization process starts with a larger total step size Δx_i to enable exploration of a wider search space, and the control parameter α_t is introduced and its value decreases proportionally with its current value, in order to progressively decrease the total step size. Specifically, α_t is calculated as $\alpha_t = (0.99)^t$ or $\alpha_t = (0.95)^t$ depending on the problem at hand. Meanwhile, the other control parameter β operates at the cell level and is determined based on the optimization problem. Its value which is always less than or equal to 1, depends on the average strain energy of the cell. As the average cell strain energy $\overline{U}_c^{\,t}$ increases, β becomes smaller. While some tuning is required for β, it is easy to determine.

In summary, the proposed approach consists of two distinct phases. During the first phase, the FSD method is employed without taking the layout variables into consideration. This results in obtaining an initial FSD solution. In the second phase, the layout optimization process begins by calculating the sensitivities for the current configuration. Specifically, the sensitivities $\frac{dU_T}{dx_i}, \frac{dU_T}{dy_i}, \frac{dU_T}{dz_i}$, and $\frac{dU_c}{dx_i}, \frac{dU_c}{dy_i}, \frac{dU_c}{dz_i}$ are computed. Once the sensitivities have been calculated, the nodes are shifted simultaneously, with each node shifting in its corresponding direction chosen from the components of the gradient. The step size for each node is given by Eq. (31). The node coordinates are then updated using the following rule:

$$x_i^{t+1} = x_i^t + \Delta x_i \tag{33}$$

This process is repeated until the layout optimization process converges to a solution that satisfies the design requirements. If, at the end of the optimization process, any of displacement constraints remain violated, the fully utilized displacement (FUD) approach [108] is employed and a uniform scaling is applied to the non-zero cross-sectional area of the structure in order to ensure that the maximum nodal displacement is within the allowable value.

5.3.3. Linking of sizing variables

For problems involving linking of sizing variables (cross-sectional areas), the approach utilized in this study is to assign the cross-sectional area of elements in a given group to the value of the area of the element with the highest stress ratio. This method ensures that the stress levels are balanced across the group.

5.3.4. Linking of layout variables

For problems involving linking of layout variables, the approach utilized in this study is to assign the coordinates of nodes in a given group to the average coordinate value of nodes in the same group. Furthermore, it was found that in problems involving the linking of both sizing and layout variables, a better optimum design could be obtained by decreasing the lower bound of the cross-sectional area by a factor of α during each iteration of the layout optimization process.

$$(\bar{A}^l)^{t+1} = \alpha * (\bar{A}^l)^t \tag{34}$$

5.3.5 Termination tolerances

The optimization algorithm incorporates termination tolerances that dictate the precision with which the algorithm ceases searching for a better solution. These tolerances are vital for ensuring that the obtained solution closely approximates the optimal one. However, caution must be exercised when employing excessively small tolerances. While they can drive the algorithm to search for even the tiniest improvement in the optimal solution, resulting in minimal weight savings, they also require a significantly higher number of evaluations of the objective function. This increase in evaluations leads to a rise in computational cost. Consequently, a trade-off exists: exceedingly small tolerances may lead to excessive computational expenses for marginal improvements in solution quality.

5.4 Computer Implementation of the Proposed Analysis and Design Optimization Strategy

The computer implementation of the proposed bi-level non-uniform CA algorithm presented in the previous section is outlined in the simplified flowchart depicted in Figure 25. It is important to note that the termination tolerances ε and ε_W are specific to the problem being addressed.

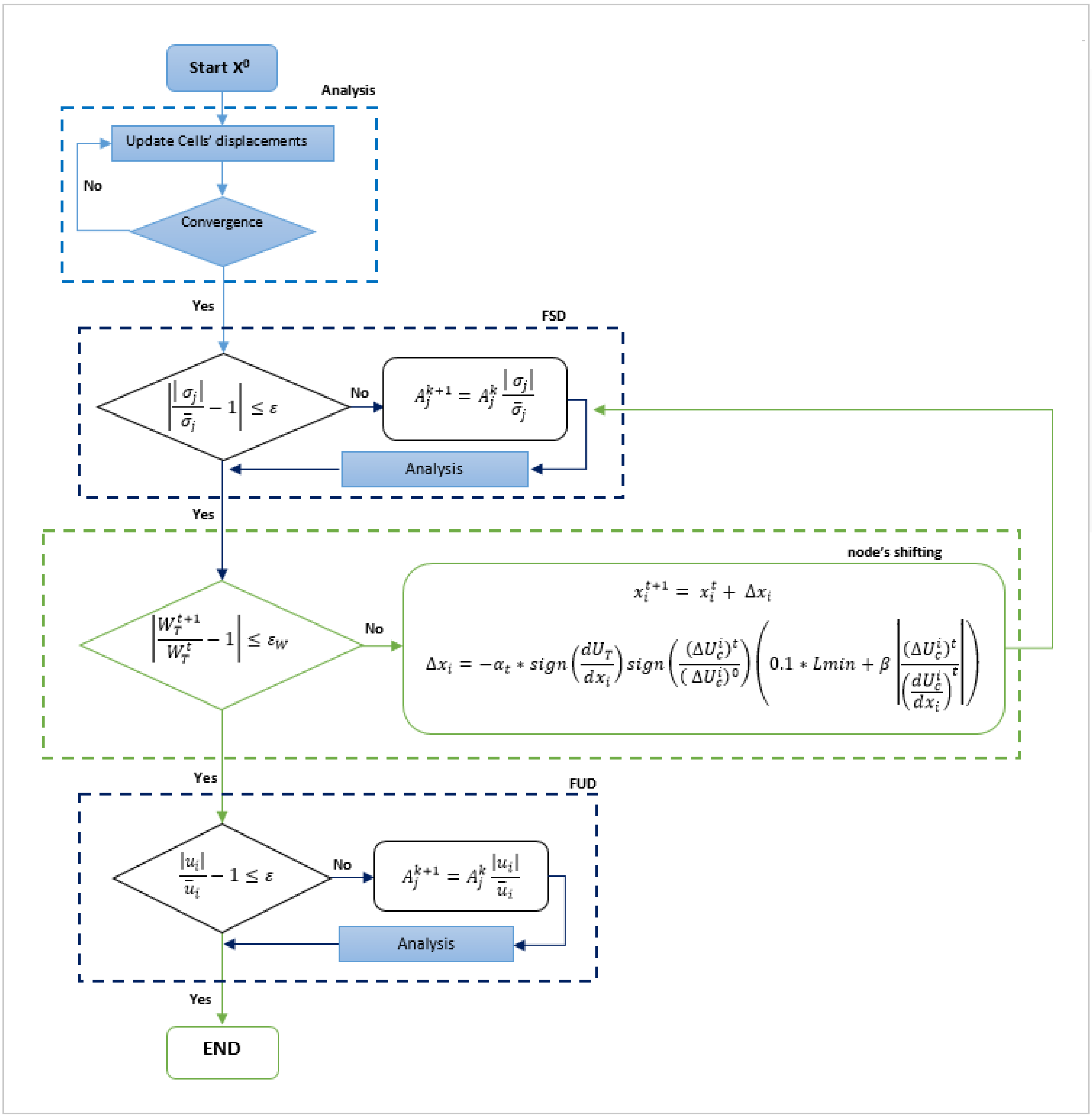


Figure 25. Flowchart of the non-uniform CA algorithm for sizing and layout optimization

5.5 Benchmark Case Studies for layout optimization

The effectiveness and accuracy of the proposed bi-level non-uniform CA method have been assessed through numerous benchmark case studies. The optimization results generated by the proposed approach are presented and compared with those based on other optimization algorithms in the existing literature. The aim is to obtain the concurrent optimal sizing, topology, and layout configuration of truss structures while adhering to both stress and displacement constraints. The design variables considered include the cross-sectional areas of the truss members and the nodal coordinates of specified joints. As previously mentioned, members whose cross-sectional areas approach the lower bounds (close to zero) are eliminated (topology optimization) during the sizing optimization.

5.5-1. Test Problem 1: 13-bar Michell Truss

This benchmark problem is a classic 13-bar, 8-node Michell truss structure, as shown in Figure 26, which has been widely studied in the literature. The material properties and problem parameters are listed in Table 12. The goal is to minimize the weight of the structure while satisfying both stress and displacement constraints by optimizing the cross-sectional areas and nodal coordinates at the top side of the truss.

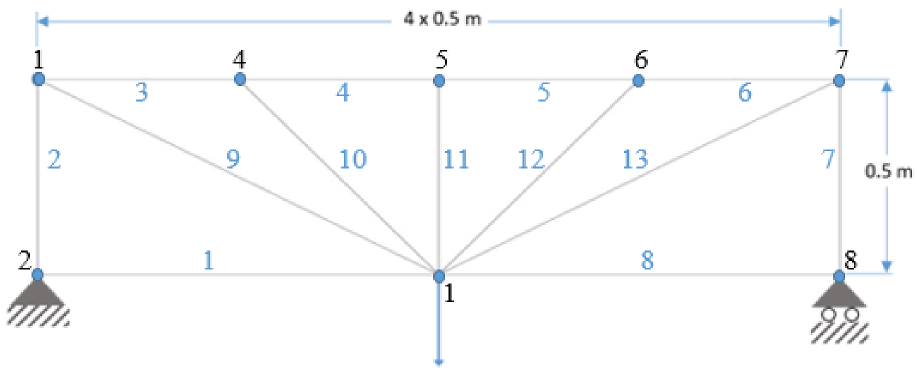


Figure 26. Schematic of the 13-bar Michell truss

Table 12. Problem data for the 13-bar Michell truss

Design variables		
Size variables: A_i; i = 1, 2, ..., 13		
Geometry variables: x_3=-x_7; x_4=-x_6; x_5; y_3=y_7; y_4=y_6; y_5		
Behaviour Constraints		
Stress constraints		
$(\sigma_t)_i \leq 240$ MPa; i = 1,..., 13		
$\|(\sigma_c)_i\| \leq 240$ MPa; i = 1,..., 13		
Displacement constraint in all direction of the coordinate system		
$\|\Delta_i\| \leq 3.8$ mm; i = 1,..., 18		
Side constraints		
Cross section areas		
$10^{-5} \leq A_i \leq 0.1$ (cm^2); i = 1, ..., 13		
Loading data		
Node	**F_x**	**F_y**
1	0.0	-200.0 KN
Material properties		
Modulus of elasticity E = 210 GPa		
Density of the material ρ = 7800 kg/m^3		
Control Parameters		
α_1 = 0.95		
β = 0.1		
Termination tolerances		
$\varepsilon = 10^{-7}$, $\varepsilon_W = 10^{-8}$		

This problem was first solved analytically by Michell, and subsequently by various researchers using different optimization techniques. The exact solution for this problem is given in [134]. Wang et al. [111] solved this problem using a bi-level evolutionary approach and reported a total weight of 20.9 kg with slightly different nodal coordinates compared to the analytical solution. The proposed methodology based on non-uniform CA achieved the exact weight of 20.9000 kg and the exact nodal coordinates after 2804 structural analyses. Table 13 compares the optimal values of sizing and layout variables obtained using the proposed method with those reported in [111]. The affected of control parameters α and β on the convergence of the algorithm are presente in Table 14. The optimized configuration and the convergence history of the proposed method for both the total strain energy and the total weight from the initial design are shown in Figures 27 and 28, respectively. Results show that the proposed optimization method rapidly converges to the exact optimal solution.

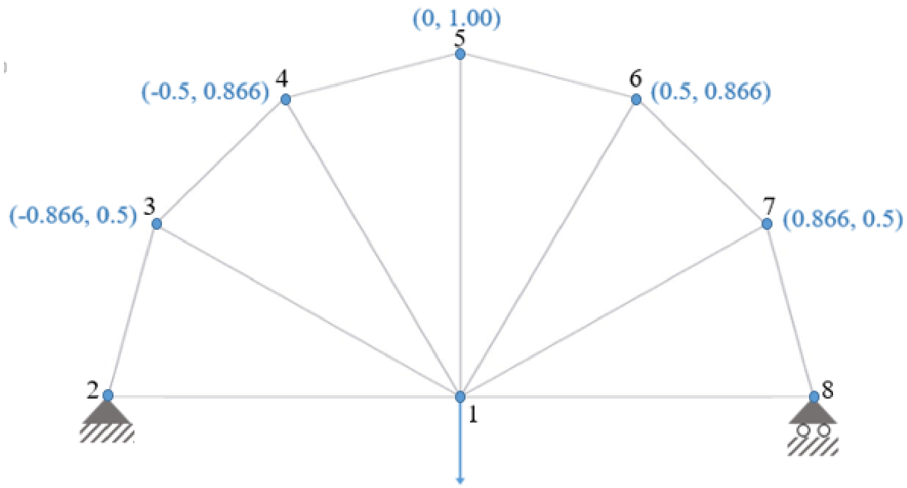


Figure 27. Optimized configuration for the 13-bar Michell truss

Table 13. Comparison of optimal cross-sections (cm²) for 13-bar Michell truss

Design variable	Exact solution [134]	Wang et al. [111]	Proposed Method
A_1, A_8 (cm²)	1.116	1.132	1.1166
A_2, A_7 (cm²)	4.314	4.318	4.3137
A_3, A_6 (cm²)	4.314	4.315	4.3125
A_4, A_5 (cm²)	4.314	4.3111	4.3110
A_9, A_{13} (cm²)	2.233	2.201	2.2283
A_{10}, A_{12} (cm²)	2.233	2.262	2.2376
A_{11} (cm²)	2.233	2.209	2.2286
Y_5 (m)	1.000	1.000	1.0005
Y_4, Y_6 (m)	0.866	0.867	0.8660
X_3, X_7 (m)	0.866	0.864	0.8660
Weight (kg)	20.90	20.90	20.9000
Max. stress constraint ratio	-	-	1.0000
No. structural analyses	-	-	2804

