Design Optimization of Lightweight Structures for Additive Manufacturing

by

Oguz Toragay

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
August 6, 2022

Keywords: Additive Manufacturing, Lightweight Structures, Topology Optimization,
Mathematical Programming, Heuristic optimization, Genetic algorithm

Copyright 2022 by Oguz Toragay

Approved by

Daniel F. Silva, Chair, Assistant Professor of Industrial and Systems Engineering, Auburn University
Aleksandr Vinel, Associate Professor of Industrial and Systems Engineering, Auburn University
Jia (Peter) Liu, Assistant Professor of Industrial and Systems Engineering, Auburn University
Nima Shamsaei, Philpott-WestPoint Stevens Distinguished Professor of Mechanical Engineering, Auburn University
Elham Mirkoohi, Assistant Professor of Mechanical Engineering, Auburn University

Abstract

Lightweight structures have many applications in different engineering areas, such as automotive, aerospace, or medical industries, among many others. Optimal design of lightweight structures deals with finding the most economical distribution of the material in the design domain. This concept becomes even more important when additive manufacturing (AM) is considered for fabrication of the parts, since it allows for exceptional freedom in the design process. In this dissertation, we consider the design optimization problem for additively manufactured planar frame structures. We specifically consider three different optimization approaches in tackling the problem of finding lightest planar frames which can withstand the external loads. We apply exact optimization methods in Chapter 3, where we propose a novel mixed integer quadratically constrained optimization model for the problem and compare its performance to the existing models from the literature. We then propose a problem-specific heuristic method in Chapter 4, which is capable of solving large-scale problems that couldn't be handled by exact optimization methods. This heuristic method is a combination of a member-node adding approach and nonlinear optimization, in which the solving process starts from a version of ground structure with a minimal number of elements and then gradually includes elements with the most promising contribution in reducing the stress in the structure. In Chapter 5 we test the ability of metaheuristics, specifically Genetic Algorithm (GA) to solve this mathematical optimization problem. In the proposed hybrid approach, we combine GA with nonlinear optimization. To this end, we designed a new encoding of the candidate solutions together with the GA operators that in addition to the stochastic nature of the GA in solving combinatorial problems, combined with the deterministic exactness of nonlinear optimization provides a novel way to solve the design problem. We conclude the dissertation by providing the main findings and future research in Chapter 6.

Acknowledgments

To my wife, for her patience and for all these years we were apart.

To my parents and sisters, for their unconditional love.

I would like to express the deepest appreciation to my Ph.D. advisor, Dr. Daniel F. Silva, for his endless support and persistent guidance. I would also like to acknowledge my committee members, Dr. Alexander Vinel, Dr. Nima Shamsaei, Dr. Peter Liu, and Dr. Elham Mirkoohi for their constructive feedback throughout the entire dissertation process. I should also express my gratitude to my co-advisor Dr. Alexander Vinel for his endless support during my Ph.D. journey. Many thanks to Dr. Richard Garnett who, as my teaching mentor, taught me how to teach undergrad level courses.

Last but not least, I would like to thank all my friends, specially (in alphabetic order), *Alison*, *Amir*, *Ana*, *Armin*, *Babek*, *Behnam*, *Farhad*, *Farnaz*, *Mehran*, *Mohammad*, *Mohammad Ali*, *Mohsen*, *Nasrin*, *Navid*, *Saina*, *Samira*, *Sanaz*, *Shahin*, *Shayan*, and *Yasin* for all the great memories, unforgettable moments, and scientific discussions that we have had.

I would like to proudly dedicate this dissertation to all people who are insisting in getting to the global optimal in the fluctuations of life, instead of being trapped in local optima.

Table of Contents

Abstract ii

Acknowledgments iii

1 Introduction and Motivation 1

1.1 Introduction 1

1.2 Contributions 4

2 State-of-the-art Literature Review 7

2.1 Exact methods related literature review 8

2.2 Heuristics related literature review 14

2.3 Metaheuristic related literature review 18

3 Discrete Topology Optimization of Additively Manufactured Lightweight Planar Frame Structures 22

3.1 Introduction 22

3.2 Mathematical Models 25

3.2.1 Problem statement and stiffness matrix decomposition 26

3.2.2 Mixed Integer Non-Linear Programming Model (MINLP) 28

3.2.3 Mixed Integer Linear Programming Model (MILP) 30

3.2.4 Mixed Integer Quadratically Constrained Programming Model (MIQCP) 32

3.2.5 Solution approach 33

3.3 Numerical Experiments 34

3.3.1 Test instances 34

3.3.2 Numerical Results . . . 37

3.3.3 Influence of the two-stage approach on MIQCP . . . 40

3.3.4 Comparison of MIQCP and MINLP models . . . 41

3.3.5 Two examples of the solved structures . . . 42

3.4 Conclusions . . . 44

4 On optimization of lightweight planar frame structures: an evolving ground structure approach . . . 46

4.1 Introduction . . . 46

4.2 Methodology . . . 50

4.2.1 General mathematical programming formulation for frame structures . . . 50

4.2.2 Relaxed master optimization model . . . 52

4.2.3 Iterative evolving ground structure algorithm . . . 54

4.3 Numerical Results . . . 57

4.3.1 Design of experiments . . . 57

4.3.2 Numerical results and observations . . . 60

4.4 Conclusions . . . 64

5 A Hybrid Genetic Algorithm for Topology Optimization of Frame Structures for Additive Manufacturing . . . 67

5.1 Introduction . . . 67

5.2 Methodology . . . 69

5.2.1 Model overview and relaxation . . . 70

5.2.2 Hybrid genetic algorithm . . . 72

5.2.3 Operators of the GA . . . 75

5.3 Numerical Results . . . 77

5.4 Conclusion . . . 80

6 Summary and Future Research 82

A Details of Mechanical relations in the modeling part 84

B Ground Structure's Pseudo code 88

C Initial and final populations 89

Bibliography 93

List of Figures

2.1 Categorization of Structural Optimization 9

3.1 Instances of ground structures 23

3.2 Designed experimental ground structures 35

3.3 All the structures resulted from solving Figure 3.2(a) ground structure using all the models 38

3.4 Optimal structures resulted from Fig.3.2 (d) 43

3.5 Optimal structures resulted from Fig.3.2 (i) 44

4.1 (a) design domain with boundary conditions and external load; (b) nodes of a ground structure; (c) a fully connected ground structure; (d) a restricted ground structure excluding non-manufacturable elements. 48

4.2 Square test instances (3×3 and 5×5 grids). 58

4.3 Cantilever test instances (coarse, medium and fine grids). 59

4.4 Bridge instances (two different build orientations). 60

4.5 Obtained structure weight and corresponding the solution times (in seconds) for the square instances for the proposed evolving ground structure method depending on load magnitude. 61

4.6 Obtained structures for the square instances for the proposed evolving ground structure method for a collection of external load magnitudes. Green points correspond to nodes added by the algorithm. 61

4.7 Original MGS, intermediate solutions and the resulting structures due to the heuristic method for the cantilever instances. Red and blue beams correspond to tension and compression respectively. Green points correspond to nodes added by the algorithm. 64

4.8 Obtained solutions for the two versions of the bridge instance with different load magnitudes. Green points correspond to nodes added by the algorithm. . . 65

5.1 Generating the initial population 70

5.2 (a) Possible paths with length ≤ 3 in a 5×5 ground structure, (b) An individual 74

5.3 (a) Square example (b) Cantilever example 78

5.4 (a) Best solution in the final population of square instance, (b) Best solution in the final population of cantilever instance 79

C.0.1 Initial population for the square instance with $k = 24$, vertival load= $400kN$ . . 89

C.0.2 Final population of square instance, best weight = 161.38 90

C.0.3 Initial population for the cantilever instance with $k = 24$, horizontal load= $10kN$ 91

C.0.4 Final population of cantilever instance, best weight = 260.97 92

List of Tables

3.1 Weight of all the structures solved by 4 models 39

3.2 Solving time of all the structures solved by 4 models (**TL**: Time Limit) 39

3.3 Optimality gap (%) of all the structures solved by 4 models 40

3.4 MIQCP with vs. without two-stage approach 41

3.5 Comparison of solving time (in second) for MIQCP vs. MINLP 42

3.6 Comparison of objective value for MIQCP vs. MINLP 42

3.7 Solution for instance Fig. 3.2(d) 43

3.8 Solution for instance Fig. 3.2(i) 44

4.1 Number of elements in FGS and MGS for a square $m \times m$ grid with two boundary conditions and single external load. 55

4.2 Solution time (T) and weight (W) of structures resulted from heuristic evolving ground structure (H) vs. exact (E) methods for square instances. Optimality gap in percent is reported instead of time for instances terminated due to time limit. "–" indicates that no feasible solution was found. "**inf**" corresponds to provably infeasible problems (for the exact grid). 62

4.3 Results for the cantilever instances. 63

4.4 Vertical and horizontal bridge results with various loads. 63

5.1 Solution for instance Fig. 5.3 78

5.2 Results for square instance with various seeds and load magnitudes 79

5.3 Results for cantilever instance with various seeds 80

Chapter 1

Introduction and Motivation

1.1 Introduction

New technologies affect our world mostly in positive ways as they bring ease to control, understand, and analyze our environment, and also allow us to get the most benefit from the scarce resources that are available to us. On the other hand, they also bring some challenges to be tackled by the scientific community. One of the recently emerged technologies, which has been able to find its own place in various branches of engineering, is Additive Manufacturing (AM). AM has its bases in prototyping, it originated from the idea of combining three-dimensional designs with the possibility of printing two-dimensional designs. In its traditional form manufacturing activities usually entailed removing or subtracting material from a bulk initial piece to reach eventually a near-net desired shape. On the other hand, in additive manufacturing, a thin layer of material is added to previous layers, based on a 3D model, to build the desired shape. As a new branch of manufacturing, in comparison to traditional manufacturing, AM relaxes some of the limitations that were conventionally imposed by traditional manufacturing on the design of fabricated parts. The most effective contribution of AM, in general, is the possibility of optimizing the distribution of material in different interior and exterior locations of a part. This aspect of fabrication of the parts was partially possible using traditional manufacturing techniques such as casting, machining, etc. but it can never be compared to the freedom that the design stage gets when additive techniques are considered for the fabrication. Material distribution control by AM brings new opportunities for design and fabrication of lightweight structures in which mechanical characteristics of the part remain the same, but volume and as a result of that, the weight of the part is reduced considerably by removing surplus material from ineffective locations of the part's interior. These lightweight parts are especially useful in some branches such as aerospace, medical, mechanical, and civil engineering. For example, as of NASA's report[1] on October 19^{th} 2020, the newly landed *Mars Perseverance Rover* took 11 additively manufactured parts to the red planet. Although we focus on the control of the material distribution throughout the design domain of the part, AM also paves the way to relax other limitations related to traditional manufacturing. For instance, structures containing enclosed hollow parts cannot be produced by conventional manufacturing techniques without a post processing or assembly step but AM can produce such fully closed internal hollows (Gibson et al. 2014).

Selecting the most beneficial distribution of material throughout the design domain can be approached as an optimization problem in which the amount of the material is minimized while some engineering limitations and constraints are forced to hold. In this dissertation, we investigate the mutual effect of optimization approaches and additive manufacturing limitations on the design of lightweight structures. More precisely, we model the design of load-bearing lightweight structures as a mathematical programming problem to optimize the weight of the structure while keeping the mechanical characteristics in some given range, with reasonable tolerances. Those mentioned mechanical characteristics can be stresses on different locations of the part, displacements caused by the external or internal loads, buckling, bending, etc. The main motivation behind this research is that using lightweight structures in many applications such as aerospace, transportation, and automotive industry can considerably save energy and reduce fuel consumption and air pollution. On the other hand, lightweight structures and specifically lattice structures (defined as meso-scale or micro-scale truss-like structures with a certain repeated arrangement in 3-D space (Tang et al. 2017; Choy et al. 2017)) have specific characteristics that cannot be achieved using parts fabricated by conventional manufacturing. For instance, in some medical applications, it is desired that the an implant's mechanical properties matches to the surrounding bone and tissue (Burton et al. 2019) or the implant reacts to the environmental forces in a specific manner that can be achieved by lattice structures rather than bulk material. These lattice structures are specific versions of the lightweight structures that can be designed and fabricated using mainly additive manufacturing.

[1] https://www.nasa.gov/feature/jpl/nasas-perseverance-rover-bringing-3d-printed-metal-parts-to-mars

In this dissertation we will consider the design optimization problem for 2D, so-called *planar* frame structures. These structures consist of discrete beam elements that are rigidly connected in the joints. Topology optimization of planar frames is known as an NP-hard problem while the AM imposed manufacturability constraints increase the complexity even more in this type of mathematical problems.

In the remaining parts of this dissertation, we investigate new modeling and solving approaches to partially handle the current limitations of this era of multidisciplinary optimization problems. To frame this work, in Chapter 2, we provide a state-of-the-art literature review of various approaches to solve the structural optimization problems and then narrow down the discussion to cover the mutual effects of optimization approaches on one hand and AM constraints on the other hand.

In Chapter 3, we compare three exact modeling approaches to find the lightweight planar frame structures designed for AM. We observed that when the size of the structure gets larger, the number of variables increases dramatically, the problem quickly becomes intractable. These models struggled to be solved even by using state-of-the-art commercial solvers. In our first article, we compare the results of three modeling approaches in terms of solving time and results' quality. From those three modeling approaches one is proposed by the author, to the best of our knowledge, for the first time in the literature. We also proposed a two-stage heuristic method to reduce the solving time while aiming for getting at least a feasible solution for the very hard instances which wouldn't be solved without the two-stage approach.

Observing the limitation of exact mathematical models in solving large-scale problems, in Chapter 4 we proposed a novel problem-specific heuristic method to solve large-scale design problems for AM. We implemented an adaptive member adding technique to solve those problem that are not possibly solvable by using exact methods. The results reported in this chapter show the that, using the proposed heuristic method, large-scale instances can be solved near-optimally, while for smaller instances the proposed method is capable of finding the optimal solution in a fraction of time than otherwise required by the exact method to find the same solution. The superiority of the heuristic method in terms of computational requirement, even for relatively simple instances is confirmed. The research discussed in this chapter has been submitted to *Optimization and Engineering journal*.

In Chapter 5, we investigate the effectiveness of metaheuristic methods on solving structural optimization problems where there are also constraints in the model imposed by AM. We especially investigate the quality of results of Genetic Algorithm (GA) metaheuristic approach for additively manufactured frame structures' design problem. Unlike the GA in its traditional form, we combined the mathematical optimization approaches together with the randomness in the assembly of generations and individuals in the GA to enhance the convergence time. The results reported in this chapter show the potential of the proposed hybrid GA in handling large-scale design problems.

1.2 Contributions

The novelty of this research, in general terms, is in combining various concepts of additive manufacturing from a mechanical engineering point of view, planar and spatial frame structures characteristics from a civil engineering perspective, and the optimization concepts from industrial engineering and operations research disciplines. In this section, we summarize all the contributions and novelty that our works bring to the literature.

In Chapter 3 we investigate the applications of exact mathematical models in solving the design problem of lightweight planar frame structures for AM. The followings are the summary of main contributions and novelties presented in Chapter 3 of this dissertation:

- Non-linear and linear models investigated from the literature for frame structures have been modified to cover the manufacturability constraints imposed by AM.
- A new mixed-integer quadratically constrained programming model is proposed for the planar lightweight frame structure design problem.
- A new two-stage solving approach to tackle the difficulty of finding an initial feasible solution for the frame structures mathematical models has been proposed.

- A comparison of various modeling approaches for frame structures in terms of accuracy and solving time has been conducted.

Observing the long solving time required by the models that are investigated in the Chapter 3, and also the inverse effect of problem's size on the performance of the available solvers for exact mathematical models, in Chapter 4 we developed a novel iterative heuristic method with the following innovations:

- The proposed approach is capable of solving large-scale problem which can not be solved using the exact method.
- The iterative nature of the proposed approach constructed upon the interactions between mechanical properties of the part on one hand and the constraints in the mathematical model on the hand which in turn allows to search the solution-space in an informed manner.
- Instead of using the traditional ground structure method, in this approach a simplified version of the ground structure with less number of candidates has been adopted.
- Adaptive member adding and removing approach has been developed which allows the addition of new nodes and candidate element to the ground structure which in turn pave the way to investigate those locations of solution-space which otherwise remained unexplored.

Observing the successful applications of metaheuristic approaches in solving the structural analysis problems mainly in the civil engineering area, in Chapter 5 we enhance one of the famous metaheuristic methods, Genetic Algorithm, to solve the problem at hand. Although the mentioned method is originally designed to solve the unconstrained optimization problems, we investigate new approaches to enable GA to solve our constrained problem. In Chapter 5 we present the results of applying GA to the problem of finding near-optimal solutions to the design problem of lightweight frame structures for AM. The main contributions of this chapter can be listed as follows:

- We modified the simple GA in order to enhance the convergence speed by applying a modification in which results from mathematical optimization model can be used to calculate the fitness values of the individuals.

- We guided the GA's search process using the problem specific information. Matrix-based representation of the structures and a novel encoding of the individuals based on the paths in the structure as a graph are some of the innovations that we implemented in the application of GA.

- GA has been used to solve the problem where both integer (binary) and continuous variables involved in the mathematical representation.

- AM imposed manufacturability constraints, such as elements' cross-sectional limitations and removing crossing-member have been considered in the proposed hybrid GA.

Chapter 6 will conclude this dissertation by summarizing the main findings and providing the road map for future research in the area of optimizing the design for additive manufacturing. In Appendix A we present the transition from mechanical properties to the mathematical models by explaining all the relations and formulations used in defining the optimization models. Appendix B gives the pseudo code for generating the ground structure which is further used in all approaches that we mention in other chapters. Appendix C depicts the initial and final population of the instances solved in Chapter 5.

Chapter 2

State-of-the-art Literature Review

In this chapter, we summarize the works from the literature that is most relevant to the design of additively manufactured lightweight structures. Two factors are considered in the ordering of the works, first we consider the chapter of this dissertation that the article most relevant, second, the type of structure that considered in the work. In Chapter 3 we consider the exact mathematical models to solve the problem of lightweight frame structure design for additive manufacturing. Next in Chapter 4 the heuristics to tackle the problem and in the Chapter 5 the selected metaheuristic method, genetic algorithm, will be considered to solve the problem. Consequently, we follow the same order in the literature review, meaning that, we first cover the most related works from the literature of implementing the exact methods, then the heuristics and finally genetic algorithm related works will be presented. For each of these categories, we first, focus on the mathematical modeling of truss structures because it can pave the way for a better understanding of the challenges that appear in the frame structures' modeling approaches. On the other hand truss structures in which the elements of the structure are pinned bar elements, although possible to be fabricated, are not appropriate for additive manufacturing. That's also because fabricating the elements separately, using the traditional manufacturing techniques, and then assemble them to get the final part, is more appropriate and cheaper than designing and fabricating the same structure as a whole, by using additive manufacturing. The next natural step in reviewing the literature is to focus on the works that consider various mathematical modeling approaches in designing the frame structures in which the elements are beam elements that are fixed on the connections or, in other words, rigid-joined structures. Finally, we focus on the applications and works related to using additive manufacturing techniques for the fabrication of the designed lightweight structures, especially frame structures.

To understand all of the works presented here, we begin this chapter by summarizing some works related to the optimization perspective to the structures. Then we discuss related works in the literature following the specific approaches that we consider for our research, i.e., applications of exact mathematical models to solve the problem, considering the heuristics to find the near-optimal solutions for large-scale problems and finally applications of metaheuristics to tackle the optimization problem in hand.

2.1 Exact methods related literature review

The concept of lightweight structures, or in other words, distribution of the material in the most economical way first appeared in Michell (1904)'s work. Since then, many advancements happened in various aspects of structural optimization. The methods under this branch of optimization can be categorized as, *Size optimization*, *Shape optimization* and *Topology optimization* (for definitions see (Querin et al. 2017) and (Bendsoe and Sigmund 2013)). Two main stream of the research in topology optimization for structures, namely for discrete and continuum structures, have been established in the literature by the appearance of the works of Dorn (1964) and Bendsoe and Sigmund (2013) respectively. Considering ground structure or design domain as the base of the optimization procedure is one of the widely used approaches in structural optimization. In this approach, design domain consists of nodal locations which are mainly assumed to be fixed, and all the candidate elements (bar or frame elements). In addition, applied load(s), boundary conditions, and some additional design restrictions are assumed to be given. According to Bendsoe and Sigmund (2013), the unknowns of the design problems are the physical size, the shape, and the connectivity of the structure. An enhanced review of the related literature is given by Stolpe (2016) for discrete structures, and by Eschenauer and Olhoff (2001), and Deaton and Grandhi (2014) for continuum structures. Stolpe (2016) further categorized the applications of modeling based on the ground structure method, into three main categories. The author also considered the objective functions of the model as the major criteria in the categorization. These divisions are given as:

- Minimum compliance (maximum stiffness) problem with volume constraint.

- Minimum weight problem with a constraint on the compliance.
- Minimum weight problem with a constraint on the nodal displacements and constraint on the allowable element stresses.

The type of problem that we consider in this work falls in the third category. The displacements are controlled while the size of the elements can be one of the variables in the mathematical model. The disappearance of the candidate connections between nodes in the ground structure specifies the type of structural optimization problem. In other words, we can categorize the structural optimization problems by checking that whether all the elements of the ground structure remain in the final solution, or some of the elements vanish during the optimization process. According to Stolpe (2016) if during the optimization iterations some of the elements of ground structure vanish, the topology of the structure changes, and this problem categorized as *topology* optimization. On the other hand, if all the given members of the ground structure remain in the final outcome of the optimization procedure, potentially with optimized size, cross-sectional area, or shape, this problem is called *shape* or *size* optimization. Figure 2.1 best describes the mentioned categorization and the type of problem that we tackle in this dissertation.

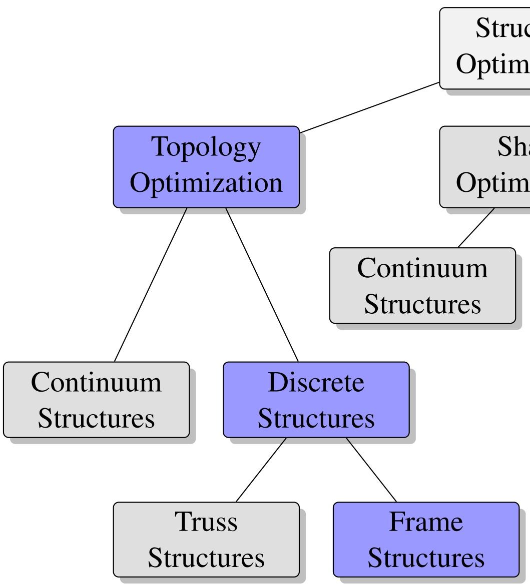


Figure 2.1: Categorization of Structural Optimization

For a better understanding of the frame structures, in the following, we first summarize some of the works in the literature related to the truss structures. Using the bar elements and consequently optimizing trusses as a discrete structure is a very popular problem in the literature. Mostly the cross-sectional area of the bars is the main variable of the truss topology optimization problems. Grossmann, Voudouris and Ghattas (1992) best described the characteristics of these problems in their own words as follow:

"Many problems in engineering design give rise to non-convex nonlinear programming (NLP) problems. Furthermore, quite often due to manufacturing constraints, design variables are restricted to take discrete values for selecting standard sizes which give rise to mixed-integer nonlinear programs (MINLP)".

Mixed Integer Linear Programming reformulation of such problems with separable objective functions and bilinearities in the constraints is given by Grossmann, Voudouris and Ghattas (1992). Since then, various approaches for tackling truss topology optimization have been contributed to the literature. Bollapragada, Ghattas and Hooker (2001) showed that for the truss structures topology optimization problems with discrete cross-sectional areas of the bar elements, the Logic-Based Branch-and-Cut method can solve larger problem instances than a branch-and-bound used for a Mixed Integer Programming (MIP) (in which non-linearity of the relaxed problems can disappear). None of these formulations contain design-dependent constraints that's why they are more sizing rather than topology problems (Stolpe and Svanberg 2003). This downside of the mentioned approaches tackled by Stolpe and Svanberg (2003) where topology optimization problem modeled as a Mixed 0-1 program with linear inequality and bilinear equality constraints which can further converted to linear inequality constraints. Stolpe later proposed a branch-and-bound method (Stolpe 2004) and a parallel branch-and-cut method (Rasmussen and Stolpe 2008) to solve this problem. Global optimization of truss structures with discrete cross-sectional areas of the bars using the Branch-and-Bound method later considered by Achtziger and Stolpe (2007). In their work, authors modeled the problem as a MIP and solved a set of benchmark problems. Objective function is weighted average of the compliance of multiple load cases while they consider a constraint on the total volume of the structure. Cerveira et al. (2013) further developed the solving approach by proposing a two-stage branch-and-bound procedure for the stiffest truss structure with discrete cross-sectional area. In addition to the MILP and MINLP, other mathematical programming methods are used to model the discrete design problems as well. Mathematical Program with Complementarity Constraints (MPCC) implemented for the truss design problem with discrete cross-sectional area by Kočvara and Outrata (2006). Authors proposed three methods to be used for numerical solution of the problem while considering the uniqueness of the displacements of the nodes and areas of the bars from ground structure in the final solution. Second Order Cone Programming (SOCP) for truss structures implemented by Lobo et al. (1998) and Makrodimopoulos, Bhaskar and Keane (2010). In the latter, authors proposed a heuristic scheme for adding members to the ground structure after each iteration of the finite element analysis.

Makrodimopoulos, Bhaskar and Keane (2010) stated that the popularity of truss structures is due to the simplicity of designing and the ease of mathematically formulating these structures. In contrast, Stolpe (2016) explaining the difficulty of the modeling of these structures from an optimization point of view by stating the followings in the authors' own words:

- Due to the number of variables and constraints, the problem can be large-scale.
- Non-convexity can be another hardness.
- Stiffness matrix is only positive semi-definite.
- There may be multiple solutions for the displacements in the design which make objective and constraint functions non-differentiable.

Notice that the number of design variables such as nodal displacements for planar frame structures is 1.5 times larger than the number of variables in truss structures. Moreover, the necessary equations for finite element analysis of frame structures are more complicated than the equations for truss structures. For example, the stiffness matrix's size for a truss structure with 10 members is 40×40 but a frame structure with the same number of elements has a stiffness matrix of size 60×60. This leads us to the conclusion that solving optimization models for frame structures' design is much harder than the truss structures.

Frame structures' optimization problem has been approached by either the Civil and Structural Engineers or by Mechanical and Aerospace Engineers. Chan (1992) best described the necessity of discretization in the optimization models for applications in civil engineering by stating that: “Structural members of a steel building structure are rarely custom-fabricated and these sections are only available in discrete standard sizes from commercial steel fabricators”. That’s why early attempts to solve structural optimization problems for frame structures generally adopt nonlinear models with continuous design variables and then round-up the results to get the nearest discrete sizes. Chan (1992) proposed an Optimality Criteria (OC) based discrete variable method for steel frames.

Mixed-integer nonlinear programming is used for modeling by (Klanšek et al. 2007), where the authors implemented Outer-Approximation/Equality-Relaxation iterative algorithm for solving the problem. Takezawa et al. (2007), by using continuous ellipsoidal cross-sectional area, optimized the rotational angle denoting the optimal principal direction of frame element implementing a procedure based on Karush-Kuhn-Tucker (KKT) conditions and the complementary strain energy concept. An extensive review of mathematical and meta-heuristic methods for optimization of frame structures is given by Saka and Geem (2013).

Most of the above works do not consider the vanishing members; they consider all of the elements in the ground structure (composed of steel framed storey and bays) to remain during the optimization procedure and in the final/optimal structure. That’s why these problems can be considered as size optimization rather than topology optimization.

For frame structure, Kureta and Kanno (2014) implemented a similar approach that Stolpe and Svanberg (2003), Rasmussen and Stolpe (2008), and Stolpe (2007) developed for truss structures. In their chapter, they considered the design of periodic frame structures with negative Poisson’s ratio. The main idea of the mentioned papers developed around the decomposition of the stiffness matrix. In Chapter 3, we propose a new modeling approach for frame structures by implementing a similar idea as the aforementioned papers in this paragraph.

