MASSACHUSETTS INSTITUTE OF TECHNOLOGY
JUN 07 2016
LIBRARIES
ARCHIVES

Volumetric Topology Optimization under Redundancy Constraints

by

Dimitrios Pagonakis
B.S., Civil and Environmental Engineering & Theoretical Mathematics,
Massachusetts Institute of Technology (2015)

Submitted to the Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degree of
MASTER OF ENGINEERING in Civil and Environmental Engineering
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2016

© Dimitrios Pagonakis, 2016. All rights reserved.
The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Author ..... Signature redacted
Department of Civil and Environmental Engineering
May 6, 2016

Certified by ..... Signature redacted
Dr. Corentin Fivet
Lecturer of Civil and Environmental Engineering
Thesis Supervisor

Certified by ..... Signature redacted
John Ochsendorf
Class 1942 Professor of Civil and Environmental Engineering & Architecture
Thesis Supervisor

Accepted by ..... Signature redacted
Heidi Nepf
Donald and Martha Harleman Professor of Civil and Environmental Engineering

Volumetric Topology Optimization under Redundancy Constraints

by

Dimitrios Pagonakis

Submitted to the Department of Civil and Environmental Engineering
on May 6, 2016, in partial fulfillment of the
requirements for the degree of
MASTER OF ENGINEERING in Civil and Environmental Engineering

Abstract

With ever-increasing pollution and scarcity of resources, structural optimization, the science of finding the optimal structural arrangements under equilibrium constraints, is becoming an increasing necessity in engineering practice. However, designers are hesitant to adopt a method that is by nature a limit state and thus potentially unreliable. This thesis embeds a level of safety, namely redundancy, within the structural optimization process. Redundancy is the ability to remove a certain number of elements from the structure without losing stability. The thesis translates this constraint into a linear mathematical optimization problem. Then, a topology optimization algorithm is developed that identifies the least volume structure with the ability to remove any element(s) while maintaining stability under the initial loading.

Besides the developed algorithm, this thesis shows the relation between the internal forces of redundant structures and their substructures, and in fact shows that it can be expressed linearly when only 1 level of redundancy is provided, and polynomial for higher levels. The algorithm is eventually implemented and extensively analyzed for a series of configurations, showing that redundant optimal shapes have considerably less volume than twice that of the pure volumetric optimal, and hence effectively combine safety with material efficiency. Overall, this thesis constitutes the early stage of a novel structural optimization algorithm that is unique to its volumetric optimization objectives.

Thesis Supervisor: Dr. Corentin Fivet
Title: Lecturer of Civil and Environmental Engineering

Thesis Supervisor: John Ochsendorf
Title: Class of 1942 Professor of Civil and Environmental Engineering & Architecture

Acknowledgments

"When you set out on your journey
to Ithaca, pray that the road is long,
full of adventure, full of knowledge."

— Constantine P. Kavafy, *Ithaca*

As my academic journey comes to an end at MIT (for now), it allows for some self-reflection. Five years ago, I arrived to the Institute excited about engineering and mathematics yet immature, confused and overwhelmed with no insight on research. After five years at MIT, thanks to my professors, the Institute's firehose and my community's support, I managed to grow as an engineer, and most importantly as a problem solver. I would like to thank all my friends and all the faculty who supported me throughout these five tough years and made sure I always stand up no matter how many times I fall.

This thesis would not have been possible without the invaluable help and support of my advisors Dr. Corentin Fivet and Professor John Ochsendorf. I would especially like to thank Dr. Fivet who inspired me with his challenging questions and spent countless hours by my side in this research journey. I look forward to the next time we cross paths in life again. I would also like to express my sincere appreciation to Professor Caitlin Mueller for all the inspiring conversations we had throughout this past year, and Professor Markus Buehler for welcoming me in his lab and making this year possible.

Finally, I would like to wholeheartedly thank my extended family here and back home, who have always been by my side. With their support anything is possible. Their love and smiles are the source of all inspiration I have in life. I will always be thankful to God, my mother Aristea, my father Konstantinos and my siblings no matter where my Ithaca lies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis introduces the concept of Volumetric Redundancy Optimization for two dimensional structural networks and particularly truss structures. The work presented here develops a computational algorithm that minimizes the material used in a truss structure for given loads while maintaining a form of structural safety, namely redundancy – the addition of structural components against fatigue or unforeseeable loads. Redundancy Optimization falls under the broad field of Structural Optimization, the science behind more lightweight and efficient structures. Through a combination of traditional structural mechanics, mathematics and optimization, the proposed algorithm has a wide spectrum of applications, ranging from minimizing the material of a truss bridge, to optimizing the shape of a hook. This thesis is structured as follows: In Chapter 1 a brief motivation of structural optimization and particularly redundancy optimization is provided. This Chapter concludes with the problem statement and potential applications of the algorithm. Chapter 2 covers a key part of the diverse literature review behind this multidisciplinary topic, and Chapter 3 explains the methodology and mathematics behind the redundancy optimization algorithm. In addition, Chapter 3 deducts the 1^{st} order redundancy solution as a linear program. The thesis then resumes with a generalization of the algorithm for higher order redundancies in Chapter 4, and presents a wide range of analysis and key takeaways from the results in Chapter 5. Finally, Chapter 6 concludes with a summary of the key findings and contributions of this work, along future work and

potential improvements in the suggested approach.