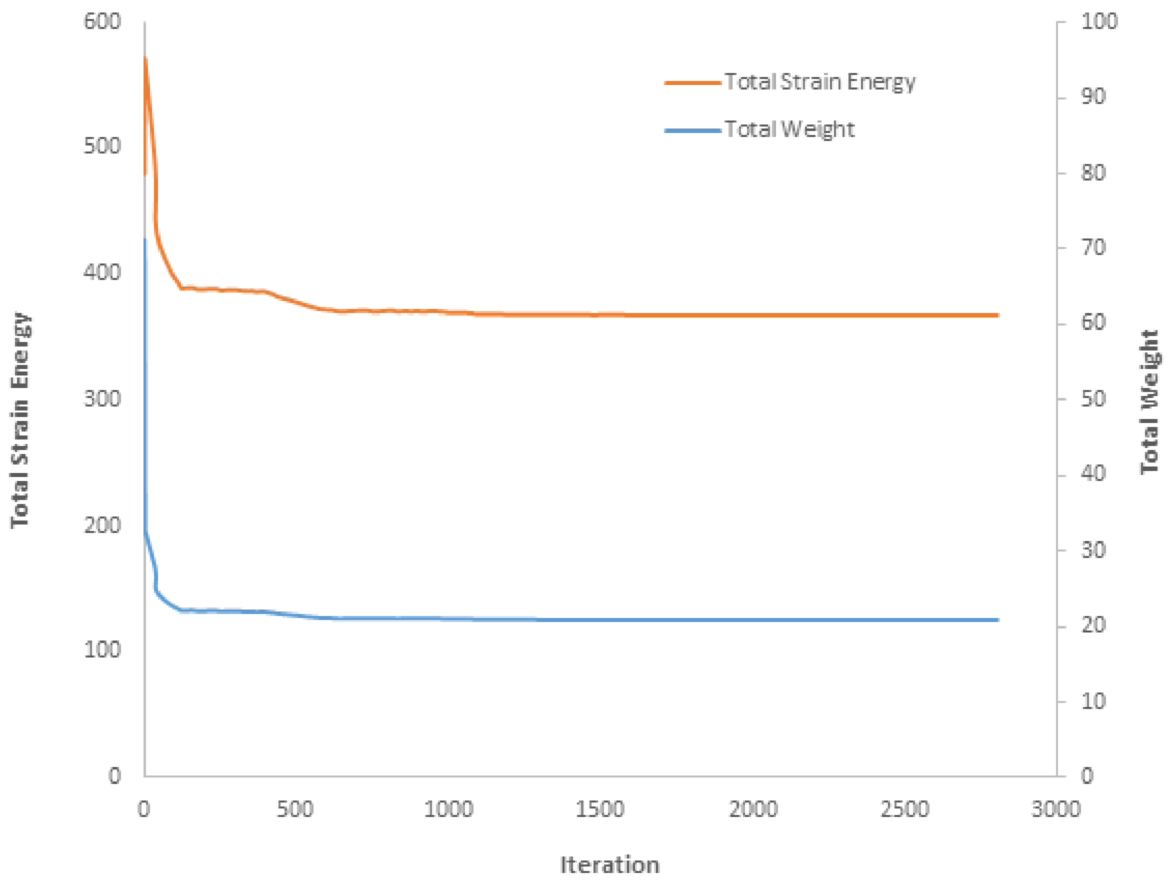


Figure 28. Convergence history for the 13-bar Michell truss

Table 14. The effect of control parameters α and β on convergence for the 13-bar Michell truss

α	β	Weight (kg)	No. structural analyses
0.90	0.01	21.2519	3495
0.90	0.10	20.9054	3305
0.95	0.01	*	-
0.95	0.10	**20.9000**	**2804**
0.99	0.01	*	-
0.99	0.10	*	-

(*) indicates a divergence of the algoithm

5.5.2. Test Problem 2: 15-Member, 8-Node Truss Structure

The 15-bar, 8-node planar truss test problem is shown in Figure 29. The material properties and problem parameters are also listed in Table 15. The minimum value for the cross-sectional area of the members is set at 0.001 in^2.

Table 15. Problem data for the 15-bar planar truss

Design variables		
Size variables: A_j; j = 1, 2, …, 15		
Geometry variables: $x_2=x_6$; $x_3=x_7$; y_2; y_3; y_4; y_6; y_7; y_8		
Behaviour constraints		
Stress constraints		
$(\sigma_t)_j \leq 25$ ksi; j = 1,…, 15		
$\lvert(\sigma_c)_j\rvert \leq 25$ ksi; j = 1,…, 15		
Side constraints		
Cross-sectional areas		
$0.001 \leq A_j \leq 30$ (in^2); j = 1, …, 15		
Nodal coordinates:		
100 in $\leq x_2 \leq$ 140 in; 220 in $\leq x_3 \leq$ 260 in;		
100 in $\leq y_2 \leq$ 140 in; 100 in $\leq y_3 \leq$ 140 in; 50 in $\leq y_4 \leq$ 90 in;		
-20 in $\leq y_6 \leq$ 20 in; -20 in $\leq y_7 \leq$ 20 in; 20 in $\leq y_8 \leq$ 60 in		
Loading data		
Node	**F_x**	**F_y**
8	0.0	-10.0 kips
Material properties		
Modulus of elasticity E = 10^4 ksi		
Density of the material ρ = 0.1 lb/in^3		
Control Parameters		
$\alpha_1 = 0.95$		
$\beta = 10^{-5}$		
Termination tolerances		
$\varepsilon = 10^{-8}$, $\varepsilon_W = 10^{-8}$		

The optimized layout configuration achieved using the proposed non-uniform Cellular Automata (CA) approach is shown in Figure 30. Table 16 also provides a comparison of the optimal values of cross-sectional areas and nodal coordinates obtained using the proposed methodology with those found by other optimization methods. The proposed approach outperforms other methods previously employed for this problem. Rahami et al. [66] utilized a single-level Genetic Algorithm (GA) and found an optimal weight of 76.6854 lb after performing 8000 structural analyses. Gholizadeh [85] employed a hybrid bi-level optimization algorithm combining CA with Particle Swarm Optimization (PSO) to find an optimal weight of 72.5143 lb after conducting 4500 structural analyses. Ahrari et al. [114] used the Fully Stressed Design based on Evolution Strategy (FSD-ES) method to achieve an optimal weight of 69.585 lb after performing 8508 structural analyses. The proposed design optimization method an optimal weight of 69.176 lb after only 240 structural analyses. It is also noted that the proposed method does not violate any of the stress constraints.

Table 17 illustrates how the control parameters α and β affect the convergence of the algorithm. The convergence history of the proposed algorithm, with respect to both total strain energy and total weight from the initial design, is shown in Figure 31. Results clearly show the rapid convergence of the proposed algorithm to the optimum solution.

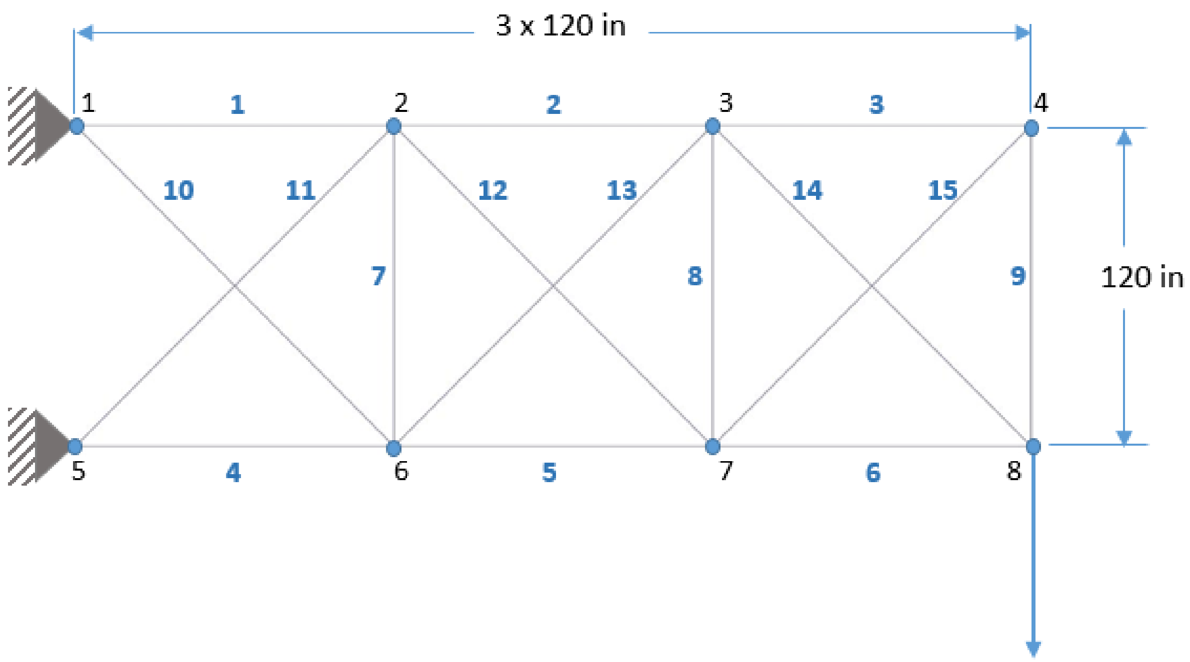


Figure 29. 15-bar, 8-node truss structure

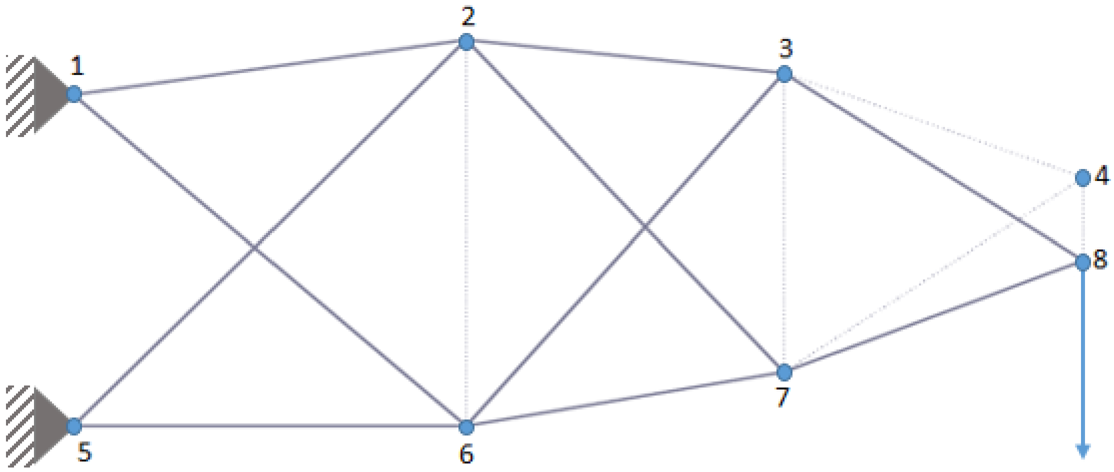


Figure 30. Optimized geometry for the 15-bar truss structure

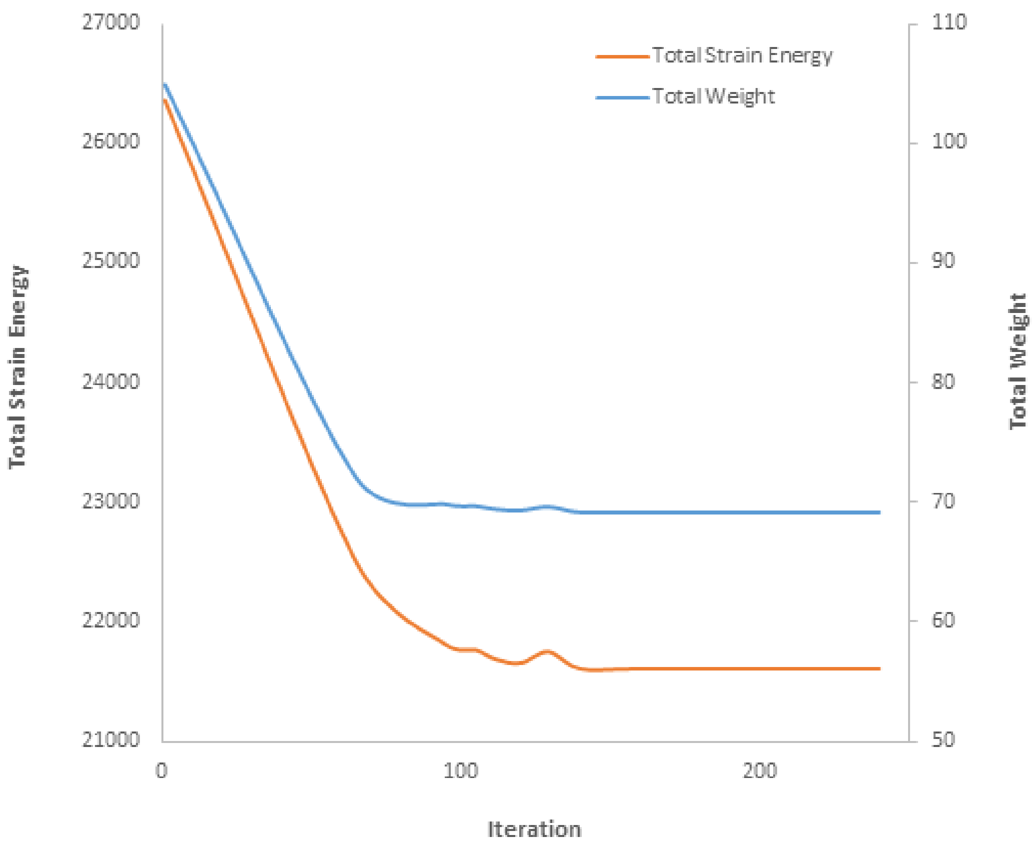


Figure 31. Convergence history for the 15-bar, 8-node planar truss

Table 16. Optimal cross-sectional areas and nodal coordinates for the 15-bar planar truss

Design variable	Rahami et al. [66]	Gholizadeh [85]	Ahrari et al. [114]	Proposed Method
A_1 (in^2)	1.0810	0.954	0.954	0.89346
A_2 (in^2)	0.5390	0.539	0.539	0.56598
A_3 (in^2)	0.2870	0.270	-	0.00100
A_4 (in^2)	0.9540	0.954	0.954	0.94409
A_5 (in^2)	0.5390	0.539	0.539	0.46634
A_6 (in^2)	0.1410	0.174	0.44	0.42026
A_7 (in^2)	0.1110	0.111	-	0.00100
A_8 (in^2)	0.1110	0.111	-	0.00100
A_9 (in^2)	0.5390	0.287	-	0.00100
A_{10} (in^2)	0.4400	0.347	0.22	0.41556
A_{11} (in^2)	0.5390	0.347	0.44	0.36191
A_{12} (in^2)	0.2700	0.220	0.22	0.09597
A_{13} (in^2)	0.2200	0.220	0.22	0.25487
A_{14} (in^2)	0.1410	0.174	0.22	0.46587
A_{15} (in^2)	0.2870	0.270	-	0.00100
X_2 (in)	101.5775	137.2216	100.0000	140.000
X_3 (in)	227.9112	259.9093	229.8186	253.350
Y_2 (in)	134.7986	123.5006	135.1354	140.000
Y_3 (in)	128.2206	110.0020	124.4261	127.920
Y_4 (in)	54.8630	59.9356	-	90.000
Y_6 (in)	-16.4484	-5.1799	-16.9664	0.000
Y_7 (in)	-13.3007	4.2193	-9.2015	19.574
Y_8 (in)	54.8572	57.8829	56.1693	60.000
Weight (lb)	76.6854	72.5143	69.585	69.176
Max. stress constraint ratio	0.9999	0.9996	1.0000	1.0000
No. structural analyses	8000	4500	8508	240

Table 17. The effect of control parameters α and β on convergence for the 15-bar planar truss

α	β	Weight (lb)	No. structural analyses
0.90	0.00001	70.175	833
0.90	0.00010	70.175	833
0.95	0.00001	**69.176**	**240**
0.95	0.00010	69.175	397
0.99	0.00001	70.276	12914
0.99	0.00010	70.276	9291

5.5.3. Problem 3: 18-Member, 11-Node Truss Structure

The third test problem tackled in this study involves an 18-member, 11-node truss structure shown in Figure 32, with material properties and problem parameters listed in Table 18. The cross-sectional area of the members is subject to a lower bound of 3.5 in^2.