Other mathematical programming approaches such as Mixed-integer second-order cone programming (Kanno 2016), Robust optimization (Changizi and Jalalpour 2017*a*) and Gradient-based optimization (Changizi and Jalalpour 2017*b*) have been used for efficiently modeling and solving frame structure optimization.

In recent years, rapid growth in the applications of additive manufacturing has led to a radical change in the way designers describe the design for fabrication. Although it has its own limitations, additive manufacturing provides freedom in the design of manufacturable complex geometries (Schmidt, Pedersen and Gout 2019). A review of medical implants produced by applying additive manufacturing for the fabrication and using topology optimization for continuum material in the design step is given by Wang et al. (2016). Authors mentioned that in the fabrication of porous metal implants a combination of topology optimization and additive manufacturing is of great interest because the design of the internal architecture of the desired parts can be optimized by topology optimization while the fabrication of optimized complicated shapes has been achieved by additive manufacturing.

Another review paper (Liu and Ma 2016) discussed manufacturing-oriented topology optimization. They further categorized the manufacturing methods which affected the topology optimization under three main categories. They investigate the effects of machining methods and injection molding/casting methods on topology optimization by figuring out the limitations that these two conventional manufacturing methods bring to the optimization of structures. Then they explained that additive manufacturing for topology optimization can be implemented for tackling these limitations.

Cansizoglu et al. (2008), used Electron Beam Melting (EBM) to fabricate three dimensional optimized frame structures. The design variable was joints' locations and the objective function was the compliance of the structure. Authors implemented Quasi-Newton line search for unconstrained optimization and sequential quadratic programming (SQP) for the models with constraints on the mechanical properties of the structure. They mentioned as a results that, they found "discrepancies between the performance of the theoretical structures and the physical EBM structures due to the layered fabrication approach". Smith et al. (2016) proposed a work-flow to start the design process from a design domain (which is not ground structure, because of continuity) by finding an optimized layout for the structure. Throughout a set of post-optimization steps, the resulting layout is converted to a continuum structure viable for additive manufacturing. In fact, the resulting layout design of simplified truss structure converted to a frame structure imposing some constraints such as buckling response and removal of crossing members (by creating nodes at the intersections). Wu, Clausen and Sigmund (2017), combined “a coating approach to obtain an optimized shell that is filled uniformly with a prescribed porous base material, and an infill approach which generates optimized, non-uniform infill within a prescribed shell” to obtain the optimized design for additive manufacturing. An iterative smoothing and projection procedure has been used for material removal in this work.

To the best of our knowledge, there are only a few works focused on the applications of additive manufacturing in the fabrication of optimized discrete structures. Most of the applications in the literature focused on the continuum structures when the material distribution among the design area is optimized by removing the material that is less desired (making holes in the continuum design space). For example Liu et al. (2017), used moving morphable components/voids topology optimization method to design graded lattice structures considering the additive manufacturability. A similar approach to optimize the lattice structure design for additive manufacturing has been done by Bai et al. (2018), where the size optimization of the strut’s cross-sectional area has been solved using the Firefly meta-heuristic search algorithm.

2.2 Heuristics related literature review

In order to handle the large number of variables in the mathematical optimization models for structural design problems, many approaches have been proposed in the literature. These include defining a penalty on the number of nodes (Parkes 1975), adaptive member adding (Gilbert and Tyas 2003; Sokół and Rozvany 2013), defining ground structures based on principal stress trajectories (Gao, Liu, Li and Qiao 2017), growing ground structure (Hagishita and Ohsaki 2009), among others. All of these heuristic approaches aim for finding a near-optimal solution to the large-scale design problems when the number of variables in mathematical optimization problems grow exponentially and the model soon become intractable for exact approaches which mainly use branch-and-cut or interior point methods.

In the “iterative adding method” in which instead of having the fully connected ground structure, the process begins with a reduced number of members in the mesh. Gao, Liu, Li and Qiao (2017) categorized “iterative adding methods” into two categories, “only member adding” and “node and member adding” methods. In this work an extension of “node and member adding” method which enables not only node addition but also node removal to enhance the problem solving time, is proposed. In the literature, "iterative adding methods" have been used to solve problems dealing with both truss and frame structures. In the following, we first discuss (generally simpler) applications to the former, and then the latter.

Truss structures: Gilbert and Tyas (2003) proposed the member adding method for the first time and showed that with the proposed adaptive layout optimization approach they could solve large-scale weight optimization problems for truss structures (e.g. $> 100,000,000$ potential members). Later this concept was implemented by He, Gilbert and Song (2019). However, the results of this approach included many cross-passing members which made the building of the resulting truss structures almost impossible. To alleviate this downside, Fairclough and Gilbert (2020) applied the idea of iteratively imposing lazy constraints in the mixed integer linear programming model aiming to reduce the number of crossover members and to obtain a simplified version of the resulting structure. Weldeyesus and Gondzio (2018) proposed a specialized primal-dual interior point method to iteratively solve the sub-problems. Member adding approach was also used by Weldeyesus et al. (2019) where the relaxation of nonlinear model for the truss structures was modeled as a semidefinite programming problem. Gao, Li, Pan, Chen and Liu (2017) extended the member adding approach by considering those members inside the principal stress trajectory area on the design domain. In this approach parts of the design domain which include members and nodes with no potential contribution to the optimal structure are removed. Due to the significantly different mechanical properties, none of these methods can be directly implemented for frame structures, where the rigidity of the joints causes moments and consequently calculating the stresses is more complicated.

Frame structures: Structural optimization for frame structures, specifically for additive manufacturing has been well-studied in the recent years (Ribeiro, Bernardo and Andrade 2021). The ground structure method (Dorn 1964) is still the main approach used for discrete structures such as truss and frame topology optimization (Gao, Liu, Li and Qiao 2017). Larsen, Sigmund and Groen (2018) used a two stage process to optimize the frame structures where, they first solve the truss analog using the homogenization-based topology optimization method, and then implement a size optimization to obtain a near-optimal frame structure. They note that with any post-processing, the resulted truss loses its optimality and the process output will be a near-optimal frame solution.

Li and Chen (2010) implemented an iterative approach to find the optimal frame structure in which the mechanical properties of the resulting structure on each iteration is calculated using a finite element analysis software. In another attempt to find the optimal frame structure, He et al. (2019) start with modeling and solving truss layout optimization as a linear programming problem and, in the second step, use nonlinear programming for geometry optimization of a frame structure by adjusting the node coordinates in the layout resulting from the first step. In their approach the number of nodes in the ground structure is constant (no node adding) but the coordinates of nodes could be changed. During this post-processing rationalization step, two issues with the resulting truss structure (many members with non-zero areas and overlapping members) were addressed. Gally et al. (2015) proposed a nonlinear semidefinite program (SDP) for the robust truss topology design problem with beam elements. They considered moment in the rigid nodes of the structure and derived the element stiffness matrices from the principle of linear momentum with the Galerkin method. Reintjes and Lorenz (2021) implemented a post-processing step on the optimized initial truss structures for the design problem of lattice structures for additive manufacturing. Member adding method together with the concept of principal stress lines have been introduced by Li and Chen (2010) for frame structures designed for additive manufacturing. While the authors used these principal lines for the design of initial structure on the design domain, they indicated the following important properties of principal stress lines for beam elements in a frame structure:

- The displacements of the nodes is proportional to the external load.
- The displacements are inversely proportional to the stiffness matrix or the elastic modulus.
- The direction of the principal stress is not related to the scaling of the external forces, nor the material type for an isotropic material within the range of elastic deformation.
- The principal stress field is mainly related to the topological variables of the given structural design such as the position of external forces and the types of constraints.

Smith, Todd and Gilbert (2013) solved the design problem of frame structures for additive manufacturing while they initially solved the problem as a truss and then defined a threshold (Euler buckling criteria) to be considered on the elements cross-sectional value to overcome the buckling on the elements. As they indicated, although using this approach (linear programming without considering the moments in the joints) is a significant simplification, there are advantages in keeping the problem linear. These advantages include the possibility of finding the guaranteed optimal solution and tackling large-scale problems. Smith et al. (2016) proposed a multi step process to design components for additive manufacturing. They implemented member adding scheme at the initial step of their approach and via a post-processing step they impose minimum area of the members, buckling constraints, and overall stability constraints to modify the resulted truss structure from previous steps for fabrication.

As we mentioned a couple of those implementations, member adding approach have been used to solve mainly truss structures. This is because, for truss structures, the optimization model can be linearized and the dual form can be easily obtained. Also because of linear form of the sub-problems, it is easy to find global optimal solutions via the interior point method (Gao, Li, Pan, Chen and Liu 2017). On the other hand, frame structures with rigid joints and constrained displacement on the nodes, can only be modeled as mixed integer nonlinear nonconvex models, for which the dual form cannot be obtained and their solution process is known to be computationally demanding. Furthermore, the additive-manufacturing-imposed constraints, make the problems even more difficult to solve. That's being said implementations of member adding approach for frame structures composed of multi-step/post-processing procedures which usually yield near-optimal solutions. In an attempt to solve these problems optimally, Toragay et al. (2022) modeled the design problem of frame structures as mixed integer quadratically constrained problem. They showed that commercial solvers are still limited by the size of the problem and only small-size problems can be solved to optimality. They also provide a comparison of three different modeling approaches to solve this problem for additive manufacturing. To the best of our knowledge, there are few member adding approaches for the frame structures, specially when those structures are being designed for additive manufacturing.

In Chapter 4, instead of solving the linear sub-problems or implement multi-step approaches, we relax some of the constraints in the nonlinear model and based on interaction between the mathematical model and mechanical properties we iteratively approximate the large-scale design problems for frame structures considering the additive manufacturability constraints. By interaction we mean, the results of mathematical model reveals the mechanical properties of the structure (specifically the displacements in the degrees of freedom) based on which we add necessary constraints to the mathematical model. We keep the non-linearity of the model as we consider the moments on the rigid joints. As mentioned by Larsen, Sigmund and Groen (2018), the location of the nodal joints has a large influence on the performance of the design. To take that into account, the proposed heuristic method iteratively adds members to the ground structure, and also adjusts the geometry of the base ground structure by adding new nodes to it. Details of the method are given in Chapter 4.

2.3 Metaheuristic related literature review

As discussed above, design optimization of frame structures for AM can be mathematically modeled as mixed integer nonlinear programming (MINLP) models. In general, there are two classes of optimization methods for these problems: stochastic search methods and deterministic search methods. Despite the state-of-the-art enhancements in commercial solvers and computational power of modern computers, using exact methods for deterministic search approaches to optimally solve these nonconvex optimization problems is still a big challenge when implementing these methods to large-scale instances (Toragay et al. 2022). One approach to handle the large-scale topology optimization problems is to design problem-specific heuristic methods which are stochastic but can find near-optimal solutions. In recent years, nature inspired metaheuristics have been adopted to solve structural optimization problems. The most important and promising ones have been surveyed by Saka (2007); Saka and Geem (2013); Zavala et al. (2014); Kaveh and Zolghadr (2014). The inadequacy of exact mathematical optimization approaches and optimality criteria methods for large instances explain the necessity of implementing metaheuristics in the structural optimization (Kicinger, Arciszewski and De Jong 2005).

On the other hand, as mentioned in (Rao and Xiong 2004), the drawbacks of stochastic methods or metaheuristic approaches include the following:

- They are independent of the specific characteristics of the problem.
- Generated information in each step is barely used in nest iterations/generations.
- They slowly converge and sometimes with a low accuracy.

Another drawback of metaheuristics is the increased solution evaluation requirement with an increase in the number of design parameters (Ahrari, Atai and Deb 2020). On the other hand applications of metaheuristics in topology optimization area are comparatively scarce, possibly because of the complexity of these problems, which demand sophisticated specialization of the standard metaheuristics. This drawback may explain why most common test problems for topology optimization are simple, small-scale problems rather than real structures with higher complexity (Ahrari, Atai and Deb 2020).

Because the essential differences between *truss* and *frame* structures were often ignored (Sui, Du and Guo 2006), and similarly to other approaches to solve the optimization of discrete structures, research on the topology optimization of frame structures using metaheuristics is very limited (An and Huang 2017). However, the promising performance of these approaches in solving topology optimization of truss structures might also be effective in frame topology optimization (Cheng and Guo 1997). Various approaches have been suggested to reduce the complexity of using metaheuristics in structural optimization problems, such as using problem-specific knowledge to customize the metaheuristics (Ahrari, Atai and Deb 2020), reducing the solution space size (Dede, Bekiroğlu and Ayvaz 2011), solving structural optimization with a layered method based on an optimization theory of multi-layer systems (Wang and Ohmori 2013), and two-level approximation (Huang and Xia 1995; Li, Chen and Huang 2014) based on approximating the fitness functions to decrease the structural analyses, amongst others.

Inspired by evolution, Genetic Algorithms (GA) are a family of metaheuristic methods to solve combinatorial problems. In GA, each candidate solution is encoded via an encoding scheme and through recombination operators critical information of individual representations are preserved from generation to generation. Each generation is composed of new offspring which are created via proportional reproductive opportunities that are given to good solutions in the previous generation (Whitley 1994).

The nature of the problem that we consider in this work, which is a design optimization with mixed type of variables, makes GA a good candidate for solving the problem. That's because, GA, unlike exact optimization methods, iterates on generations of designs rather than on a single design (Balling, Briggs and Gillman 2006). This characteristic of the GA can potentially yield a variety of good designs in the final generation. For a designer in the conceptual design stage, having the choice of different topological ideas is beneficial (Balling, Briggs and Gillman 2006). According to Erbatur et al. (2000) the main three components in the operation of a GA are:

- the creation of an initial pool of designs,
- combination of the designs in a pool in order to produce better designs,
- obtaining new generations of designs.

Various modifications have been suggested to interpret and convey all the necessary data for the success of GA whenever this method has been implemented for topology optimization. For instance, binary encoding of discrete cross-sectional values for truss structures has been implemented in (Rajeev and Krishnamoorthy 1992) where a modified objective function has been defined to take in account the total constraint violation which in turn convert the constrained problem to an unconstrained one. Tree encoding of structures has been implemented in (Madeira, Pina and Rodrigues 2010) where the authors used the finite element program ABAQUS (Smith 2009) to compute the fitness for individuals. They considered finite elements in a continuum structure where the connectivity of the individual is guaranteed via graph theory concepts such as minimum spanning tree. Specific operators such as simulated binary crossover and parameter-based mutation has been defined for the GA used for truss structures where a real-number encoding is implemented (Deb and Gulati 2001). A modified GA based on the segmentation of the design domain has been proposed in Khodzhaiev and Reuter (2021) to simultaneously optimize the size, shape and topology of transmission towers. A variable length genome was defined for each segment of the design domain. The fitness of each individual was calculated via the application of third-party FEM software. Azad and Hasançebi (2013) proposed an upper bound strategy for reducing the number of structural analyses in metaheuristic based design optimization of steel frame structures. For the most recent applications of metaheuristic methods in structural optimization problems we refer the readers to the literature, such as Azad (2021); Kaveh and Ghafari (2019); Talatahari and Azizi (2020)

Increasing applications of hybrid GAs in the literature validate these approaches' applicability. These approaches, formed by the combination of a GA with local search methods (Seront and Bersini 2000), combine the advantages of random search and deterministic search and provide increased performance when compared to a GA in its simple form or local search algorithms alone. Hybrid methods can improve the convergence speed and computational efficiency.

In Chapter 5 instead of using the traditional encoding that has been used for the GA implementation in structural optimization, we propose a novel path-based encoding. This encoding is designed based on the similarity of the planar structures and graph networks containing all the connections between nodes. This idea is tailored based on the fact that all the elements in the structure which are located on the principal stress lines can contribute to the distribution of the stresses among the structure. Using this encoding will reduce the computational burden of defining and operating on other types of encoding such as binary encoding. This aims to alleviate the intractability of GA for large-scale problems in structural optimization. The proposed encoding paves the way for the construction of a novel hybrid GA which is more powerful than the traditional approaches which suffer from very large number of variables in structural optimization problems.

Chapter 3

Discrete Topology Optimization of Additively Manufactured Lightweight Planar Frame Structures[1]

3.1 Introduction

Design of lightweight structures is an important task for many engineering fields such as mechanical, civil, aerospace, and biomedical engineering. It includes applications in the aerospace, automotive, and medical industries. Preliminary steps of the design process for load-bearing structures usually include:

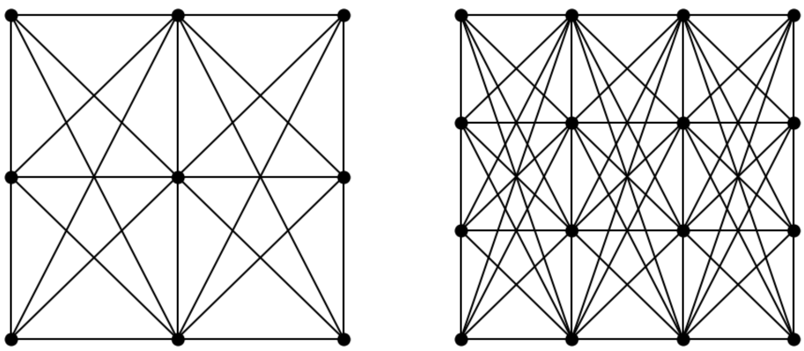
- Identifying the ground structure in which all the potential elements of the design domain are defined.
- Identifying loads and boundary conditions on the design space.
- Considering the type of material in the fabrication stage (physical characteristics).
- Identifying the expected mechanical properties.

In general, based on the type of elements that are considered for the fabrication of the designed structure, design tasks can be divided into two main categories, continuous and discrete design. For instance, design of a truss structure consists of bar elements and a frame structure consists of beam elements are two examples of discrete design. In these examples, the elements of the structure are discrete, most of the time predefined (pre-fabricated), bar or beams elements. Bar elements can carry axial forces while beam elements can, in addition, convey bending moments while exposed to external loads. Truss structures consists of pin-joined bar elements, while frame structures include beam elements that are rigid-joint at the end-nodes of the elements.

[1]This chapter has been published as a journal article in the *Structural and Multidisciplinary Optimization* journal (Toragay et al. 2022)

In recent years, additive manufacturing (AM) has had a significant impact on the design of lightweight structures, since it removes many of the limitations that conventional manufacturing imposed on the fabrication stage (Liu et al. 2018). For instance, for truss and frame structures, AM is capable of fabricating the bar and beam elements of various shapes, sizes and cross-sectional areas. More generally, AM provides significant control on the density of material through the design domain. At the same time, it has its own limitations as well. For instance, when fabricating a frame structure, the angle between the element and printing plane is important and must satisfy certain constraints.

In this chapter, we focus on modeling and solution approaches to the problem of design of load-bearing planar frame structure. We assume that the ground structure includes all of the potential discrete beam elements. The considered ground structure consists of nodes or connection points (joints) for which the coordinates are fixed and given. Figure 3.1 shows two instances of ground structures with 9 nodes and 24 beams in (a) and 16 nodes and 60 beams in (b). Note that in a designed ground structure with $n \times n$ grid, the number of potential elements is $n(n^2 - 1)$.



(a) 3×3 ground structure (b) 4×4 ground structure

Figure 3.1: Instances of ground structures

The design problem then consists of finding a subset of the beam elements and their respective cross-sectional areas, such that it can tolerate a given set of external nodal loads, while minimizing the total weight. The problem combines discrete (which elements are selected) and continuous (cross-sectional areas) decision variables. It is further complicated by the nonlinear relationship between node displacement and structural variables, especially for the case of frame structures (as opposed to the relatively simpler case of truss structures).

This combination of discrete and continuous variables, and non-linear constraints makes the problem exceptionally computationally challenging. To address this challenge we propose a mathematical model, formulated as a Mixed Integer Quadratically Constrained Programming problem, which, to the best of our knowledge, has never been implemented for design of frame structures for AM before. Further, we implement a two-stage, hybrid approach to solve these models more effectively. The proposed approach takes advantage of relative simplicity of a discretized version of the problem in order to improve solvability of the continuous model.

Our computational results show that the proposed model solves the problem more efficiently than the linearized models and faster than the other non-linear models. The necessity of this research comes from the fact that although in recent years commercial solvers enhance their ability in solving large scale problems, in some cases they are still limited by the problem size. The aforementioned complex problem in hand is one of that kind in which state-of-the-art solvers still can not solve large scale problem. In this study we evaluate the capacity of enhanced solvers in handling the design problem in frame structures when solving this problem which modeled using efficient implementation of different modeling approaches. Specially we investigate a new approach to model this problem based on MIQCP modeling approach.

This chapter is organized as follows. After a brief explanation of the design problem, section 3.2 gives the linear and nonlinear models that are suggested in the literature for the frame structures' design problem. We further enhance these models by adding several constraints to adjust them for the AM related conditions. We also explain in this section, the proposed novel mathematical model in the form of MIQCPs for the design problem of lightweight planar frame structures. In section 3.3, we present the results of a set of experimental instances using linear, quadratic, and nonlinear models. We compare the models in terms of the solution quality (resulting structures' weight) and computational effort. Section 3.4 concludes this chapter by summarizing the main results and findings.

3.2 Mathematical Models

We consider three modeling approaches to the frame structures' optimization problem, referred to as: Mixed Integer Non-Linear Programming (MINLP), Mixed Integer Quadratically Constrained Programming (MIQCP) and Mixed Integer Linear Programming (MILP). Main ideas of MINLP and MILP models are adapted from the literature with additional improvements and modifications that enable these models to cover manufacturability constraints as well. Detail of these modifications are given in the corresponding sections. In the proposed MIQCP model we use semi-continuous variables for defining the beam elements' cross-sectional areas. First, we present the notation used throughout the rest of the discussion and give a formal problem statement used for all three models.

List of variables, parameters, and sets

Sets:

$\mathbb{S}_{\text{dof}}$ Set of all degrees of freedom in the ground structure ($3 \times$ number of nodes)

$\mathbb{S}_e$ Set of all potential elements in ground structure

$\mathbb{S}_n$ Set of all nodes in the ground structure

$\mathcal{D}_{(k)}$ Set of all degrees of freedom on node k

$\mathcal{H}_{(k)}$ Set of all elements in the neighbourhood of a node k

$\mathcal{S}$ Set of all (e_1, e_2) pairs with $e_1, e_2 \in \mathbb{S}_e$ such that e_1 may intersects e_2

Parameters:

a_{max} Maximum possible cross-sectional area of each beam

a_{min} Minimum possible cross-sectional area of each beam

u_{max} Maximum possible displacement on each degree of freedom

l_e Preliminary length of beam element e in the ground structure

$\mathcal{N}_{(e)}$ (n_1, n_2) pair of nodes with $n_1, n_2 \in \mathbb{S}_n$ such that element e connects n_1 and n_2

I_e Moment of inertia of element e

θ_e Angle between beam element e and horizontal line

$\mathbf{k}_e$ Stiffness matrix of element e

$\mathbf{K}$	Structure's stiffness matrix	
E	Young modulus of elasticity	
d	Number of all degrees of freedom in the structure	
p_j	External load(rotation or axial force) on degree of freedom j	
$\mathbf{P}$	Vector of all external nodal loads in the structure	
ρ	Density of the used material	

Design Variables:

$\mathbf{a}$	Vector of all elements' cross-sectional area	
a_e	Circular cross-sectional area of the beam element e	$\in \mathbb{R}^+$
x_e	Indicates the inclusion of beam element e in the resulted structure	$\in \{0, 1\}$
x_{ec}	Indicates that profile c is chosen for element e in the resulted structure	$\in \{0, 1\}$
y_k	Indicates the inclusion of node k in the resulted structure	$\in \{0, 1\}$

State Variables:

I_e	Moment of inertia of element e	
$\mathbf{I}$	Vector of moment of inertia of all elements	
u_j	Displacement on degree of freedom j	$\in \mathbb{R}$
$\mathbf{u}$	Vector of displacements on all degrees of freedom	$\in \mathbb{R}^{d\times 1}$

3.2.1 Problem statement and stiffness matrix decomposition

We consider the following general form of mathematical models for the frame structures. We use this general form of mathematical models to find the lightest additively manufactured planar frame structure composed of beam elements that can tolerate the external nodal load while keeping the displacements in all connected nodes in some specific, given ranges.

$$\begin{aligned} \text{minimize} \quad & \text{Weight of the structure} \\ \text{subject to} \quad & \mathbf{K}(\mathbf{a}, \mathbf{I})\mathbf{u} = \mathbf{P} \\ & \text{Manufacturability constraints} \end{aligned}$$

in which the first constraint represents the equilibrium equations that needs to be held in all the feasible solutions. These equations relate the displacement in the degrees of freedom to the related external nodal loads. Second constraint represent all the manufacturability related constraints including the constraints on the cross-sectional areas of the elements, stress on the elements, and limitation on the displacements in the degrees of freedom. As well as constraints to remove the hanging members, and constraints to remove the over-crossing members.

As the displacements are assumed to be small in these structures, the Euler-Bernoulli beam elements can be used in analyzing the mechanical behavior and characteristics of the structure. We generate the ground structure based on the given length, width and number of nodes in each dimensions of the desired structure using the pseudo code given in Appendix B. Each beam element of the structure connects two nodes. Each node in the structure has 3 degrees of freedom. By those degrees of freedom we mean displacements in x and y directions, and the rotational displacement in the node. Hence, each element has 6 degrees of freedom, 3 for each end-node. In truss structures, each end-node of a bar element have only 2 degrees of freedom because the bar elements in truss structures are pin-joint, not rigidly connected, so do not generate moments due to rotational displacements. Unlike truss structures, in frame structures, where the end-nodes are rigidly connected, elements are fixed in their end-nodes so the rotational displacement causes moment forces along the elements. These forces in turn contribute to the stress on the elements that may cause bending or buckling. Truss and frame structures are not only different because of the type of elements that they are made of, they are also different in the complication of the analysis because of having different stiffness matrices. Element stiffness matrix in frame structures are 6×6 which can be decomposed (Stolpe and Svanberg (2003); Stolpe (2004); Rasmussen and Stolpe (2008); Kanno (2016)). Although the details of mechanical relations that we used in the models are included in Appendix A, it worth mentioning the followings here to the sake of smoothness in understanding the models. The stiffness matrix of the structure can be decomposed as the following equation:

$$\mathbf{K} = \sum_{e \in \mathbb{S}_e} k_{e1}\mathbf{b_{e1}}(\mathbf{b_{e1}^\top}) + k_{e2}\mathbf{b_{e2}}(\mathbf{b_{e2}^\top}) + k_{e3}\mathbf{b_{e3}}(\mathbf{b_{e3}^\top}), \tag{3.2.1}$$

in which k_{ei} for each element e are defined as follow:

$$k_{e1} = \frac{a_e E}{l_e}, \qquad k_{e2} = \frac{3a_e^2 E}{4\pi l_e}, \qquad k_{e3} = \frac{a_e^2 E}{4\pi l_e}, \tag{3.2.2}$$

and $\mathbf{b_{ei}}$ is defined as:

$$\mathbf{b_{ei}} = \mathbf{T}_e^\top \times \mathbf{Trans}_e^\top \times \hat{\mathbf{b}}_{\mathbf{ei}}. \tag{3.2.3}$$

Notice that in equations (3.2.3), $\mathbf{b_{ei}} \in \mathbb{R}^{d\times 1}$ is a $d \times 1$ vector from which $\mathbf{K} \in \mathbb{R}^{d\times d}$ matrix can be generated. For the definitions of the matrices $\mathbf{T}_e^\top$, $\mathbf{Trans}_e^\top$, and $\hat{\mathbf{b}}_{\mathbf{ei}}$ that are used in the equation (3.2.3) we refer the reader to Appendix A. We implement this decomposition of the stiffness matrix in the mathematical models explained in this section.