1.1 Motivation for Structural Optimization

The rising cost and energy intensiveness of structural materials makes it imperative for researchers and engineers to invent novel building methods that are performed in a more economical, sustainable and environmental manner. According to the U.S. Energy Information Administration (EIA), the building sector consumes nearly half of all energy in the US (Fig. 1-1). Furthermore, the building sector accounts for nearly half of the U.S. CO_2 emissions, while the transportation sector comes second with about a third of the total emissions of CO_2 (Fig. 1-2).

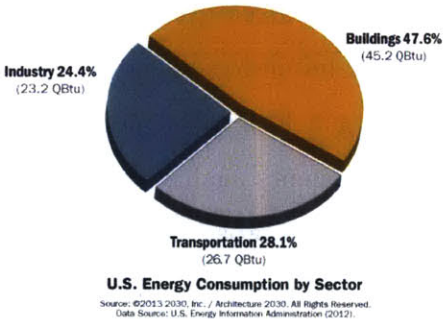

Figure 1-1: The building sector is responsible for nearly half US energy use

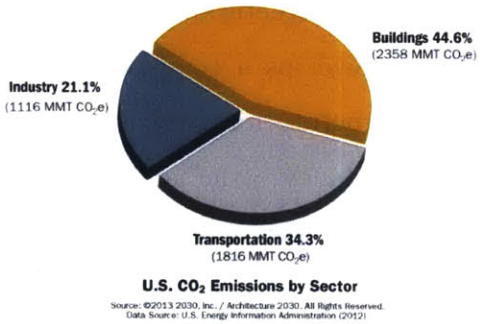

Figure 1-2: Buildings are the largest source of CO_2 emissions in U.S.

Therefore, given the global climate crisis, scarceness of resources and the financial challenges our society is facing, the need for more sustainable structures cannot be put aside. For this reason, we have to develop a deeper understanding of the underlying connection between form and forces, in order to design buildings that are both structurally feasible and material-efficient.

Indeed, finding ways to minimize the amount of material used, and hence the cost of a building, has been a topic of active research for more than a century. The first one to study the relationship between topological shapes and forces is Maxwell (Maxwell, 1869). In early 20^{th} century, Michell developed a method to find optimal geometrical shapes of framing truss structures that minimizes material and self-weight

(Michell, 1904). Since then we know that when the geometry of a structure is wisely designed, *the material properties do not particularly matter.* In other words, any cheap material could be used as a structural basis, given that the geometry of the design was developed in such a topologically optimal way. Since the 16th century, masons have empirically developed methods of compression-only arches that make structures lighter and more stable. Nevertheless, until today a robust answer to structural optimality is not always analytically known. This is a hot topic in Building Technology, Structural Engineering and Optimization Research, as there is a large margin for improvements on the efficiency, constructibility and costs of building.

Even though structural optimization is a very prolific academic topic, it is still fairly primitive and foreign to the greater construction industry (with a few companies – exceptions). Several reasons is the case are: the very intricate designs structural optimization often proposes (Schafer et al., 2005), the fairly conservative atmosphere among structural engineering firms, and the lack of reliability and structural safety among optimized solutions. That is, since structural optimization minimizes the material used, it is by definition a limit state, meaning that it reaches the material to its limits. Hence, any additional safety net is removed for the sake of material efficiency and even tiny fluctuations on the load cases or partial damage in the structure can unexpectedly propagate to the building's global failure (Fig.1-3). Thus, even if structural optimization is a reality, practically very little construction firms would take the risk in implementing such solutions on a commercial scale. There is a clear need to embed safety of a building and additional rigidity within the optimization process itself. Structural redundancy, the use of additional members and/or materials in construction to provide additional layers of safety, has long been regarded as a desirable property to ensure the safety of structural systems. Building codes usually get away with structural redundancy by adding rough factors of safety and the addition of material in a structure to account for unforeseeable uncertainties and fatigue. However, this can sometimes have the opposite effect, as the addition of material in certain locations can increase the stress certain critical elements take, and eventually induce failure or buckling. The majority of optimization procedures primarily deal with

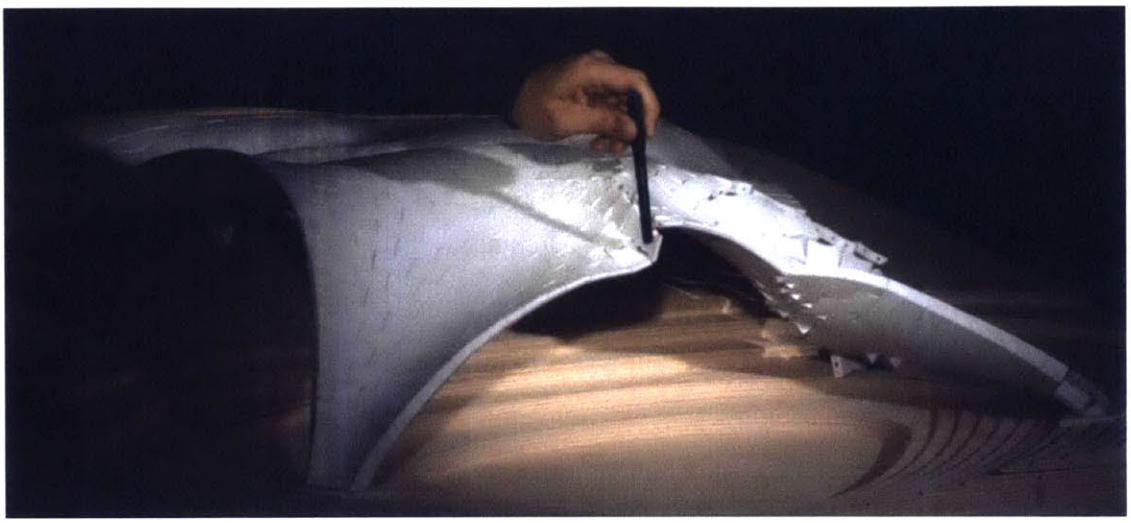

Figure 1-3: Discrete element vault model – an element's failure results in the vault's global collapse (Rippmann et al., 2013)