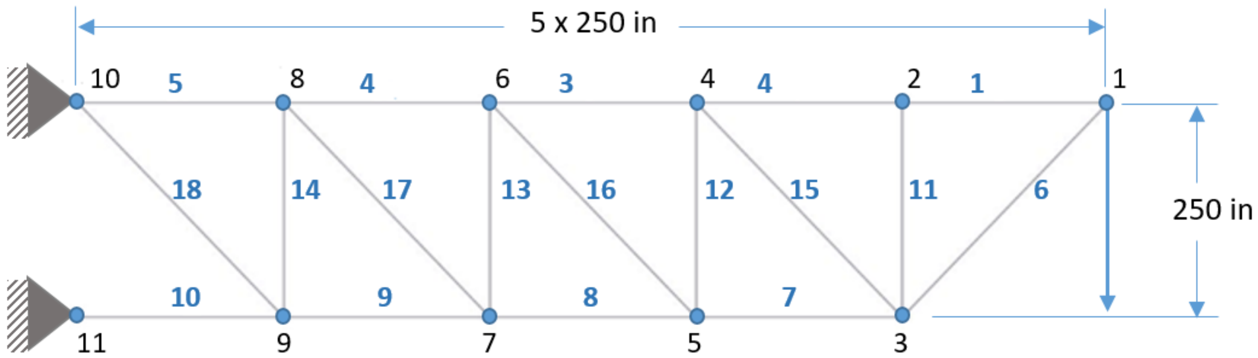


Figure 32. 18-bar, 11-node truss structure

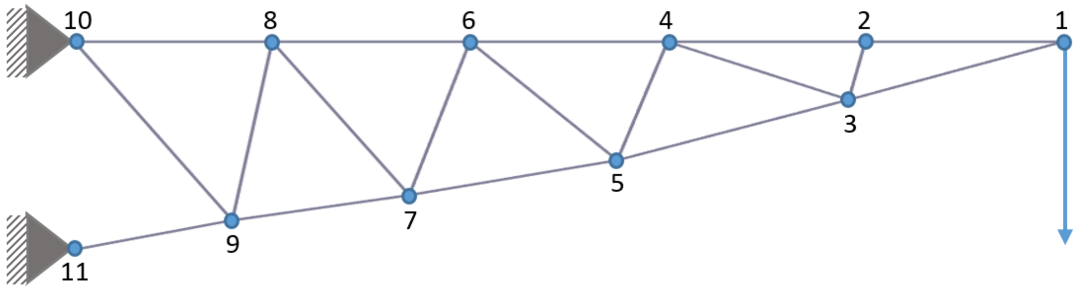


Figure 33. Optimized geometry for the 18-bar truss structure

Table 18. Problem data for the 18-bar planar truss

Design variables		
Size variables $A_1=A_4=A_8=A_{12}=A_{16}$; $A_2=A_6=A_{10}=A_{14}=A_{18}$; $A_3=A_7=A_{11}=A_{15}$; $A_5=A_9=A_{13}=A_{17}$		
Geometry variables: x_3; x_5; x_7; x_9; y_3; y_5; y_7; y_9		
Behaviour constraints		
Stress constraints		
$(\sigma_t)_j \leq 20$ ksi; j = 1,..., 18		
$\lvert(\sigma_c)_j\rvert \leq 20$ ksi; j = 1,..., 18		
Euler buckling Stress constraints		
$\lvert(\sigma_c)_j\rvert \leq 4E_jA_j/L_j^2$; j = 1,..., 18		
Side constraint Cross section areas:		
$3.5 \leq A_j \leq 18$ (in^2); j = 1, ..., 25		
Nodal coordinates:		
775 in $\leq x_3 \leq$ 1225 in; 525 in $\leq x_5 \leq$ 975 in; 275 in $\leq x_7 \leq$ 725 in; 25 in $\leq x_9 \leq$ 475 in;		
-225 in $\leq y_3, y_5, y_7, y_9 \leq$ 245 in		
Loading data		
Node	$\mathbf{F_x}$	$\mathbf{F_y}$
1, 2, 4, 6, 8	0.0	-20.0 kips
Material properties		
Modulus of elasticity E = 10^4 ksi		
Density of the material ρ = 0.1 lb/in^3		
Control Parameters		
$\alpha_1 = 0.95$		
$\beta = 10^{-7}$		
Termination tolerances		
$\varepsilon = 10^{-5}$, $\varepsilon_W = 10^{-5}$		

Figure 33 displays the optimal layout configuration achieved through the proposed bi-level non-uniform Cellular Automata (CA) methodology. Table 19 also provides a comparison of the optimal results obtained using the proposed methodology with those obtained using other optimization methods [36, 66, 85, 126]. As it can be realized, the proposed approach resulted in a significantly lower optimal weight compared to Hansen et al. [36] who used a method based on first-order Taylor series expansions of the member forces and stresses, as well as compared to Rahami et al. [66] who employed GA, and Gholizadeh [85] who used a hybrid CA and PSO method.

Hansen et al. [36] achieved a minimum weight of 4505.0 lb after only eight structural analyses, while Rahami et al. [66] reported a total weight of 4530.68 lb after 8000 structural analyses. Gholizadeh [85] obtained a slightly lower weight of 4512.365 lb after 4500 structural analyses. Flager et al. [126] found an optimal weight of 4321.52 lb after 65870 structural analyses, using the Fully Constrained Design (FCD) method combined with a gradient-based optimization method. Although Flager et al. [126] achieved an optimal weight that was much smaller than previously reported values; it was obtained by violating the maximum stress constraint ratio (1.075) and required an exceedingly large number of analyses. In contrast, the proposed method obtained an optimal configuration with an overall weight of 4480.77 lb after 6574 structural analyses, without violating any stress or buckling constraints. The effect of control parameters α and β on the algorithm convergence is presented in Table 20. Figure 34 also shows the convergence history of the proposed method for both the total strain energy and the total weight from the initial design.

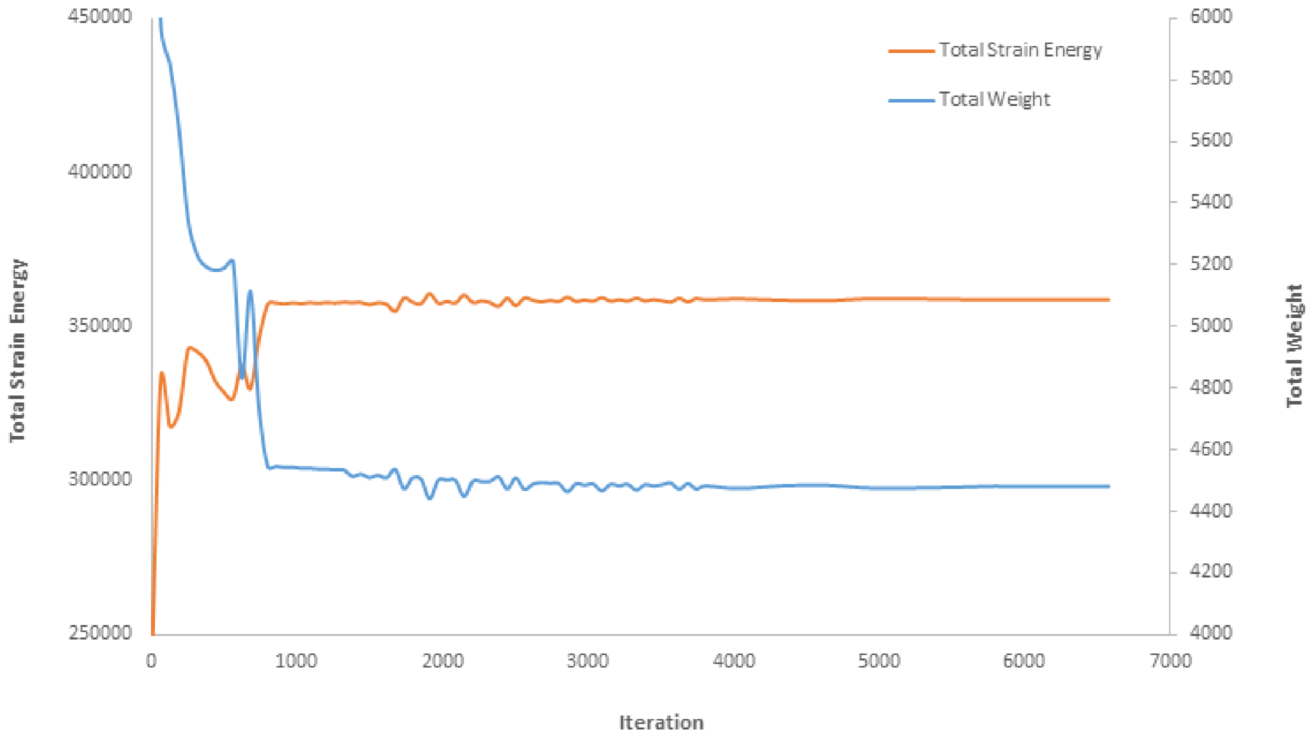


Figure 34. Convergence history for the 18-bar planar truss

Table 19. Optimal cross-sectional areas and nodal coordinates for the 18-bar planar truss

Design variable	Hansen et al. [36]	Rahami et al. [66]	Gholizadeh [85]	Flager [126]	Proposed Method
A_1 (in²)	12.76	12.25	12.50	11.25	10.159
A_2 (in²)	17.77	17.50	17.50	16.75	18.171
A_3 (in²)	5.55	5.75	5.75	5.75	9.008
A_4 (in²)	3.26	3.25	3.75	4.25	3.000
X_3 (in)	881.4	917.4475	907.2491	907.0477	983.06
Y_3 (in)	178.8	193.7899	179.8671	177.9134	180.19
X_5 (in)	628.9	654.3243	636.7871	635.1578	687.95
Y_5 (in)	124.9	159.9436	141.8271	136.8110	107.45
X_7 (in)	390.5	424.4821	407.9442	405.7088	417.87
Y_7 (in)	66.8	108.5779	94.0559	87.2047	65.554
X_9 (in)	313.2	208.4691	198.7897	198.3859	204.52
Y_9 (in)	45.0	37.6349	29.5157	21.4567	35.215
Weight (lb)	4505.0	4527.6952	4512.365	4321.52	4480.77
Max. stress constraint ratio	-	-	-	1.075	1.000
No. structural analyses	8	8000	4500	65870	6574

Table 20. The effect of control parameters α and β on convergence for the 18-bar planar truss

α	β	Weight (lb)	No. structural analyses
0.90	10^{-7}	5603.36	7018
0.90	10^{-6}	5603.3	12474
0.95	10^{-7}	**4480.77**	**6574**
0.95	10^{-6}	4480.54	13595
0.99	10^{-6}	5040.268	32670

5.5.4. Test Problem 4: 25-bar, 10-node space truss

This problem is a widely known benchmark space truss, which has been tackled using various optimization techniques. The structure is illustrated in Figure 35, and its material properties and problem parameters are provided in Table 21. This problem involves the linking of both the cross-sectional areas and the nodal coordinates. The minimum cross-sectional area of members for this problem is not fixed. It is initially set at 0.3 in^2 and it gradually decreases throughout the layout optimization process to a minimum value of 10^{-4} in^2 at a rate β.

Table 21. Problem data for the 25-bar space truss

Design variables			
Size variables A_1; $A_2 = A_3 = A_4 = A_5$; $A_6 = A_7 = A_8 = A_9$; $A_{10} = A_{11}$; $A_{12} = A_{13}$; $A_{14} = A_{15} = A_{16} = A_{17}$; $A_{18} = A_{19} = A_{20} = A_{21}$; $A_{22} = A_{23} = A_{24} = A_{25}$			
Geometry variables: $x_4 = x_5 = -x_3 = -x_6$; $x_8 = x_9 = -x_7 = -x_{10}$; $y_3 = y_4 = -y_5 = -y_6$; $x_7 = x_8 = -x_9 = -x_{10}$; $x_3 = x_4 = x_5 = x_6$			
Behaviour constraints			
Stress constraints: $(\sigma_t)_j \le 40$ ksi; $j = 1,\dots, 25$; $\lvert(\sigma_c)_j\rvert \le 40$ ksi; $j = 1,\dots, 25$			
Displacement constraint in all direction of the coordinate system: $\lvert\Delta_i\rvert \le 0.35$; $i = 1,\dots, 18$			
Side constraints:			
Cross-sectional areas: $10^{-4} \le A_i \le 1.2$ (in^2); $i = 1, \dots, 25$			
Nodal coordinates: 20 in $\le x_4 \le$ 60 in; 40 in $\le x_8 \le$ 80 in; 40 in $\le y_4 \le$ 80 in; 100 in $\le y_8 \le$ 140 in; 90 in $\le z_4 \le$ 130 in			
Loading data			
Node	$\mathbf{F_x}$	$\mathbf{F_y}$	$\mathbf{F_z}$
1	1.0 kips	-10.0 kips	-10.0 kips
2	0.0	-10.0 kips	-10.0 kips
3	0.5 kips	0.0	0.0
6	0.6 kips	0.0	0.0
Material properties			
Modulus of elasticity $E = 10^4$ ksi			
Density of the material $\rho = 0.1$ lb/in^3			
Control Parameters			
$\alpha_1 = 0.99$			
$\beta = 10^{-3}$			
Termination tolerances			
$\varepsilon = 10^{-3}$, $\varepsilon_W = 10^{-6}$			

The optimal layout obtained using the proposed method is presented in Figure 36, along with a comparison of optimal values of cross-sectional areas and nodal coordinates with those obtained using other optimization methods [36, 66, 85, 114] in Table 22. Hansen et al. [36] used a method based on first-order Taylor series expansions of member forces and stresses, and reported a minimum weight of 128.3 lb after seven structural analyses. In comparison, Rahami et al. [66] obtained a weight of 120.1149 lb after 10000 structural analyses using GA, while Gholizadeh [85] achieved a smaller weight of 117.2227 lb after 4500 structural analyses using a hybrid CA and PSO. Ahrari et al. [114] also reported a smaller overall optimal weight of 114.417 lb after 10000 structural analyses.

On the other hand, the proposed method achieved an optimal configuration with an overall weight of 113.789 lb, which is smaller than those reported in previous studies. The proposed method required 7409 structural analyses and did not violate any stress or displacement constraints. The convergence of the algorithm under different control parameters α and β is presented in Table 23. The convergence history of the proposed method for both the total strain energy and the total weight from the initial design is also depicted in Figure 37.