3.2.2 Mixed Integer Non-Linear Programming Model (MINLP)

The MINLP formulation is applied by directly applying the stiffness matrix decomposition to the general formulation, as well as enforcing AM manufacturability constraints.

$$\underset{a_e}{\text{minimize}} \quad W_N = \sum_{e\in\mathbb{S}_e} \rho a_e l_e \tag{minlp0}$$

subject to

$$\left(\sum_{e\in\mathbb{S}_e} a_e k_{e1}\mathbf{b_{e1}}(\mathbf{b_{e1}^\top}) + a_e^2 k_{e2}\mathbf{b_{e2}}(\mathbf{b_{e2}^\top}) + a_e^2 k_{e3}\mathbf{b_{e3}}(\mathbf{b_{e3}^\top})\right)_i \mathbf{u} = p_i \forall i \in \mathbb{S}_{\text{dof}} \tag{minlp1}$$

$$-u_{max} y_k \le u_j \le u_{max} y_k \qquad \forall k \in \mathbb{S}_n \ \ \forall j \in \mathcal{D}_{(k)} \tag{minlp2}$$

$$a_{min} x_e \le a_e \le a_{max} x_e \qquad \forall e \in \mathbb{S}_e \tag{minlp3}$$

$$x_{e_1} + x_{e_2} \le 1 \qquad \forall (e_1, e_2) \in \mathcal{S} \tag{minlp4}$$

$$2y_k \le \sum_e x_e \qquad \forall k \in \mathbb{S}_n \ \ \forall e \in \mathcal{H}_{(k)} \tag{minlp5}$$

$$2x_e \le y_{n_1} + y_{n_2} \qquad \forall e \in \mathbb{S}_e \ \ (n_1, n_2) \in \mathcal{N}_{(e)} \tag{minlp6}$$

$$x_e \in \{0, 1\} \qquad \forall e \in \mathbb{S}_e \tag{minlp7}$$

$$y_k \in \{0, 1\} \qquad \forall k \in \mathbb{S}_n \tag{minlp8}$$

The variables can be split into two categories: *decision variables* are the independent variables related to element placement and their cross-sectional areas; and *state variables*, which are the dependent variables determining the structure response to the external load, i.e., the internal forces, displacements, etc. Note that fixing the decision variables, fully determines the state variables (albeit, through nonlinear equations). The objective function (minlp0), denoted as W_N, gives the total weight of the structure where ρ is the material density. Constraint (minlp1) enforces the equilibrium equations by employing the stiffness matrix decomposition discussed above. More details on the matrix generation are given in Appendix A. Note that by decomposing the stiffness matrix it is possible to separate the variable portion from the fixed portion in the matrix.

To limit the deformation of the structure under the external load, we bound the displacement on the degrees of freedom to be in a pre-defined range in constraint (minlp2). Note that in order to account for the fact that displacement on any nodes can be non-zero if and only if the node is connected to the resulted structure by at least one element, in the model we define a binary variable y_k for each node k in the ground structure to indicate the connectivity of node k to the body of the resulted structure. In this constraint, without loss of generality we assume that $u_{min} = -u_{max}$.

The rest of the constraints explicitly account for some of the AM-related limitations. First, in addition to the upper bound on the cross-sectional area (a_{max}), a lower bound must be enforced, due to the laser and powder interactions. Note though that the lower bound is only applicable to the beams that are present in the structure. Consequently, we define a binary variable, x_e , referring to whether beam e is included. Then constraint (minlp3) enforces the cross-sectional area bounds. Note that this constraint also eliminates the possibility of vanishing elements.

Unlike truss structures, in planar frame structures where the connections are fixed or welded (rigid joints), it is not desired (or even feasible) to have intersecting elements. Constraint (minlp4) eliminates those, by listing all pairs of beams that cannot be simultaneously included in the structure.

Constraint (minlp5) makes sure that the problem does not contain any "hanging elements", which here are defined as nodes with only single beam attached (with the exception of boundary nodes and external load locations). Note that in the design of ground structures we considered another important manufacturibility constraint of the structures which is elements' angle. All the potential elements in the ground structure designed to have horizontal angle $\geq 45°$.

Overall, the problem given is then a mixed-integer non-convex nonlinear programming problem (non-convex property refers to the non-convex continuous relaxation rather than the non-convexity due to integer variables). In general it cannot be linearized without loss of generality. One possible approach, discussed in the following section, is to restrict the continuous cross-sectional areas to only a discrete set of values, in which case linearization is possible.

3.2.3 Mixed Integer Linear Programming Model (MILP)

As noted earlier, traditionally, primary reason for discretizing potential cross-sectional areas (or diameters) of the structural elements in topology optimization was the fact that these elements were usually manufactured and available in pre-defined standard cross-sectional sizes and shapes. AM naturally relaxes this requirement, allowing for continuous decision variables. At the same time, if discretization is applied, then it is possible to linearize the formulation above. Naturally, the resulting problem is not equivalent to the MINLP formulation. Note also that linearization described below relies on introduction of additional binary variables, and hence, it is not clear whether the computational benefit of relaxing nonlinear constraints outweighs the burden due to the extra variables. Our model is similar to the linear model that has been proposed by Kureta and Kanno (2014). We modify it in order to be able to compare the results to the solutions of nonlinear models.

We define the discrete cross-sectional area candidate set as $C = \{0, c_1, c_2, ..., c_{pr}\}$. In general, this set can be element-specific without changing the structure of the model. pr[2] is the number of different non-zero discrete cross-sectional area that we define in the set C (note that $|C| = pr + 1$). For each element $e \in \mathbb{S}_e$ and each $c \in C$, we also define a profile which includes the cross-sectional area of the element (a_{ec}), moment of inertia I_{ec} (calculated using A.0.2) and

[2] abbreviation for profile

length of the element (l_e). We can then define binary variable:

$$x_{ec} = \begin{cases} 1 & \text{if profile } c \text{ is chosen for element } e \\ 0 & \text{otherwise.} \end{cases} \tag{3.2.5}$$

Finally, we can then modify the definitions of k_{ei} as:

$$k_{ec1} = \frac{a_{ec}E}{l_e}, \qquad k_{ec2} = \frac{3a_{ec}^2 E}{4\pi l_e}, \qquad k_{ec3} = \frac{a_{ec}^2 E}{4\pi l_e} \tag{3.2.6}$$

As a result, the values in (3.2.6) are no longer dependent on the variables, and so can be pre-calculated for each profile/element combination. Consequently, we can linearize all constraints with the usual "big-M" method.

$$\underset{a_e}{\text{minimize}} \quad W_L = \sum_{e \in \mathbb{S}_e} \rho l_e \sum_{c \in C} a_{ec} \cdot x_{ec} \tag{milp0}$$

subject to

$$\sum_{e \in \mathbb{S}_e} \sum_{c=0}^{pr+1} k_{ec1} v_{ec1} \mathbf{b_{e1}}(i) + k_{ec2} v_{ec2} \mathbf{b_{e2}}(i) + k_{ec3} v_{ec3} \mathbf{b_{e3}}(i) = p_i \qquad \forall i \in \mathbb{S}_{\text{dof}} \tag{milp1}$$

$$-M\left(1 - \sum_{c=1}^{pr+1} x_{ec}\right) \leq \sum_{c=1}^{pr+1} v_{ecj} - \mathbf{b_{ej}^\top u} \leq M\left(1 - \sum_{c=1}^{pr+1} x_{ec}\right) \qquad \forall e \in \mathbb{S}_e \quad j \in \{1,2,3\} \tag{milp2}$$

$$-M\left(\sum_{c=1}^{pr+1} x_{ec}\right) \leq \sum_{c=1}^{pr+1} v_{ecj} \leq M\left(\sum_{c=1}^{pr+1} x_{ec}\right) \qquad \forall e \in \mathbb{S}_e \tag{milp3}$$

$$\sum_{c \in C} x_{ec} = 1 \qquad \forall e \in \mathbb{S}_e \tag{milp4}$$

$$\sum_{c=1}^{pr+1} x_{ec} \leq \frac{1}{2}(y_{n_1} + y_{n_2}) \qquad \forall e \in \mathbb{S}_e \ (n_1, n_2) \in \mathcal{N}_{(e)} \tag{milp5}$$

$$-u_{max} \cdot y_k \leq u_j \leq y_k \cdot u_{max} \qquad \forall k \in \mathbb{S}_n \quad \forall j \in \mathcal{D}_{(k)} \tag{milp6}$$

$$x_{e_1} + x_{e_2} \leq 1 \qquad \forall (e_1, e_2) \in \mathcal{S} \tag{milp7}$$

$$2y_k \leq \sum_{e} \sum_{c=1}^{pr+1} x_{ec} \qquad \forall k \in \mathbb{S}_n \quad \forall e \in \mathcal{H}_{(k)} \tag{milp8}$$

$$x_{ec} \in \{0,1\} \qquad \forall e \in \mathbb{S}_e \ \forall c \in C \tag{milp9}$$

$$y_k \in \{0,1\} \qquad \forall k \in \mathbb{S}_n \tag{milp10}$$

Here we define a new variable v_{ecj} as the elongation of element e with profile c in direction j, where $j \in \{1, 2, 3\}$ which represents horizontal (x), vertical (y), and rotational directions respectively.

The presented formulation modifies constraint (milp1) and adds constraints (milp2), (milp3) and (milp4). Constraint (milp1) reformulates the equilibrium equation in the linear form. Constraint (milp2) ensures that that $v_{ecj} = \mathbf{b}_{\mathbf{ej}}^{\top}\mathbf{u}$ if element e is present in the structure. Constraints (milp3) restricts the elongation of each element that is not selected to 0. Finally, constraint (milp4) forces exactly one profile from the set C to be selected for element e. The rest of the constraints are directly analogous to the MINLP model.

3.2.4 Mixed Integer Quadratically Constrained Programming Model (MIQCP)

Finally, we consider a modification for the MINLP model, that allows to reformulate it as a mixed-integer (non-convex) quadratically constrained optimization problem. To the best of our knowledge this formulation has not been considered in the literature before.

Recall that a semi-continuous variable is a special kind of variable in a mathematical program, that is restricted to either zero or a continuous value from a pre-specified interval (which does not include zero). In our case, it naturally lends itself to representing corss-sectional areas, since those are restricted to be either zero or between a_{min} and a_{max}. While in general a semi-continuous variable can be replaced with a pair of a binary and a continuous variable and a "big-M" constraint, some modern optimization solvers allow for direct modeling with such variables. This in addition to the recent progress in non-convex quadratically constrained optimization motivates the MIQCP formulation below.

$$\underset{a_e}{\text{minimize}} \quad W_Q = \sum_{e \in \mathbb{S}_e} \rho a_e l_e \tag{miqcp0}$$

subject to

$$\left(\sum_{e \in \mathbb{S}_e} a_e k_{e1} \mathbf{b_{e1}} (\mathbf{b_{e1}^{\top}}) + z_e k_{e2} \mathbf{b_{e2}} (\mathbf{b_{e2}^{\top}}) + z_e k_{e3} \mathbf{b_{e3}} (\mathbf{b_{e3}^{\top}}) \right)_i \mathbf{u} = p_i \quad \forall i \in \mathbb{S}_{\text{dof}} \tag{miqcp1}$$

$$z_e = a_e^2 \qquad \forall e \in \mathbb{S}_e \tag{miqcp2}$$

$$-M\left(\sum_e a_e\right) \le u_j \le M\left(\sum_e a_e\right) \qquad \forall j \in \{j \in \mathbb{S}_{\text{dof}} | j \in \mathcal{D}_{(k)}, \forall e \in \mathcal{H}_{(k)}\} \tag{miqcp3}$$

$$-u_{max} \le u_j \le u_{max} \qquad \forall j \in \mathbb{S}_{\text{dof}} \tag{miqcp4}$$

$$a_{e_1} \cdot a_{e_2} = 0 \qquad \forall (e_1, e_2) \in \mathcal{S} \tag{miqcp5}$$

$$a_e \le a_{max} \cdot y_{n_1} \cdot y_{n_2} \qquad \forall (n_1, n_2) \in \mathcal{N}_{(e)} \tag{miqcp6}$$

$$2y_k \cdot a_{min} \le \sum_e a_e \qquad \forall k \in \mathbb{S}_n \quad \forall e \in \mathcal{H}_{(k)} \tag{miqcp7}$$

$$a_e \in \{0\} \cup [a_{min}, a_{max}] \qquad \forall e \in \mathbb{S}_e \tag{miqcp8}$$

$$z_e \in \{0\} \cup [a_{min}^2, a_{max}^2] \qquad \forall e \in \mathbb{S}_e \tag{miqcp9}$$

Using semi-continuous variables for cross-sectional areas eliminates the need for binary variables x_e used in MINLP formulation. Further, in order to preserve the quadratic structure of the constraints, we also introduce an auxiliary variable z_e, which is set to the square of a_e by constraint (miqcp2). With these, it is now possible to reformulate the problem as a MIQCP. Compared to MINLP formulation, some of the constraints have to be updated to eliminate variables x_e. Specifically, constraint (miqcp3) enforces the limits on the displacements with a "big-M" technique, constraint (miqcp5) removes intersecting beams, constraint (miqcp6) ensures that both ends of an existing beam are included in the structure, and constraint (miqcp7) eliminates hanging beams.

3.2.5 Solution approach

Since most optimization algorithms can usually greatly benefit from either a warm-start or a quality initial solution, we also propose a simple numerical procedure described here, aimed at overcoming the computational challenge. Specifically, we consider a two-stage framework, where in the first stage we solve the simplest possible version of the model: linear formulation with just two profile levels, which can be expected to be relatively computationally inexpensive; then feed the resulting feasible solution (a valid upper bound) to the three considered formulations. Specifically, in stage one, we define $C = \{0, c_{max}\}$ and solve problem (milp0)–(milp10). It is guaranteed that: 1) if the problem is infeasible, then the underlying design problem is infeasible; 2) the obtained optimal solution is feasible to both MINLP and MIQCP and any version of MILP with $\{0, c_{max}\} \subset C$. Finally, the resulting optimal solution can be provided as the incumbent to any of the three formulations (since, MIQCP formulation does not directly include variables x_{ij}, it needs to be amended in this case, as described in Section 3.3).

3.3 Numerical Experiments

In this section we present the numerical results of using the proposed models and two-stage solving approach on a set of test instances. First we show the designed experiment, then the models for solving these instances are given and finally we show the effects of each of design inputs on the performance of each model.

3.3.1 Test instances

A problem instance in question is defined by: the ground structure (number and position of nodes), external loads/boundary conditions (locations and magnitude), material properties and the set C of cross-sectional area candidates (MILP model only). Next we briefly describe each component separately.

Ground structures. Observe that, as defined, the problem allows for ground structures of any shape or form. At the same time, in this case study, for the sake of streamlining the discussion, we only consider regular structures defined as 3×3 (with 9 nodes and 24 elements), 4×4 (with 16 nodes and 60 elements), and 5×5 (with 25 nodes and 120 elements) grids. We assume $50mm \times 50mm$ design domain (with proportionally adjusted inter-nodal distance). The ground structures (along with external loads discussed below) are depicted on Figure 3.2. Note that, all potential beams at angle less than $\geq 45°$ to the ground are excluded due to manufacturability restrictions.

Boundary conditions and external loads. We consider three pairs of boundary conditions and external loads, which in combination with three ground structures results in the nine cases given on Figure 3.2, where blue nodes represent the fixed locations (displacements set to 0) vertical upward and downward loads are applied at the green and red nodes respectively. For each of nine cases, we apply three load magnitudes: 25, 50, and 75 kN for the total of 27 test

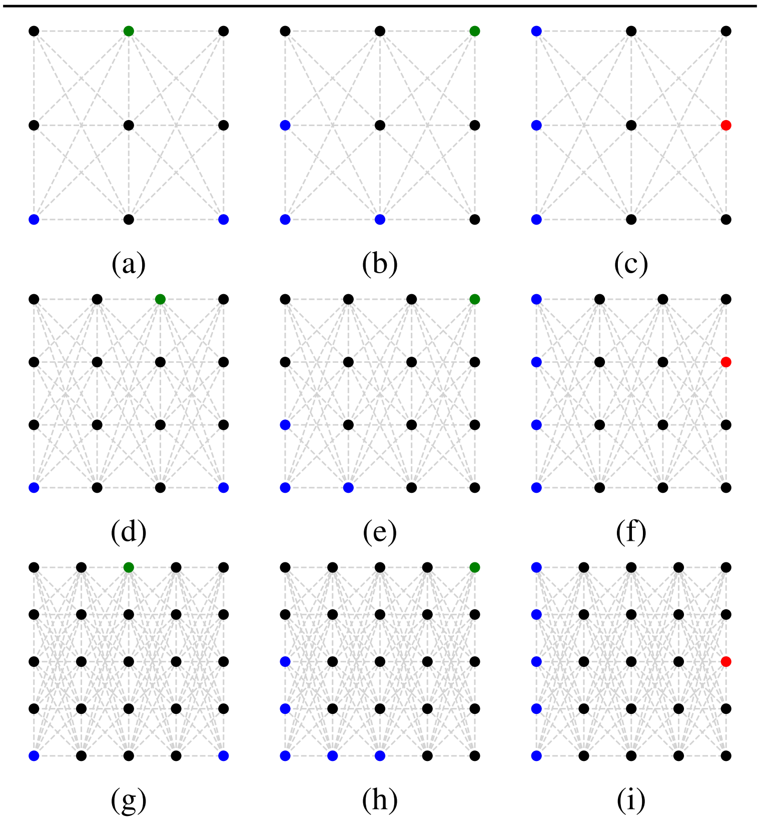


Figure 3.2: Designed experimental ground structures

instances. The loads (and displacement limits) are selected so that the lowest results in fairly trivial structure with just a handful of beams, while the heaviest requires a more substantial structure close to infeasible. It is also worth noting that due to symmetry and since we keep the size of the design space constant, 3×3 structures are restrictions of 5×5 cases, and so we expect that the latter should result in strictly lighter solutions.

Material properties. The material related mechanical parameters are, $E = 109$ GPa (Young's modulus of elasticity for steel) which is assumed to be fixed for all members, displacements are limited to $[-0.095\,\text{mm}, +0.095\,\text{mm}]$ in horizontal and vertical directions. We also assume beam elements in the structure with circular cross-sections for which the radius does not change along the members. Without loss of generality, we assume material density (ρ) as 1.

Candidate sets. All 27 instances were solved with the two exact formulations (MINLP and MIQCP) and three versions of the MILP model with the following cross-sectional area candidate sets:

- $Set_1 = \{0, 0.2, 0.5\}$ (3 candidate cross-sectional radius)
- $Set_2 = \{0, 0.2, 0.3, 0.4, 0.5\}$ (5 candidate cross-sectional radius)
- $Set_3 = \{0, 0.2, 0.25, 0.30, 0.35, 0.40, 0.45, 0.5\}$ (8 candidate cross-sectional radius)

Naturally, all three include 0 as an option to allow for excluding some of the beams, and $Set_1 \subset Set_2 \subset Set_3$. Consequently, we expect that each optimal solution to the subsequent MILP version closer approximates the exact continuous case. On the other hand, each results in more binary variables, and thus, is more computationally expensive. Through the rest of this section we will refer to the models as:

- NL: Mixed integer non-linear model with continuous and binary variables (Sec. 3.2.2).
- QD: Mixed integer quadratically constrained model with semi-continuous variables (Sec. 3.2.4).
- CS1: Linear model that is solved together with Set_1 (Sec. 3.2.3).
- CS2: Linear model that is solved together with Set_2 (Sec. 3.2.3).
- CS3: Linear model that is solved together with Set_3 (Sec. 3.2.3).

Note also that we attempt to solve each instance and each formulation with and without the two-stage initial solution heuristic described above.

Implementation details and solver selection. All model construction has been implemented in Python Optimization Modeling Objects (Pyomo) (Hart et al. 2017; Hart, Watson and Woodruff 2011) and Gurobi's interface for Python (Gurobipy) (Gurobi Optimization 2021). Once modelled, all instances were solved with off-the-shelf commercial solvers, selected as follows. All experiments unless specifically mentioned were performed on a 64-bit Windows 10 Enterprise operating system running on a desktop computer with, Intel® Core™ i7-10700K CPU @ 3.80 GHz, 64.0 GB installed memory, with solver time limit set to 5 hours (18000 seconds).

MINLP is a general mixed-integer programming problem with non-convex continuous relaxation. While there exist a few available exact solvers, these can be unreliable and their relative performance is often instance-dependent. For our experiments we selected BARON

(Sahinidis 2017), which is often considered as one of the best available general purpose global optimization solvers (Neumaier et al. 2005; Kronqvist et al. 2019).

MILP formulation results in a standard mixed-integer linear problem. While it is still NP-hard and computationally challenging, there exist reliable and efficient available solvers. Three of the popular choices that we have considered are: CPLEX, Gurobi and Xpress. We chose to use Gurobi, since: 1) in our preliminary experiments we did not observe consistent pattern in relative performance of the linear solvers for our instances; and 2) Gurobi is the only one that also allows for non-convex MIQP problems.

MIQCP formulation improves on the MINLP by only using quadratic constraints, which introduces additional structure to the problem. Note that due to this structure, we would expect this formulation to outperform the MINLP version even when used with a general global solver. Further, beginning with version 9.1 Gurobi allows for direct application to non-convex quadratic constrained problems. Consequently, we select it for this formulation. Note that Gurobi also allows for explicit use of semi-continuous variables. To facilitate the replication of our results, we have shared all Pyomo, Gurobipy, and Python codes for all models as well as the ground structure generation codes in the author's Github account[3].

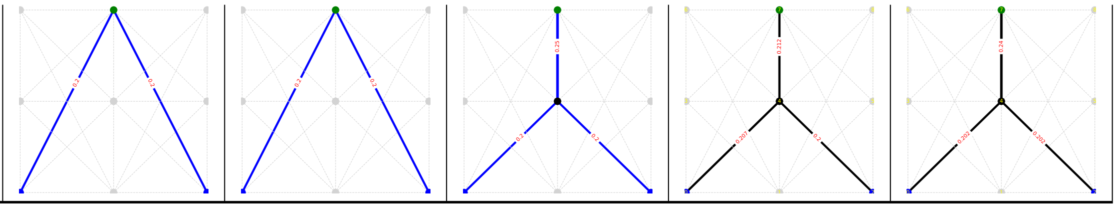
3.3.2 Numerical Results

While discussing the results of the experiments, we first mention our overall observations and conclusions, and then elaborate on those one at a time.

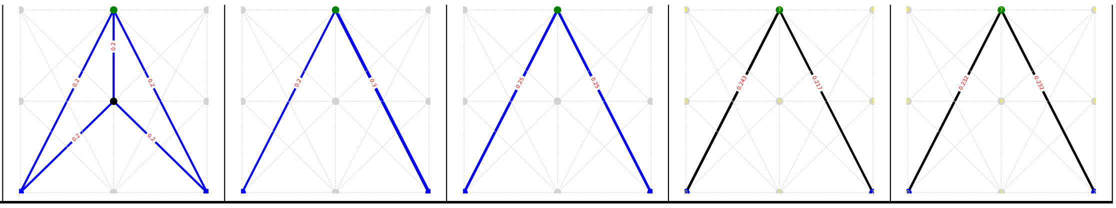
General observations. All resulting structures for each of 27 instances and 5 models are given in the same Github repository. For the purpose of illustration here we only depict on Figure 3.3 all load/model combinations for the case of the ground structure shown in Figure 3.2(a).

In general, the resulting structures can be compared in terms of three metrics: total weight, solution time and optimality gap. Observe that in theory, formulations MINLP and MIQCP are equivalent and hence at optimally should produce structures of the same total weight, while potentially requiring different computational effort. On the other hand, MILP is a restricted

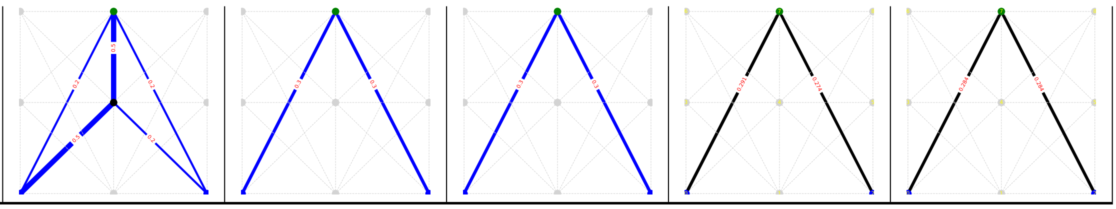
[3]https://github.com/oguztoragay



Model	CS1	CS1	CS2	CS2	CS3	CS3	QD	QD	NL
Weight	14.0496	14.0496	14.0496	14.0496	13.7945	13.7945	12.7216	12.7216	13.0247
Time	0.66	0.63	0.58	0.61	1.44	0.86	0.85	1.21	28.563
Gap	0	0	0	0	0	0	0	0	0
Load	25	25	25	25	25	25	25	25	25
WS	Yes	No	Yes	No	Yes	No	Yes	No	No



Model	CS1	CS1	CS2	CS2	CS3	CS3	QD	QD	NL
Weight	26.077	26.077	22.8306	22.8306	21.9525	21.9525	18.6705	18.6705	18.8614
Time	1.77	1.95	1.71	1.76	5.29	4.79	1.11	1.52	134.022
Gap	0	0	0	0	0	0	0	0	0
Load	50	50	50	50	50	50	50	50	50
WS	Yes	No	Yes	No	Yes	No	Yes	No	No



Model	CS1	CS1	CS2	CS2	CS3	CS3	QD	QD	NL
Weight	65.8955	65.8955	31.6117	31.6117	31.6117	31.6117	28.0921	28.0921	28.292
Time	17.14	16.36	9.54	9.46	63.77	57.77	1.97	2.77	723.117
Gap	0	0	0	0	0	0	0	0	0
Load	75	75	75	75	75	75	75	75	75
WS	Yes	No	Yes	No	Yes	No	Yes	No	No

Figure 3.3: All the structures resulted from solving Figure 3.2(a) ground structure using all the models

formulation, and so at optimally it should produce a structure heavier than the optimal MINLP structure. Further, the more options for cross-sectional areas are considered the closer the MILP should approximate the exact optimal solution. Note though that since a time limit of 5 hour was allied to all instances, these relationships may not hold for all cases in the experiments below. In Tables 3.1, 3.2, and 3.3 we respectively report total weight of the resulted structure, solution time for the models and the optimality gap when time limit reached by the solver. Note that we exclude MINLP model from the results after observing that this model cannot solve most of the models in the allocated time limit.

		Structures from Fig. 3.2								
Load	Model	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
25	CS1	14.05	70.18	77.18	12.70	106.15	79.21	14.05	50.03	81.88
	CS2	14.05	38.10	59.63	12.70	84.10	69.05	14.05	43.34	70.51
	CS3	13.80	38.10	56.09	12.70	82.54	68.27	13.80	38.64	75.32
	QD	13.58	33.69	54.34	12.70	97.40	65.52	13.58	40.48	55.27
50	CS1	26.08	86.67	150.34	29.07	202.62	130.34	26.07	86.67	133.55
	CS2	22.83	79.60	110.36	24.84	202.62	112.14	22.83	79.60	110.35
	CS3	21.95	77.09	110.36	22.84	202.62	110.36	19.97	77.54	110.63
	QD	18.86	72.04	114.12	21.46	202.62	129.02	18.88	78.54	131.05
75	CS1	65.89	130.58	189.61	44.74	-	174.19	39.15	148.53	166.48
	CS2	31.61	130.58	168.41	35.50	-	158.94	31.61	144.99	150.67
	CS3	31.61	126.85	164.68	34.01	-	157.24	29.19	143.25	150.99
	QD	28.29	126.68	163.41	32.19	-	174.10	28.61	148.46	166.39

Table 3.1: Weight of all the structures solved by 4 models

From Table 3.1 we can infer that there is a direct relationship between the magnitude of external load and the weight of structure. The trend of increasing weight from top to bottom of each column (compare the blocks of 4) shows this observation. Also in each block there is a descending trend. That is, for each load magnitude, "weight of CS1" $\geq$ "weight of CS2" $\geq$ "weight of CS3" $\geq$ "weight of QD".

		Structures from Fig. 3.2								
Load	Model	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
25	CS1	0.76	57.17	10.98	2.52	**TL**	**TL**	23.38	**TL**	**TL**
	CS2	1.15	65.69	444.59	4.04	**TL**	**TL**	28.13	**TL**	**TL**
	CS3	0.87	630	**TL**	4.88	**TL**	**TL**	62.98	**TL**	**TL**
	QD	1.29	217.46	**TL**	25.34	**TL**	**TL**	**TL**	**TL**	**TL**
50	CS1	1.00	24.70	41.99	78.45	**TL**	**TL**	**TL**	**TL**	**TL**
	CS2	0.91	1568.6	6043.4	100.4	**TL**	**TL**	15240	**TL**	**TL**
	CS3	1.54	**TL**	**TL**	120.92	**TL**	**TL**	1488	**TL**	**TL**
	QD	1.59	**TL**	**TL**	702.11	**TL**	**TL**	**TL**	**TL**	**TL**
75	CS1	2.34	31.97	17.19	696.28	-	**TL**	**TL**	**TL**	**TL**
	CS2	1.83	3184	8429	4372	-	**TL**	**TL**	**TL**	**TL**
	CS3	6.55	**TL**	**TL**	**TL**	-	**TL**	**TL**	**TL**	**TL**
	QD	2.44	**TL**	**TL**	4184	-	**TL**	**TL**	**TL**	**TL**

Table 3.2: Solving time of all the structures solved by 4 models (**TL**: Time Limit)

Table 3.2 summarizes the solving time for all instances. We use **TL** when the model reaches the dedicated 5 hours (18000 seconds) solving time. For these situation we reported best known result. We can infer that regardless of the external load's magnitude, solving time for the MIQCP model is relatively higher than the other models. Also, comparing CS1 , CS2 , and CS3 , for the linear models, the higher the number of variables the longer solving time is.