material/labor cost minimization under first order equilibrium constraints, without further studying post buckling or partial failure of the structural system (Okasha et al., 2009). Hence, the need to incorporate redundancy and additional safety measures in structural optimization is evident. It is worth mentioning that as of now there is neither a uniform theory of structural redundancy nor any widely agreed definitions (Schafer et al., 2005). Hence, instead of proceeding with a very sophisticated and specific redundancy definition, we use a simple constraint that can provide the higher levels of safety pure structural optimization lacks.

1.2 Problem Statement

This thesis examines a special case of structural redundancy, inspired by the *Mathematical Bridge* in Queen's college, University of Cambridge (Caston, 2012), as a partial solution to safer structural optimization. We decide to simply define redundancy, as the ability to remove any elements of the truss system while maintaining stability structure under the same external loads. Embedding this constraint inside structural optimization would result in not just lightweight, but also very safe structures, while providing us with a better understanding of how to wisely allocate redundancy inside

a structure. Hence, the problem this thesis tackles is:

What is the least volume truss network, such that any n elements can be removed, while the structure remains stable?	(1.1)

This problem is formulated mathematically using topology optimization and is solved analytically under a material plasticity assumption. The solution is provided both for $n = 1$ and $n \geq 2$ elements, however computationally the solutions that are presented are only for $n = 1$ and $n = 2$ or first and second order redundancy as defined in the next Chapter.

1.3 Feasibility – Applications

Most recent reliability-based optimization research embeds robustness i.e. uncertainty in the applied loads when optimizing a structure to account for additional safety. Very little research is done in *redundancy* optimization compared to *robustness*. Even though accounting for robustness allows a design that can withstand several uncertain loading scenarios, a redundant structure is the more tangible counterpart, where even if some of it fails, the rest of the structure remains stable. This is something a robust design does **not** guarantee. What is more, several need for solving such a problem are provided below.

1.3.1 Bridge Design & Maintenance

The majority of US bridge failures has been accounted due to a defect of a *critical element* and due to poor maintenance. According to US infrastructure news in 2015 (Faturechi et al., 2015):

"The crossings are kept standing by engineering design, not supported with brute strength or redundant protections like their more modern counterparts. Bridge regulators call the more risky spans "fracture critical," meaning that if a single, vital com-

ponent of the bridge is compromised, it can crumple. Thousands of bridges around the U.S. may be a mistake away from collapse, even if the global spans are deemed structurally sound".

Developing an algorithm that can set a bridge design to be as lightweight as possible, while neutralizing the criticality of all of its structural components, can create bridges that are very strong and fracture resistant.

What is more, maintaining conventional bridges usually costs millions of dollars in material and delays, as typically the whole bridge needs to be supported and then step by step renovated. Implementing a redundant optimization algorithm would allow each structural component of a new structure to be replaced immediately, without any additional major external support or having to block traffic for extended periods of time. Hence, bridges could benefit a lot from such a structural solution.

1.3.2 Transmission Towers

Transmission towers are a typical example of structural optimization. In fact, the current shape of large steel transmission towers around the world is a result of volume optimization (An et al., 2010) (Fig.1-4). Nevertheless, when a component of the tower fails, the whole structure fails, inducing huge indirect costs from power surges. Hence building a tower that can withstand its design loads while a set of elements have failed, would allow enough time for its appropriate maintenance.

1.3.3 Target Structures – Government Buildings

Many buildings including government buildings can frequently be targets of terrorists groups. Hence, they need to be constructed in a very robust and redundant way to ensure the safety of their tenants. A high level of redundancy on the optimization would allow the engineers to design structures that are still the most efficient possible while having a very large degree of redundancy. For example if the structure was designed for 100^{th} degree redundancy, i.e. more than any 100 structural components could be removed while the structure still stands, then it would make it extremely

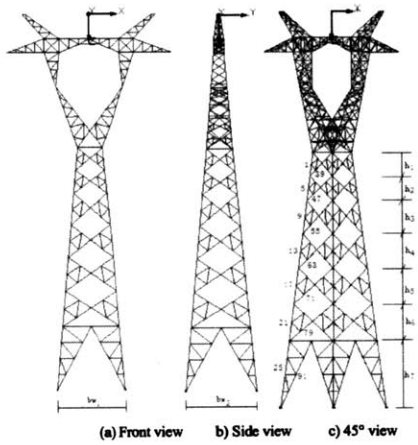

Figure 1-4: An optimized transmission tower for wind and gravity loadings (An et al., 2010).

hard for any terror group to target a particular part of the structure and induce global damage.

1.4 Summary

Redundancy accounting structural optimization, is a fairly immature topic. Providing an answer in (1.1) could shed some light in the unexplored field of redundant structures, while providing some insight on the relationship between forces inside a damaged substructure and the resulting shape. In Section 5 several visualizations are shown that showcase these relations. Next, Chapters 3 and 4, the mathematical backbone of this thesis, follow.