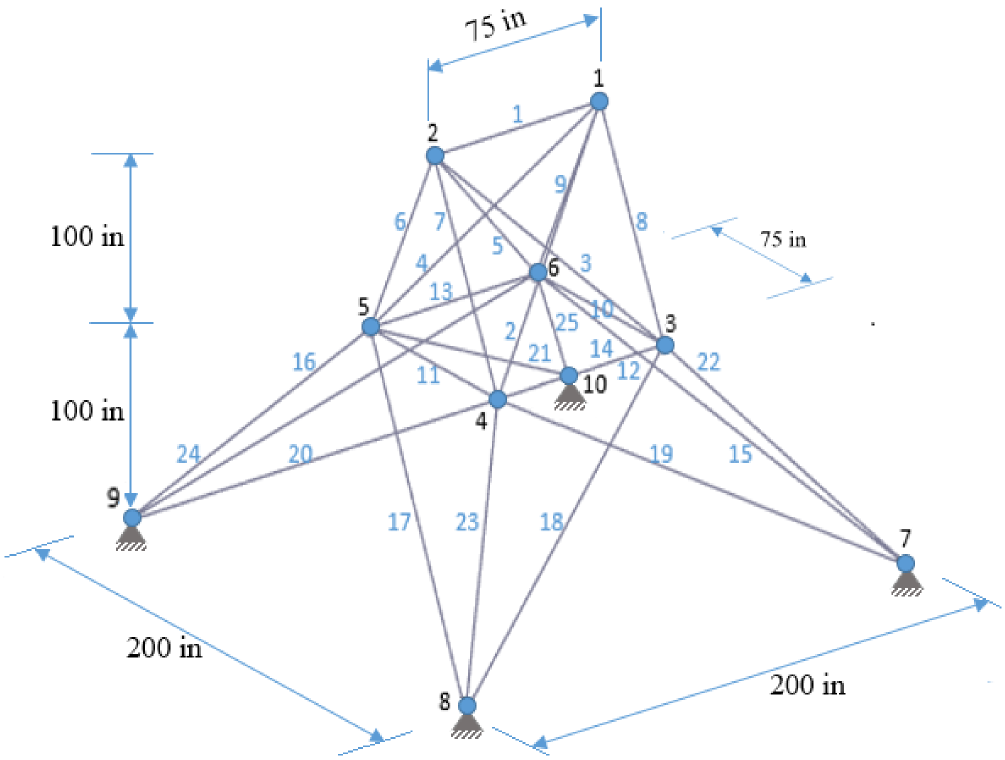


Figure 35. 25-bar, 10-node space truss

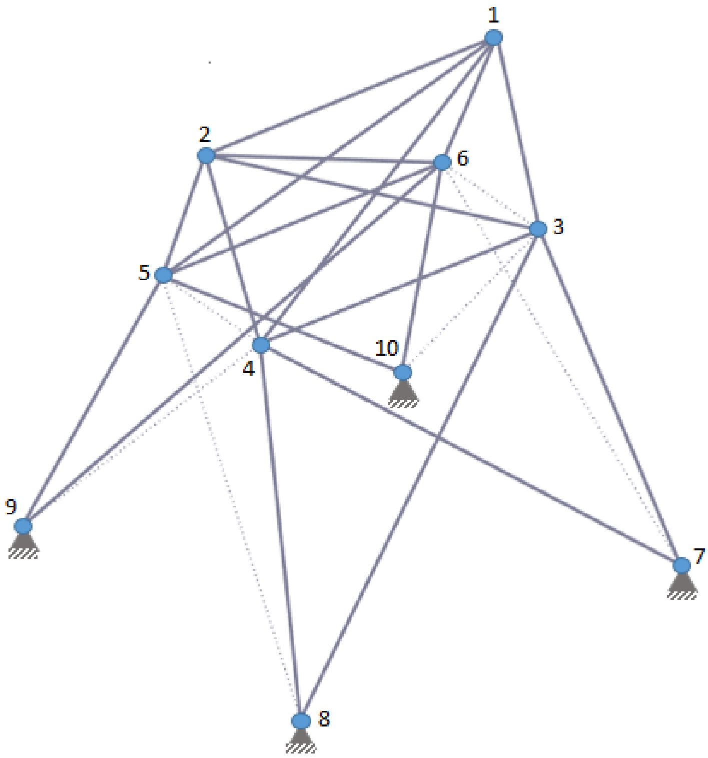


Figure 36. Optimized geometry for the 25-bar space truss

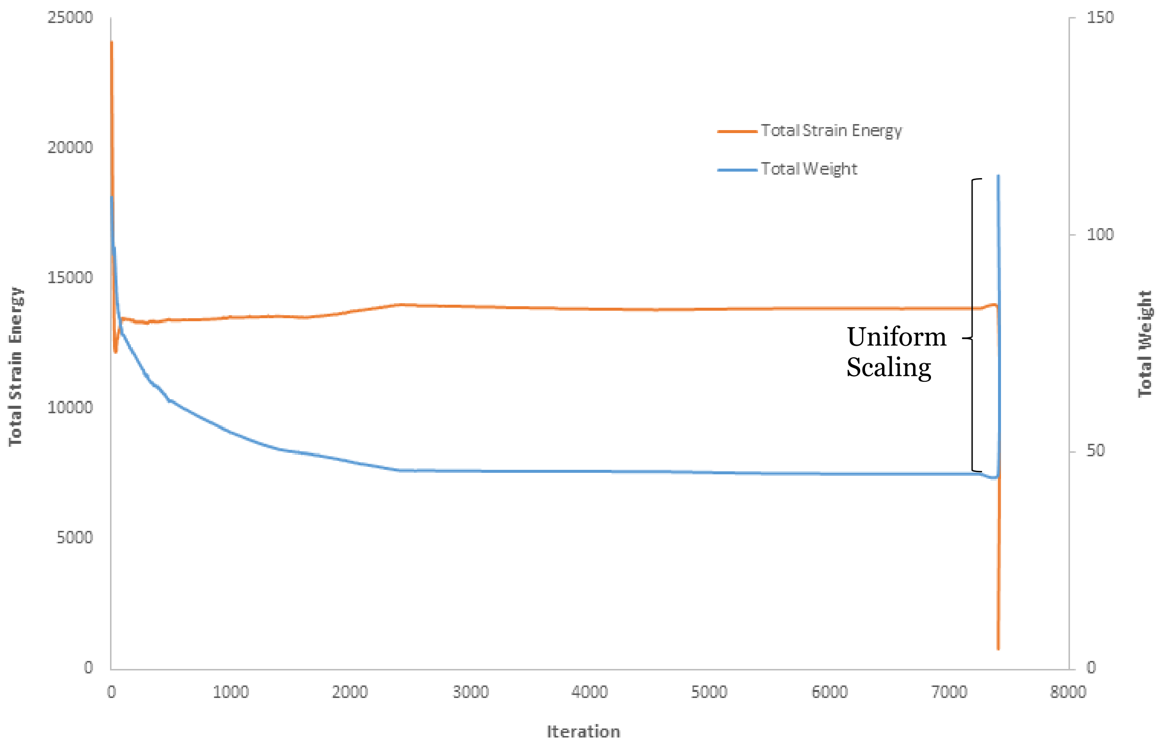


Figure 37. Convergence history for the 25-bar space truss

Table 22. Optimal cross-sectional areas and nodal coordinates for the 25-bar space truss

Design variable	Hansen et al. [36]	Rahami et al. [18]	Gholizadeh [24]	Ahrari et al. [22]	Proposed Method
A_1 (in^2)	0.010	0.1	0.1	-	0.01252
A_2 (in^2)	0.487	0.1	0.1	0.1	0.09893
A_6 (in^2)	0.836	0.9	1.0	0.9	0.93692
A_{10} (in^2)	0.021	0.1	0.1	-	0.00416
A_{12} (in^2)	0.123	0.1	0.1	-	0.03231
A_{14} (in^2)	0.084	0.1	0.1	0.1	0.00416
A_{18} (in^2)	0.698	0.2	0.1	0.1	0.16649
A_{22} (in^2)	0.548	0.9	0.9	1.0	1.01115
X_4 (in)	23.7	32.9609	36.9520	38.8713	35.960
Y_4 (in)	49.3	53.6141	54.5786	61.5207	49.090
Z_4 (in)	97.7	129.8648	129.9758	119.1785	130.000
X_8 (in)	27.5	43.6204	51.7317	49.4146	48.7777
Y_8 (in)	96.4	137.2674	139.5316	137.9423	139.590
Weight (lb)	128.3	120.1149	117.227	114.417	113.789
Max. stress constraint ratio	-	-	-	0.4490	0.3268
No. structural analyses	7	10000	4500	10000	7409

Maximum displacement = 0.35000

Table 23. The effect of control parameters α and β on convergence for the 25-bar space truss

α	β	Weight (lb)	No. structural analyses
0.90	0.010	155.762	11019
0.90	0.100	151.456	10519
0.95	0.010	147.505	11492
0.95	0.100	139.406	13094
0.99	0.001	**113.789**	**7409**
0.99	0.010	122.732	11142

5.5.5. Test Problem 5: 77-bar, 40-node truss bridge

This test problem involves a 77-member, 40-node bridge structure shown in Figure 38. The bridge structure spans a distance of 6000 in and has 21 panel joints equally spaced at 300 in. The applied forces include self-weight and loads given in Table 24. Flager et al. [126] solved this problem using a single geometry variable, which is the y-coordinate of the top joints of the truss, and discrete sizing variables chosen from a list of W-shape profiles. The buckling constraints were based on the AISC-ASD standard. In this study, the proposed algorithm is based on the fully stressed approach, making it unsuitable for tackling discrete standard profiles. Therefore, we solve the problem using continuous sizing variables and compute the Euler buckling allowable stress in the same way as in test problem 3. Material properties and problem parameters are also given in Table 24, with a lower bound of 1.0 in² set for cross-sectional areas.

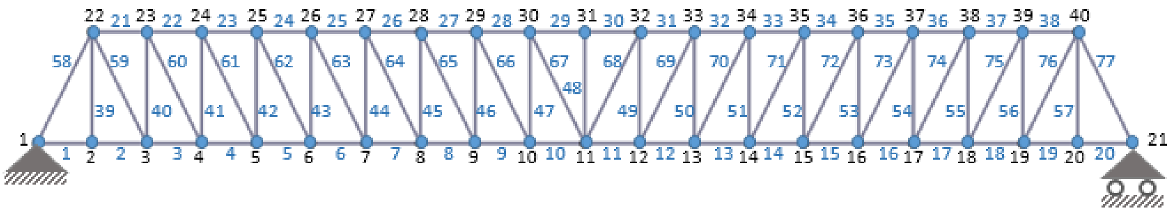


Figure 38. 77-bar, 40-node truss bridge

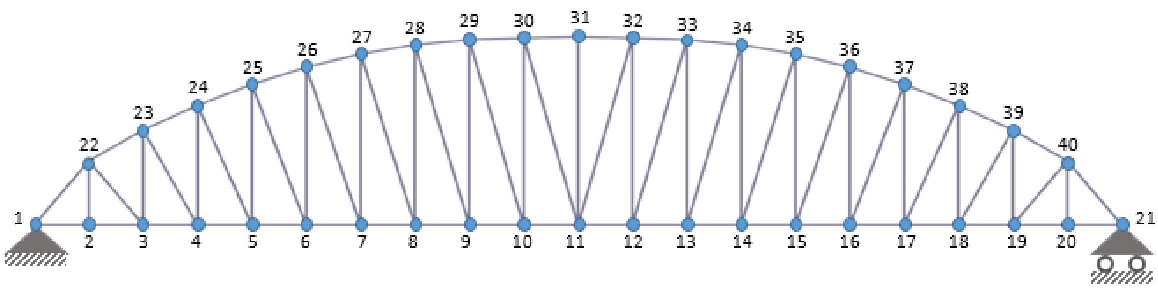


Figure 39. Optimized geometry for the 77-bar space truss

Table 24. Problem data for the 77-bar space truss

Design variables		
Size variables: A_j; j = 1, 2, ..., 77		
Geometry variables: y_{22}; y_{23}; y_{24}; y_{25}; y_{26}; y_{27}; y_{28}; y_{29}; y_{30}; y_{31}		
Behaviour constraints		
Stress constraints: $(\sigma_t)_j \le 21.6$ ksi; j = 1,..., 77; $\lvert(\sigma_c)_j\rvert$ varies; j = 1,..., 77;		
Displacement constraint in all direction of the coordinate system: $\lvert\Delta_i\rvert \le 10.0$ in; i = 1,..., 40		
Euler buckling Stress constraints: $\lvert(\sigma_c)_j\rvert \le 4E_jA_j/L_j^2$; j = 1,..., 77		
Side constraints:		
Cross-sectional areas: $1.0 \le A_j \le 250$ (in^2); j = 1, ..., 77		
Nodal coordinates: 0.0 in $\le y_i \le$ 1000 in; i =22,..., 40		
Loading data		
Node	**F_x**	**F_y**
i= 2,..., 20	0.0	-0.6 kips
Material properties		
Modulus of elasticity E = 29000 ksi		
Yield stress = 36 ksi		
Density of the material ρ = 0.1 lb/in^3		
Control Parameters		
$\alpha_1 = 0.95$		
$\beta = 10^{-2}$		
Termination tolerances		
$\varepsilon = 10^{-5}$, $\varepsilon_W = 10^{-5}$		

The proposed method has generated the optimal layout, cross-sectional areas, and nodal coordinates for the bridge structure, which are presented in Figure 39 and Table 25, respectively.

In their study, Flager et al. [126] obtained an optimal overall weight of 511,037 lb after performing 7553 structural analyses. In contrast, the proposed method achieves a weight of 320038 lb which is considerably smaller than the overall weight reported in [126], despite addressing the buckling constraints differently. Additionally, the proposed algorithm reaches the optimal solution in just 5707 structural analyses without violating any stress or displacement constraints.

Table 26 demonstrates how the control parameters α and β affect the algorithm convergence, while Figure 40 depicts the convergence history of the proposed method for both the total strain energy and total weight from the initial design.