We further report in Table 3.3 the optimality gap of each model in solving the instances. Gap $= 0$ shows the optimality of the model in solving that specific instance, Gap > 0 shows that the model can not prove the optimality and "dash line" indicate that instance/load/model combination is infeasible. The design structures in Figure 3.2(e), is infeasible when the external load is chosen as 75 kN. The infeasibility of the "stage one" for this structure/load combination revealed that there is no solution for the original problem.

		Structures from Fig. 3.2								
Load	Model	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
25	CS1	0	0	0	0	57.3	62.90	0	62.06	76.36
	CS2	0	0	0	0	64.06	62.71	0	58.34	74.06
	CS3	0	0.01	19.16	0	70.02	65.88	0	55.90	79.42
	QD	0	0	9.37	0	68.72	58.33	2.45	64.79	75.03
50	CS1	0	0	0	0	67.65	72.41	10.02	75.34	85.73
	CS2	0	0.01	0	0	84.28	76.59	0	76.85	82.66
	CS3	0	28.49	53.52	0	86.93	78.56	0	76.76	84.38
	QD	0	17.99	38.61	0.01	91.23	78.13	17.83	78.98	87.77
75	CS1	0	0	0	0	-	66.78	42.47	83.18	87.48
	CS2	0	0.01	0	0	-	80.59	39.90	85.65	86.48
	CS3	0	46.76	60.77	19.55	-	83.52	33.53	86.72	86.88
	QD	0	31.16	39.8	0.01	-	78.90	39.52	85.64	88.17

Table 3.3: Optimality gap (%) of all the structures solved by 4 models

The discussion here shows the trade off between, on one hand, number of members in the cross-sectional candidate set, and on the other hand the problem's solving time and quality of the result. By increasing the number of members in the set, models can find better solutions (lighter structures), while solving problems with more variables takes more time and computational effort.

3.3.3 Influence of the two-stage approach on MIQCP

In Table 3.4 we report the solution time for 54 combinations of structure/load/approach ($9 \times 3 \times 2$) when we consider approaches $= \{$"with two-stage", "without two-stage"$\}$ and the loads $= \{25\,\text{kN}, 50\,\text{kN}, 75\,\text{kN}\}$, for all the 9 ground structure instances given in Figure 3.2. We set a solution time limit of 3600 seconds for all problems. We report the optimal solution unless the solution time reaches to 3600. For that cases (for instance column (d) load 75) the best known solution in the termination of the solving process have been reported. Note that in Table 3.4, 4 out of 9 instances could not be solved without using the two-stage approach when the load $= 25$. This number for load $= 50$ increases to 5 out of 9 and for load $= 75$ reaches to 7 out of 9. In total for $26/27$ structure/load combinations the "with two-stage" approach can find at least a feasible solution (8 are optimal) while "without two-stage" approach can find at least a feasible solution for only $11/27$ instances (8 are optimal). In $7/8$ instances that both approaches can find the optimal solution, "with two-stage" approach outperform "without two-stage" approach in terms of solving time. The largest improvement by 69% happened in ground structure (g) as shown in Figure 3.2.

		Structures from Fig. 3.2								
Load	Approach	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
25	two-stage	0.59	8.52	909.39	30.04	3600	3600	470.76	3600	3600
	one-stage	0.96	10.90	947.62	21.61	-	-	1504	-	-
50	two-stage	0.88	3600	3600	117.83	3600	3600	3600	3600	3600
	one-stage	1.27	3600	-	158.51	-	-	3600	-	-
75	two-stage	1.77	3600	3600	3600	-	3600	3600	3600	3600
	one-stage	1.90	-	-	3600	-	-	-	-	-

Table 3.4: MIQCP with vs. without two-stage approach

3.3.4 Comparison of MIQCP and MINLP models

The main difference between MIQCP and MINLP models, which is also one of the contributions of this article, is that, we defined the semi-continuous variables for the cross-sectional area and by that, we eliminate some of the binary variables from the MINLP model. To verify the usability of the MIQCP model versus the MINLP model, we compare the solution time and the results of these models for all the ground structure instances given in Figure 3.2. Again for this set of experiment we consider the solving time limit as 3600 seconds for all problems. We solve the MINLP models using the commercial solver BARON (in which MILP solver is set to be CPLEX and NLP solver is IPOPT), while for the MIQCP the solver is Gurobi 9.1. Table 3.5 presents the solving time of MINLP compared to MIQCP which is solved using two-stage approach.

		Structures from Fig. 3.2 (Solving time)								
Load	Model	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
25	MIQCP	0.59	8.52	909.39	30.04	3600	3600	470.76	3600	3600
	MINLP	5.83	3600	3600	3600	-	3600	-	-	-
50	MIQCP	0.88	3600	3600	117.83	3600	3600	3600	3600	3600
	MINLP	28.98	3600	3600	3600	-	3600	3600	-	-
75	MIQCP	1.77	3600	3600	3600	-	3600	3600	3600	3600
	MINLP	286.35	-	3600	3600	-	3600	-	-	-

Table 3.5: Comparison of solving time (in second) for MIQCP vs. MINLP

MINLP model can find at least a feasible solution for $15/27$ instances while this number for the MIQCP model is $26/27$. Also MINLP model could prove the optimality in only 3 instances while MIQCP model using two-stage approach could find 8 optimal solution in the dedicated solution time limit of one hour. In Table 3.6 the structural weight of the solved ground structure instances are reported. When both models can find the optimal solution, the reported weight are the same (ground structure (a) with all the load options). In $24/27$ solved instances we can see that MIQCP outperform MINLP in terms of structural weight of the solved (at least feasible) instances.

		Structures from Fig. 3.2 (weight)								
Load	Model	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
25	MIQCP	13.58	33.73	42.29	12.70	108.22	69.35	12.19	45.92	59.01
	MINLP	13.58	39.19	65.34	17.76	-	86.11	-	-	-
50	MIQCP	18.86	70.96	114.06	19.01	202.62	137.78	20.38	86.67	149.80
	MINLP	18.86	80.55	125.10	21.46	-	164.6	35.18	-	-
75	MIQCP	28.29	130.58	163.67	29.05	-	147.2	49.4	130.6	166.5
	MINLP	28.29	-	169.14	32.19	-	218.7	-	-	-

Table 3.6: Comparison of objective value for MIQCP vs. MINLP

3.3.5 Two examples of the solved structures

Here we show two more instances that we solved using all the models. For the QD model in solving both instances, we set the following parameters for Gurobi: `NonConvex` = 2, `Threads` = 16, `FeasibilityTol` = $1e^{-5}$, `Heuristics` = 1, `MIPFocus` = 1, `Presolve` = 2, `PreQLinearize` = 2, and `TimeLimit` = 7200. We reduced feasibility tolerance to enforce finding a feasible intermediate solution in both linear and quadratic models which causes slight differences in the solutions comparing to the results that we provided previously in Table 3.1, 3.2, and 3.3.

The ground structure of the first instance is shown in Figure 3.4 where we indicate the boundary conditions with blue nodes and the 50 kN load with an upward arrow on the green node. In the remaining sub-figures of Figure 3.4, we show the optimal solutions which obtained using CS1, CS2, CS3, QD, and NL (best known result is shown). Note that in Figure 3.4 we magnify the cross-sectional areas to be visually notable. The following Table 3.7 shows the solution time and weight of the optimal structure resulted from these models.

Model	CS1	CS2	CS3	QD	NL
Solving time	341.36	326.66	400.31	369.86	7200
Optimal weight	29.07	24.84	22.84	21.46	21.46

Table 3.7: Solution for instance Fig. 3.2(d)

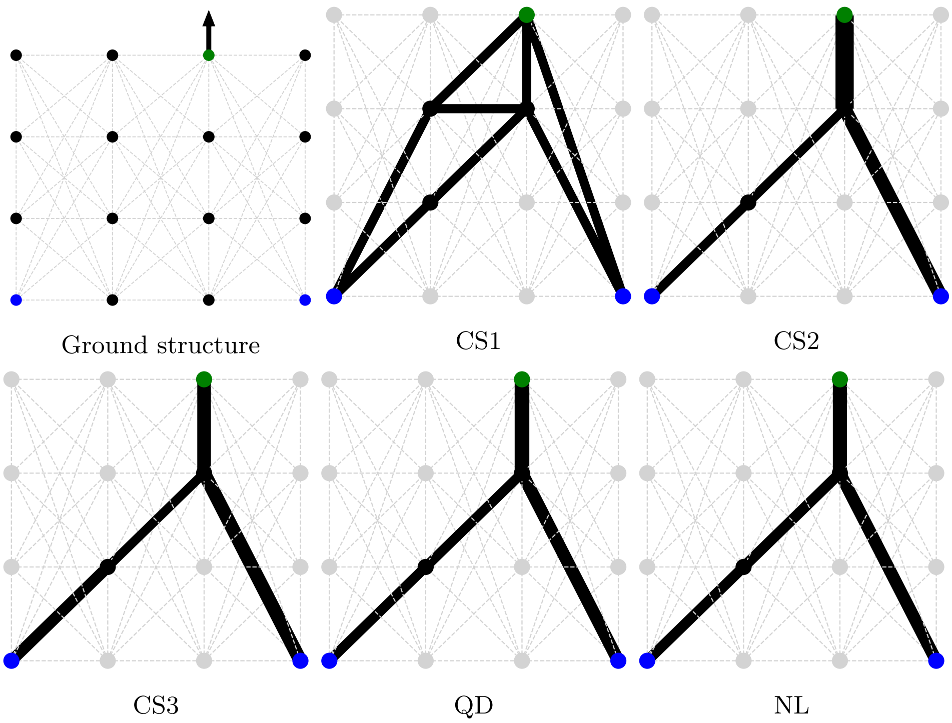


Figure 3.4: Optimal structures resulted from Fig.3.2 (d)

In the second example we solved a 5×5 structure. Load magnitude is changed to 25 kN which is indicated with a downward arrow on the ground structure of Figure 3.5. Table 3.8 shows the best weight that found by the models in the limited solving time as well as the optimality gap at the termination of solving process. Notice that none of the models could guarantee the optimality in the dedicated solution time. Moreover the NL model could not generate a feasible solution during the specified time.

Model	CS1	CS2	CS3	QD	NL
Best weight	76.42	54.44	52.82	68.67	-
Optimality gap	79.6%	73.72%	74.76%	79%	-

Table 3.8: Solution for instance Fig. 3.2(i)

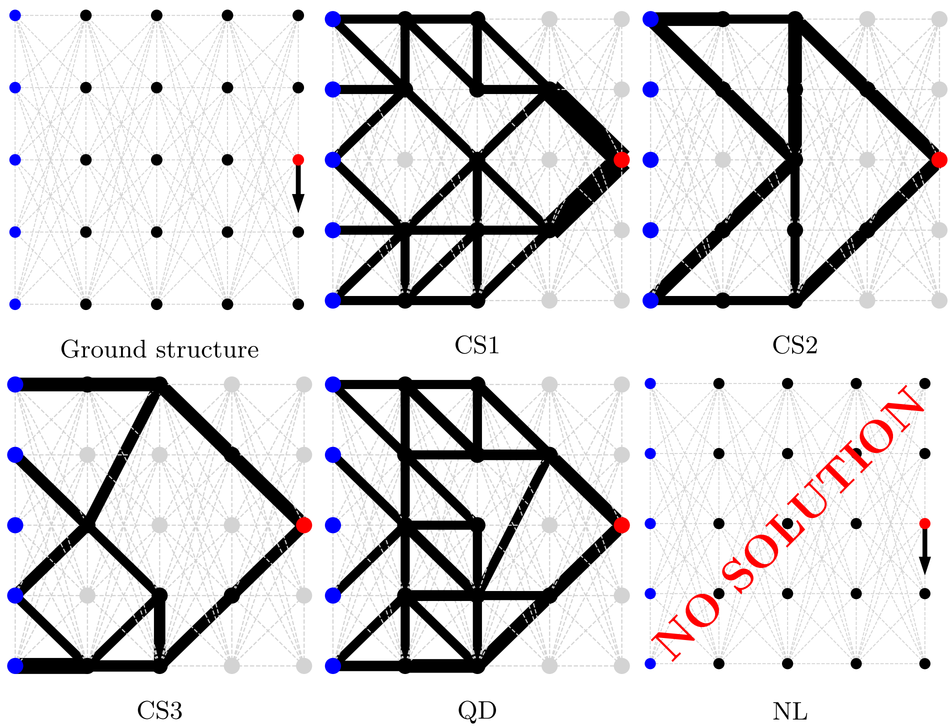


Figure 3.5: Optimal structures resulted from Fig.3.2 (i)

3.4 Conclusions

In this chapter we modeled and solved the design problem of planar frame structures for additive manufacturing fabrication. We modeled the problem using three different mathematical modeling approaches and compare the results of these models in terms of structures' weight and the solving time. Although for all the models the objective function is a linear combination of the elements' volume, because of the non-linearity in the equilibrium equations in the constraint, the base models is a nonlinear, non-convex model (MINLP). Using "big M" method and decomposition of the stiffness matrix we linearized the model using the cross-sectional sets

SC1, SC2, SC3. All three formulation can be solved with off-the-shelf commercial solvers. Note though, each poses significant computational challenges. Specifically, MINLP formulation is explicitly non-convex, and while there exist general-purpose global optimization tools (e.g., BARON, used in our experiments), they are still rather limited in their ability to solve practical problems, in the absence of some particular problem structure. MIQCP formulation partially alleviates this issue by relying on quadratic constraints only. At the same time, the formulation is still non-convex and so still poses significant computational challenge. All these models could be successfully implemented to solve various ground structures. Our results show that the proposed quadratic model out-performs the MINLP model and in most cases (including all the cases that reached optimality) resulted in lighter structures compared to the linearized models. Using quadratic model together with the proposed two-stage approach to solve the designed experimental instances, we could find at least a feasible solution for 26 out of 27 structure/load combinations. We observe that two-stage approach helps solving the problem in two directions:

- Infeasibility of the first stage proves the infeasibility of the instance.
- At least a feasible solution can be found for all of the instances for which the first stage resulted in a solution.

Finally, the linear formulation can rely on extremely efficient modern MIP solvers (CPLEX, Gurobi, etc). The drawback of this formulation is that in order to eliminate non-convexity it introduces a large number of binary variables, especially if more than a handful of profile options is considered. Consequently, while linear, this model is still hard to solve, as evidenced by our experiments discussed in this chapter.

Chapter 4

On optimization of lightweight planar frame structures: an evolving ground structure approach[1]

4.1 Introduction

Lightweight structures have many applications in different engineering areas such as, automotive, aerospace or medical industries, among many others. Optimal design of lightweight structures becomes even more important when additive manufacturing (AM) is considered for fabrication of the parts, since it allows for exceptional freedom in the design process. For example, functionally graded lattice structures have been considered for design and fabrication of patient-specific orthopedic implants considering individual parameters of synthetic bone grafts such as porosity, pore size, shape, and permeability for better biological performance (Becker et al. 2009). The aim of this chapter is to revisit mathematical optimization-based approaches to the problem of designing light-weight frame structures, focusing on additive manufacturing and its corresponding manufacturability constraints.

We specifically consider a particular approach to design AM lightweight parts – mathematical optimization of planar frame structures. Weaver and Gere (2012) defined a planar frame as a collection of members lying in a single plane where the joints between members are rigid (as opposed to truss structures, where connections are not rigid). The forces acting on a frame and the translations of the frame, naturally, are in the plane of the structure. The internal stress resultants, acting at any cross section of a frame member, may consist in general of a bending moment, a shearing force, and an axial force. Frames are different from the trusses in terms of conveying the moments in their rigidly connected members. In AM frames the designer can choose any size or shape for the members, making the problem distinct from the usual steel frame design (with column and story) that have been studied in civil engineering and the

[1]This chapter, as a journal article, is under review with the *Optimization and Engineering* Journal

construction industry. In the latter case, the elements can only be selected from a pre-defined set of commercially available cross-sections and the design requirements were imposed from specific design codes such as the “steel construction manual” by American Institute of Steel Construction(AISC 2005).

For the considered AM frame structure optimization problem we assume a given design domain with given boundary conditions and static external loads. We also adopt the well known *ground structure method*, which is often used in the literature for analysis of discrete structures (Dorn 1964; Bendsoe and Sigmund 2013). Here, a grid consisting of nodes and potential connections between nodes (candidate members or beam elements) on the design domain is given. Figure 4.1(a) shows an example of a design domain, which is meshed into a 5×9 grid in Figure 4.1(b). Note that in general the design domain is not restricted to any particular shape, but we will use rectangular areas without loss of generality to simplify the discussion. Using this grid, Figure 4.1(c) shows the ground structure with all possible connections. Note that, importantly, not all of those members are valid for the design process because of AM imposed manufacturability constraints, which limit the extent to which overhanging layers are allowed. We will assume that potential elements at angles to the ground below certain critical value are disallowed. Without loss of generality, following (Wang, Yang, Yi and Su 2013), we will use $45°$ as the threshold in this research effort. Consequently, Figure 4.1(d) shows the ground structure with all the invalid candidate members excluded.

The problem then is to find the lightest load-bearing planar frame structure, while controlling the displacement on all nodes over the design domain. The decision variables in the mathematical model for this design problem are the subset of beams selected from the ground structure and the cross-sectional areas of the elements. Equilibrium equations are used to explain the governing mechanical behaviour and characteristics of the selected structure. These equations need to hold for a design to be feasible. In addition, we also explicitly model manufacturability constraints, which include: disallowing over-crossing elements, lower and upper bounds on the cross-sectional area of each element, and a minimum build angle of the beams in the resulting structure. Both of these sets of constraints (mechanical equilibrium and manufacturability) play important roles in the particular computational challenge of solving the problem under consideration, as will be discussed below.

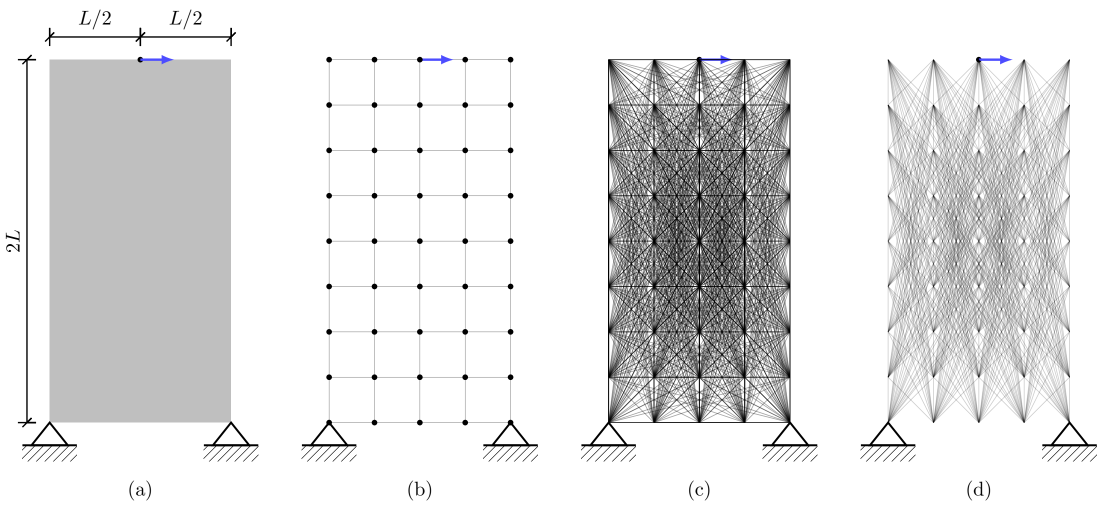


Figure 4.1: (a) design domain with boundary conditions and external load; (b) nodes of a ground structure; (c) a fully connected ground structure; (d) a restricted ground structure excluding non-manufacturable elements.

The resulting optimization problem can be formulated as a mixed integer nonlinear mathematical program with a nonconvex continuous relaxation (from here on we will refer to such problems as simply nonconvex mixed-integer nonlinear programs, or nonconvex MINLP). In general, having an order of $\mathcal{O}(n^2)$ variables (where n is number of nodes in the ground structure), such models are known to be very hard to solve to optimality, even given significant recent progress in both software and hardware tools. In a previous effort, Toragay et al. (2022) investigated the potential to employ off-the-shelf commercial solvers to obtain exact solutions by considering different approaches to formulating the mathematical model itself. We have observed that even the most powerful solvers are still severely limited by instance size, and global optimal solutions can only be obtained for very small structures (up to at most 16 nodes in the ground structure depending on the instance).

Consequently, the goal in this chapter is to consider heuristic approaches that can achieve good (or even near-optimal) solutions in a reasonable run time. Specifically, we propose a problem-specific iterative heuristic method to solve larger-scale problems of this type. The approach starts with a severely reduced ground structure (referred to as minimal ground structure) with relaxed manufacturability constraints, for which a (locally optimal) solution can be obtained easily. We then iteratively add promising elements to the ground structure and re-solve the resulting problem. Importantly, we avoid having to model the master problem with integer variables by reformulating manufacturability constraints. Specifically, we avoid intersecting beams by adding a new structure node at the places where intersections occur, a technique that has not been explored in the literature before for frame structures specifically. The build angle constraint is accounted for through the design of the ground structure, and the bounds on cross-sectional elements are enforced directly.

The main contribution of the chapter is in developing a heuristic approach to lightweight planar frame optimization problem. We propose an iterative method that allows for both element and node addition. Unlike existing approaches, our algorithm is not based on solving a truss reformulation, and instead relies on obtaining locally optimal solutions to nonlinear frame subproblems. Consequently, we argue that it is better suited for identifying near-optimal solutions to the overall design problem, since it does not rely on postprocessing. Further, our approach allows for a collection of AM-specific manufacturability constraints, which, in general, significantly complicate the problem. Most importantly, we are able to avoid the issue of crossing beams by allowing for new node creation, which further improves solution quality.

The remainder of this chapter is organized as follows. After briefly outlining the design problem in general, in Section 4.2 we first present the general form of the master mathematical model and then the relaxed version of it, obtained by relaxing manufacturability constrains and implementing a stiffness matrix decomposition technique. We then present the proposed heuristic approach. We test the performance of the proposed method in solving various design optimization benchmark problems as reported in Section 4.3. Section 4.4 concludes the chapter by summarizing the main findings and providing a road map for future research.

4.2 Methodology

4.2.1 General mathematical programming formulation for frame structures

The problem of designing a lightweight additively manufactured frame structure under given load is constructed as a mixed integer, nonlinear, nonconvex program, for which the relaxations are also non-convex. As discussed in Smith, Todd and Gilbert (2013), the existence of moments at the joints is the main reason for the need for nonconvexity, and the problem is known to be NP-hard (Wang, Wang, Yang, Liu, Tong, Tong, Deng, Chen and Liu 2013). The mechanical behavior of the structure can be described through the equilibrium equations. These define the mechanical interactions between the external nodal loads and the elongations on the elements, or equivalently, displacements on the end nodes, which in turn cause the internal stresses on the elements. From a structural analysis point of view, these equations should be coupled and considered for all the elements simultaneously. To this end, the equilibrium equations can be written in the following matrix form, where $\mathbf{K}$ represents the stiffness matrix of the structure, $\mathbf{U}$ indicates the vector of displacements on the degrees of freedom and $\mathbf{P}$ is the vector of external nodal loads,

$$\mathbf{KU} = \mathbf{P}. \tag{4.2.1}$$

The expression is a system of N equations, where N is the total number of degrees of freedom in the structure, $\mathbf{K}$ is a $N \times N$ matrix while the $\mathbf{U}$ and $\mathbf{P}$ are $N \times 1$ vectors. The equilibrium equations, coupled with constraints bounding node displacements can serve to ensure that the resulting structure is mechanically acceptable. Another set of constraints arises from the manufacturability limitations imposed by AM. As mentioned before these constraints eliminate over-crossing members, impose bounds on the cross-sectional areas of each element, and aim for reducing failure in the fabrication process of AM by controlling the print angle of the elements in the structure.

The primary design variables are cross-sectional areas for each potential element (including zero, if an element is not included). Secondary variables can be defined for node displacements, elements of matrix $\mathbf{K}$, etc., which are all directly determined based on the values of primary variables through the equilibrium equations. Finally, note that in order to model the manufacturability constraints, binary variables should also be defined for each node and each element indicating their presence or absence, which significantly complicates the model.

The problem can be categorized as a topology optimization problem in which both geometry and size of the structure can be modified. For the categorization of other types of problems in the field of structural optimization we refer to Dorn (1964) and Bendsoe and Sigmund (2013). The following optimization problem outlines the resulting model,

$$\textbf{minimize} \text{ weight of the structure} \tag{4.2.2a}$$
$$s.to \tag{4.2.2b}$$
$$\mathbf{KU} = \mathbf{P} \tag{4.2.2c}$$
$$u_{min} \leq \mathbf{u} \leq u_{max} \tag{4.2.2d}$$
$$\text{manufacturability constraints.} \tag{4.2.2e}$$

A detailed description of how each element can be mathematically expressed is given by Toragay et al. (2022), where three variations are presented: a general nonlinear, a linearized version (with a discrete set of element diameters) and a quadratic version. As previously discussed, exact globally optimal solution can only be obtained for very small instances. Two distinct sources of computational challenges can be identified: non-convex equilibrium equations and binary variables required to formulate manufacturability constraints.

In the remainder of this section we present an approach to constructing an iterative approximation scheme, where a specially constructed relaxed master problem is repeatedly re-solved for gradually more advanced ground structures. The relaxed problem is constructed by eliminating manufacturability constraints, which allows us to reformulate the problem without binary variables. Note that while obtaining a provably global optimal solution for a non-convex problem (e.g., equality constraints with nonlinear terms) is still a challenging optimization problem, in the proposed scheme we only require locally optimal solutions on each iteration. Next, we present a detailed mathematical description of the relaxation and then discuss the main algorithm.