Chapter 2

Literature Review

In this section, the concepts of Topology Optimization and Redundancy/Robustness are encountered. A brief introduction and literature review of each topic/theory is provided followed by an expository on Plastic Layout Optimization and the GRAND framework, which constitute a major part of the computational work developed in this thesis.

2.1 Topology Optimization

Topology Optimization is a mathematical approach that implements the use of finite element methods. to optimize a material layout within a given design space, boundary and loading conditions. In this particular practice, topology optimization generates an initial mesh/density – *a ground structure* – with all possible connections among the structure and iteratively removes elements/reduces the density until it minimizes the objective function (usually compliance or volume), subject to the desired constraints.

The generic formulation of the topology optimization problem for compliance minimization of statically loaded structures proposed by Sigmund (Sigmund, 2001) takes

the form

$$\begin{aligned} \min_{t} \quad & c(t) = \mathbf{U}^{\mathbf{T}}\mathbf{K}\mathbf{U} \\ \text{s.t.} \quad & \mathbf{K}\mathbf{U}(t) = \mathbf{F} \\ & \mathbf{M}t \leq d \\ & t \geq 0, \quad u \in \mathbb{R}^N, t \in \mathbb{R}^m, \end{aligned} \tag{2.1}$$

where $u = diag(\mathbf{U})$ the vector of displacements for each node, $\mathbf{K}$ the stiffness matrix of the system, $\mathbf{F}$ the external applied force vector and $\mathbf{M}, d$ some upper limit factors for the volume vector t of each truss in the system.

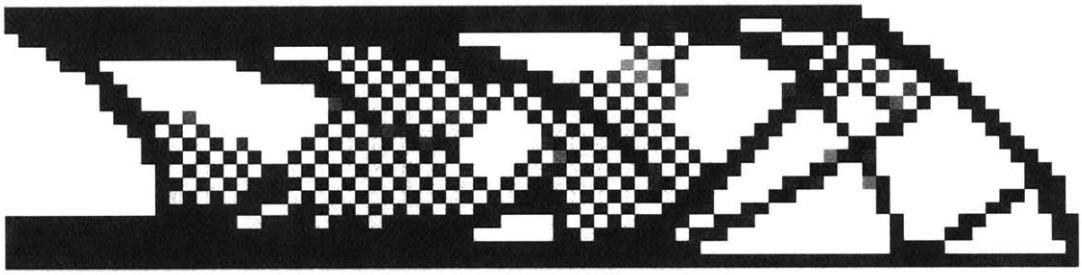

Figure 2-1: Matlab output of generic TopOpt (Sigmund, 2001) code for an 80×20 FE grid with $1/2$ of the volume as a maximum attained volume constraint.

It is worth noting at this point that Sigmund's algorithm, like the majority of topology optimization ones, begins with a Finite Element Check board where every element is initially existent to its maximum volume t[1]. Then, given fixities, loads and boundary conditions, each volume element is minimized subject to the given constraints. This ground structure is initialized using a filtering technique to ensure existence of solutions and to create a refined starting mesh (Sigmund et al., 1998). Sigmund admits that this filter has not yet been proven to ensure existence of solutions and that it is rather empirically found through several practical design solutions (Sigmund, 2001) . Most importantly, the current filter does not necessarily guarantee *mesh-independency* and this is currently one of the greatest drawbacks of ground structures. Nevertheless, experience has shown that the current filter is sufficiently independent for traditional topology optimization problems.

[1]This check board is the corresponding Ground Structure in continuum space

The positive side of Sigmund's paper is that his 99 Matlab code is very straightforward and customizable for multiple loading cases, different conditions and fixities. However, for the objective of this research, TopOpt is not applicable given that truss systems cannot be simply modeled as finite element blocks. Instead, a discrete topology optimization with nodes and branches would be more useful.

Freund gives several equivalent convex, semidefinite and second order conic formulations of this problem on which the basic assumptions align with a ground structure of not finite blocks but rather points and connections (Freund, 2004).

Freund's convex formulation takes the following topology optimization form

$$\begin{aligned} \min_{f,t} \quad & \textstyle\sum_{t_k>0} \frac{1}{2}\frac{1}{t_k}\frac{L_k^2}{E_k} f_k^2 \\ \text{s.t.} \quad & \textstyle\sum_{t_k>0} a_k f_k \;=\; -F \\ & Mt \;\leq\; d \\ & t \;\geq\; 0 \\ & f \in \mathbb{R}^m \quad , \quad t \in \mathbb{R}^m. \end{aligned} \tag{2.2}$$

The truss design problem is to choose the values of the volumes $t = (t_1, ..., t_m)$ on the bars so that the optimal solution (the objective function represents the compliance of the truss) is minimized, subject to the constraints on t: $Mt \leq d$ and $t \geq 0$, where the constraint Mt is simply a volumetric scaling constraint. a_k are the rows of the connectivity – geometry matrix of the truss system. Hence each column has exactly two nonzero entries. This problem can be solved analytically, yet computationally it is fairly costly for large refined Ground Structures (GS).