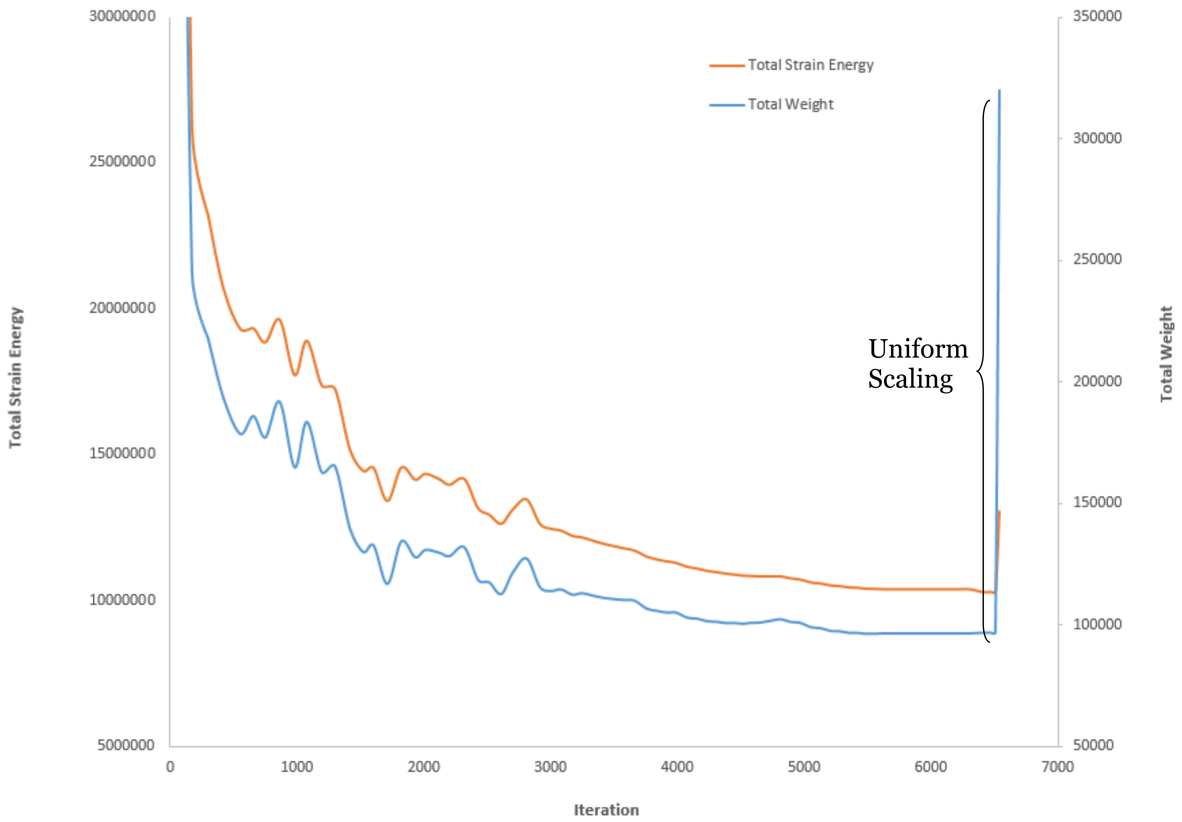


Figure 40. Convergence history for the 77-bar space truss

Table 25. Optimal cross-sectional areas and nodal coordinates for the 77-bar space truss

Design variable	Flager et al. [15]	Proposed Method
Geometry variables (in)		
Y_{22}	608.49	332.82
Y_{23}	608.49	492.48
Y_{24}	608.49	621.01
Y_{25}	608.49	729.20
Y_{26}	608.49	818.24
Y_{27}	608.49	886.96
Y_{28}	608.49	934.54
Y_{29}	608.49	961.56
Y_{30}	608.49	970.74
Y_{31}	608.49	976.81

Size variables (in^2)		
A_1	20.0	144.840
A_2	20.0	144.840
A_3	38.8	185.800
A_4	56.8	209.130
A_5	75.6	223.600
A_6	83.3	233.940
A_7	101.0	241.620
A_8	109.0	248.240
A_9	117.0	254.440
A_{10}	125.0	259.880
A_{21}	42.7	125.980
A_{22}	61.8	136.170
A_{23}	75.6	142.420
A_{24}	89.6	146.110
A_{25}	101	148.680
A_{26}	109.0	150.820
A_{27}	117.0	153.290
A_{28}	125.0	156.000
A_{29}	125	156.540
A_{39}	14.6	16.822
A_{40}	74.0	64.567
A_{41}	67.7	70.234
A_{42}	55.8	64.237
A_{43}	44.7	56.953
A_{44}	39.9	42.977
A_{45}	29.4	36.449
A_{46}	25.9	35.833
A_{47}	-	15.763
A_{48}	-	10.625
A_{58}	24.0	162.670
A_{59}	14.6	60.991
A_{60}	117.0	44.596
A_{61}	44.7	33.029
A_{62}	39.9	26.945
A_{63}	35.3	22.105

A_{64}	29.1	20.448
A_{65}	23.2	20.107
A_{66}	17.9	18.096
A_{67}	14.6	3.245
Weight (lb)	**511,037**	**320,038**
Violated Constraints	**0**	**0**
No. structural analyses	**7553**	**5707**

Maximum displacement = 10.000 in

Table 26. The effect of control parameters α and β on convergence for the 77-bar space truss

α	**β**	**Weight (lb)**	**No. structural analyses**
0.90	0.010	*	-
0.95	0.010	**320,038**	**5707**
0.95	0.100	*	-

(*) indicates a very slow convergence

5.5.6. Test Problem 6: 258-bar, 60-node space truss bridge

This test problem addresses the 258-member, 60-node bridge structure shown in Figure 41, which has been solved by Decamps et al. [135]. The bridge spans a distance of 14 m and consists of 1 x 1 x 1 equally spaced modules. The supports are located at nodes 1, 2, 29 and 30, and are blocked in all three directions. The forces applied to this structure involve downward loads of magnitude 1 applied to the lower nodes of the structure. Additionally, a lateral horizontal force of 0.2 is applied at each of the lower nodes, on one side of the bridge and perpendicular to its main axis. The design variables include the cross-sectional area of the 258 members, and the x, y and z coordinates of the upper nodes. These nodes are allowed to move within a bounding box of 2 x 2 x 10 meters around each node's original position. The allowable stress and the Young modulus also have a value of 1. Control parameters and the termination tolerances for this problem, are given in table 27.

Table 27. Control parameters and termination tolerances for the 258-bar space truss

Control Parameters
$\alpha_1 = 0.995$
$\beta = 10^{-5}$
Termination tolerances
$\varepsilon = 10^{-6}$, $\varepsilon_W = 10^{-6}$

The optimal layout as well as nodal coordinates and optimal cross-sectional areas obtained using the proposed method are shown in Figure 42, Table 28 and Table 29, respectively. Decamps et al. [135] reported an optimal volume of 408.807, which was obtained after 1414 iterations. The proposed method achieves an optimal volume of 389.53 which is about 5% smaller than the overall volume reported in [135]. Moreover, the proposed algorithm required 65432 structural analyses to converge to the optimal solution without violation of any stress. The effect of algorithmic parameters α and β on the convergence of the algorithm are presented in Table 30. The convergence history of the proposed method for both the total strain energy and the total weight from the initial design are shown in Figure 43.

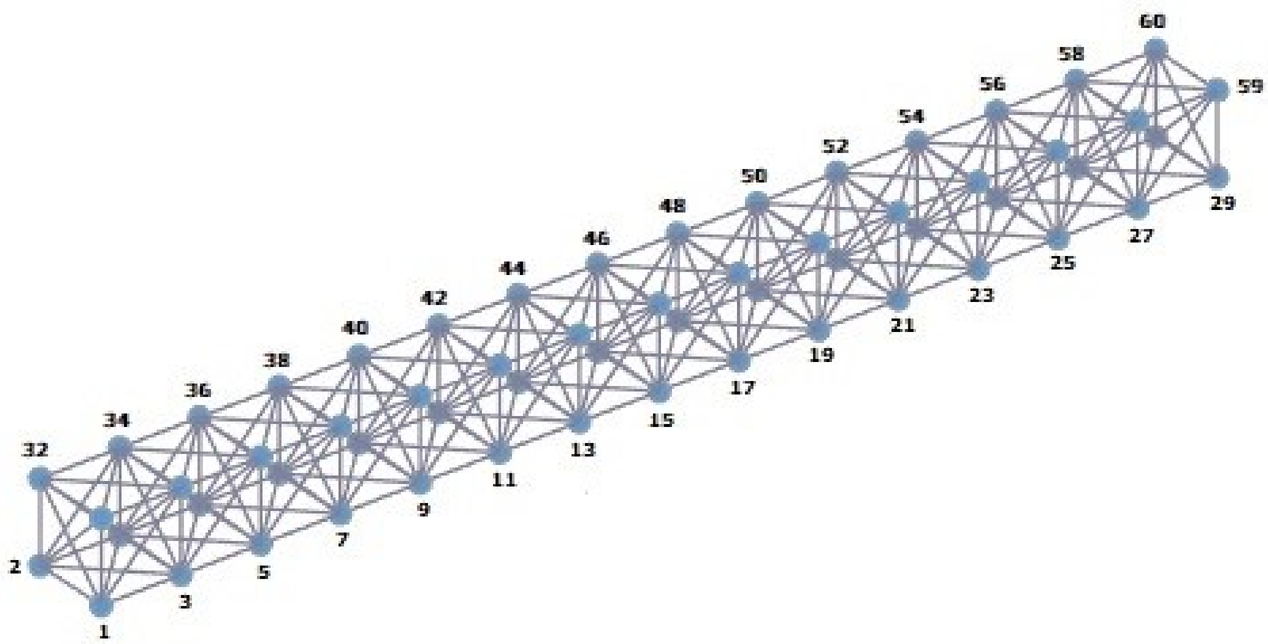
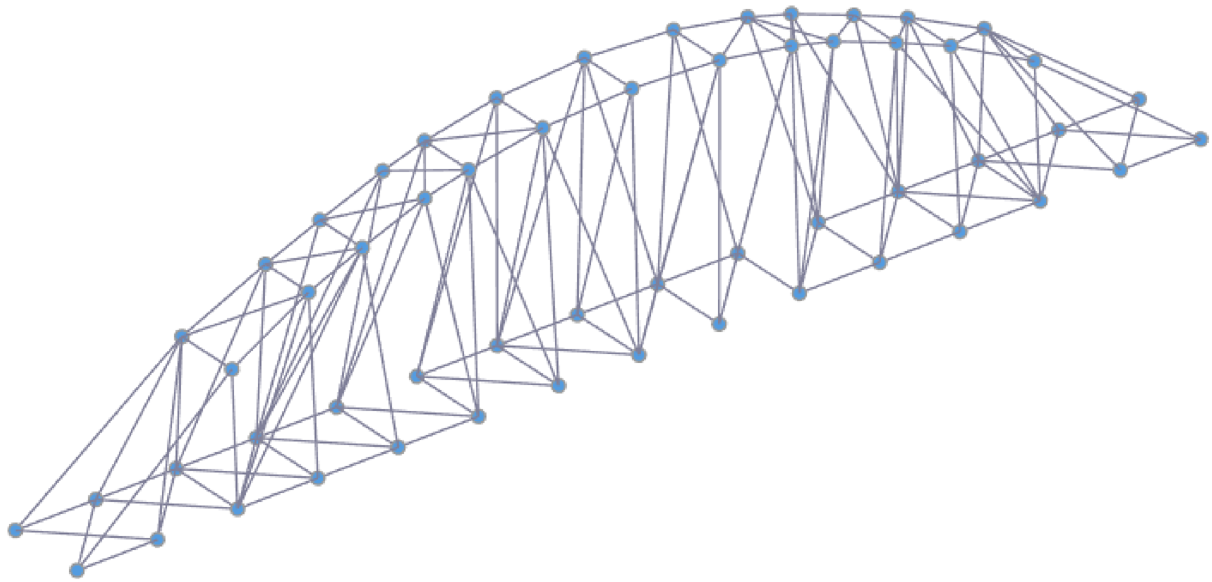
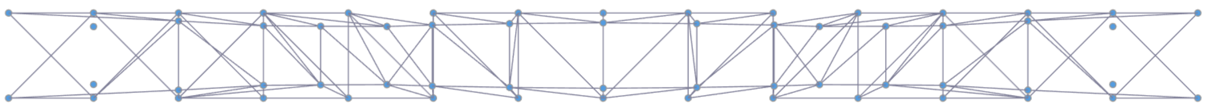


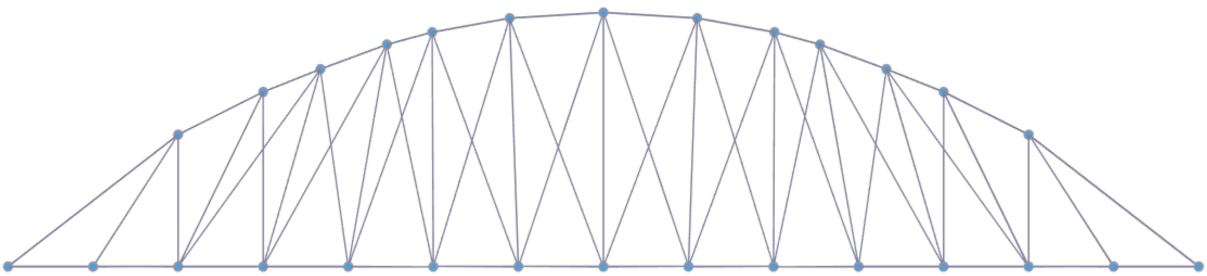
Figure 41. 258-bar, 60-node truss bridge



(a) 3D view of the optimized truss.



(b) Top view of the optimized truss.



(c) Front view of the optimized truss.

Figure 42. Optimized geometry for the 258-bar space truss

Table 28. Optimal cross-sectional areas for the 258-bar space truss

Member no.	End Node 1	End Node 2	A_j	σ_j	Member no.	End Node 1	End Node 2	A_j	σ_j
1	1	2	0.0050	0.0000	45	8	9	0.3311	-1.0000
2	1	3	1.4412	1.0000	46	8	10	0.1831	1.0000
3	1	4	0.7187	1.0000	47	8	36	0.5870	1.0000
4	1	31	0.0050	0.0226	48	8	37	0.1938	1.0000
5	1	32	0.0050	0.0127	49	8	38	0.1771	1.0000
6	1	33	10.6170	-1.0000	50	8	40	0.0792	1.0000
7	2	3	1.1198	-1.0000	51	9	10	0.0314	-1.0000
8	2	4	0.7057	-1.0000	52	9	11	0.1273	-1.0000
9	2	31	0.0050	-0.1610	53	9	12	0.0050	-0.5000
10	2	32	0.0050	-0.1721	54	9	37	1.0076	1.0000
11	2	34	10.6140	-1.0000	55	9	39	0.0050	0.4309
12	3	4	0.0050	0.0000	56	9	40	0.0050	0.2382
13	3	5	0.0050	-0.0021	57	9	41	0.0050	0.2511
14	3	6	0.0089	0.9990	58	10	11	0.0941	-1.0000
15	3	31	0.0050	0.2532	59	10	12	0.0050	0.0000
16	3	33	0.0050	0.0393	60	10	38	0.0050	0.6527
17	3	34	1.3281	1.0000	61	10	39	0.1666	1.0000
18	3	35	0.0050	-0.8576	62	10	40	0.4320	1.0000
19	4	5	0.8039	-1.0000	63	10	42	0.4419	1.0000
20	4	6	0.2736	-1.0000	64	11	12	0.1964	-1.0000
21	4	32	0.0050	0.2927	65	11	13	0.0050	0.0416
22	4	33	0.0050	0.0318	66	11	14	0.0050	0.5208
23	4	34	1.1878	1.0000	67	11	39	0.9618	1.0000
24	4	36	0.0050	0.4140	68	11	41	0.0536	1.0000
25	5	6	0.0357	1.0001	69	11	42	0.0050	0.7067
26	5	7	0.0698	0.9999	70	11	43	0.0050	0.7029
27	5	8	0.3984	1.0000	71	12	13	0.1133	-1.0000
28	5	33	2.2934	1.0000	72	12	14	0.0471	1.0000
29	5	35	0.0443	-1.0000	73	12	40	0.0050	0.6246
30	5	36	0.0050	-0.1587	74	12	41	0.9174	1.0000
31	5	37	1.5418	-1.0000	75	12	42	0.0050	0.8674
32	6	7	0.1465	-1.0000	76	12	44	0.1222	1.0000
33	6	8	0.1940	-1.0000	77	13	14	0.1667	-1.0000
34	6	34	0.9408	1.0000	78	13	15	0.0050	-0.0393
35	6	35	0.0050	0.3946	79	13	16	0.0050	0.4803
36	6	36	0.0630	1.0000	80	13	41	0.1523	1.0000
37	6	38	0.0050	0.6598	81	13	43	0.8514	1.0000
38	7	8	0.1699	-1.0000	82	13	44	0.0050	0.8388
39	7	9	0.0342	-1.0001	83	13	45	0.0050	0.6581
40	7	10	0.0050	-0.0001	84	14	15	0.0388	-1.0000
41	7	35	1.0002	1.0000	85	14	16	0.1014	1.0000
42	7	37	0.0050	0.4206	86	14	42	0.0050	0.6622
43	7	38	0.0050	0.2712	87	14	43	0.6086	1.0000
44	7	39	0.0050	-0.1807	88	14	44	0.3849	1.0000

Table 28. Optimal cross-sectional areas for the 258-bar space truss (cont'd).