4.2.2 Relaxed master optimization model

For a given ground structure, i.e., a set of candidate edges $\mathcal{E}$ and candidate nodes $\mathcal{N}$, a relaxed model, which considers the equilibrium equations and feasibility of the structure (i.e., bounds on node displacements), is given as follows. The notation is explained in more detail below.

$$\text{minimize} \quad \sum_e a_e l_e, \tag{4.2.3a}$$

$$\text{subject to} \quad \left(\sum_e a_e k_{e1} \mathbf{b_{e1}} (\mathbf{b_{e1}^\top}) + a_e^2 k_{e2} \mathbf{b_{e2}} (\mathbf{b_{e2}^\top}) + a_e^2 k_{e3} \mathbf{b_{e3}} (\mathbf{b_{e3}^\top}) \right)_j \mathbf{u} = p_j \tag{4.2.3b}$$

$$0 \leq a_e \leq a_{max} \tag{4.2.3c}$$

$$-u_{max} \leq u_j \leq u_{max} \tag{4.2.3d}$$

$$a_e,\ u_j \in \mathbb{R}. \tag{4.2.3e}$$

Variables a_e and parameters l_e give respectively the cross-sectional area and the initial length of beam element $e \in \mathcal{E}$. Recall that each beam element $e \in \mathcal{E}$ has 6 degrees of freedom (three for each end node). Consequently, we can define u_j representing the displacement for each degree of freedom j. Objective function (4.2.3a) minimizes the volume of the structure which is directly proportional to its weight. Constraint (4.2.3c) bounds the cross-sectional area for all candidate beams, while (4.2.3d) restricts the displacements on all of the degrees of freedom in the structure. Note that a non-zero lower bound on cross-sectional area would require a binary variable, and so we do not include it in the relaxation. We also consider, without loss of generality, that $u_{min} = -u_{max}$. Finally, (4.2.3b) is a formulation of the equilibrium equations in the matrix form, using a decomposition scheme proposed in Kureta and Kanno (2014) and Hirota and Kanno (2015), which is obtained as follows. Components of the stiffness matrix for element $e \in \mathcal{E}$ are defined as:

$$\widehat{\mathbf{b_{e1}}} = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad \widehat{\mathbf{b_{e2}}} = \begin{pmatrix} 0 \\ 2/l_e \\ 1 \\ 0 \\ -2/l_e \\ 1 \end{pmatrix}, \quad \widehat{\mathbf{b_{e3}}} = \begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad \forall e \in \mathbb{S}_e, \tag{4.2.4}$$

$$k_{e1} = \frac{a_e E}{l_e}, \qquad k_{e2} = \frac{3 I_e E}{l_e}, \qquad k_{e3} = \frac{I_e E}{l_e}. \tag{4.2.5}$$

As a result, the member stiffness matrix in local coordinates can be obtained as follows,

$$\mathbf{k}_e = \sum_{z=1}^{3} k_{ez} \widehat{\mathbf{b}}_{\mathbf{ez}} (\widehat{\mathbf{b}}_{\mathbf{ez}}^{\top}), \tag{4.2.6}$$

where $\widehat{\mathbf{b}}_{\mathbf{ez}}(\widehat{\mathbf{b}}_{\mathbf{ez}}^{\top})$ represents the outer product of the two vectors $\widehat{\mathbf{b}}_{\mathbf{ez}} \in \mathbb{R}^{6\times 1}$ and $\widehat{\mathbf{b}}_{\mathbf{ez}}^{\top} \in \mathbb{R}^{1\times 6}$, which results in a matrix in $\mathbb{R}^{6\times 6}$. Considering circular cross-sections we can define the moment of inertia $I_e = a_e^2/4\pi$ and rewrite the equations in (4.2.5) and the decomposition of (4.2.6) as follows,

$$\mathbf{k}_e = \frac{a_e E}{l_e} \widehat{\mathbf{b}}_{\mathbf{e1}} (\widehat{\mathbf{b}}_{\mathbf{e1}}^{\top}) + \frac{3 a_e^2 E}{4\pi l_e} \widehat{\mathbf{b}}_{\mathbf{e2}} (\widehat{\mathbf{b}}_{\mathbf{e2}}^{\top}) + \frac{a_e^2 E}{4\pi l_e} \widehat{\mathbf{b}}_{\mathbf{e3}} (\widehat{\mathbf{b}}_{\mathbf{e3}}^{\top}). \tag{4.2.7}$$

Here we omit some technical steps for the sake of brevity but the details on transforming the stiffness matrices from local to global coordinate system and building the structure's stiffness matrix K can be found in Toragay et al. (2022), which results in the following expression:

$$\mathbf{K} = \sum_{e \in \mathbb{S}_e} a_e k_{e1} \mathbf{b}_{\mathbf{e1}} (\mathbf{b}_{\mathbf{e1}}^{\top}) + a_e^2 k_{e2} \mathbf{b}_{\mathbf{e2}} (\mathbf{b}_{\mathbf{e2}}^{\top}) + a_e^2 k_{e3} \mathbf{b}_{\mathbf{e3}} (\mathbf{b}_{\mathbf{e3}}^{\top}), \tag{4.2.8}$$

used in (4.2.3b) to generate the equilibrium equations.

Solving this problem to global optimality gives the lightest frame structure, restricted to the given ground structure, which can withstand the given external load, while ignoring manufacturability constraints (most notably, crossing elements). Further, we expect that, even for this relaxed version, a (locally) optimal solution can only be obtained efficiently for relatively restricted ground structures. Consequently, the proposed iterative heuristic scheme should, on one hand, provide a way to repair manufacturability constraints for a given solution of the relaxed problem, and, on the other, give an algorithm for adding promising candidate beams and/or nodes to the ground structure, which are two ideas that we discuss in detail next.

4.2.3 Iterative evolving ground structure algorithm

To handle the number of variables in large-scale problems Gilbert and Tyas (2003) suggested that instead of considering the full ground structure (FGS) with all the candidate elements in place, the optimization process can start from a minimal ground structure (MGS), in which most of the candidate elements are eliminated and only a (relatively) small number of essential elements exist. Although Gilbert and Tyas (2003) proposed this approach for solving large-scale lightweight truss problems, the central ideas can be adapted for frame structures as well. We follow a similar idea in the proposed heuristic. However, in addition to adding members/elements to the ground structure, we also allow for adding nodes, whenever it is necessary. Although adding nodes and related elements to the ground structure in each iteration increases the complexity of the problem and the number of variables, it also serves as an approach to eliminate one of the manufacturability constraints from the model. In other words, instead of having a constraint to remove the crossing members (this constraints require binary variables to be defined for each element), enforce this requirement by defining new nodes on the intersection of those members that appear in each iteration's solution with $a_e > 0$ and intersect.

Algorithm 1 outlines the iterations of the proposed approach. Critical elements of the algorithm are explained in detail below.

MGS generation. There are different possible approaches to choose the candidate members in the MGS from which the algorithm is initiated (Gilbert and Tyas 2003). We enforce connectivity of all nodes in the MGS by choosing all vertical FGS elements, and all the elements that are connected to the boundary (fixed nodes) or load nodes. An example of constructed MGS can be found in the first column of Figure 4.7, which presents computational results for a particular instance. The resulting ground structure is guaranteed to include all nodes, and connect load and boundary nodes. Naturally, it severely restricts the possible structure, and will need to be updated with other promising elements in the remaining steps of the algorithm. On

Algorithm 1 Evolving ground structure algorithm

Input: Grid with a defined set of nodes and boundary condition
Output: Near-optimal lightweight frame structure on the given grid

```
Generate FGS ← all the possible candidates with the current set of nodes
Generate MGS ← guarantee the connectivity of all nodes in the grid
Solve initial NLP
while Manufacturability condition ≠ satisfied do
    while Inner loop condition ≠ satisfied do
        Calculate proportional elongation for all members ∈ FGS \ MGS
        Choose additional candidates
        Update MGS
        Solve NLP
        if improvement < threshold ∨ FGS \ MGS = ∅ then
            Inner loop condition = satisfied
    if set of crossing members ≠ ∅ then
        Add new nodes to the set of nodes
        Eliminate previously added nodes if not in the current structure
        Update FGS
        Update MGS
    else
        Manufacturability condition = satisfied
```

the other hand, this approach significantly reduces the number of potential elements, and hence, the number of variables in the mathematical programming formulation. For example, Table 4.1 gives the number of elements in such an MGS and FGS for a square $m \times m$ grid with two fixed boundary nodes (in the corners), and one load node. This reduction in the number of candidate elements directly translates into reduction in the computational time to solve the master problem in each iteration. Finally, MSG should be selected in such a way that the master NLP is feasible. This then guarantees that each NLP solved in the iterations of the algorithm are also feasible. Note that since the master problem allows for crossing elements, MSG constructed as described is likely to satisfy this condition.

m^2	Grid	FGS	MGS	% reduction
9	3×3	18	12	34%
49	7×7	410	42	90%
121	11×11	2346	226	91%
625	25×25	60096	1237	98%

Table 4.1: Number of elements in FGS and MGS for a square $m \times m$ grid with two boundary conditions and single external load.

Evaluation of the resulting structure. For the current MSG, relaxed master problem is solved with a general-purpose nonlinear solver. As discussed above, global optimum may not be possible to obtain within reasonable time due to non-convexity of the problem. Hence, locally converging methods can be used. A solution to the problem then gives the weight of the current structure, which represents the quality of the solution. We then evaluate ways to improve this solution by adding new elements to the MGS and recover manufacturability constraints through node addition.

Member adding. In this step, the algorithm should determine the most promising elements among those that are in the FGS but not in the current MGS. In the case of truss structures solved in the literature, the dual solution can be used for this purpose, since the underlying problem is linear. In our case, dual solution is not useful due to non-convexity of the problem. Instead, we propose the following procedure. After solving the NLP master problem in each iteration, the displacements in all degrees of freedom and the proportional elongation of each possible element in the FGS are calculated. Naturally, elongation of the elements reveals the stress paths in the structure, or in other words, elements in the FGS (but not in the MGS) that would be most elongated would also be the most impactful, if added to MGS. Consequently, to choose the new members, we sort the FGS elements in a decreasing elongation order, and then add a certain number of them in that order. The number of new elements added at each step is a hyper-parameter in our algorithm and can be selected by trail and error. For our implementation we use the following formula:

$$\text{Number of new elements} = min\Big\{100, 0.1 \times |FGS|, |FGS - MGS|\Big\},$$

where $|.|$ denotes the number of elements in the set. In other words, the number of new elements is no more than 100, at most 10% of the FGS elements, or all FGS elements that are still not in the MGS. These are then added to the current MSG.

Inner loop stopping criterion. We continue resolving the master problem and adding new elements until either no significant (above defined threshold) improvement in the objective function have been observed in two consecutive iterations, or no elements in the FGS are remaining (i.e., elongated violating displacement constraint), and hence no future improvement is possible. The inner loop then concludes with a feasible solution that may include crossing elements, and hence may not be manufacturable. The next step is designed to repair manufacturability.

Location of potential new nodes. For each pair of crossing elements we either introduce a new node located at the intersection point and add the corresponding elements to FGS and MGS. Importantly, this may lead to computational issues, since the algorithm may result in creating nodes that are too close to the existing nodes. To avoid this, after node creation, we check if there exist clusters of nodes within a certain threshold distance from each other. If so, each node cluster is replaced with a single node, which inherits all the corresponding members in the FGS and MGS. For our implementation, we use the maximum allowed displacement (u_{max}) as the threshold value. In addition, in each iteration we also check if the previously added nodes (ones not in the original grounds structure) appear in the current result. If not, those nodes are eliminated from the set of nodes for the next iteration, in order to reduce the computational challenge of solving NLP. At the end of this step, the ground structure should be regenerated by updating FGS and MGS following the same rules as before.

Stopping criterion. The algorithm stops if there are no crossing members in the optimal result of current iteration.

4.3 Numerical Results

4.3.1 Design of experiments

Material properties. In all cases the following mechanical characteristics are considered:

- maximum allowed displacement in any direction is 0.095 mm,
- cross-sectional lower and upper bounds are $\pi(0.2)^2$ and $\pi(0.5)^2$ mm (min./max. radius 0.2, 0.5 mm),
- density of material (without loss of generality) is set to 1,
- modulus of elasticity is 109000 GPa.

The value for the maximum displacement is calculated based on the material properties of steel to remain in the elastic area.

Test instances. We use three sets of instances. The first, which we will refer to as *square* instances, are used to compare against the exact global optimization method proposed in Toragay et al. (2022). The corresponding ground structure is depicted on Figure 4.2, where a $5cm \times 5cm$ design domain is considered. The number of eligible elements in the FGS is 18 and 116 respectively. The external nodal load for these instances is imposed upward (shown with a blue arrow), the build direction is bottom-up (all elements that cannot be manufactured due to printing angle constraint are removed from the FGS).

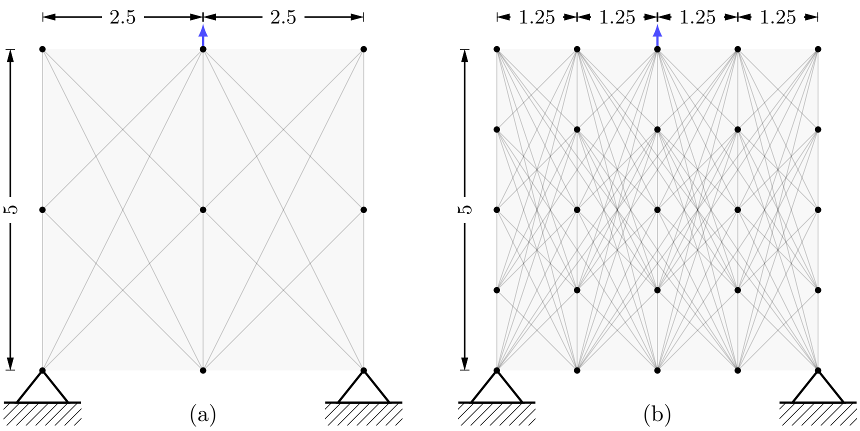


Figure 4.2: Square test instances (3×3 and 5×5 grids).

The second set of instances is based on a Michell *cantilever* beam example from literature and is depicted in Figure 4.3, with three grids. Note that the external nodal load in these instances is located in the middle of the top row of the grid. For the sake of maintaining readability, the case of fine grid does not depict candidate beams in the FGS. The number of elements in FGS varies from 460 to 67,872 for the coarse and fine grids respectively. Note that the cantilever is depicted vertically, which coincides with the AM build direction. With a $-90°$ rotation, similar to the ones that were analyzed in the literature before (for truss structures), e.g., (Prager 1977; Stolpe 2007; Sokół 2011; Shahabsafa et al. 2021). These instances are too large to allow for exact global solution. While the instances themselves have been considered in the literature before, direct comparison to the results previously reported cannot be made, since these are usually solved as trusses. At the same time, the resulting topology from the truss and frame problems are very similar.

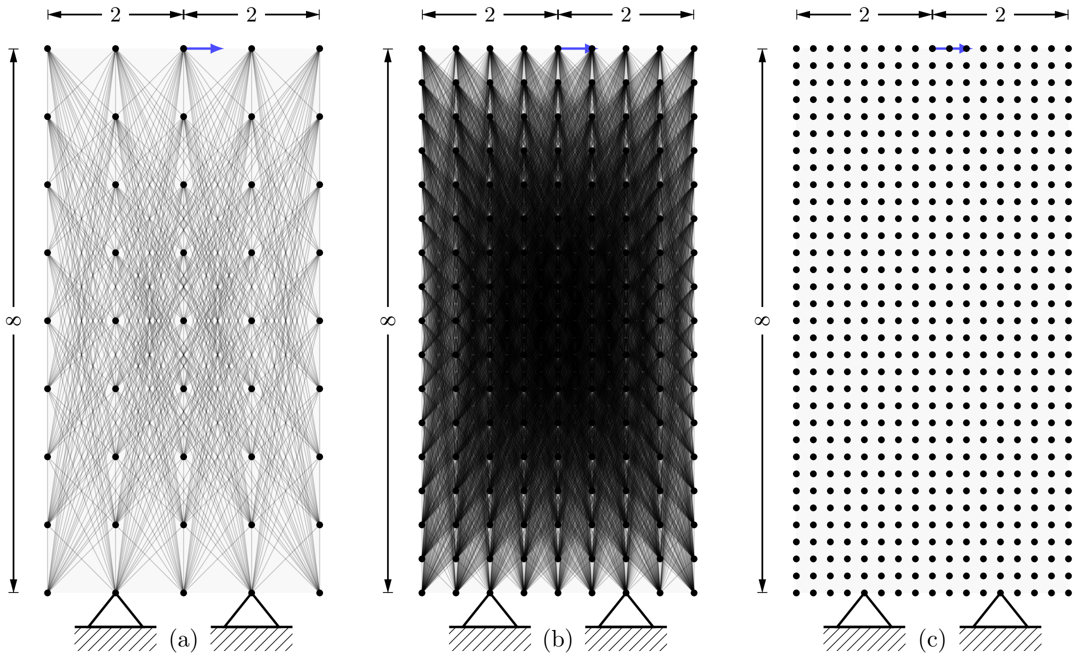


Figure 4.3: Cantilever test instances (coarse, medium and fine grids).

The final set of instances that we will refer to as *bridge* instances, is given on Figure 4.4. Observe that these correspond to identical design space and external loads, but represent two different build directions. The distinction is significant due to the manufacturability requirement on beam printing angle. The two instances consist of the same number of nodes and 1036 elements for instance (a) and 256 elements for instance (b). Note that in the bridge instances there are two external nodal loads. Instance (a) is reported in Sokół and Rozvany (2013) as a benchmark in solving multiple load truss-structure design problem.

Implementation details. The algorithm was implemented in Python (Van Rossum and Drake 2009) version 3.8. The mathematical optimization problems are modelled in Pyomo version 6.0.1 (Hart, Watson and Woodruff 2011; Hart et al. 2017) which is an open-source algebraic modeling language for Python. All experiments (unless specifically mentioned) were performed on a desktop computer with Intel® Xeon® CPU E3-1241 v3 @ 3.50 Gh, 14 GB installed memory, running on a 64-bit Windows 10 Pro OS. To solve the NLP models in each iteration we used **Ipopt** (Wächter and Biegler 2006) together with **pardiso** (Schenk et al. 2001) as the internal linear solver. We have set constraint violation tolerance to 10^{-6} (default 10^{-4}),

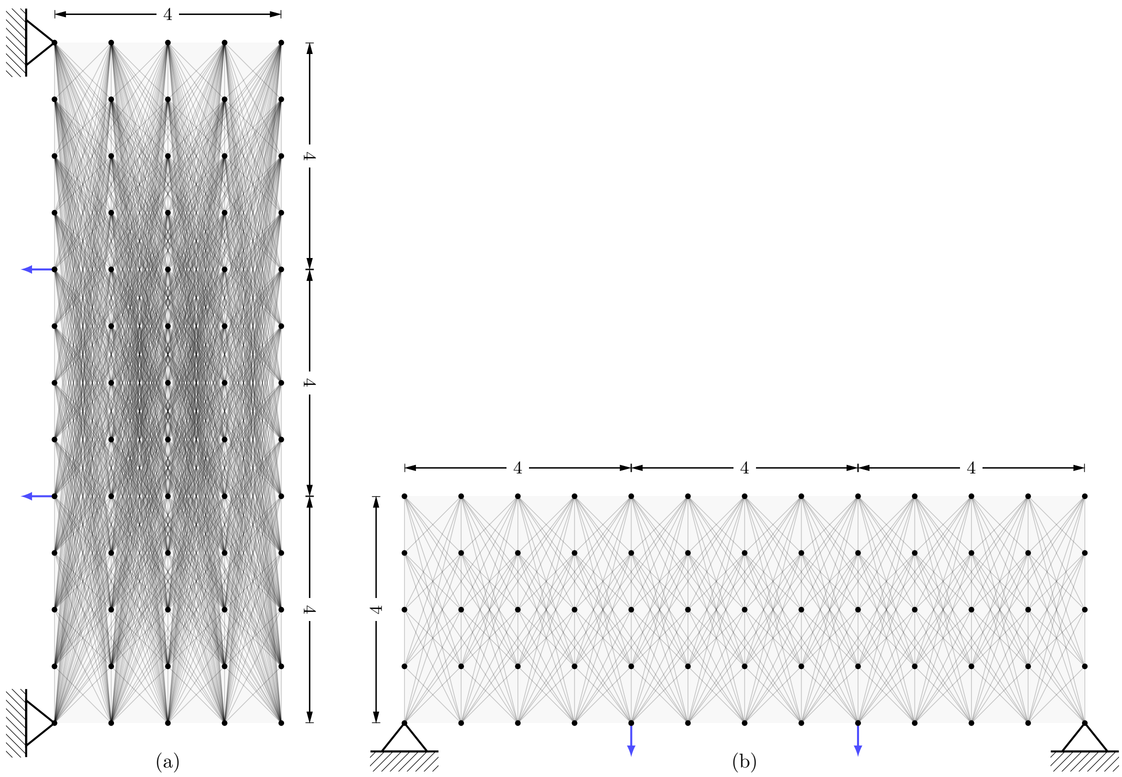


Figure 4.4: Bridge instances (two different build orientations).

and convergence tolerance to 10^{-6} (default 10^{-8}). In the implementation of the exact method we follow the same setup as in Toragay et al. (2022). Mixed integer quadraticaly constrained programming problems are solved using Gurobi version 9.5.1 (Gurobi Optimization 2021) with 2 hour (7200 seconds) time limit. We do not apply a time limit to the heuristic algorithm, since it did not exceed 2 hours in any of the test cases.

4.3.2 Numerical results and observations

Square instances. For these instances, we vary the external load in the interval $[100kN, 400kN]$ with $20kN$ increments, resulting in the total of 32 instances. Each instance is solved with both the heuristic and the exact methods. The solutions are then compared in terms of solution time and structure weight. First, focusing on performance of the heuristic itself, Figure 4.5 gives a comparison between 3×3 and 5×5 instances, giving the weight of the best found structure and computational times. Note that both cases share the same design domain, i.e., the resulting solutions can be directly compared. Some of the structures resulted from the heuristic method are depicted in Figure 4.6. Naturally, finer ground structure allows for lighter solutions, which is observed for the cases with higher external loads. Observe that the solution time for the heuristic is relatively low. Note also that in the case of 3×3 grid and the highest load, the algorithm takes advantage of the ability to add new nodes to the ground structure (depicted in green), which explains relatively high computational effort in this case.

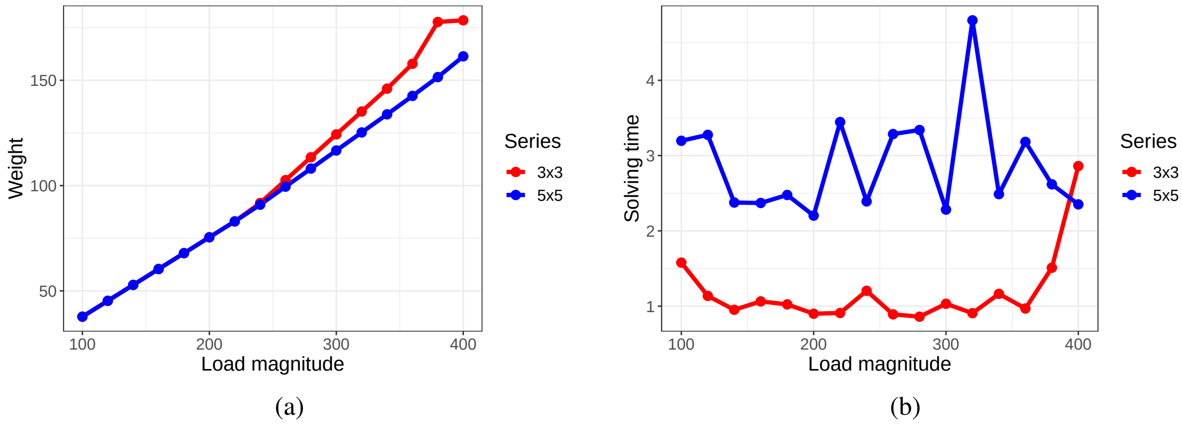


Figure 4.5: Obtained structure weight and corresponding the solution times (in seconds) for the square instances for the proposed evolving ground structure method depending on load magnitude.

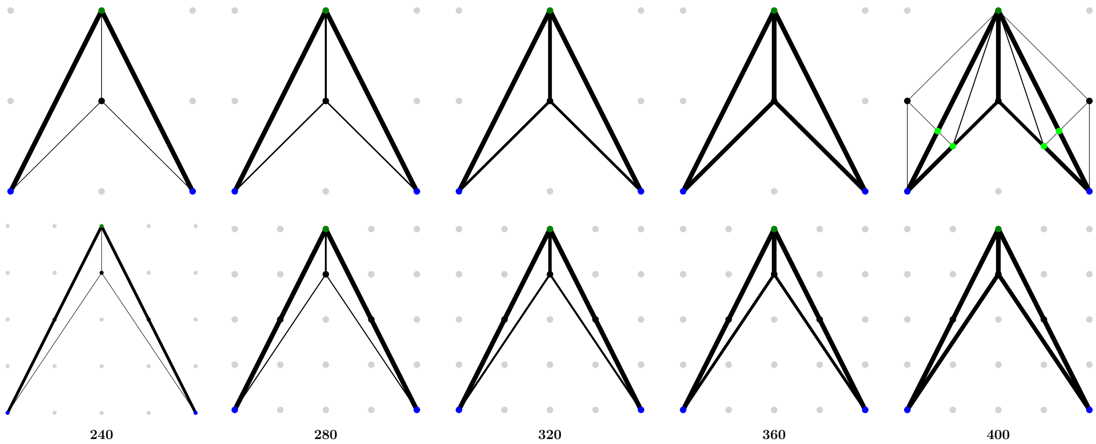


Figure 4.6: Obtained structures for the square instances for the proposed evolving ground structure method for a collection of external load magnitudes. Green points correspond to nodes added by the algorithm.

Table 4.2 reports on the comparison between the exact and heuristic methods. Here we report resulting structure weight (rows W) and solution time (rows T) for the two structures and 16 external loads. Note that in many cases the exact method did not converge within the time limit. In this case, whenever possible we report the optimality gap (in percent) and the corresponding weight for the best (suboptimal) solution is reported with an asterisk (*). In some cases, the exact method did not find any feasible solutions, in which case we use symbol "–". Finally, in two cases, the exact method reported that the instances are infeasible, i.e., no structure within the constraints of the ground structure and manufacturability can withstand the load, which are indicated with "inf". Note that for these two cases, heuristic method is able to return a feasible solution, since it allows for introduction of additional nodes, which enables more possible solutions. In other words, in addition to letting us avoid binary variables to eliminate crossing elements, this feature of the methodology is able to improve on the limitations due to fixed ground structure (to some extent).

Ins.	Results	Method	Load values															
			100	120	140	160	180	200	220	240	260	280	300	320	340	360	380	400
3×3	W	H	37.7	45.3	52.8	60.4	67.9	75.4	83.0	99.3	102.6	113.5	124.3	135.2	146.0	157.9	178.8	184.5
		E	37.7	45.3	52.8	60.4	67.9	75.4	83.0	94.4	102.6	113.4	124.3	135.2	146.0	157.9	**inf**	**inf**
	T	H	0.72	0.76	0.64	0.66	0.70	0.73	0.74	0.98	0.70	0.65	0.70	0.68	0.70	0.60	2.07	1.98
		E	9.70	7.90	12.3	17.4	18.1	15.6	16.4	28	28.8	30	23.6	35	38.6	23.3	–	–
5×5	W	H	37.7	45.3	52.8	60.4	67.9	75.4	83.0	100.1	100.8	108.1	116.7	125.2	133.8	142.4	151.5	161.4
		E	41.6*	51.5*	60.35*	69.9*	80.3*	–	–	–	–	–	–	–	–	–	–	–
	T	H	1.83	1.98	2.44	1.30	3.17	1.35	1.83	1.39	2.15	2.01	1.65	2.31	1.58	1.32	3.23	2.07
		E	48%	57%	60%	64%	67%	–	–	–	–	–	–	–	–	–	–	–

Table 4.2: Solution time (T) and weight (W) of structures resulted from heuristic evolving ground structure (H) vs. exact (E) methods for square instances. Optimality gap in percent is reported instead of time for instances terminated due to time limit. "–" indicates that no feasible solution was found. "**inf**" corresponds to provably infeasible problems (for the exact grid).

Comparing the performance of the two methods, observe that only in one case (240kN load and 3×3 grid) the exact solution is lighter compared to heuristic. In all other cases, the heuristic has either found the global optimum, or found a feasible solution that is better than an intermediate suboptimal solution reported by the exact method when it terminated due to time limit. In all cases the solution time is significantly lower for the heuristic method. Overall, we can conclude that the exact approach can only solve very small instances to optimality (which are almost always also solved to optimality by the heuristic), and consequently, the heuristic is preferred in virtually all cases.

Cantilever instances. The cantilever instances (coarse, normal, and fine grids, as specified above) are solved with a common load of $10kN$, imposed across the $x-$axis (horizontally). Figure 4.7 gives the initial MGS, intermediate steps, and the final best obtained solution for the three cases. As before, green indicates added nodes, and blue and red beams correspond to tension and compression respectively. Table 4.3 reports the weight as well as total solution time for the best obtained structure. The results illustrate the trade-off between solution quality (weights) and solution time that depends on the coarseness of the grid. In this case, the finest grid results in 6.67% lighter structure, while taking considerably longer time to solve. Note that the node-adding feature of the algorithm, allows for circumventing grid limitations to some extent. The fine grid instance is the largest one solved in our experiments, indicating that the approach is capable of finding solutions for problems with 10,000+ elements and 100+ of nodes.

grid	# of Nodes	# of Elements	Solution time (sec)	Weight
coarse	45	460	27.27	183.34
normal	153	5088	319.94	175.65
fine	561	67872	2282.57	171.11

Table 4.3: Results for the cantilever instances.