2.1.1 Plastic Layout Optimization

Zegard and Paulino (Zegard et al., 2014) converted Freund's optimization formulation to a linear optimization problem by adding a plasticity assumption, i.e. no explicit compatibility or stress-strain relations are assumed. Instead, each internal axial force has to lie within a range of critical tensile and compressive forces. This particular formulation is quite elegant and the computational power and breadth of

its solutions are worth the cost of the additional plasticity assumption. The plasticity volumetric optimization formulation is given by

$$\begin{aligned} \min_{a} \quad & V = \mathbf{l^T a} \\ \text{s.t.} \quad & \mathbf{A^T f} = \mathbf{F} \\ & -\sigma_C a_i \leq f_i \leq \sigma_T a_i, \ \forall i = 1, 2...N_e, \end{aligned} \tag{2.3}$$

where $\mathbf{l}$ the truss lengths vector, $\mathbf{a}$ the truss cross section vector, $\mathbf{A}$ the nodal equilibrium matrix of size $N_n \times N_e$, $\mathbf{f}$ the internal force vector, $\mathbf{F}$ the external applied forces, and σ_C, σ_T the stress limits in compression and tension respectively. In the geometry matrix $\mathbf{A}$, each row represents one node and each column one element. In optimization theory, it is well known that the optimum is reached when the constraints are active, i.e. the equality holds. Thus, the intuitive solution would be to reduce the cross section of each member if $f_i < \sigma_T a_i$ or if $f_i > -\sigma_C a_i$, since the stress constraint must be active for all variables at the optimum. Hence, following Hemp's suggestion (Hemp, 1973) of introducing slack variables in the stress constraints, converts the inequalities into equalities:

$$f_i + 2\frac{\sigma_0}{\sigma_C} s_i^- = \sigma_T a_i \tag{2.4a}$$

$$-f_i + 2\frac{\sigma_0}{\sigma_T} s_i^+ = \sigma_C a_i, \tag{2.4b}$$

where all the variables can be written in terms of the slack variables:

$$a_i = \frac{s_i^+}{\sigma_T} + \frac{s_i^-}{\sigma_C} \tag{2.5a}$$

$$n_i = s_i^+ - s_i^-, \tag{2.5b}$$

where n_i the internal force vector. These manipulations convert the problem to a linear programming problem with $\mathbf{s}^+, \mathbf{s}^-$ variables:

$$\begin{aligned} \min_{\mathbf{s}^+, \mathbf{s}^-} \quad & V = \mathbf{l^T}\left(\tfrac{\mathbf{s}^+}{\sigma_T} + \tfrac{\mathbf{s}^-}{\sigma_C}\right) \\ \text{s.t.} \quad & \mathbf{A^T}(\mathbf{s}^+ - \mathbf{s}^-) = \mathbf{F} \\ & s_i^+, s_i^- \geq 0 \end{aligned} \tag{2.6}$$

Note that for each element i only one of s_i^+ and s_i^- is nonzero. If element i is in tension then $s_i^+ > 0$, and if in compression then $s_i^- > 0$. Defining the stress limit ratio as $\kappa = \frac{\sigma_T}{\sigma_C}$ the final linear formulation of the *plastic topology optimization problem* (Sokol et al., 2012) (Tyas et al., 2010) is:

$$\begin{aligned} \min_{\mathbf{s}^+,\mathbf{s}^-} \quad & V = \mathbf{l}^\mathbf{T}\left(\mathbf{s}^+ + \mathbf{s}^-\right) \\ \text{s.t.} \quad & \mathbf{A}^\mathbf{T}(\mathbf{s}^+ - \mathbf{s}^-) = \mathbf{F} \\ & s_i^+, s_i^- \geq 0 \end{aligned} \tag{2.7}$$

This formulation is selected as one of the currently most appropriate ways to model the particular redundancy problem. The ground structure approach developed in accordance to this formulation (Zegard et al., 2014) has significantly more distinct assumptions compared to traditional topology optimization (Sigmund, 2001), such as:

- Bars with section areas instead of finite elements with given volume
- Accurately counts the number of trusses used
- Accounts for co-linear trusses and removes them accordingly
- Can avoid occurrence of singular topologies, one of the TopOpt's most significant drawbacks (Sokol et al., 2012)

Hence, the Plastic Layout Optimization is inherited for the development of the redundant optimization formulation. Zegard's Plastic Layout Formulation proves to be computationally efficient and it is the only discrete linear topology optimization. In addition, a Matlab script has been developed by Zegard and Paulino (Zegard et al., 2014), GRAND, that allows for the linear plastic topology optimization of any 2D or 3D structure. More on the computational aspect of this algorithm are encountered in Section 2.3. On the next section, the notions of Redundancy and Robustness are introduced, and their relation to topology optimization.

2.2 Structural Redundancy & Robustness

Now that the basic notions of topology optimization have been established, a brief introduction on robustness and redundancy concepts is presented. Both redundancy and robustness fall under the general umbrella of reliability engineering. Reliability engineering deals with the estimation, prevention and management of high levels of "lifetime" engineering uncertainty and risks of failure from over-stressing/overloading or fatigue.

Definition 1. Robustness is the ability of a structure to withstand multiple unforeseen loading cases that it has not been primarily designed for.

In simple words, it is the ability of the structure to withstand several load cases with given uncertainty of occurrence. On the other hand, redundancy is a similar notion but does not account for uncertain loadings. Instead, Redundancy provides a more general safety net as defined by Kanno et al (Kanno, 2011):

Definition 2. Redundancy of a structure is defined as the extent of degradation the structure can suffer without losing some specified elements of its functionality.

Kanno et al, provide a survey of redundancy concepts with their respective definitions.