Member no.	End Node 1	End Node 2	A_j	σ_j	Member no.	End Node 1	End Node 2	A_j	σ_j
89	14	46	0.0317	1.0001	133	21	51	0.0050	0.4309
90	15	16	0.1850	-1.0000	134	21	52	0.0050	0.2382
91	15	17	0.0050	-0.0393	135	21	53	1.0076	1.0000
92	15	18	0.0388	-1.0000	136	22	23	0.0050	-0.0001
93	15	43	0.0424	1.0001	137	22	24	0.1831	1.0000
94	15	45	0.9174	1.0000	138	22	50	0.4419	1.0000
95	15	46	0.0050	0.8412	139	22	51	0.1666	1.0000
96	15	47	0.0424	1.0001	140	22	52	0.4320	1.0000
97	16	17	0.0050	0.4803	141	22	54	0.0050	0.6527
98	16	18	0.1014	1.0000	142	23	24	0.1699	-1.0000
99	16	44	0.0050	0.7770	143	23	25	0.0698	0.9999
100	16	45	0.5786	1.0000	144	23	26	0.1465	-1.0000
101	16	46	0.4383	1.0000	145	23	51	0.0050	-0.1807
102	16	48	0.0050	0.7770	146	23	53	0.0050	0.4206
103	17	18	0.1667	-1.0000	147	23	54	0.0050	0.2712
104	17	19	0.0050	0.0416	148	23	55	1.0002	1.0000
105	17	20	0.1133	-1.0000	149	24	25	0.3984	1.0000
106	17	45	0.0050	0.6581	150	24	26	0.1940	-1.0000
107	17	47	0.8514	1.0000	151	24	52	0.0792	1.0000
108	17	48	0.0050	0.8388	152	24	53	0.1938	1.0000
109	17	49	0.1523	1.0000	153	24	54	0.1771	1.0000
110	18	19	0.0050	0.5208	154	24	56	0.5870	1.0000
111	18	20	0.0471	1.0000	155	25	26	0.0357	1.0001
112	18	46	0.0317	1.0001	156	25	27	0.0050	-0.0021
113	18	47	0.6086	1.0000	157	25	28	0.8039	-1.0000
114	18	48	0.3849	1.0000	158	25	53	1.5418	-1.0000
115	18	50	0.0050	0.6622	159	25	55	0.0443	-1.0000
116	19	20	0.1964	-1.0000	160	25	56	0.0050	-0.1587
117	19	21	0.1273	-1.0000	161	25	57	2.2934	1.0000
118	19	22	0.0941	-1.0000	162	26	27	0.0089	0.9990
119	19	47	0.0050	0.7029	163	26	28	0.2736	-1.0000
120	19	49	0.0536	1.0000	164	26	54	0.0050	0.6598
121	19	50	0.0050	0.7067	165	26	55	0.0050	0.3946
122	19	51	0.9618	1.0000	166	26	56	0.0630	1.0000
123	20	21	0.0050	-0.5000	167	26	58	0.9408	1.0000
124	20	22	0.0050	0.0000	168	27	28	0.0050	0.0000
125	20	48	0.1222	1.0000	169	27	29	1.4412	1.0000
126	20	49	0.9174	1.0000	170	27	30	1.1198	-1.0000
127	20	50	0.0050	0.8674	171	27	55	0.0050	-0.8576
128	20	52	0.0050	0.6246	172	27	57	0.0050	0.0393
129	21	22	0.0314	-1.0000	173	27	58	1.3281	1.0000
130	21	23	0.0342	-1.0001	174	27	59	0.0050	0.2532
131	21	24	0.3311	-1.0000	175	28	29	0.7187	1.0000
132	21	49	0.0050	0.2511	176	28	30	0.7057	-1.0000

Table 28. Optimal cross-sectional areas for the 258-bar space truss (cont'd).

Member no.	End Node 1	End Node 2	A_j	σ_j	Member no.	End Node 1	End Node 2	A_j	σ_j
177	28	56	0.0050	0.4140	221	44	45	0.0050	-0.0685
178	28	57	0.0050	0.0318	222	44	46	4.1682	-1.0000
179	28	58	1.1878	1.0000	223	45	46	0.0988	1.0000
180	28	60	0.0050	0.2927	224	45	47	12.2780	-1.0000
181	29	30	0.0050	0.0000	225	45	48	0.0050	-0.0685
182	29	57	10.6170	-1.0000	226	46	47	0.0050	-0.6469
183	29	59	0.0050	0.0226	227	46	48	4.1682	-1.0000
184	29	60	0.0050	0.0127	228	47	48	0.0571	1.0000
185	30	58	10.6140	-1.0000	229	47	49	12.2900	-1.0000
186	30	59	0.0050	-0.1610	230	47	50	0.1344	-1.0000
187	30	60	0.0050	-0.1721	231	48	49	0.0050	0.5420
188	31	32	0.0050	0.0142	232	48	50	4.2528	-1.0000
189	31	33	0.0050	-0.1454	233	49	50	0.0192	1.0000
190	31	34	0.0050	0.0288	234	49	51	12.3680	-1.0000
191	32	33	0.0050	-0.1505	235	49	52	0.1698	-1.0000
192	32	34	0.0050	0.0150	236	50	51	0.1914	1.0000
193	33	34	0.2216	1.0000	237	50	52	4.7167	-1.0000
194	33	35	9.3952	-1.0000	238	51	52	0.0050	0.2519
195	33	36	0.0050	0.2479	239	51	53	11.7370	-1.0000
196	34	35	1.3116	-1.0000	240	51	54	0.8811	-1.0000
197	34	36	6.8594	-1.0000	241	52	53	0.0050	0.3151
198	35	36	0.3997	1.0000	242	52	54	5.0798	-1.0000
199	35	37	10.1010	-1.0000	243	53	54	0.5438	1.0000
200	35	38	0.0050	0.9148	244	53	55	10.1010	-1.0000
201	36	37	0.9494	-1.0000	245	53	56	0.9494	-1.0000
202	36	38	5.8744	-1.0000	246	54	55	0.0050	0.9148
203	37	38	0.5438	1.0000	247	54	56	5.8744	-1.0000
204	37	39	11.7370	-1.0000	248	55	56	0.3997	1.0000
205	37	40	0.0050	0.3151	249	55	57	9.3953	-1.0000
206	38	39	0.8811	-1.0000	250	55	58	1.3116	-1.0000
207	38	40	5.0798	-1.0000	251	56	57	0.0050	0.2479
208	39	40	0.0050	0.2519	252	56	58	6.8594	-1.0000
209	39	41	12.3680	-1.0000	253	57	58	0.2216	1.0000
210	39	42	0.1914	1.0000	254	57	59	0.0050	-0.1454
211	40	41	0.1698	-1.0000	255	57	60	0.0050	-0.1505
212	40	42	4.7167	-1.0000	256	58	59	0.0050	0.0288
213	41	42	0.0192	1.0000	257	58	60	0.0050	0.0150
214	41	43	12.2900	-1.0000	258	59	60	0.0050	0.0142
215	41	44	0.0050	0.5420					
216	42	43	0.1344	-1.0000					
217	42	44	4.2528	-1.0000					
218	43	44	0.0571	1.0000					
219	43	45	12.2780	-1.0000					
220	43	46	0.0050	-0.6469					

Table 29. Optimal nodal coordinates for the 258-bar space truss

Node no.	x	y	z
33	2.0000	0.0934	1.5504
34	2.0000	0.9067	1.5504
35	3.0000	0.1501	2.0524
36	3.0000	0.8499	2.0524
37	3.6731	0.1555	2.3209
38	3.6731	0.8445	2.3209
39	4.4539	0.1586	2.6097
40	4.4539	0.8414	2.6097
41	4.9876	0.1419	2.7547
42	4.9876	0.8581	2.7547
43	5.8976	0.1252	2.9200
44	5.8976	0.8748	2.9200
45	7.0000	0.1150	2.9864
46	7.0000	0.8850	2.9864
47	8.1024	0.1252	2.9200
48	8.1024	0.8748	2.9200
49	9.0124	0.1419	2.7547
50	9.0124	0.8581	2.7547
51	9.5461	0.1586	2.6097
52	9.5461	0.8414	2.6097
53	10.3270	0.1555	2.3209
54	10.3270	0.8445	2.3209
55	11.0000	0.1501	2.0524
56	11.0000	0.8499	2.0524
57	12.0000	0.0934	1.5504
58	12.0000	0.9067	1.5504

*Nodes 31, 32, 59, 60 are not presented because all the bars connected to them have zero cross-sectional area.

Table 30. The effect of algorithmic parameters α and β on convergence for the 258-bar space truss

α	β	Optimal Volume	No. structural analyses
0.995	10^{-3}	*	Diverge
0.995	10^{-5}	**389.53**	**65432**
0.999	10^{-5}	*	-

(*) indicates a very slow convergence

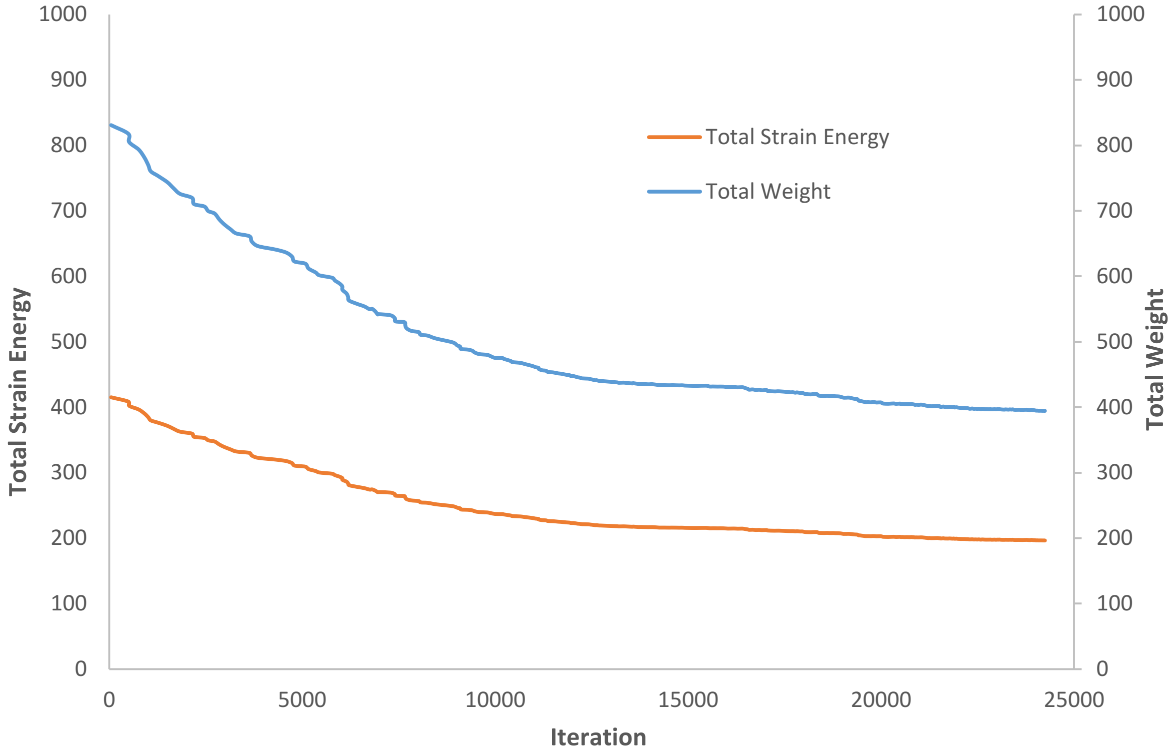


Figure 43. Convergence history for the 258-bar space truss

5.6 Conclusion

The problem of minimizing the weight of truss structures under stress and displacement constraints through simultaneous optimization of sizing, topology, and layout has been discussed in this Chapter. The proposed approach is a bi-level algorithm that uses an unconventional non-uniform cellular automata method, with an alternating procedure to couple sizing and layout during the optimization process.

To evaluate the proposed algorithm, several benchmark problems were tested. The results demonstrate that the proposed algorithm outperforms other methods in terms of both efficiency and accuracy. The superiority of the proposed algorithm can be attributed to its ability to handle complex problems and its capability to obtain optimal solutions with fewer structural analyses compared to other optimization methods.

Chapter 6
Contribution, Conclusions, and Future work

6.1 Contributions

This research study proposes a non-uniform cellular automata (CA) framework for minimizing the weight of truss structures, taking into account stress, displacement, and buckling constraints. CA is a type of computational model that consists of a grid of cells, each of which can have a finite number of states, and a set of rules that determine how the cells change their states over time. Unlike conventional CA, which relies on identical cells, the non-uniform CA framework proposed in this dissertation is designed to handle different types of cells with different properties and behaviors, allowing for more flexibility in the optimization process. The proposed non-uniform CA framework was first formulated and implemneted for the topology and sizing optimization of truss structures and was later extended to solve the simulanaous sizing, topology and layout optimization problem by adding the nodal coordinates to the vector of design variables.