Bridge instances. The bridge instances are solved with three load magnitudes $\{25, 50, 75\}$ kN. These instances are designed to emphasize the importance of build direction due to manufacturability constraints. Figure 4.8 gives the final solutions for the six instances, and Table 4.4 reports the weights and solution times. Observe that vertically built bridge in all three load cases results in both significantly lower weight and shorter solution times. The result is not surprising, since this built orientation allows for incorporating horizontal (or close to horizontal) elements. This instance also showcases the ability of our approach to handle instances with multiple loads and/or boundary conditions.

	Load	25	50	75
Time (sec)	vertical (a)	28.68	70.79	83.57
	horizontal (b)	70.72	45.61	260.11
Weight	vertical (a)	123.51	256.34	412.04
	horizontal (b)	194.60	471.12	803.25

Table 4.4: Vertical and horizontal bridge results with various loads.

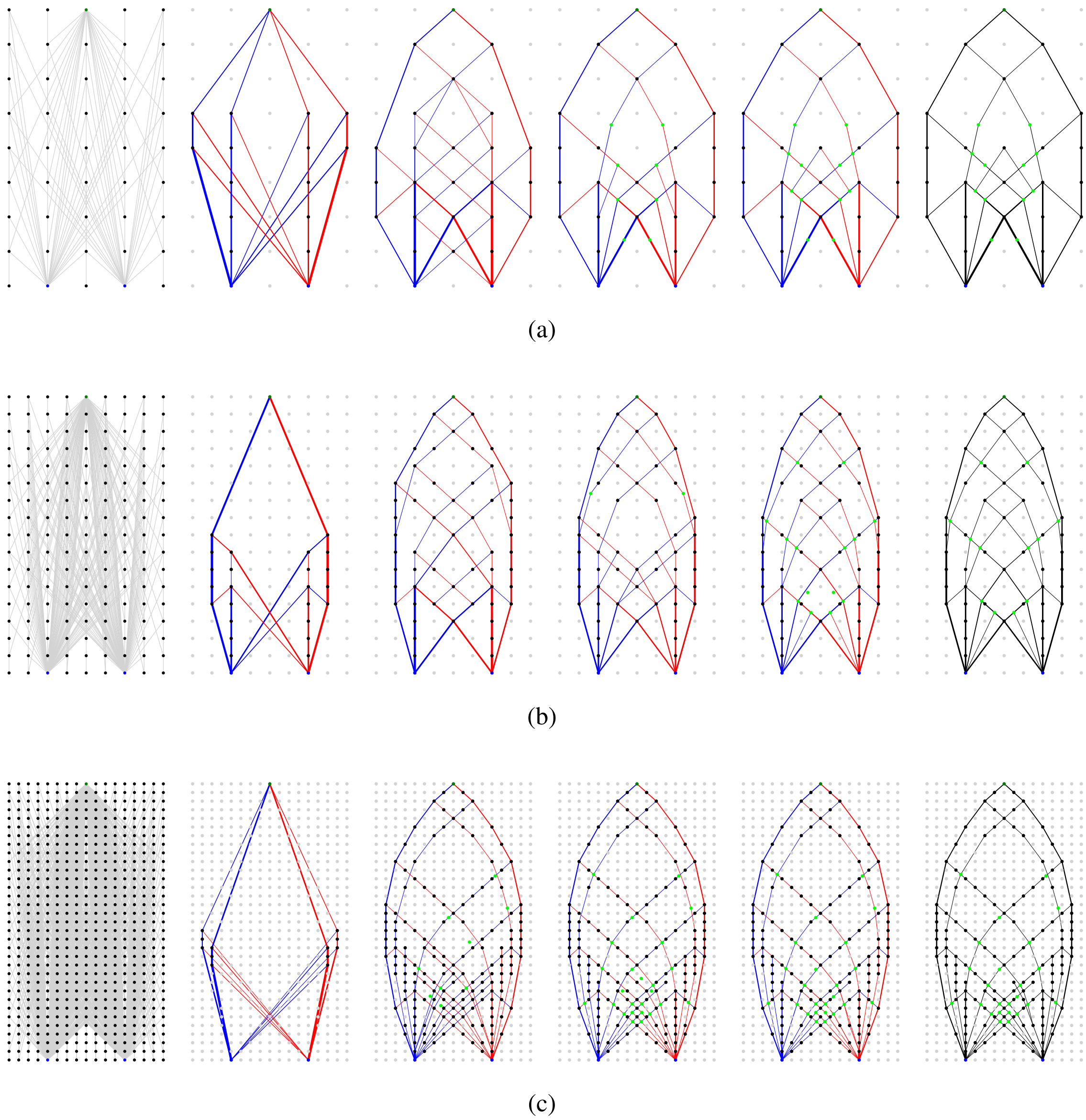


Figure 4.7: Original MGS, intermediate solutions and the resulting structures due to the heuristic method for the cantilever instances. Red and blue beams correspond to tension and compression respectively. Green points correspond to nodes added by the algorithm.

4.4 Conclusions

We proposed an iterative heuristic algorithm to solve the nonlinear, nonconvex optimization problem of designing lightweight load-bearing frame structures for additive manufacturing. Our numerical experiments show that using the proposed heuristic method, relatively large-scale instances (up to 10,000s of elements) can be solved. We observe that for small instances with known optimal solutions, the proposed method is usually capable of finding the exact

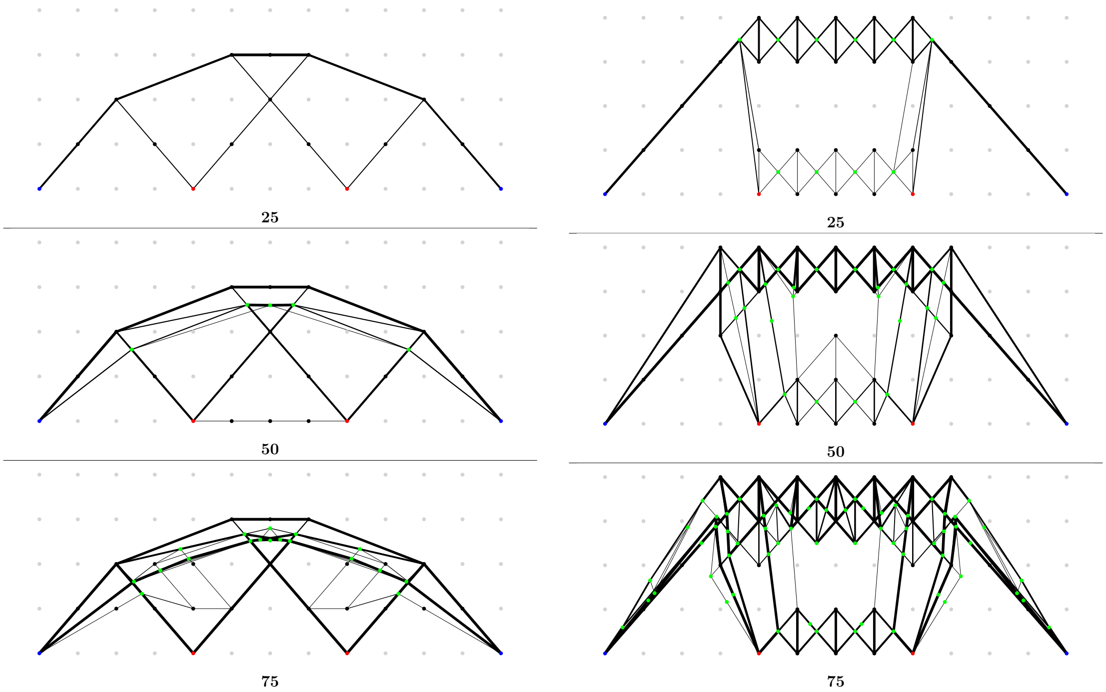


Figure 4.8: Obtained solutions for the two versions of the bridge instance with different load magnitudes. Green points correspond to nodes added by the algorithm.

global optimum, while using significantly less computational resources. The superiority of the heuristic method in terms of computational requirement, even for relatively simple cases, is observed in the tested instances.

We showed that the heuristic method can solve large-scale instances with different load magnitudes, load locations, and various ground structures. The main contributions of the heuristic approach developed in this chapter are as follows:

- iteratively expanding the ground structure by adding both members and nodes;
- relaxation of the most challenging manufacturability constraints in the mathematical model to be able to solve large-scale instances;
- taking into account the moments in the rigid joints of the structure to avoid postprocessing.

Note that we could not compare the heuristic and exact methods in solving large-scale instances because none of those instances can be solved to optimality using the exact method.

Nor could we compare the results of benchmark problems solved here directly to the results reported in the literature because those instances have been either solved as truss structures, or considered different (sometimes conflicting) sets of constraints in the mathematical models to account for structure manufacturability or feasibility. Consequently, a thorough comparison of the existing methods for frame optimization requires a significant additional modeling effort, and hence is beyond the scope of the current study. On the other hand, the field would benefit from creating a repository of both standard test instances and implementations of the algorithms, which could be a direction of future study.

Several other directions for either improving the proposed method or implementing alternative ideas can be identified. First, while our algorithm allows for node addition, node coordinates are largely fixed. Naturally, allowing for variable coordinates significantly complicates the formulation, making a straightforward implementation of the underlying NLP intractable. On the other hand, a carefully designed iterative procedure, where node adjustment and NLP solution alternate could be possible.

Secondly, while the method here is designed as a problem-specific heuristic, general-purpose metaheuristics have been previously considered in the literature. Implementation of such approaches is often severely complicated by the curse of dimensionality. At the same time, such approaches could benefit from the mathematical programming formulations presented here and in Toragay et al. (2022), paving way for potentially efficient matheuristics implementations.

Chapter 5

A Hybrid Genetic Algorithm for Topology Optimization of Frame Structures for Additive Manufacturing

5.1 Introduction

In Chapter 3 of this dissertation, exact optimization approaches were implemented to optimize the designing problem of lightweight frame structures for additive manufacturing (AM). We concluded that these approaches are able to solve only small-scale problems. To handle this downside of the exact methods, in Chapter 4 a problem specific heuristic method have been proposed to solve large-scale problem. This novel heuristic outperformed the exact methods and inspired us to closely inspect the applications of metaheuristics for the design problems. A literature review in the implementation of metaheuristics for topology optimization problems revealed that most of the applications consider truss structures because the frame structures are more complicated and metaheuristics, in their pure form, do not have promising outcomes when implemented for topology optimization problems. Observing this gap in the literature, in this chapter, we tackle the problem of designing lightweight frame structures for AM. We specifically consider a metaheuristic approach, Genetic Algorithm (GA), to solve the mathematical model of weight minimization for planar frame structures when the AM imposed manufacturability constraints are considered. As depicted in Figure 4.1, a design domain and fixed boundary conditions (a), a defined grid of nodes (b), a discrete ground structure composed of all candidate elements (c), and a ground structure excluding all the non-eligible elements from it (d), are necessary for the initialization of the design problem for discrete structures. Reaching the best design for lightweight structures requires evolving the geometry of the structure, in addition to just optimizing the properties of the elements such as cross-section of the members Raich and Ghaboussi (2000). Therefore, topology optimization, which is the combination of the geometry (layout) and size optimization for discrete structures is considered in this chapter.

Note that the mathematical modeling of this problem is possible via implementing the ground structure approach proposed by Dorn (1964) and the decomposition of the stiffness matrix proposed by (Kanno 2016). For the details of decomposition method that we implemented in this chapter, we refer the readers to Appendix A of this dissertation.

Evolution-inspired metaheuristics, specifically Genetic Algorithms (Holland 1992; Golberg 1989) are a good fit for the topology optimization problems, in the sense that, discrete variables, which increase the complexity in the mathematical models, can be effectively handled using GA. In addition, the stochastic nature of the GA avoids the drawback of mathematical optimization, where the latter is prone to getting trapped in the local optima (Alkhatib, Jazar and Golnaraghi 2004). On the other hand GA itself with mixed type of variables, in its pure form, is more complex and can also be prone to local optima Rocha and Neves (1999).

To tackle this handicap of the simple GA, in this chapter, we propose a modification which accelerates the convergence of the algorithm benefiting from both stochastic and deterministic search scheme, i.e., both GA and mathematical optimization methods' strengths. Our proposed modification combines the results of mathematical optimization models in the form of nonlinear programming, representation of the structures as graphs and matrices, and constraint relaxation, to obtain the fitness values of the individuals.

The main issue in the implementation of GA for various problems is to find a clever representation (encoding) of the candidate solutions. We propose a novel encoding based on a combination of graph theory and matrix algebra to represent the individuals and define operators in the proposed hybrid GA. We call this the *path-based encoding* and will explain it in details in Section 5.2.2. Note that a brief review of the most related works from the literature is presented in Section 2.3 of Chapter 2. The remainder of this chapter is as follows. To explain the details of the methodology, Section 5.2 includes the overview of the mathematical model proposed for the problem, the hybrid GA proposed in this chapter, and details of operators specifically-defined for the hybrid GA. The numerical results of the implementing GA on various test instances have been given in Section 5.3. We conclude the findings and draw the road map for future work in Section 5.4.

5.2 Methodology

In all structural analysis problems there are some governing equations or inequalities that need to be satisfied for an accurate analysis of the structure's mechanical behaviour and characteristics. These governing equations for the elastic analysis of frame structures are called equilibrium equations. Any approach to solve the topology optimization problem should ensure that the suggested solutions satisfy these equations. In evolutionary algorithms, such as GA, all the individuals in the population need to have a fitness score which shows how good the individual is, in terms of representing a solution for the problem. Originally designed for unconstrained optimization problems, GA can still be used for constrained optimization problems such as topology optimization. One of the approaches to handle the constrains in these type of optimization problem is to define penalty functions for those individuals that do not fully satisfy the constraints. Although these functions reduce the credibility of an individuals in terms of fitness value, they enable the evolving of the inherent heritage of each individual to the next generations. In its general form a penalty function can be defined as follows:

$$\begin{cases} 0 & \text{if the individual satisfies all the constraints} \\ K_h(x) & \text{if the individual violate constraint } h. \end{cases} \tag{5.2.1}$$

in which x is the individual for which each constraint h in the model needs to be satisfied. K is a function that generates a large number, usually proportional to the magnitude of the violation of constraint h. This value for the minimization problems is a positive value that is added to the objective function. To calculate the fitness values in this work, we define the fitness function as the weight of the structure plus constraint penalties. We precisely consider an individual as a fit one if it satisfies the constraints and also minimizes the weight. To this end, we use the optimization model's solutions as individual's fitness value and we also include some of the solutions which do not completely satisfy the constraints. Those individuals encounter a penalty and will always have a higher fitness value (worse individual in a minimization problem). As a topology optimization problem we have both binary and continuous variables in our model. The presence and absence of each of the candidate elements in the ground structure can be modeled as a binary variable. Cross-sectional area of the beam elements are the continuous variables of the model. As depicted in Figure 5.1, in a layered manner, we first find the binary variables $x^{(i)}$ using the GA and the values of the continuous variables $y^{(i)}$ are found by solving a nonlinear optimization problem (abbreviated as NLP).

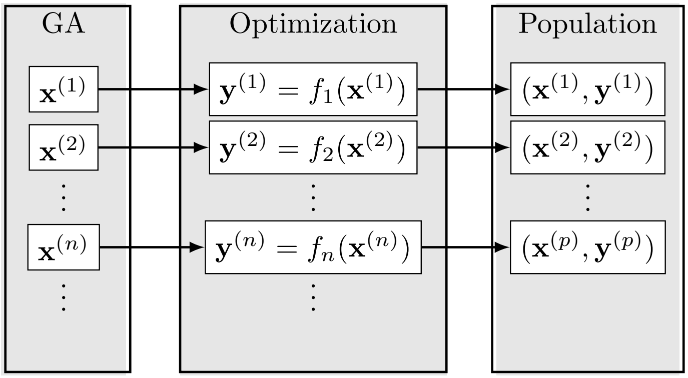


Figure 5.1: Generating the initial population

The approach has to be layered because finding the continuous cross-sectional variables in the optimization model depends on knowing which candidate elements selected by the GA are going to be part of that specific individual or solution. The layered format and the binary variables' decision in the first layer are necessary because the manufacturability constraints can actually be handled in the first layer. These constraints, imposed by additive manufacturing, consist of eliminating the over crossing members in the structure. More precisely, after GA chooses the elements from ground structure, if there are crossing members then a node is added to the ground structure with the coordinate of crossing point. Doing so, we can ensure that all the selected individuals can possibly fabricated by AM if they show a good performance in terms of the fitness function.

5.2.1 Model overview and relaxation

The problem of finding the economical distribution of the material in the design domain, or more specifically finding the optimal combination of beam element locations and sizes in the ground structure is an NP-hard problem. A solution approach for this problem using exact mathematical programming frame work is presented in Chapter 3 where the problem is modeled as a mixed integer nonlinear programming. The general form of the suggested model is given in 4.2.2.

The aim of this model is to minimize the weight (or volume) of the structure while keeping the displacements of the nodes in the structure in a controlled region. As mentioned previously GA are designed for unconstrained optimization problems (Rajeev and Krishnamoorthy 1992). We use a penalty function to transform the above mentioned constrained problem into an unconstrained one. To this end, the following modifications are necessary on the mathematical model:

- The constraints on the displacement need to be relaxed.
- Manufacturability constraints need to be handled separately, as there is no tractable way to create penalties for these violations.

The relaxation of the constraints on the displacement can be done by adding auxiliary variables to the right hand side on those constraints. For instance the following set of constraints shows a relaxed form of constraint 4.2.2d in the above mentioned model and will replace this constraints in the relaxed model:

$$\begin{cases} u_{min} - \boldsymbol{x_l} \leq \mathbf{u}, \\ \mathbf{u} \leq u_{max} + \boldsymbol{x_u}, \\ \boldsymbol{x_l} \geq 0, \\ \boldsymbol{x_u} \geq 0. \end{cases} \tag{5.2.2}$$

$\boldsymbol{x_l}$ and $\boldsymbol{x_u}$ can then be used in the penalty function described in 5.2.1 in order to calculate the penalties and include them in the objective/fitness function. Using the defined variables, mathematical expression of the penalty function is as follows:

$$K_h(x) = \sum_h (M\boldsymbol{x_l} + M\boldsymbol{x_u}) \qquad \forall x \in \text{population} \tag{5.2.3}$$

note that M is a large positive number. Manufacturability constraints consist of eliminating the cross-over elements and lower/upper bounds on the cross-sectional variables for all elements.

These constraints require binary variables to be assigned to each element. Cross-section limiting constraints are kept in the model but we relax the crossing member elimination and instead of ignoring the generated candidate individuals we keep them in the population with some modifications. By that, we mean applying Algorithm 2 on the generated individuals.

Algorithm 2 Modifying crossing members

Input: Randomly generated individual
Output: Feasibility of fabrication by AM
  Find all the crossing member pairs
  Define a new node on the crossing point
  Remove the crossing elements from solution
  Replace each removed element with two new elements
  Return the new solution with no crossing members

We keep cross-sectional area limiting constraints in the model because these constraints do not need binary variables anymore. Note that those binary variables have been decided in the first layer of the method by the GA. Note also that the build angle of the elements should be controlled in order to decrease the build failure of the structure during the AM fabrication process. This angles (α_e) for each element e should satisfy $45^\circ \leq \alpha_e \leq 135^\circ$. We considered this manufacturability constraint in the design of ground structure from which the paths are selected. It's guaranteed that all the elements in the chosen paths satisfy the angle constraint.

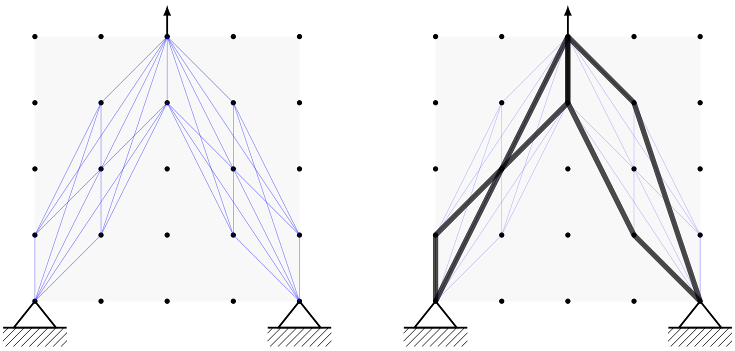
5.2.2 Hybrid genetic algorithm

This metaheuristic algorithm is an iterative procedure to find a near-optimal solution for the unconstrained combinatorial optimization problem. The whole algorithm can be summarized as follows.

A Genetic Algorithm starts with an initial population of individuals each of which is an encoded representation of a solution for the problem. This population is generated randomly and for each of the individuals a fitness value is calculated. The associated score describes the adaptability of each individual to survive. The better the fitness value, the higher the individual's chance to survive in the evolution process. For a minimization problem (as we are dealing in this work) a lower fitness value describes a better/lighter solution. After the establishment of the initial population, in each of the following iterations individuals are evaluated based on their fitness and through a selection process some individuals are selected as parents for the next generation. Reproduction of the individual for the next generation can be carried out via crossover and mutation operators of the algorithm. A replacement operator then substitutes new offspring for the individuals from the previous generation with worse fitness values. We explain all the details and operators of the hybrid GA below.

Encoding of the solutions: A representation for the solutions of the problem is necessary to encode each potential solution as an unique individual. This step is one of the most crucial factors in the success of implementing a GA (Kumar 2013). Many encoding schemes such as binary representation of the elements, value representation of the elements, permutation encoding, etc. have been used for the encoding of the structures each of which have its own downsides (Bekiroğlu, Dede and Ayvaz 2009). For example binary coding is not tractable when there are large number of variables in the problem (elements in the ground structure) which produce a long string of 0 or 1 values. Alternatively, the binary representation of nodes needs additional information to be carried on with each individual that explain the connections in the structure. Considering the large number of variable in the problem and based on the similarities of the planar discrete structures, on one hand, and graphs, on the other, we introduce a novel path-based representation to encode the solutions. To the best of our knowledge, this is the first time a path based representation is used to encode a GA for topology optimization.

The idea for path-based encoding arises from considering the fact that any element in the ground structure, in order to be useful in distributing the stress, needs to be placed on the principal stress lines of the structure. In other words, only paths connecting load nodes to any of the fixed nodes in the structure contribute to reducing the stresses. After defining the ground structure, we listed all the possible paths in the ground structure connecting each fixed node to each load node. Each individual will be a combination of paths. Figure 5.2(a) shows all the simple paths with length ≤ 3, in a 5×5 ground structure while 5.2(b) depicts a randomly generated individual. We only consider so-called *simple* paths, that is, paths where there won't be any loops inside the path. Furthermore, we impose a length parameter for the generated paths.



(a)

(b)

Figure 5.2: (a) Possible paths with length ≤ 3 in a 5×5 ground structure, (b) An individual

Path-based encoding has two main benefits. First, it guarantees the absence of any *hanging* elements in the structure. That is, elements that are only connected to a single node and do not contribute to supporting the load. Second, it mimics the principal stress paths in the structure. Also, by defining the individual using path-based encoding instead of modifying each element in the cross-over or mutation process and checking the connectivity and feasibility of the mutated individual, we can directly modify paths (with all the elements on it) which in turn reduce the computational complexity of the process. That is, connectivity is guaranteed to be preserved in the cross-over and mutation operators, removing the computational burden of checking connectivity for each generated individual.

Individuals. Each individual in the population is expressed as a combination of paths that connect all fixed nodes to all load nodes in the ground structure. For instance, the example individual shown in 5.2(b) has 2 paths connecting the fixed point on the left to the load node and 2 for connecting the fixed node on the right to the load node. A concatenated list of all paths gives the elements in the ground structure which are selected to appear in this specific individual.

Initial population. The convergence and performance of GA are critically affected by the initial population (Toğan and Daloğlu 2008). Starting the algorithm from an initial population which includes better individual will decrease the number of generations that are necessary to find near-optimal solutions. As a parameter of the GA, the size of initial population k, can be decided prior to the analysis. In the proposed hybrid GA, we choose k distinct individuals that are feasible for the relaxed optimization model as the initial population. To do this, we choose random combinations of paths to generate the individuals (ensuring each fixed node is connected to each load node) and then solve the relaxed model considering the specifications of that structure. Considering the geometry that the individual has, and the constraints of the model imposed on the cross-sectional areas, the mathematical model's results yield the optimal cross-sectional areas of the elements for that individual. As shown in 5.1, both binary and continuous variables are determined during this process for each individual. Moreover, the solution of the relaxed model reveals $z^{\boldsymbol{i}}$, the fitness values of individual $\boldsymbol{i}$ (which also includes the penalties for violating any constraints in the model). The objective function of the relaxed model, shown in Equation 5.2.4, reveals the details of the initialization process. Note also that all the individuals in the initial population are selected to be unique, to allow diversity in the population.

$$z^{\boldsymbol{i}} = \sum_{e \in \boldsymbol{i}} a_e l_e + \sum_{h \in \text{Constraints}} K_h(\boldsymbol{i}) \tag{5.2.4}$$

in which, $\boldsymbol{i}$ is the individual to be assessed, the first summation calculates the weight of ind, and second summation represents total penalty values. Obviously $z^{\boldsymbol{i}} \geq \sum a_e l_e$ because $\sum K_h(\boldsymbol{i}) \geq 0$ and also by comparing the real weight of the individual to $z^{\boldsymbol{i}}$, the presence of penalty values for the individual can be detected. Considering the feasibility of the individuals in their candidacy for the initial population guarantees a better start point for the algorithm, but also makes the initialization process longer.

5.2.3 Operators of the GA

Here we explain the details of all operators that we use in the GA with an emphasize on the implementations of those operator on the specific encoding (path-based representation) that we propose in this work.

Selection. To produce offspring for the new generation, parents are selected randomly from the current generation. For this selection we choose two unique sets of parents (S_1 and S_2) with $k/2$ number of individuals in each of those. From each combination of pairs (p_1, p_2) where $p_1 \in S_1$ and $p_2 \in S_2$ a new child is generated to be considered for the next generation.

From those $k^2/16$ children we select the $k/4$ unique best children to be directly added to the next generation. Note that these numbers were selected via experimentation and were tested for $k \geq 4$.

Crossover. As mentioned before, for each child we have a pair of parents which consists of paths connecting each fixed node to each load node. The crossover operator compares the paths in p_1 and p_2 and select the ones that appear in both parents to be added to the set of paths in the child. Also, paths from p_1 and p_2 that are only present in one parent are selected at random to be added to the child in order to guarantee the integrity of the newly generated child. At a high level, the crossover operator is described by the following pseudo-code.

```
function CROSSOVER(p1, p2)
    Compare the paths in p1 and p2,
    Add the shared paths to child's paths set,
    For all other paths randomly choose some,
    return Child individual.
```

Mutation. A mutation operator is used to generate $k/4$ additional individual for the next generation. We implement mutation on the $k/2$ worse individuals in the current generation and $k/8$ worse generated children from the current generation. The mutation of each individual follows the pseudo-code below. Note that for the selection of a path to be removed from

```
function MUTATION(i)
    Compare the fitness value (z^i) to the real weight of the individual i,
    if z^i = real weight then
        Remove randomly selected path
    else
        Add randomly selected path
    return Mutated i.
```

an individual we consider all the paths from each of the fixed nodes and compare their numbers. Then choose the fixed nodes with larger number of paths and based on their length we assign a probability for each of the paths on that fixed node and randomly choose one of them. Obviously there is a direct relation between number of paths and stiffness of the structure on one hand, and weight of the structure on the other hand. This trade off between stiffness and weight is the focus of the mutation process. In individuals with larger fitness value than their real weight, it can be inferred that there are violated constraints in the mathematical model and the structure needs to be stiffer to decrease the constraint violation. In this case, we add more paths to the individual. Again we compare the number of paths from each of the fixed nodes and choose the one with less connected path. From all the paths in the path set that are not currently in the structure we choose a path based on probability proportional to their length. Conversely, if no constraints are violated, we can infer this individual is heavier than necessary (as its fitness is poor) so we randomly remove a path, checking that connectivity is not broken.

Elitism. Some of the individuals in each generation are selected to be directly added to the next generation. In this work, this number is considered as $k/2$ where k is the number of individuals in the population. The reason of adding these individual directly to the next generation is all of them show good performance having better fitness value and inherent those characteristics that should be carried in the next generations.

Stopping criteria. Stopping criteria used in the literature include: a fixed number of iterations, improvement in some consecutive generations, number of similar individual in the population, etc. We consider a combination of quality of the individual in one hand and the number of iteration on another hand. We consider two parameters to stop the hybrid GA; maximum number of generations in the process, and maximum number of generation without improvement. Whenever one of these numbers is reached, the process will stop and report the final generation.