2.2.1 Indeterminacy as a measure of Structural Redundancy

The most classical measure of structural redundancy is the degree of static indeterminacy, as it implies the existence of alternative loads paths. The degree of indeterminacy is defined by

$$s = N - \text{rank}H, \tag{2.8}$$

where N the number of internal forces, $H \in \mathbb{R}^{d \times N}$ the geometry matrix (as defined in the Plastic Layout Formulation) and d the number of degrees of freedom of displacements. Even though alternative load paths seem like a promising approach to redundancy, given that a failure of an element would not limit all pathways of a load to the fixities, Frangopol et al (Frangopol et al., 1987) and Pandey et al (Pandey et

al., 1997) noted that this can actually be a deceiving metric, inaccurately evaluating the true performance of a structural system. Instead Frangopol (Frangopol et al., 1987) defined the strength factor

$$r = \frac{l_{intact}}{l_{intact} - l_{damaged}}, \tag{2.9}$$

where l_{intact} the ultimate strength of the intact structure and $l_{damaged}$ the ultimate strength of the damaged structure. They asserted the the strength factor r performed better as a factor of redundancy in a structural system compared to indeterminacy. Such a factor was naturally extended to a probabilistic uncertainty counterpart by Okasha and Frangopol (Okasha et al., 2009) as:

$$r = \frac{P(D) - P(C)}{P(C)} \tag{2.10}$$

where $P(C)$ the probability of global collapse and $P(D)$ the probability of a structural element failure. This ratio then defines a residual strength index that has been used as a measure of redundancy.

Schafer et al (Schafer et al., 2005) implemented their own engineering demand parameter (EDP) based on eigenvalue buckling analysis:

$$(\mathbf{K}_l - \lambda_{cr}\mathbf{K}_g(\mathbf{P}))\phi = 0, \tag{2.11}$$

where $\mathbf{K}_l$ the elastic stiffness and $\mathbf{K}_g$ the geometric stiffness. $\mathbf{K}_g$ is a function of the internal forces that develop due to $\mathbf{P}$. The buckling load is $\lambda_{cr}\mathbf{P}$ where λ_{cr} is a scalar multiplied by the reference load and ϕ is the mode shape of the buckling load. The elastic buckling load multiplier λ_{cr} is considered an attractive metric by Schafer since it is a single scalar metric and it is related formally to stability but also gives a sense of buckling constraints. Schafer then tests outs several frame examples and through single element removal measures how λ_{cr} varies and checks whether this accurately measures redundancy of the reduced structure (Fig.2-2).

Safari et al (Safari, 2012) and Guilani et al (Guilani et al., 2014) present the

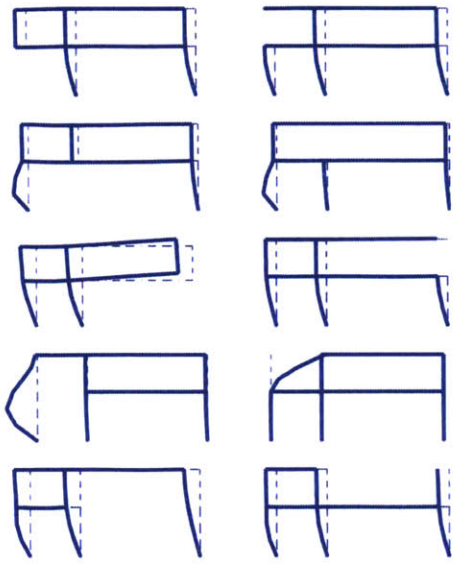

Figure 2-2: Measuring stability degradation and redundancy in damaged structures (Schafer et al., 2005).

redundancy allocation problem (RAP). RAP involves the simultaneous selection of components and a system-level design configuration that can collectively meet all design constraints in order to optimize some objective functions, such as system cost and/or reliability. Through a multi-objective optimization (Safari, 2012) or Genetic Algorithms (Guilani et al., 2014) and stochastic functions they provide case-specific measures of the optimal allocation of redundancy in a structure. Surprisingly as of now there has still not been a universally accepted redundancy definition (Ghista, 1966) (Zhou, 2014). Instead, researchers develop highly sophisticated metrics that can hardly be generalized and used in every single case. The most general common ground behind redundancy is that it has to do with the allocation of additional–redundant material in order to withstand unexpected degradation. Hence, for the remaining of this thesis, a broader simplified metric of redundancy is introduced:

Definition 3. n^{th} order redundancy of a structure is the ability to remove any $n \in \mathbb{N}$ elements of the structure while maintaining stability.

This simple and widely applicable redundancy metric is in accordance with most, if not all, much more complicated uncertainty-accounting metrics.

2.2.2 Robustness and Topology Optimization

Besides the theoretical nature of universally accepted robustness and redundancy terms, researchers have actively attempted to find ways to account for such notions when designing the shape of a structure. Ghista (Ghista, 1966) recognizes that topology optimization alone reaches the structure to a limit state removing any robustness in the structure. Most researchers (Save et al., 1989) (Ben-tal et al., 1997) (Frangopol et al., 1987) primarily focus on optimizing a structure not just for a single load case, but under uncertainty, they add stochastic load cases and optimize for them. For instance Ben-Tal and Nemirovski (Ben-tal et al., 1997) designed a Semidefinite program that optimizes the rigidity of a truss system with respect to both given loading scenarios and small probabilistic occasional loads. The key contribution from this research is that they designed an innovative *robustness constraint.* The robustness constraint stems from the ability of the designer/engineer to embed a small finite set of loads she/he is especially interested in (the primary loads), into a more massive set that contains additionally occasional loads of perhaps much smaller magnitude (secondary loads), and optimizes for the truss $t \in T$ which minimizes the worst-case compliance $c^M(t)$ taken with respect to the extended set M of loading scenarios. The researchers picked an ellipsoid for the set M centered at the origin (in order to get a tractable solution):

$$M = QW_q \equiv \{Qe | e \in \mathbb{R}^q, e^T e \leq 1\}, \tag{2.12}$$

where Q given $n \times q$ "scale" matrix and W_q is the unit Euclidean ball in R^q. The corresponding robust optimization problem then takes the form: *find* $t \in T$ *which minimizes the compliance*

$$c^M(t) = \max_{e^T e \leq 1} \max_{x \in \mathbb{R}^n} [2(Qe)^T x - x^T A(t) x]. \tag{2.13}$$

where x the vector of nodal displacements and t the bar volume vector. The solution of this semidefinite program gives very interesting two and three dimensional results (Fig. 2-3).