The main contributions of this reseach dissertation may eb asuumarzied as:

- The development of a non-uniform cellular automata framework for topology and sizing and layout optimization of truss structures that takes into account stress, displacement, and buckling constraints.
- The implementation of a new bi-level approach for the solution of the layout optimization problem, where an alternating procedure was used to couple sizing and layout during the optimization process, thus improving the quality of the optimized designs. The FSD approach was effectly used to update the sizing variables, while the layout variables were updated based on an optimality criterion based on the the strain energy distribution within the structure, and using the standard steepest descent method.
- The application of the proposed framework on different 2-D and 3-D truss structures, including relatively large structures, which shows its versatility and potential for various engineering problems.

- The efficiency and performance of the developed algorithm based on the proposed non-uniform cellular automata (CA) framework were demonstrated through several benchmark test problems. Results unanimously showed the superior performance of the proposed non-uniform CA in terms of both efficiency and accuracy compared with other optimization algorithms.

In conclusion, non-uniform cellular automata is a powerful tool for optimizing truss structures. By exploring a wide range of design possibilities and simulating the behavior of individual cells, non-uniform CA can help engineers and designers to identify the most efficient and cost-effective truss structure for a given set of performance requirements. As non-uniform CA continues to evolve and improve, it is likely that it will play an increasingly important role in the design of complex truss structures.

6.2 Major Conclusions

Here are some of the major conclusions from this research study:

- It was shown that the developed design optimization methodlogy based on the non-uniform cellular automata paradime can be effectively used for sizing, topology and layout design optimization of plane and space truss structures subjected to displacement, stress and buckling constraints. Through several benchmark problmes, it was demonstrated that the proposed methodology can generate better optimal designs in terms of both efficency and accuracycompared with other traditional optimization methods.
- In layout optimization problems, the performance of the non uniform CA algorithm is sensitive to the selection of control parameters. These parameters thus need to be carefully fine tuned for convergence.
- The proposed approach can identify truss configurations that are not intuitive and would likely not be identified through manual design exploration.

Overall, we conclude that non-uniform cellular automata can be an effective and efficient tool for the optimization of truss structures, enabling designers to identify novel solutions that are both material-efficient and structurally sound. However, it is important to consider that it may have some limitations when applying this approach to practical design problems.

6.3 Recommendation for the future works

While this research has initiated a major leap toward implementation of CA in structural design optimization, there are some potential recommendations for future work regarding the use of non-uniform cellular automata:

- Optimization of real-world truss structures: Most test problems in this study are idealized truss structures. Future work could focus on the optimization of real-world truss structures, such as those used in bridges and cranes.
- Investigation of alternative objectives: While we have focused on minimizing the weight of the structure, future work could explore alternative objectives such as maximizing the strength or increasing the load carrying capacity. .
- Investigation of multi-objective optimization: Explore the optimization of truss structures for multiple objectives, where the algorithm seeks to find a balance between these competing objectives such as strength, stability, and cost-effectiveness.
- Refining the algorithm to effectively address optimization problems involving discrete cross-sections.
- Incorporation of more design constraints: Investigate the inclusion of additional design constraints such as deflection, vibration, and global buckling into the non-uniform CA optimization process.
- Incorporation of real-world constraints: In real-world engineering applications, there are often practical constraints that must be considered, such as material availability, manufacturing limitations, and construction constraints. Future work could explore the incorporation of these types of constraints into the non-uniform CA optimization process, and investigate how they might affect the resulting truss designs.
- Investigation of hybrid optimization techniques: Explore the possibility of integrating the proposed framework with other optimization algorithms, to improve the efficiency and accuracy of truss optimization.
- Extension to other types of structures: While this non-uniform CA has been developed specifically for truss structures, future work could explore the applicability of the non-uniform CA to other types of structural systems, such as frames or shells or in general continuum structures.

- Investigation of uncertainty and variability: Investigate the effect of uncertainty and variability in the design parameters on the performance of the non-uniform CA approach.
- Investigation of the impact of initial design: Investigate how the initial design of the truss structure affects the performance of the non-uniform CA approach.
- Exploration of parallel computing techniques: The computational complexity of the non-uniform CA optimization can be a limiting factor. Future work could explore the use of parallel computing techniques to improve the efficiency of the optimization process.
- Integration of machine learning: Explore the integration of machine learning techniques into the non-uniform CA optimization process. This could involve using machine learning to learn from previous designs and generate designs that are more efficient in the future.
- Validation and comparison with newer methods: While non-uniform CA has shown promising results in truss optimization, it is important to validate its effectiveness and efficiency against newer methods such as deep learning and reinforcement learning.

References

[1] A. Kaveh and S. Talatahari, Principles of structural optimization. Springer Science & Business Media, 2010.

[2] X. Tao, Y. Mao, and H. Liu, Structural optimization: Theory, methods, and applications. Springer, 2017.

[3] Y.-C. Chang and P.-H. Wen, “A review of heuristics for structural optimization,” Engineering with Computers, vol. 32, no. 3, pp. 331-346, 2016.

[4] P.-H. Wen and Y.-C. Chang, “Particle swarm optimization for truss structure optimization,” Journal of Ambient Intelligence and Humanized Computing, vol. 9, no. 3, pp. 203-212, 2018.

[5] Q. Liu and K. Y. Lam, Structural optimization: Techniques and Applications. John Wiley & Sons, 2019.

[6] M. P. Bendsøe, Topology optimization: theory, methods, and applications. Springer, 2003.

[7] O. Sigmund, “Material interpolation schemes in topology optimization,” Structural and multidisciplinary optimization, vol. 15, no. 2, pp. 120-133, 1998.

[8] P. Bouvry, Handbook of metaheuristics. Springer, 2011.

[9] D. E. Goldberg, Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, 1989.

[10] J. Kennedy and R. Eberhart, “Particle swarm optimization,” Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942-1948, 1995.

[11] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” Science, vol. 220, no. 4598, pp. 671-680, 1983.

[12] M. Dorigo and T. Stützle, Ant colony optimization. The MIT Press, 2004.

[13] J. von Neumann, Theory of self-reproducing automata. University of Illinois Press, 1966.

[14] J. Conway, “The game of life,” Scientific American, vol. 223, no. 4, pp. 4-23, 1970.

[15] U. Frisch, B. Hasslacher, and Y. Pomeau, “Lattice-gas automata for the Navier-Stokes equation,” Phys Rev Lett, vol. 56, no. 14, pp. 1505, 1986.

[16] J. Kari, “Theory of cellular automata: A survey,” Theoretical Computer Science, vol. 334, no. 1-3, pp. 3-33, 2005.

[17] S. Eubank, H. Guclu, V. A. Kumar, M. V. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang, “Modelling disease outbreaks in realistic urban social networks,” Nature, vol. 429, no. 6988, pp. 180-184, 2004.

[18] H. Krawinkler, "Optimization techniques in structural engineering," International Journal of Solids and Structures, vol. 39, no. 10, pp. 2827-2841, 2002.
[19] P. B. Luh and C. H. Chen, "Structural optimization using cellular automata," Journal of Structural Engineering, vol. 120, no. 7, pp. 2041-2058, 1994.
[20] K. Saitou, K. Izui, and T. Yamada, "Structural optimization by cellular automata," in Proceedings of the 1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, vol. 1, 1999, pp. 540-545.
[21] J. Luo and J. Yang, "A review on structural optimization using cellular automata," Structural and Multidisciplinary Optimization, vol. 54, no. 5, pp. 1253-1272, 2016.
[22] C. H. Chen and P. B. Luh, "Structural optimization by genetic algorithm and cellular automata," Computers & Structures, vol. 63, no. 6, pp. 1209-1226, 1997.
[23] S. Wolfram, Cellular automata and complexity: collected papers. Westview Press, 1994.
[24] A. Ilachinski, Cellular automata: a discrete view of the world. John Wiley & Sons, 2001.
[25] M. Mitchell, Complexity: a guided tour. Oxford University Press, 2012.
[26] E. Ahmed, A. A. Ali, M. A. Ahmed, and D. Y. Kim, "An Overview of Cellular Automata Framework for Optimization of Truss Structures," International Journal of Concrete Structures and Materials, vol. 14, no. 1, pp. 29, 2020.
[27] S. Sikder, S. Chakraborty, and S. Saha, "Optimization of Truss Structures Using Cellular Automata: A Comprehensive Review," Archives of Computational Methods in Engineering, vol. 25, no. 3, pp. 677-696, 2018.
[28] L. A. Schmit, "Structural design by systematic synthesis," Proc. 2nd Conference on Electronic Computation, ASCE, New York, 1960, pp. 105-122.
[29] W. Dorn, "Automatic design of optimal structures," J. de Mecanique, vol. 3, pp. 25-52, 1964.
[30] L. A. Schmit, "Structural synthesis. 1959-1969 A decade of progress," A paper presented at Japan-U.S. Seminar on Mafrix Methods of Structural Analysis and Design, 25-30 August 1969, Tokyo, Japan.
[31] R. T. Haftka, Z. Gürdal, and M. P. Kamat, Elements of Structural Optimization, 3rd ed. Dordrecht: Kluwer Academic Publishers, 1992.

[32] P. Pedersen, "On the optimal layout of multi-purpose trusses," Computers & Structures, vol. 2, no. 5-6, pp. 695-712, 1972.

[33] G. N. Vanderplaats and F. Moses, "Structural optimization by methods of feasible directions," Computers and Structures, vol. 3, pp. 739–755, 1973.
[34] H. Adeli and O. Kamal, "Efficient optimization of space trusses," Computers and Structures, vol. 24, pp. 501–511, 1986.
[35] Christensen, P. W., & Klarbring, A. (2009). An Introduction to Structural Optimization. Springer.
[36] S. R. Hansen and G. N. Vanderplaats, "Approximation method for configuration optimization of trusses," AIAA Journal, vol. 28, no. 1, pp. 161-168, 1990.
[37] N. L. Pedersen and A. K. Nielsen, "Optimization of practical trusses with constraints on eigenfrequencies, displacements, stresses, and buckling," Structural and Multidisciplinary Optimization, vol. 25, no. 5, pp. 436-445, 2003.
[38] L. Lamberti and C. Pappalettere, "Move limits definition in structural optimization with sequential linear programming. Parts I & II," Computers and Structures, vol. 81, pp. 197-238, 2003.
[39] Lamberti, L., & Pappalettere, C. (2004). Improved sequential linear programming formulation for structural weight minimization. Computer Methods in Applied Mechanics and Engineering, 193, 3493-3521.
[40] L. A. Schmit Jr. and B. Farshi, "Some approximation concepts for structural synthesis," AIAA Journal, vol. 12, no. 5, pp. 692–699, 1974.
[41] Jie, J. T. H., & Tabatabaei, S. S. (2016). Optimization of truss structures under uncertain loading using non-linear programming. Journal of Structural Engineering and Mechanics, 59(2), 259-272.
[42] Jie, J. T. H., Tabatabaei, S. S., & Akbari, T. (2020). Application of non-linear programming techniques for the optimization of truss structures with discrete variables. Journal of Constructional Steel Research, 168, 198-211.
[43] A. D. Belegundu and J. S. Arora, "A study of mathematical programming methods for structural optimization, part I: theory, part II: numerical results," International Journal for Numerical Methods in Engineering, vol. 21, pp. 1583–1623, 1985.
[44] B. H. V. Topping, "Shape optimization of skeletal structures: a review," Journal of Structural Engineering, vol. 109, no. 8, pp. 1933-1951, 1983.

[45] W. Prager and J.E. Taylor, "Problems of optimal structural design," Journal of Applied Mechanics, vol. 90, no. 1, pp. 102-106, 1968.

[46] W. Prager and R.T. Shield, "Optimal design of multi-purpose structures," International Journal of Solids and Structures, vol. 4, no. 5, pp. 469-475, 1968.
[47] V.B. Venkayya, "Design of optimum structures," presented at the Conference on Computer Oriented Analysis of Shell Structures, Palo Alto, California, August 1970.
[48] Venkaya, V.B., "Application of Optimality Criteria Approaches to Automated Design of Large Practical Structures," AGARD, 1973. 12. Kiusalaas, J., "Minimum Weight Design of Structures via Optimality Criteria," NASA TN D-7115, 1972
[49] Kiusalaas, J., "Minimum Weight Design of Structures via Optimality Criteria," NASA TN D-7115, 1972.
[50] Allwood, R. J., & Chung, Y. S. (1984). Minimum-weight design of trusses by an optimality criteria method. International Journal for Numerical Methods in Engineering, 20, 697-713.
[51] Patnaik, S. N., Guptill, J. D., & Berke, L. (1995). Merits and limitations of optimality criteria method for structural optimization. International Journal for Numerical Methods in Engineering, 38, 3087-3120.
[52] S. N. Patnaik, R. M. Coroneos, J. D. Guptill, and D. A. Hopkins, "Comparative evaluation of different optimization algorithms for structural design application," Int. J. Numer. Meth. Eng., vol. 39, pp. 1661-1674, 1996.
[53] Ananthanarayanan, N. K., & Ramakrishnan, G. (2001). Modified optimality criteria method for structural optimization of trusses. Structural and Multidisciplinary Optimization, 23(3), 197-203.
[54] Li, G., Li, M., Liang, X., & Zhang, W. (2018). Design optimization of truss structures based on improved optimality criteria method. Advances in Mechanical Engineering, 10(3), 1-11.
[55] Teng, Y., & Qin, Q. H. (2019). Multi-objective optimization of truss structures using an improved bi-objective optimality criteria algorithm. International Journal of Structural Stability and Dynamics, 19(9), 1950133.
[56] M. P. Bendsoe and O. Sigmund, "Topology optimization: theory, methods, and applications," Springer, 2004.
[57] A. Kaveh and M. Ilchi Ghazaan, "Truss sizing optimization using a convex programming technique," Structural and Multidisciplinary Optimization, vol. 33, no. 1, pp. 45-52, 2007.

[58] A. H. Gandomi and X. S. Yang, "Convexity of structural optimization problems: A review," Structural and Multidisciplinary Optimization, vol. 47, no. 5, pp. 693-717, 2013.
[59] C. Fleury, "CONLIN: An efficient dual optimizer based on convex approximation concept," Structural Optimization, vol. 1, pp. 81-89, 1989.
[60] K. Svanberg, "Optimization of geometry in truss design," Computer Methods in Applied Mechanics and Engineering, vol. 28, no. 1, pp. 63-80, 1981.
[61] W. H. Zhang, M. Domaszewski, and C. Fleury, "A new mixed convex approximation method with applications for truss configuration optimization," Structural Optimization, vol. 15, no. 3, pp. 237-241, 1998.
[62] Y. Zhang and Y. Li, "An improved semidefinite programming relaxation for truss topology optimization," Structural and Multidisciplinary Optimization, vol. 43, no. 5, pp. 599-610, 2011.
[63] Y. Pan and J. Wang, "Truss topology optimization with discrete design variables using convex relaxation methods," Structural and Multidisciplinary Optimization, vol. 57, no. 1, pp. 221-235, 2018.
[64] H. Wang, Y. Zhu, and Y. M. Xie, "A convex relaxation method for truss topology optimization with discrete variables," Computers & Structures, vol. 204, pp. 34-46, 2018.
[65] K. Deb and S. Gulati, "Design of truss-structures for minimum weight using genetic algorithms," Finite Elements in Analysis and Design, vol. 37, pp. 447-465, 2001.
[66] H. Rahami, A. Kaveh, and Y. Gholipour, "Sizing, geometry and topology optimization of trusses via force method and genetic algorithm," Engineering Structures, vol. 30, no. 9, pp. 2360-2369, 2008.
[67] W. Zuo, J. Bai, and B. Li, "A hybrid OC-GA approach for fast and global truss optimization with frequency constraints," Applied Soft Computing, vol. 14, pp. 528-535, 2014.
[68] S.-J. Wu and P.-T. Chow, "Integrated discrete and configuration optimization of trusses using genetic algorithms," Computers & Structures, vol. 55, pp. 695-702, 1995.
[69] M. R. Ghasemi, E. Hinton, and R. D. Wood, "Optimization of trusses using genetic algorithms for discrete and continuous variables," Engineering Computations, vol. 16, no. 3, pp. 272-301, 1999.