5.3 Numerical Results

In this section we present the results for two experimental structures that we designed and solved using the hybrid GA approach. Figure 5.3 shows the two illustrative instances that are solved by implementing the process. These are *square* (Fig. 5.3 (a)) and *cantilever* (Fig. 5.3 (a)) instances. We choose these instances in order to be able to compare the results to what we had in previous chapters. For the square example GA found the optimal solution that we showed in Chapter 4. For the cantilever example the best solution that we could find using heuristic approach in Chapter 4 was 183.34 while there we allowed additional nodes to be considered and the shape could change accordingly. Here we did not include the additional nodes during the process except in calculation of fitness values. The best solution that GA could find for the cantilever instance is 260.97.

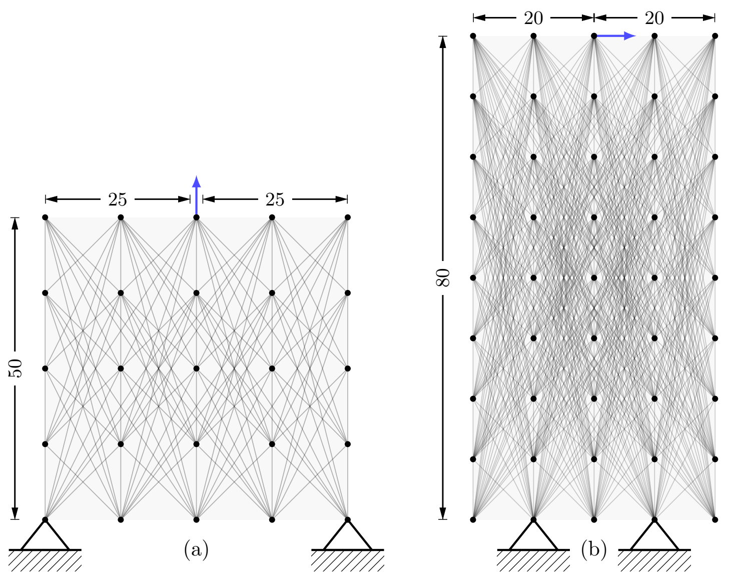


Figure 5.3: (a) Square example (b) Cantilever example

Table 5.1 shows the example ground structures' characteristics and best found solutions. Number of individuals in the population is $k = 24$. Each individual in the initial population consist of 3 and 4 paths from each fixed node respectively for square and cantilever examples. External loads are 400kN vertical for square and 10kN horizontal for the cantilever. Maximum allowable displacement on the degrees of freedom set to be 0.095mm for both structures in order to keep them in the elastic region. Without loss of generality we consider material density $\rho = 1$ and modulus of elasticity $E = 109000$ GPa for both structure.

Instance	grid	size (mm^2)	# of nodes	# of element	paths	path lengths	best weight
Square	5×5	2500	25	116	22	3	161.38
Cantilever	5×9	3200	45	460	890	4	260.97

Table 5.1: Solution for instance Fig. 5.3

The initial and final generations of these two examples are given in Appendix C. Figure 5.4 depicts the final solution of the test instances. In this figure, we colored the initial ground structure nodes in the final solution in green while the position of the additional nodes is shown in red. Note that we consider these additional nodes only in the calculation of fitness function and these nodes are not consider for next generations.

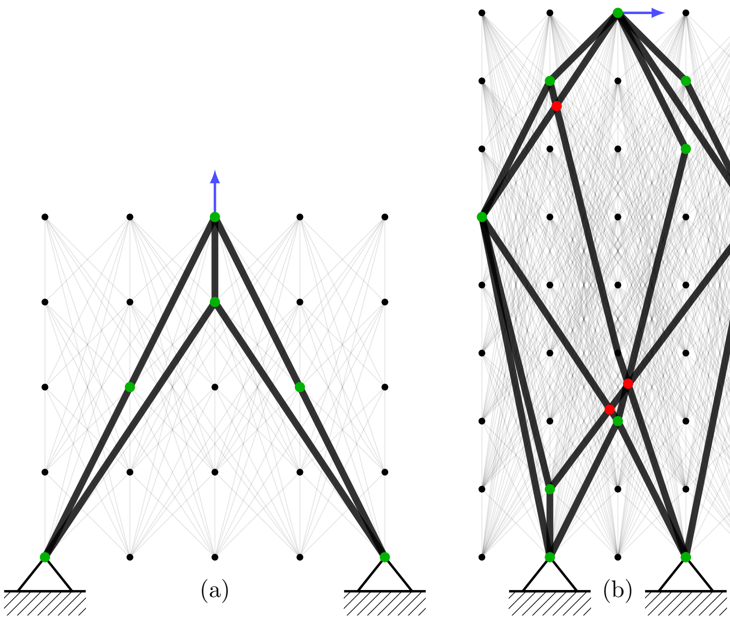


Figure 5.4: (a) Best solution in the final population of square instance, (b) Best solution in the final population of cantilever instance

To test the performance of the proposed hybrid GA, we tried various random seeds and reported the results in the following Table 5.2. We also tried different load magnitudes and include the results of all combinations of seed, load magnitude in the table.

Load / Seed	200	250	300	350	400
S_1	11 gen., 75.44	13 gen.,95.98	14 gen., 116.65	13 gen., 138.11	13 gen., 161.38
S_2	11 gen., 75.44	12 gen.,95.98	13 gen., 116.65	15 gen., 138.11	15 gen., 161.38
S_3	12 gen., 75.44	12 gen.,95.98	14 gen., 116.65	18 gen., 138.11	11 gen., 770.19
S_4	10 gen., 75.44	12 gen.,95.98	19 gen., 116.65	19 gen., 138.11	13 gen., 161.38
S_5	12 gen., 75.44	20 gen.,95.98	22 gen., 116.65	10 gen., 159.98	13 gen., 161.38

Table 5.2: Results for square instance with various seeds and load magnitudes

In two of the combinations, (S_5, 350) and (S_3, 400), the algorithm could not find the best solution that could be achieved by other seeds. We also tried various seeds for the cantilever instance and report the results in Table 5.3.

Seed	S_1	S_2	S_3	S_4
Generation/weight	24 gen., 366.63	31 gen., 307.17	12 gen., 373.77	43 gen., 260.97

Table 5.3: Results for cantilever instance with various seeds

Results in Table 5.3 shows that best achieved GA's result for the cantilever instance is 42% worse than the best known solution achieved by the proposed heuristic in Chapter 4. Modification of the operators or the self-tuning of the parameters for the hybrid GA can be two approach to alleviate the weakness that we experience with the GA.

5.4 Conclusion

In this chapter we proposed a novel hybrid genetic algorithm metaheuristic approach to solve the nonlinear design optimization problem for additively manufactured lightweight planar frames. The novelty of the proposed algorithm are as follow:

- Implementing a nonlinear optimization approach to calculate the fitness values.
- A novel encoding of the individuals in the population based on the paths in the ground structure.
- Modified operators for the hybrid GA that can handle the path-based encoding.

Our numerical results showed that although the proposed hybrid GA is able to find solutions for large-scale problems, the accuracy and consistency of the results still needs in depth research. These large-scale instances could not be solved using the exact methods that we present in Chapter 3. On the other hand, the results of hybrid GA is comparable to the results that we obtained for the same instances using the heuristic method that we proposed in Chapter 4. We conclude that the proposed path-based encoding is a good alternative for the traditional encoding that has been used for the structural optimization problems in the literature which makes hybrid GA a promising candidate for solving the engineering optimization problem that we considered here. More research can be done to accelerate the process and test its performance against traditional GA approaches. Another continuation of this research can be determining a self-tuning approach for the parameters that are currently used in the proposed algorithm.

Chapter 6

Summary and Future Research

In Chapter 3 we mathematically modeled the design optimization of lightweight frame structures for additive manufacturing as a mixed integer, nonlinear, nonconvex optimization problem. Because of nonlinearity and nonconvexity, these models are in the NP-hard category and cannot easily be solved to optimality. Despite technological advances in computational resources and commercial solvers, still only small-scale instances of these problems can be solved to optimality. We proposed a new modeling approach in the form of a mixed integer quadratically constrained optimization problem. We showed that the proposed approach, although still limited by the number of variables in the problem, performs better than the existed mathematical models in the literature. Observing the limitations of exact mathematical models in solving the lightweight frame structures' design, in Chapter 4 we proposed a novel, problem-specific iterative heuristic method to address the problem. This heuristic approach is based on the member-node adding scheme in which instead of considering all the possible candidates in the ground structure, a sub-problems with a small number of members is chosen and gradually, trough a selection process, candidate elements which were not in the current ground structure are selected and added to the existing ground structure aiming for better results. Although this approach does not guarantee the global optimality of the solutions, the results are near-optimal solutions achieved in a fraction of the time which is otherwise required by the exact methods to find the answer. For small-scale instances for which we could find the exact global optimal solutions, the heuristic method is also capable of finding those results. On the other hand, this approach enables us to find solution for those large-scale problems which couldn't be solved using the exact methods. This problem-specific approach can not be easily adopted for other topology optimization problem raised when the assumptions or the constraints set change in the problem. To alleviate this downside general-purpose well-known metaheuristic such as genetic algorithm or particle swarm optimization can be implemented.

As an alternative to our heuristic approach, in Chapter 5 a modification of genetic algorithm (GA), a metaheuristic method, is proposed. In this method we combined the stochastic nature of the GAs with the deterministic manner of the exact methods to better guide the process to the best solution. In other words, we applied the solutions of the exact nonlinear methods to calculate the fitness value for each individual generated randomly by the GA. Through a novel path-based encoding of the individuals we guaranteed the connectivity and feasibility of the randomly generated initial population in the proposed hybrid GA, and maintained these characteristics throughout the duration of the GA. Our numerical results show that although GA is a powerful metaheuristic method to solve the combinatorial optimization problems, when it is implemented to solve the design problems, it still has some limitations. Further research in the applications of metaheuristics in the topology optimization area is suggested. Moreover all the proposed approaches in this dissertation are designed to solve the 2D planar frame structures. Modification and application of these methods to 3D additively manufactured frames would be a good extension of this research.

Appendix A

Details of Mechanical relations in the modeling part

Using the relations between displacement and external nodal loads from matrix analysis of the frame structures, we can define the stiffness matrix of each beam element in its local coordinate system as follow:

$$\mathbf{k}_e(a_e, I_e) = \begin{pmatrix} \frac{a_e E}{l_e} & 0 & 0 & -\frac{a_e E}{l_e} & 0 & 0 \\ 0 & \frac{12 I_e E}{l_e^3} & \frac{6 I_e E}{l_e^2} & 0 & -\frac{12 I_e E}{l_e^3} & \frac{6 I_e E}{l_e^2} \\ 0 & \frac{6 I_e E}{l_e^2} & \frac{4 I_e E}{l_e} & 0 & -\frac{6 I_e E}{l_e^2} & \frac{2 I_e E}{l_e} \\ -\frac{a_e E}{l_e} & 0 & 0 & \frac{a_e E}{l_e} & 0 & 0 \\ 0 & -\frac{12 I_e E}{l_e^3} & -\frac{6 I_e E}{l_e^2} & 0 & \frac{12 I_e E}{l_e^3} & -\frac{6 I_e E}{l_e^2} \\ 0 & \frac{6 I_e E}{l_e^2} & \frac{2 I_e E}{l_e} & 0 & -\frac{6 I_e E}{l_e^2} & \frac{4 I_e E}{l_e} \end{pmatrix} \tag{A.0.1}$$

$\mathbf{k}_e \in \mathbb{R}^{6\times 6}$ is a symmetric positive definite matrix in which, a_e and I_e represents the two cross-sectional properties, namely area and moment of inertia, of the element e (Kassimali 2012). In order to reduce the number of decision variables in our mathematical models, we can relate the moment of inertia for circular beams to their cross-sectional area as follows (r_e is the radius of element e):

$$I_e = \frac{\pi r_e^4}{4} = \frac{a_e^2}{4\pi} \tag{A.0.2}$$

Equilibrium state of the structure, where the internal forces and external nodal loads are balancing each other, can be illustrated in the model by using equilibrium equations:

$$\mathbf{K}\vec{U} = \vec{P} \tag{A.0.3}$$

Equation (A.0.3) represents d similar constraints in the mathematical model. In (A.0.3), $\mathbf{K} \in \mathbb{R}^{d\times d}$ is the global stiffness matrix of the structure. $\vec{U}$ and $\vec{P}$ are $d \times 1$ vectors representing respectively, displacements and the external nodal loads on all degrees of freedom. Although we will explain it later in detail, it should be noted here that $\mathbf{K}$ is constructed using the members' stiffness matrices in a global coordinate system. Based on the boundary conditions, for some of the degrees of freedom j, $u_j = 0$ and for some degrees of freedom j, p_j is a variable representing reaction forces on the fixed nodes. Based on the mentioned boundary condition effects, we express each equation of (A.0.3) in the form of the constraints in mathematical models as follow:

$$\sum_{j=1}^{d} \mathbf{K}_{ij} \check{u}_j = \check{p}_i \qquad \forall i \in \mathbb{S}_{\text{dof}} \tag{A.0.4}$$

where $\check{u}_j$ and $\check{p}_i$ define as:

$$\check{u}_j = \begin{cases} u_j & \text{If } j \text{ is not fixed and free to move} \\ 0 & \text{If } j \text{ belongs to a boundary node and fixed} \end{cases} \tag{A.0.5}$$

$$\check{p}_i = \begin{cases} p_i & \text{If } i \text{ doesn't belong to a boundary node} \\ RF_i & \text{If } i \text{ belongs to a boundary node} \end{cases} \tag{A.0.6}$$

We assume that external loads exist only on the end-nodes of the elements not along them. Loads also assumed to be concentrated dead loads which means they are deterministic, fixed, given, and do not change during the analysis. Moreover in (A.0.6), RF_i, is the reaction force in the direction of degree of freedom i. Also note that, RF_is are defined only on fixed degrees of freedom. $\mathbf{k}_e$ in (A.0.1), and as a consequence, $\mathbf{K}$ in (A.0.3), include a_e and I_e which are decision variables in the model. In finite element analysis, to avoid matrix inversion in solving the system of equations in (A.0.3), usually different decomposition methods such as LU decomposition, Cholesky decomposition, or Singular Value Decomposition are used. Here, following the approach that Stolpe and Svanberg (2003); Stolpe (2004); Rasmussen and Stolpe (2008); Kanno (2016) suggested, we decompose the stiffness matrices in (A.0.1). We define the following vectors together with the eigenvalues:

$$\hat{\mathbf{b_{e1}}} = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad \hat{\mathbf{b_{e2}}} = \begin{pmatrix} 0 \\ 2/l_e \\ 1 \\ 0 \\ -2/l_e \\ 1 \end{pmatrix}, \quad \hat{\mathbf{b_{e3}}} = \begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad \forall e \in \mathbb{S}_e \,, \tag{A.0.7}$$

$$k_{e1} = \frac{a_e E}{l_e}, \qquad k_{e2} = \frac{3 I_e E}{l_e}, \qquad k_{e3} = \frac{I_e E}{l_e}. \tag{A.0.8}$$

The member stiffness matrix in local coordinates can be obtained using the following matrix multiplication:

$$\mathbf{k}_e = \sum_{z=1}^{3} k_{ez} \hat{\mathbf{b}}_{\mathbf{ez}} (\hat{\mathbf{b}}_{\mathbf{ez}}^\top), \tag{A.0.9}$$

where $\hat{\mathbf{b}}_{\mathbf{ez}}(\hat{\mathbf{b}}_{\mathbf{ez}}^\top)$ represents the outer product of the two vectors $\hat{\mathbf{b}}_{\mathbf{ez}} \in \mathbb{R}^{6\times1}$ and $\hat{\mathbf{b}}_{\mathbf{ez}}^\top \in \mathbb{R}^{1\times6}$, which results in a matrix in $\mathbb{R}^{6\times6}$. Using equation (A.0.2) we can replace I_e with $a_e^2/4\pi$ and rewrite the decomposition in (A.0.9) as:

$$\mathbf{k}_e = \frac{a_e E}{l_e} \hat{\mathbf{b}}_{\mathbf{e1}} (\hat{\mathbf{b}}_{\mathbf{e1}}^\top) + \frac{3 a_e^2 E}{4\pi l_e} \hat{\mathbf{b}}_{\mathbf{e2}} (\hat{\mathbf{b}}_{\mathbf{e2}}^\top) + \frac{a_e^2 E}{4\pi l_e} \hat{\mathbf{b}}_{\mathbf{e3}} (\hat{\mathbf{b}}_{\mathbf{e3}}^\top), \tag{A.0.10}$$

in which:

$$k_{e1} = \frac{a_e E}{l_e}\,, \qquad k_{e2} = \frac{3 a_e^2 E}{4\pi l_e}\,, \qquad k_{e3} = \frac{a_e^2 E}{4\pi l_e}. \tag{A.0.11}$$

To be used in the construction of the structure's stiffness matrix, vectors in (A.0.7) need to be transformed to global coordinates. Considering the geometry of each beam element, we define the following transformation matrices:

$$\mathbf{Trans}_e = \begin{pmatrix} c & s & 0 & 0 & 0 & 0 \\ -s & c & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c & s & 0 \\ 0 & 0 & 0 & -s & c & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}_{6\times6}$$

in which $c = cos(\theta_e)$, $s = sin(\theta_e)$. To assemble the global stiffness matrix of the structure, we define the element specified $\mathbf{T}_e \in \mathbb{R}^{6\times d}$ matrices. Each $\mathbf{T}_e$ is a $6 \times d$ binary location matrix that relates the element's degrees of freedom to the degrees of freedom in the structure. Consider element e which connects node 2 (degrees of freedom: $\{4, 5, 6\}$) and node 4 (degrees of freedom: $\{10, 11, 12\}$), the degrees of freedom of element e are: $\{4, 5, 6, 10, 11, 12\}$ and the $\mathbf{T}_e$ matrix for this element can be defined as:

$$\mathbf{T}_3 = \begin{array}{c} \begin{matrix} \mathit{1} & \mathit{2} & \mathit{3} & \mathit{4} & \mathit{5} & \mathit{6} & \mathit{7} & \mathit{8} & \mathit{9} & \mathit{10} & \mathit{11} & \mathit{12} & \mathit{13} & \mathit{14} & \ldots & \mathit{d} \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \ldots & 0 \end{pmatrix}_{6\times d} \end{array}$$

The following transformation of the $\hat{\mathbf{b}_\mathbf{e}}$ matrices are used in the generating of equilibrium equations:

$$\mathbf{b_e} = \mathbf{T}_e^\top \times \mathbf{Trans}_e^\top \times \hat{\mathbf{b}}_\mathbf{e}\,, \tag{A.0.12}$$

for which as mentioned before, $\mathbf{T}_e^\top \in \mathbb{R}^{d\times 6}$, $\mathbf{Trans}_e^\top \in \mathbb{R}^{6\times 6}$ and $\hat{\mathbf{b}_\mathbf{e}} \in \mathbb{R}^{6\times 1}$, so $\mathbf{b_e} \in \mathbb{R}^{d\times 1}$.

Using these equations and values in (A.0.11), equation (A.0.4) can be written as (notice that $\mathbf{b_{e1}} \wedge \mathbf{b_{e1}^\top}$ and $\mathbf{b_{e2}} \wedge \mathbf{b_{e2}^\top}$ and $\mathbf{b_{e3}} \wedge \mathbf{b_{e3}^\top} \in \mathbb{R}^{d\times d}$):

$$\sum_{j\in\mathbb{S}_{\text{dof}}} \left(\sum_{e\in\mathbb{S}_e} k_{e1}\mathbf{b_{e1}}(\mathbf{b_{e1}^\top}) + k_{e2}\mathbf{b_{e2}}(\mathbf{b_{e2}^\top}) + k_{e3}\mathbf{b_{e3}}(\mathbf{b_{e3}^\top}) \right)_{ij} \check{u}_j = \check{p}_i\,, \qquad \forall i \in \mathbb{S}_{\text{dof}} \tag{A.0.13}$$

In the above equation, the inner summation is the global stiffness matrix of the structure that was generated using the relations (A.0.7) through (A.0.12).

Appendix B

Ground Structure's Pseudo code

Algorithm 3 Pseudo code for generating ground structure

Inputs: m, n, Node coordinates, E
Output: Ground Structure composed of all the potential beam elements

function GS(m, n)
    $nodes \leftarrow m \cdot n$
    $DoF \leftarrow 3 \cdot nodes$
    for $i \leftarrow 1$ to $nodes$ **do**
        for $j \leftarrow i$ to $nodes$ **do**
            if $ij \leftarrow OK$ **then**
                $elements \leftarrow elements + ij$
                Calculate $L : Length$, $\theta : Angle$, $sin(\theta)$, $cos(\theta)$
                Generate $\hat{b_{1e}}$, $\hat{b_{2e}}$, $\hat{b_{3e}}(6 \times 1)$
                Generate $k_{1e} = \frac{E}{L_e}, k_{2e} = \frac{3E}{4\pi L_e}, k_{3e} = \frac{E}{4\pi L_e}$
                Generate $T_e(6 \times d)$: 1 for DoFs and zero everywhere
                Generate t_e: Transformation matrix (6×6)
                Calculate b_{1e}, b_{2e}, $b_{3e}(d \times 1)$ as $({T_e}^\top \times {t_e}^\top \times \hat{b_{ie}})$

Appendix C

Initial and final populations

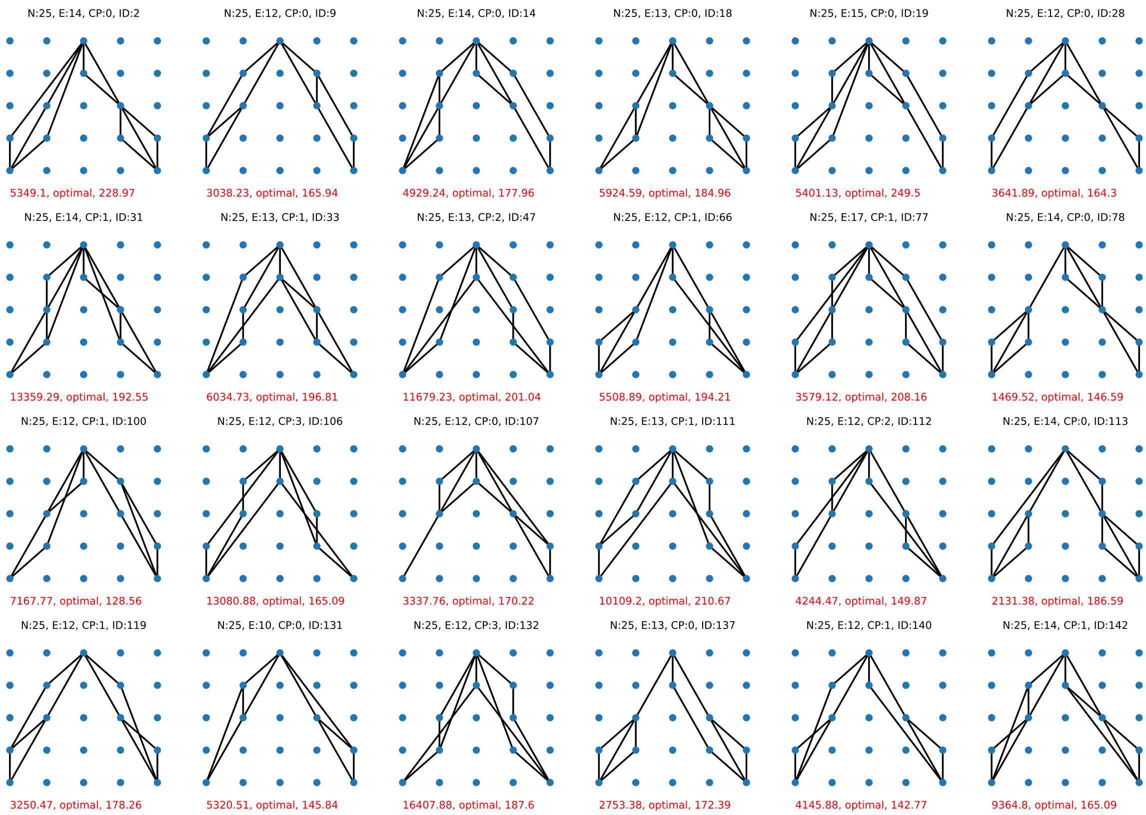


Figure C.0.1: Initial population for the square instance with $k = 24$, vertival load$= 400kN$

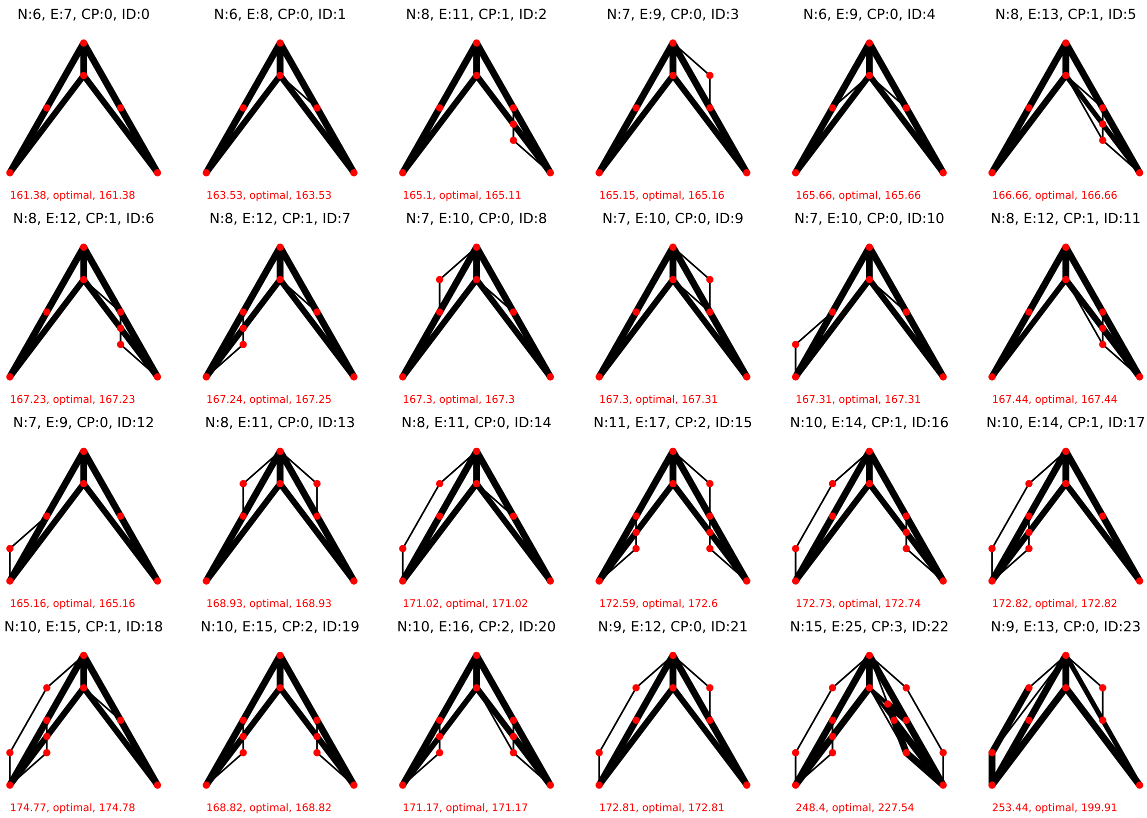


Figure C.0.2: Final population of square instance, best weight = 161.38

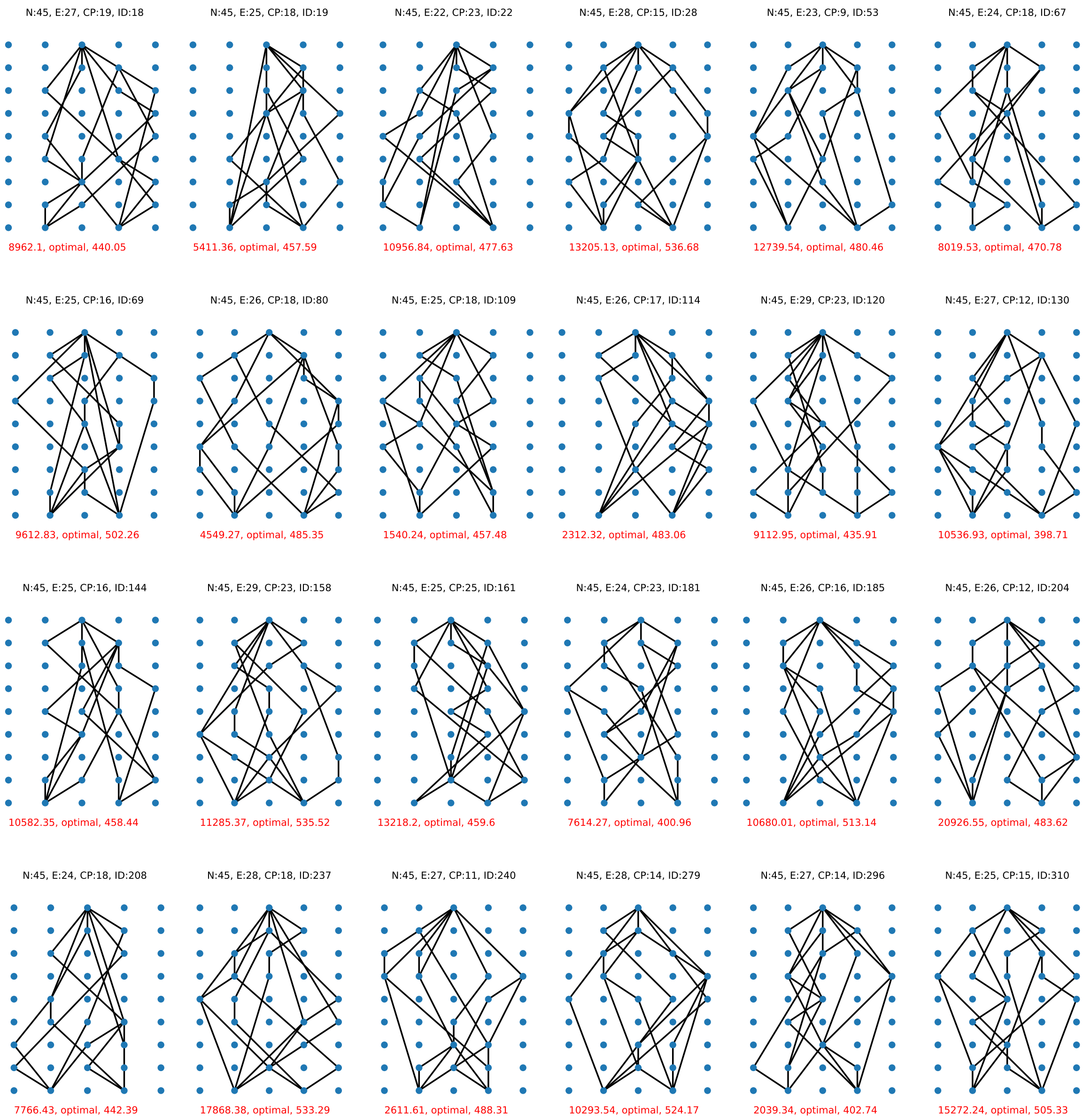


Figure C.0.3: Initial population for the cantilever instance with $k = 24$, horizontal load= $10kN$