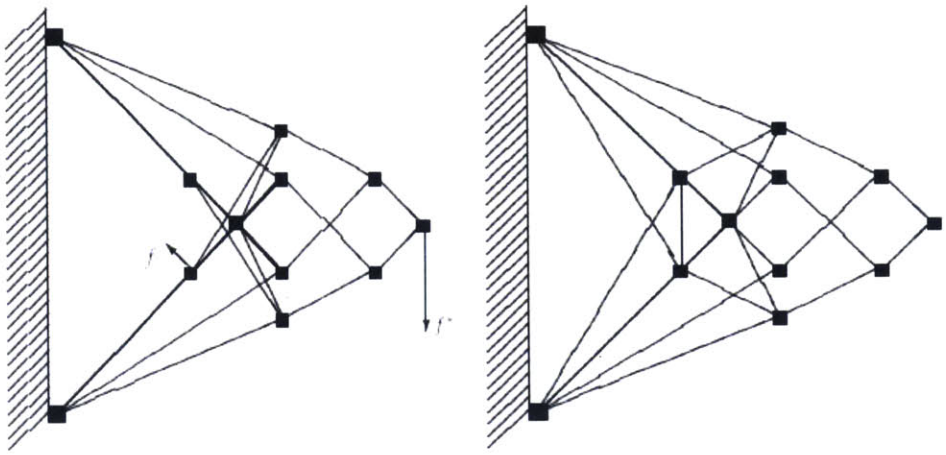

Figure 2-3: Bental's Robust Truss of Michell Cantilever (right) vs. sole Optimal (left)

Robust topology optimization has also been considered from a volumetric perspective. Mohr et al (Mohr et al., 2012) performed robust topology optimization with regard to the volume of a truss for multiple load cases given continuous probability distributions (Fig. 2-4).

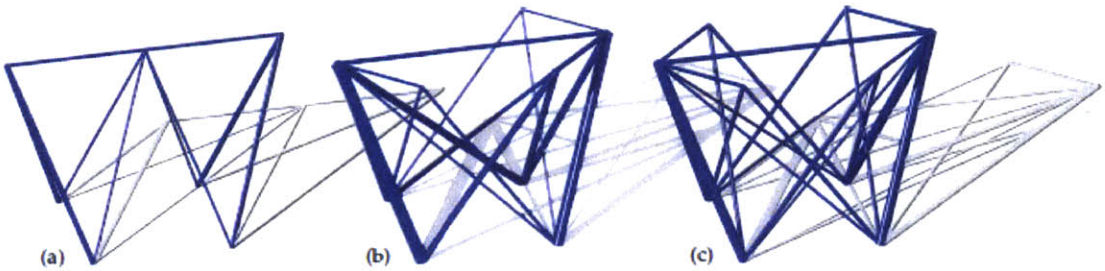

Figure 2-4: (a) Volumetric Optimal (b),(c) Equivalent robust optimal solutions (Mohr et al., 2012)

Interestingly enough, redundancy in topology optimization has been minimally researched, possibly given the difficulty to embed globally accepted redundancy constraints in the optimization formulation and acquire a tractable problem. Nevertheless, we will show in this thesis that combining definition 3 and the computational tools of Plastic Layout Optimization presented in Section 2.3, can result in a well-defined Redundancy Topology Optimization problem.

2.3 GRAND Topology Optimization

Zegard and Paulino (Zegard et al., 2014) developed a Matlab code that performs volumetric topology optimization for arbitrary grid structures. The code implements the plasticity assumption and plastic layout optimization as discussed in Section 2.1.1 to convert the topology optimization into a linear program. The algorithm works as follows:

- Input initial Ground Structure and level of connectivity
- Get loads, fixities, nodes and bars – from base mesh and construct the connectivity matrix of the Ground Structure
- Solve linear optimization for $L^T \cdot \vec{f}$ subject to stability constraint $A^T \vec{f} = F$
- Acquire optimal force vector and through fully stressed assumption obtain member thicknesses
- Draw the optimal structure

Zegard generates discrete Ground Structures in a highly efficient way by inputting the level of connectivity of a ground structure i.e. to what extent each node is connected with each other, and also by removing any co-linear trusses. Figure 2-5 shows how the base mesh and the levels of connectivity work.