[70] C. Camp, S. Pezeshk, and G. Cao, "Optimized design of two-dimensional structures using a genetic algorithm," Journal of Structural Engineering, vol. 124, no. 5, pp. 551-559, 1998.
[71] T. Dede, S. Bekiroglu, and Y. Ayvaz, "Weight minimization of trusses with genetic algorithm," Applied Soft Computing, vol. 11, pp. 2565-2575, 2011.
[72] G.-C. Luh and C.-Y. Lin, "Optimal design of truss structures using ant algorithm," Structural and Multidisciplinary Optimization, vol. 36, no. 4, pp. 365-379, 2007.
[73] S. Gholami and S. Zahedi, "A modified ant colony optimization algorithm for truss optimization," Engineering Structures, vol. 42, pp. 61-69, 2012.
[74] S. Ramezani, A. Kaveh, and R. Farmani, "Truss optimization using a new ant colony algorithm," Advances in Engineering Software, vol. 83, pp. 23-31, 2015.
[75] S. K. Azad and O. Hasançebi, "Discrete sizing optimization of steel trusses under multiple displacement constraints and load cases using guided stochastic search technique," Struct Multidiscip Optim, vol. 52, no. 2, pp. 383–404, 2015.
[76] Azad, S. K., & Hasançebi, O. (2016). "Discrete size optimization of steel trusses using guided stochastic search technique with multi-objective application." Advances in Engineering Software, vol. 98, pp. 10-26.
[77] S. Azam, M. S. Islam, A. B. M. A. Islam, and T. H. Dinh, "Optimal Design of Truss Structures Using Guided Stochastic Search Method," in 2018 International Conference on Innovations in Science, Engineering and Technology (ICISET), 2018, pp. 1-6.
[78] A. Kaveh and B. Ahmadi, "Sizing, geometry and topology optimization of trusses using force method and supervised charged system search," Structural Engineering and Mechanics, vol. 50, no. 3, pp. 365-382, 2014.
[79] P. Civicioglu and E. Besdok, "Charged system search algorithm for optimal truss design considering displacement constraints," Journal of Structural Engineering, vol. 142, no. 3, paper 04015147, 2016.
[80] Civicioglu, P., & Besdok, E. (2017). "Charged system search algorithm for truss optimization under displacement constraints." Journal of Structural and Multidisciplinary Optimization, vol. 56, no. 2, pp. 409-424.
[81] G.-C. Luh and C.-Y. Lin, "Optimal design of truss-structures using particle swarm optimization," Computers & Structures, vol. 89, no. 23-24, pp. 2221-2232, 2011.
[82] P. C. Fourie and A. A. Groenwold, "The particle swarm optimization algorithm in size and shape optimization," Structural and Multidisciplinary Optimization, vol. 23, no. 4, pp. 259-267, 2002.

[83] A. Kaveh, T. Bakhshpoori, and E. Afshari, “Hybrid PSO and SSO algorithm for truss layout and size optimization considering dynamic constraints,” Structural Engineering and Mechanics, vol. 54, no. 3, pp. 453-474, 2015.
[84] A. Mortazavi and V. Toğan, “Simultaneous size, shape, and topology optimization of truss structures using integrated particle swarm optimizer,” Structural and Multidisciplinary Optimization, vol. 54, no. 4, pp. 715-736, 2016.
[85] S. Gholizadeh, “Layout optimization of truss structures by hybridizing cellular automata and particle swarm optimization,” Computers & Structures, vol. 125, pp. 86-99, 2013.
[86] A. Kaveh and S. Talatahari, “Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures,” Computers & Structures, vol. 87, no. 5-6, pp. 267-283, 2009.
[87] L. Lamberti, “An efficient simulated annealing algorithm for design optimization of truss structures,” Computers & Structures, vol. 86, no. 19-20, pp. 1936-1953, 2008.
[88] X. S. Yang, S. Deb, and S. Fong, "Metaheuristic optimization for truss structures with discrete variables using a discrete glowworm swarm optimization algorithm," Computers & Structures, vol. 88, no. 21-22, pp. 1417-1429, 2010.
[89] A. H. Gandomi and A. H. Alavi, "Simulated annealing-based optimization of truss structures," Structural Engineering and Mechanics, vol. 39, no. 5, pp. 649-668, 2011.
[90] L. F. F. Miguel, R. H. Lopez, and L. F. F. Miguel, “Multimodal size, shape, and topology optimization of truss structures using the Firefly algorithm,” Advances in Engineering Software, vol. 56, pp. 23-37, 2013.
[91] D. Kumar, D. K. Singh, S. Kumar, and M. Thakur, "Optimal design of truss structures using firefly algorithm," International Journal of Civil Engineering and Technology, vol. 8, no. 5, pp. 1159-1173, 2017.
[92] S. M. Ali, H. F. Khitam, A. A. Hadi, and M. A. Sadiq, "Optimization of truss structures using hybrid firefly algorithm with particle swarm optimization," Arabian Journal for Science and Engineering, vol. 43, no. 11, pp. 6007-6022, 2018.
[93] A. Kaveh and S. Talatahari, “Size optimization of space trusses using Big Bang-Big Crunch algorithm,” Computers & Structures, vol. 87, pp. 1129-1140, 2009.
[94] K. G. Kiran and J. Hemanth, "Optimal design of truss structures using Big Bang–Big Crunch optimization," Structural Engineering and Mechanics, vol. 52, no. 1, pp. 143-159, 2014.

[95] H. Haddadpour, G. G. Amiri, and A. Ramezani, "Optimal design of truss structures using Big Bang–Big Crunch algorithm considering buckling constraints," Structural Engineering and Mechanics, vol. 58, no. 4, pp. 699-716, 2016.
[96] S. O. Cho, K. S. Lee, and J. H. Kim, "Optimal design of truss structures using harmony search algorithm," Engineering Structures, vol. 27, no. 7, pp. 1042-1052, 2005.
[97] J. Lim, K. S. Lee, and Y. Park, "Structural optimization using harmony search algorithm with varying harmony memory consideration rate," Computers & Structures, vol. 86, no. 21-22, pp. 2067-2075, 2008.
[98] A. Kaveh and S. M. Javadi, "Shape and size optimization of trusses with multiple frequency constraints using harmony search and ray optimizer for enhancing the particle swarm optimization algorithm," Acta Mechanica, vol. 225, no. 6, pp. 1595-1605, 2014.
[99] A. Kaveh and S. Talatahari, "Size optimization of truss structures using differential evolution," Advances in Engineering Software, vol. 41, no. 4, pp. 572-580, 2010.
[100] S. Chowdhury and R. Halder, "Optimal design of truss structures using differential evolution algorithm," International Journal of Engineering, Science and Technology, vol. 4, no. 4, pp. 9-16, 2012.
[101] V. Ho-Huu, T. Nguyen-Thoi, T. Vo-Duy, and T. Nguyen-Trang, "An adaptive elitist differential evolution for optimization of truss structures with discrete design variables," Computers & Structures, vol. 165, pp. 59-75, 2016.
[102] O. Sigmund, "On the usefulness of non-gradient approaches in topology optimization," Struct. Multidiscip. Optim., vol. 43, no. 5, pp. 589-596, 2011.
[103] M. Stolpe, "To bee or not to bee—comments on "Discrete optimum design of truss structures using artificial bee colony algorithm"," Struct. Multidiscip. Optim., vol. 44, pp. 707-711, 2011.
[104] W. Zhang, C. Zhang, L. Zhang, and G. Guikai, "Comparison of gradient and nongradient algorithms in the structural optimization course," Int. J. Mech. Eng. Educ., 2018.
[105] W. Zuo, J. Bai, and B. Li, "A hybrid OC-GA approach for fast and global truss optimization with frequency constraints," Applied Soft Computing, vol. 14, pp. 528-535, 2014.
[106] A. Kaveh and S.M. Javadi, "Shape and size optimization of trusses with multiple frequency constraints using harmony search and ray optimizer for enhancing the

particle swarm optimization algorithm," Acta Mechanica, vol. 225, no. 6, pp. 1595-1605, 2014.
[107] S. Gholizadeh, "Layout optimization of truss structures by hybridizing cellular automata and particle swarm optimization," Computers & Structures, vol. 125, pp. 86-99, 2013.
[108] S. N. Patnaik, A. S. Gendy, L. Berke, and D. A. Hopkins, "Modified Fully Utilized Design (MFUD) Method for Stress and Displacement Constraints," International Journal for Numerical Methods in Engineering, vol. 41, no. 7, pp. 1171-1194, 1998.
[109] L. Gil and A. Andreu, "Shape and Cross-Section Optimisation of a Truss Structure," Computers & Structures, vol. 79, no. 7, pp. 681-689, 2001.
[110] G. N. Vanderplaats and F. Moses, "Automated Design of Trusses for Optimum Geometry," Journal of the Structural Division, vol. 98, no. 3, pp. 671-690, 1972.
[111] D. Wang, W. H. Zhang and J. S. Jiang, "Combined Shape and Sizing Optimization of Truss Structures," Computational Mechanics, vol. 29, no. 4, pp. 307-312, 2002.
[112] R. H. Gallagher and O. C. In Zienkiewicz, Optimum Structural Design: Theory and Applications, London: Wiley, 1973.
[113] R. Razani, "The Behaviour of the Fully Stressed Design of Structures and its Relationship to Minimum Weight Design," AIAA Journal, vol. 3, pp. 2262-2268, 1965.
[114] A. Ahrari, A. Atai and K. Deb, "Simultaneous Topology, Shape and Size Optimization of Truss Structures by Fully Stressed Design Based on Evolution Strategy," Engineering Optimization, vol. 47, no. 8, pp. 1063-1084, 2015.
[115] P. Hajela, "Implications of artificial life simulations in structural analysis and design," in 39th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference and exhibit, American Institute of Aeronautics and Astronautics, 1998.
[116] E. Kita and T. Toyoda, "Structural design using cellular automata," Struct Multidiscip Optim, vol. 19, no. 1, pp. 64-73, 2000.
[117] Z. Gürdal and B. Tatting, "Cellular Automata for design of truss structures with linear and nonlinear response," in 41st structures, structural dynamics, and materials conference and exhibit, American Institute of Aeronautics and Astronautics, 2000.
[118] B. Tatting and Z. Gürdal, "Cellular automata for design of two-dimensional continuum structures," in 8th symposium on multidisciplinary analysis and optimization, American Institute of Aeronautics and Astronautics, 2000.

[119] M. M. Abdalla and Z. Gürdal, "Structural design using Cellular Automata for eigenvalue problems," Struct Multidiscip Optim, vol. 26, no. 3-4, pp. 200-208, 2004.
[120] O. E. Canyurt and P. Hajela, "A SAND approach based on cellular computation models for analysis and optimization," Eng Optim, vol. 39, no. 4, pp. 381-396, 2007.
[121] A. Faramarzi and M.-H. Afshar, "Application of cellular automata to size and topology optimization of truss structures," Scientia Iranica A, vol. 19, no. 3, pp. 373-380, 2012.
[122] B. Bochenek and K. Tajs-Zielin´ska, "Minimal compliance topologies for maximal buckling load of columns," Struct Multidiscip Optim, vol. 51, no. 5, pp. 1149-1157, 2015.
[123] M. Sipper, "Co-evolving non-uniform cellular automata to perform computations," Phys-Sect D, vol. 92, no. 3, pp. 193-208, 1996.
[124] G. Y. Vichniac, P. Tamayo, and H. Hartman, "Annealed and quenched inhomogeneous cellular automata (INCA)," J Stat Phys, vol. 45, no. 5-6, pp. 875-883, 1986.
[125] M. Ming and X. Renwei, "An efficient method of truss design for optimum geometry," Computers & Structures, vol. 35, pp. 115-119, 1990.
[126] F. Flager, A. Adya, J. Haymaker, and M. Fischer, "A bi-level hierarchical method for shape and member sizing optimization of steel truss structures," Computers & Structures, vol. 131, pp. 1-11, 2014.
[127] A. Kaveh and S. M. Javadi, "Shape and size optimization of trusses with multiple frequency constraints using harmony search and ray optimizer for enhancing the particle swarm optimization algorithm," Acta Mechanica, vol. 225, no. 6, pp. 1595-1605, 2014.
[128] A. Kaveh and M. Ilchi Ghazaan, "Layout and size optimization of trusses with natural frequency constraints using improved ray optimization algorithm," Iranian Journal of Science and Technology Transactions of Civil Engineering, vol. 39, no. 2, pp. 395-408, 2015.
[129] S. O. Degertekin, L. Lamberti, and I. B. Ugur, "Sizing, layout and topology design optimization of truss structures using the Jaya algorithm," Applied Soft Computing, vol. 70, pp. 903-928, 2018.
[130] S. O. Degertekin, L. Lamberti, and I. B. Ugur, "Discrete sizing/layout/topology optimization of truss structures with an advanced Jaya algorithm," Applied Soft Computing, vol. 79, pp. 363-390, 2019.

[131] C. S. Horstmann, Object-Oriented Design and Patterns, 3rd ed. John Wiley & Sons, 2016.
[132] Slotta DJ, Tatting B, Watson LT, Gürdal Z, Missoum S. Convergence analysis for cellular automata applied to truss design. Eng Comput. 2002;19(8):953–69.
[133] P.A. Makris and C.G. Provatidis, "Weight minimisation of displacement-constrained truss structures using a strain energy criterion," Comput Methods Appl Mech Eng, vol. 191, no. 19-20, pp. 2187-2205, 2002.
[134] O. M. Querin, "Evolutionary structural optimisation: stress based formulation and implementation," University of Sydney, Sydney, Australia, 1997.
[135] Descamps, B., & Filomeno Coelho, R. (2013). A lower-bound formulation for the geometry and topology optimization of truss structures under multiple loading. *Structural and multidisciplinary optimization*, *48*, 49-58.