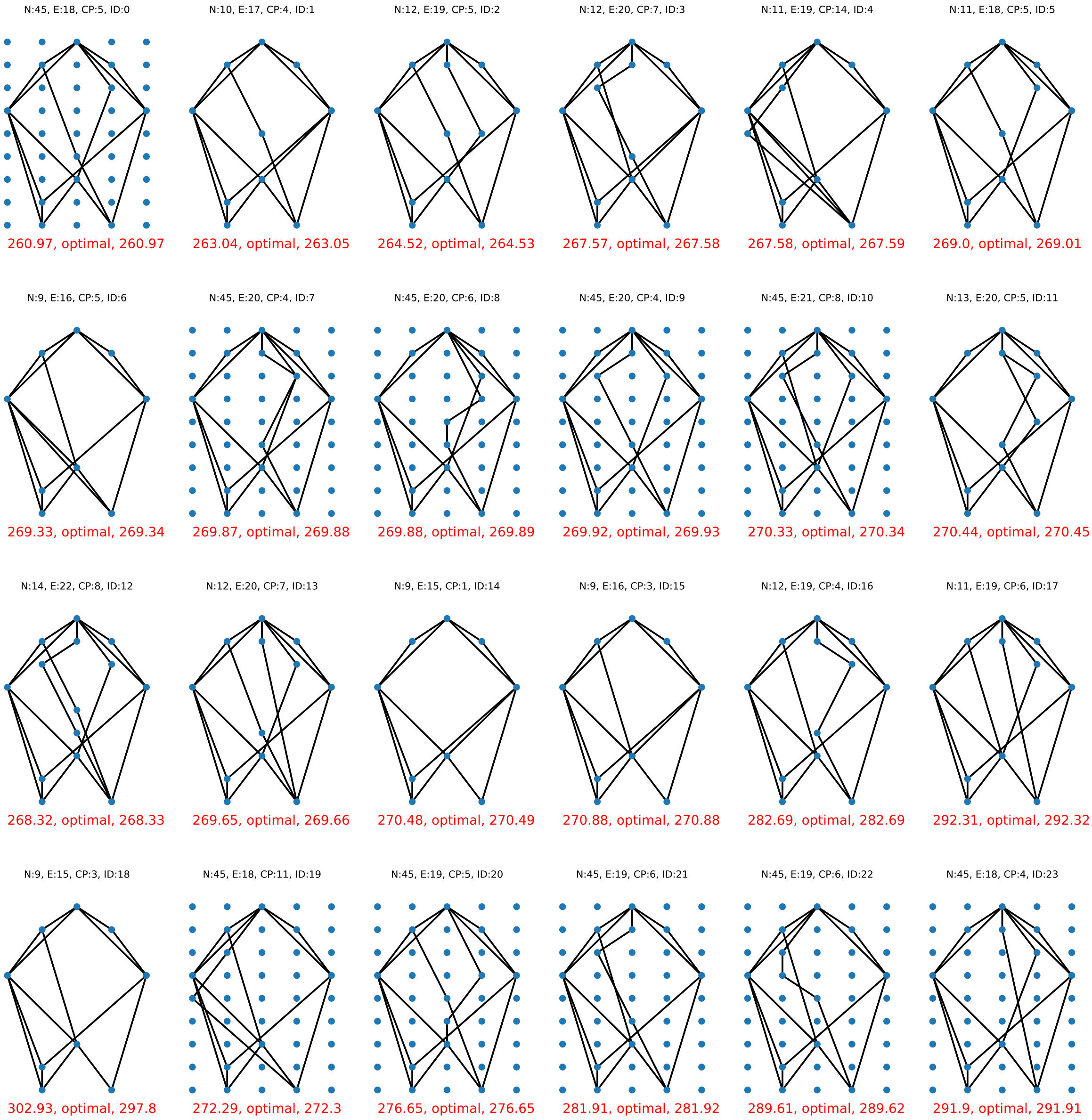


Figure C.0.4: Final population of cantilever instance, best weight = 260.97

Bibliography

Achtziger, Wolfgang and Mathias Stolpe. 2007. "Truss topology optimization with discrete design variables—guaranteed global optimality and benchmark examples." *Structural and Multidisciplinary Optimization* 34(1):1–20.

Ahrari, Ali, Ali-Asghar Atai and Kalyanmoy Deb. 2020. "A customized bilevel optimization approach for solving large-scale truss design problems." *Engineering Optimization* 52(12):2062–2079.

AISC. 2005. *Steel construction manual*. Amer Inst of Steel Construction.

Alkhatib, Rabih, G Nakhaie Jazar and M Farid Golnaraghi. 2004. "Optimal design of passive linear suspension using genetic algorithm." *Journal of Sound and vibration* 275(3-5):665–691.

An, Haichao and Hai Huang. 2017. "Topology and sizing optimization for frame structures with a two-level approximation method." *AIAA Journal* 55(3):1044–1057.

Azad, S Kazemzadeh and OĞUZHAN Hasançebi. 2013. "Upper bound strategy for metaheuristic based design optimization of steel frames." *Advances in Engineering Software* 57:19–32.

Azad, Saeid Kazemzadeh. 2021. "Design optimization of real-size steel frames using monitored convergence curve." *Structural and Multidisciplinary Optimization* 63(1):267–288.

Bai, Long, Junfang Zhang, Xiaohong Chen, Changyan Yi, Rui Chen and Zixiang Zhang. 2018. "Configuration optimization design of Ti6Al4V lattice structure formed by SLM." *Materials* 11(10):1856.

Balling, Richard J, Ryan R Briggs and Kevin Gillman. 2006. "Multiple optimum size/shape/topology designs for skeletal structures using a genetic algorithm." *Journal of structural engineering* 132(7):1158–1165.

Becker, Stephan T, Hendrik Bolte, Oliver Krapf, Hermann Seitz, Timothy Douglas, Sureshan Sivananthan, Jörg Wiltfang, Eugene Sherry and Patrick H Warnke. 2009. "Endocultivation: 3D printed customized porous scaffolds for heterotopic bone induction." *Oral oncology* 45(11):e181–e188.

Bekiroğlu, Serkan, Tayfun Dede and Yusuf Ayvaz. 2009. "Implementation of different encoding types on structural optimization based on adaptive genetic algorithm." *Finite Elements in Analysis and Design* 45(11):826–835.

Bendsoe, Martin Philip and Ole Sigmund. 2013. *Topology optimization: theory, methods, and applications*. Springer Science & Business Media.

Bollapragada, Srinivas, Omar Ghattas and John N Hooker. 2001. "Optimal design of truss structures by logic-based branch and cut." *Operations Research* 49(1):42–51.

Burton, Hanna E, Neil M Eisenstein, Bernard M Lawless, Parastoo Jamshidi, Miren A Segarra, Owen Addison, Duncan ET Shepherd, Moataz M Attallah, Liam M Grover and Sophie C Cox. 2019. "The design of additively manufactured lattices to increase the functionality of medical implants." *Materials Science and Engineering: C* 94:901–908.

Cansizoglu, Omer, Ola LA Harrysson, Harvey A West, Denis R Cormier and Tushar Mahale. 2008. "Applications of structural optimization in direct metal fabrication." *Rapid Prototyping Journal* .

Cerveira, Adelaide, Agostinho Agra, Fernando Bastos and Joaquim Gromicho. 2013. "A new Branch and Bound method for a discrete truss topology design problem." *Computational Optimization and Applications* 54(1):163–187.

Chan, C-M. 1992. "An optimality criteria algorithm for tall steel building design using commercial standard sections." *Structural optimization* 5(1-2):26–29.

Changizi, Navid and Mehdi Jalalpour. 2017*a*. "Robust topology optimization of frame structures under geometric or material properties uncertainties." *Structural and Multidisciplinary Optimization* 56(4):791–807.

Changizi, Navid and Mehdi Jalalpour. 2017*b*. "Stress-based topology optimization of steel-frame structures using members with standard cross sections: Gradient-based approach." *Journal of Structural Engineering* 143(8):04017078.

Cheng, Geng Dong and Xiao Guo. 1997. "ε-relaxed approach in structural topology optimization." *Structural optimization* 13(4):258–266.

Choy, Sing Ying, Chen-Nan Sun, Kah Fai Leong and Jun Wei. 2017. "Compressive properties of Ti-6Al-4V lattice structures fabricated by selective laser melting: Design, orientation and density." *Additive Manufacturing* 16:213–224.

Deaton, Joshua D and Ramana V Grandhi. 2014. "A survey of structural and multidisciplinary continuum topology optimization: post 2000." *Structural and Multidisciplinary Optimization* 49(1):1–38.

Deb, Kalyanmoy and Surendra Gulati. 2001. "Design of truss-structures for minimum weight using genetic algorithms." *Finite elements in analysis and design* 37(5):447–465.

Dede, Tayfun, Serkan Bekiroğlu and Yusuf Ayvaz. 2011. "Weight minimization of trusses with genetic algorithm." *Applied Soft Computing* 11(2):2565–2575.

Dorn, W. 1964. "Automatic design of optimal structures." *J. de Mecanique* 3:25–52.

Erbatur, Fuat, Oğuzhan Hasançebi, Ilker Tütüncü and Hakan Kılıç. 2000. "Optimal design of planar and space structures with genetic algorithms." *Computers & Structures* 75(2):209–224.

Eschenauer, Hans A and Niels Olhoff. 2001. "Topology optimization of continuum structures: a review." *Appl. Mech. Rev.* 54(4):331–390.

Fairclough, H and M Gilbert. 2020. "Layout optimization of simplified trusses using mixed integer linear programming with runtime generation of constraints." *Structural and Multidisciplinary Optimization* 61(5):1977–1999.

Gally, Tristan, Christopher M Gehb, Philip Kolvenbach, Anja Kuttich, Marc E Pfetsch and Stefan Ulbrich. 2015. Robust truss topology design with beam elements via mixed integer nonlinear semidefinite programming. In *Applied Mechanics and Materials*. Vol. 807 Trans Tech Publ pp. 229–238.

Gao, Ge, Yaobin Li, Hui Pan, Limin Chen and Zhenyu Liu. 2017. "An effective members-adding method for truss topology optimization based on principal stress trajectories." *Engineering Computations* .

Gao, Ge, Zhen-yu Liu, Yao-bin Li and Yan-feng Qiao. 2017. "A new method to generate the ground structure in truss topology optimization." *Engineering Optimization* 49(2):235–251.

Gibson, Ian, David Rosen, Brent Stucker and Mahyar Khorasani. 2014. *Additive manufacturing technologies*. Vol. 17 Springer.

Gilbert, Matthew and Andrew Tyas. 2003. "Layout optimization of large-scale pin-jointed frames." *Engineering computations* .

Golberg, David E. 1989. "Genetic algorithms in search, optimization, and machine learning." *Addion wesley* 1989(102):36.

Grossmann, IE, VT Voudouris and O Ghattas. 1992. Mixed-integer linear programming reformulations for some nonlinear discrete design optimization problems. In *Recent advances in global optimization*. Princeton University Press pp. 478–512.

Gurobi Optimization, LLC. 2021. "Gurobi Optimizer Reference Manual.".
URL: *http://www.gurobi.com*

Hagishita, Takao and Makoto Ohsaki. 2009. "Topology optimization of trusses by growing ground structure method." *Structural and Multidisciplinary Optimization* 37(4):377–393.

Hart, William E., Carl D. Laird, Jean-Paul Watson, David L. Woodruff, Gabriel A. Hackebeil, Bethany L. Nicholson and John D. Siirola. 2017. *Pyomo–optimization modeling in python*. Vol. 67 second ed. Springer Science & Business Media.

Hart, William E, Jean-Paul Watson and David L Woodruff. 2011. "Pyomo: modeling and solving mathematical programs in Python." *Mathematical Programming Computation* 3(3):219–260.

He, Linwei, Matthew Gilbert, Thomas Johnson and Tom Pritchard. 2019. "Conceptual design of AM components using layout and geometry optimization." *Computers & Mathematics with Applications* 78(7):2308–2324.

He, Linwei, Matthew Gilbert and Xingyi Song. 2019. "A Python script for adaptive layout optimization of trusses." *Structural and Multidisciplinary Optimization* 60(2):835–847.

Hirota, Masayuki and Yoshihiro Kanno. 2015. "Optimal design of periodic frame structures with negative thermal expansion via mixed integer programming." *Optimization and Engineering* 16(4):767–809.

Holland, John H. 1992. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

Huang, H and RW Xia. 1995. "Two-level multipoint constraint approximation concept for structural optimization." *Structural optimization* 9(1):38–45.

Kanno, Yoshihiro. 2016. "Mixed-integer second-order cone programming for global optimization of compliance of frame structure with discrete design variables." *Structural and Multidisciplinary Optimization* 54(2):301–316.

Kassimali, Aslam. 2012. *Matrix Analysis of Structures SI Version*. Cengage Learning.

Kaveh, A and A Zolghadr. 2014. "Comparison of nine meta-heuristic algorithms for optimal design of truss structures with frequency constraints." *Advances in Engineering Software* 76:9–30.

Kaveh, A and MH Ghafari. 2019. "Geometry and sizing optimization of steel pitched roof frames with tapered members using nine metaheuristics." *Iranian Journal of Science and Technology, Transactions of Civil Engineering* 43(1):1–8.

Khodzhaiev, Marat and Uwe Reuter. 2021. "Structural optimization of transmission towers using a novel Genetic Algorithm approach with a variable length genome." *Engineering Structures* 240:112306.

Kicinger, Rafal, Tomasz Arciszewski and Kenneth De Jong. 2005. "Evolutionary computation and structural design: A survey of the state-of-the-art." *Computers & structures* 83(23-24):1943–1978.

Klanšek, Uroš, Tomaž Žula, Zdravko Kravanja and Stojan Kravanja. 2007. "MINLP optimization of steel frames." *Adv Steel Constr* 3(3):689–705.

Kočvara, Michal and Jiří V Outrata. 2006. "Effective reformulations of the truss topology design problem." *Optimization and Engineering* 7(2):201–219.

Kronqvist, Jan, David E Bernal, Andreas Lundell and Ignacio E Grossmann. 2019. "A review and comparison of solvers for convex MINLP." *Optimization and Engineering* 20(2):397–455.

Kumar, Anit. 2013. "Encoding schemes in genetic algorithm." *International Journal of Advanced Research in IT and Engineering* 2(3):1–7.

Kureta, Rui and Yoshihiro Kanno. 2014. "A mixed integer programming approach to designing periodic frame structures with negative Poisson's ratio." *Optimization and Engineering* 15(3):773–800.

Larsen, SD, O Sigmund and JP Groen. 2018. "Optimal truss and frame design from projected homogenization-based topology optimization." *Structural and Multidisciplinary Optimization* 57(4):1461–1474.

Li, Dongfang, Shenyan Chen and Hai Huang. 2014. "Improved genetic algorithm with two-level approximation for truss topology optimization." *Structural and Multidisciplinary Optimization* 49(5):795–814.

Li, Yongqiang and Yong Chen. 2010. Beam structure optimization for additive manufacturing based on principal stress lines. In *2010 International Solid Freeform Fabrication Symposium*. University of Texas at Austin.

Liu, Chang, Zongliang Du, Weisheng Zhang, Yichao Zhu and Xu Guo. 2017. "Additive manufacturing-oriented design of graded lattice structures through explicit topology optimization." *Journal of Applied Mechanics* 84(8).

Liu, Jikai, Andrew T Gaynor, Shikui Chen, Zhan Kang, Krishnan Suresh, Akihiro Takezawa, Lei Li, Junji Kato, Jinyuan Tang, Charlie CL Wang et al. 2018. "Current and future trends in topology optimization for additive manufacturing." *Structural and Multidisciplinary Optimization* 57(6):2457–2483.

Liu, Jikai and Yongsheng Ma. 2016. "A survey of manufacturing oriented topology optimization methods." *Advances in Engineering Software* 100:161–175.

Lobo, Miguel Sousa, Lieven Vandenberghe, Stephen Boyd and Hervé Lebret. 1998. "Applications of second-order cone programming." *Linear algebra and its applications* 284(1-3):193–228.

Madeira, JF, HL Pina and HC Rodrigues. 2010. "GA topology optimization using random keys for tree encoding of structures." *Structural and Multidisciplinary Optimization* 40(1):227–240.

Makrodimopoulos, Athanasios, Atul Bhaskar and Andy J Keane. 2010. "Second-order cone programming formulations for a class of problems in structural optimization." *Structural and Multidisciplinary Optimization* 40(1-6):365.

Michell, Anthony George Maldon. 1904. "LVIII. The limits of economy of material in frame-structures." *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 8(47):589–597.

Neumaier, Arnold, Oleg Shcherbina, Waltraud Huyer and Tamás Vinkó. 2005. "A comparison of complete global optimization solvers." *Mathematical programming* 103(2):335–356.

Parkes, EW. 1975. "Joints in optimum frameworks." *International Journal of Solids and Structures* 11(9):1017–1022.

Prager, William. 1977. "Optimal layout of cantilever trusses." *Journal of Optimization Theory and Applications* 23(1):111–117.

Querin, Osvaldo M, Mariano Victoria, Cristina Alonso Gordoa, Rubén Ansola and Pascual Martí. 2017. *Topology design methods for structural optimization*. Butterworth-Heinemann.

Raich, AM and J Ghaboussi. 2000. "Evolving structural design solutions using an implicit redundant genetic algorithm." *Structural and Multidisciplinary Optimization* 20(3):222–231.

Rajeev, Srijith and CS Krishnamoorthy. 1992. "Discrete optimization of structures using genetic algorithms." *Journal of structural engineering* 118(5):1233–1250.

Rao, Singiresu S. and Ying Xiong. 2004. "A Hybrid Genetic Algorithm for Mixed-Discrete Design Optimization." *Journal of Mechanical Design* 127.
URL: *https://doi.org/10.1115/1.1876436*

Rasmussen, MH and Mathias Stolpe. 2008. "Global optimization of discrete truss topology design problems using a parallel cut-and-branch method." *Computers & Structures* 86(13-14):1527–1538.

Reintjes, Christian and Ulf Lorenz. 2021. "Bridging mixed integer linear programming for truss topology optimization and additive manufacturing." *Optimization and Engineering* 22(2):849–893.

Ribeiro, Tiago P, Luís FA Bernardo and Jorge Andrade. 2021. "Topology optimisation in structural steel design for additive manufacturing." *Applied Sciences* 11(5):2112.

Rocha, Miguel and José Neves. 1999. Preventing premature convergence to local optima in genetic algorithms via random offspring generation. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer pp. 127–136.

Sahinidis, N. V. 2017. *BARON 21.1.13: Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual.

Saka, Mehmet Polat and Zong Woo Geem. 2013. "Mathematical and metaheuristic applications in design optimization of steel frame structures: an extensive review." *Mathematical problems in engineering* 2013.

Saka, MP. 2007. "Optimum design of steel frames using stochastic search techniques based on natural phenomena: a review." *Civil engineering computations: tools and techniques, Saxe-Coburg Publications, Stirlingshire, UK* pp. 105–147.

Schenk, Olaf, Klaus Gärtner, Wolfgang Fichtner and Andreas Stricker. 2001. "PARDISO: a high-performance serial and parallel sparse linear solver in semiconductor device simulation." *Future Generation Computer Systems* 18(1):69–78.

Schmidt, Martin-Pierre, Claus BW Pedersen and Christian Gout. 2019. "On structural topology optimization using graded porosity control." *Structural and Multidisciplinary Optimization* 60(4):1437–1453.

Seront, Grégory and Hugues Bersini. 2000. A new GA-local search hybrid for continuous optimization based on multi level single linkage clustering. In *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation*. pp. 90–95.

Shahabsafa, Mohammad, Ramin Fakhimi, Weiming Lei, Sicheng He, Joaquim RRA Martins, Tamás Terlaky and Luis F Zuluaga. 2021. "Truss topology design and sizing optimization with guaranteed kinematic stability." *Structural and Multidisciplinary Optimization* 63(1):21–38.

Smith, Christopher J, Matthew Gilbert, Iain Todd and Fatos Derguti. 2016. "Application of layout optimization to the design of additively manufactured metallic components." *Structural and Multidisciplinary Optimization* 54(5):1297–1313.

Smith, CJ, I Todd and M Gilbert. 2013. Utilizing additive manufacturing techniques to fabricate weight optimized components designed using structural optimization methods. In *2013 International Solid Freeform Fabrication Symposium*. University of Texas at Austin.

Smith, Michael. 2009. *ABAQUS/Standard User's Manual, Version 6.9*. United States: Dassault Systèmes Simulia Corp.

Sokół, T and GIN Rozvany. 2013. "On the adaptive ground structure approach for multi-load truss topology optimization." *10th world congresses of structural and multidisciplinary optimization* .

Sokół, Tomasz. 2011. "A 99 line code for discretized Michell truss optimization written in Mathematica." *Structural and Multidisciplinary Optimization* 43(2):181–190.

Stolpe, Mathias. 2004. "Global optimization of minimum weight truss topology problems with stress, displacement, and local buckling constraints using branch-and-bound." *International journal for numerical methods in engineering* 61(8):1270–1309.

Stolpe, Mathias. 2007. "On the reformulation of topology optimization problems as linear or convex quadratic mixed 0–1 programs." *Optimization and Engineering* 8(2):163–192.

Stolpe, Mathias. 2016. "Truss optimization with discrete design variables: a critical review." *Structural and Multidisciplinary Optimization* 53(2):349–374.

Stolpe, Mathias and Krister Svanberg. 2003. "Modelling topology optimization problems as linear mixed 0–1 programs." *International Journal for Numerical Methods in Engineering* 57(5):723–739.

Sui, Yunkang, Jiazheng Du and Yingqiao Guo. 2006. "Independent continuous mapping for topological optimization of frame structures." *Acta Mechanica Sinica* 22(6):611–619.

Takezawa, A, S Nishiwaki, K Izui and M Yoshimura. 2007. "Structural optimization based on topology optimization techniques using frame elements considering cross-sectional properties." *Structural and Multidisciplinary Optimization* 34(1):41–60.

Talatahari, Siamak and Mahdi Azizi. 2020. "Optimal design of real-size building structures using quantum-behaved developed swarm optimizer." *The Structural Design of Tall and Special Buildings* 29(11):e1747.

Tang, Yunlong, Guoying Dong, Qinxue Zhou and Yaoyao Fiona Zhao. 2017. "Lattice structure design and optimization with additive manufacturing constraints." *IEEE Transactions on Automation Science and Engineering* 15(4):1546–1562.

Toğan, Vedat and Ayşe T Daloğlu. 2008. "An improved genetic algorithm with initial population strategy and self-adaptive member grouping." *Computers & Structures* 86(11-12):1204–1218.

Toragay, Oguz, Daniel F Silva, Alexander Vinel and Nima Shamsaei. 2022. "Exact global optimization of frame structures for additive manufacturing." *Structural and Multidisciplinary Optimization* 65(3):1–18.

Van Rossum, Guido and Fred L. Drake. 2009. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace.

Wächter, Andreas and Lorenz T Biegler. 2006. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming." *Mathematical programming* 106(1):25–57.

Wang, Di, Yongqiang Yang, Ziheng Yi and Xubin Su. 2013. "Research on the fabricating quality optimization of the overhanging surface in SLM process." *The International Journal of Advanced Manufacturing Technology* 65(9-12):1471–1484.

Wang, Huaguo and Hiroshi Ohmori. 2013. "Elasto-plastic analysis based truss optimization using genetic algorithm." *Engineering Structures* 50:1–12.

Wang, Weiming, Tuanfeng Y Wang, Zhouwang Yang, Ligang Liu, Xin Tong, Weihua Tong, Jiansong Deng, Falai Chen and Xiuping Liu. 2013. "Cost-effective printing of 3D objects with skin-frame structures." *ACM Transactions on Graphics (ToG)* 32(6):1–10.

Wang, Xiaojian, Shanqing Xu, Shiwei Zhou, Wei Xu, Martin Leary, Peter Choong, Ma Qian, Milan Brandt and Yi Min Xie. 2016. "Topological design and additive manufacturing of porous metals for bone scaffolds and orthopaedic implants: A review." *Biomaterials* 83:127–141.

Weaver, William and James M Gere. 2012. *Matrix analysis framed structures*. Springer science & business media.

Weldeyesus, Alemseged Gebrehiwot and Jacek Gondzio. 2018. "A specialized primal-dual interior point method for the plastic truss layout optimization." *Computational optimization and applications* 71(3):613–640.

Weldeyesus, Alemseged Gebrehiwot, Jacek Gondzio, Linwei He, Matthew Gilbert, Paul Shepherd and Andrew Tyas. 2019. "Adaptive solution of truss layout optimization problems with global stability constraints." *Structural and Multidisciplinary Optimization* 60(5):2093–2111.

Whitley, Darrell. 1994. "A genetic algorithm tutorial." *Statistics and computing* 4(2):65–85.

Wu, Jun, Anders Clausen and Ole Sigmund. 2017. "Minimum compliance topology optimization of shell–infill composites for additive manufacturing." *Computer Methods in Applied Mechanics and Engineering* 326:358–375.

Zavala, Gustavo R, Antonio J Nebro, Francisco Luna and Carlos A Coello Coello. 2014. "A survey of multi-objective metaheuristics applied to structural optimization." *Structural and Multidisciplinary Optimization* 49(4):537–558.