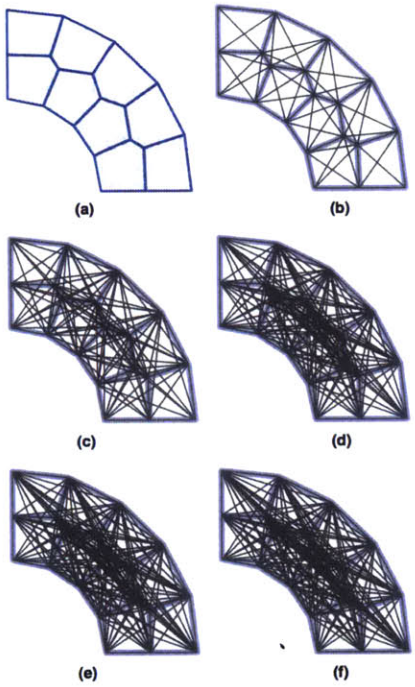

Figure 2-5: Ground Structure connectivity level generation: (a) Base mesh. (b) Level 1 connectivity. (c) Level 2. (d) Level 3. (e) Level 4. (f) Level 5. (Zegard et al., 2014)

GRAND is one of the fastest discrete Topology Optimization algorithms that allows the optimization of complicated initial Ground Structures (Figure 2-6), and is the backbone of the Redundant Topology Optimization algorithm developed in this thesis.

2.4 Summary

This section provided a brief overview of the theories that are milestone of this thesis, namely: Discrete and Continuous Ground Structure Topology Optimization, Robustness & Redundancy as well as the GRAND computational framework in Matlab. Combining techniques between these fields and inspired by a new straightforward redundancy definition (3), the following sections explain the novel Redundancy Optimization Algorithm.

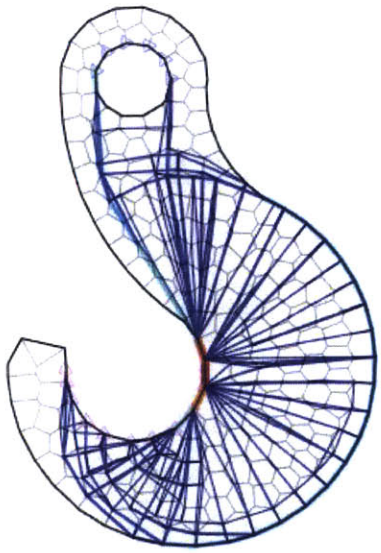

Figure 2-6: Optimal hook structure overlaid on initial Ground Structure (Zegard et al., 2014)

Chapter 3

Mathematical Formulation & Implementation of First Order Redundancy

In the aforementioned literature review, a variety of different structural volumetric or strain energy optimizations were covered. In addition, several optimizations have been performed that take robustness into account, either from a stochastic loading perspective (where the loading vectors have uncertainty functions embedded in them), or multiple loading conditions under probabilistic distributions. The work within this thesis is from the perspective of element redundancy, and the ability to identify structures that are volumetrically optimal, and yet allow the simultaneous removal of any element(s) and stability. In this chapter, the mathematical formulation for a general structural redundancy problem is provided, and further solved and modeled for a single element removal (1^{st} order redundancy). Both continuous and integer redundancy optimizations are encountered in order to compare and contrast the structural efficiency and performance of each algorithm. The Chapter concludes with some state of the art remarks and key ideas.

3.1 Methodology – Plastic Layout

The following development of the structural redundancy problem draws from the plasticity assumption (Zegard et al., 2014) described in Chapter 2 and the Ground Structure approach (Hagishita et al., 2009). Sacrificing elasticity in the particular case is worth it, given the large simplification that incurs by deriving a linear program. In fact, plastic design makes even more sense when dealing with redundancy, since both deal with *ultimate limit states.* Indeed, safety through redundancy is a matter of stability under extreme conditions and hence it is not matter of seviceability. The plastic volumetric optimization formulation in (2.7) with the additional redundancy constraints can be used to generate the initial ground structure to perform the desired optimization. The first step in solving the problem, is to mathematically interpret the redundancy constraint of *equilibrium for any substructure generated by removing one element.*

Before proceeding with the formulation, the following definition is provided, to clarify the process to follow.

Definition 4. Redundancy of degree i implies that the number of any elements that can be removed from the system is i, while maintaining stability in the rest of the structure.

Definition 5. In an initial ground structure even though full connectivity is initially assumed, meaning that all nodes of the initial structure are connected with each other, a collinearity check is always performed, removing trusses that are collinear and accounting for only one element (Fig. 3-1).

Definition 6. The number of nodes and elements of the ground structure are N_n, N_e respectively. E_k, t_k, l_k, a_k, f_k and u_k the Young's modulus, volume, length, cross sectional area, internal force and displacement of element k respectively.

In this section, the *level* 1 *redundancy* constraint is formulated mathematically in order to be compatible with the plastic layout optimization. For simplicity, the constraint of traditional topology optimization is revisited and then adjusted for the plastic formulation.

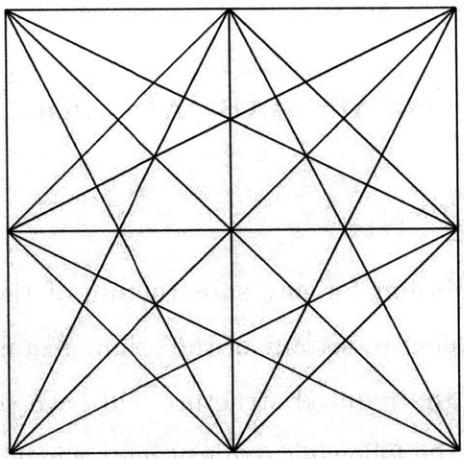

Figure 3-1: Fully connected 2×2 square ground structure.

The 2D equilibrium equation for a system of trusses is

$$\mathbf{Af} = \mathbf{F}, \tag{3.1}$$

where $\mathbf{A}$ the $N_n \times N_e$ node equilibrium matrix, excluding fixities[1] and $\mathbf{F}$ the $N_n \times 1$ external loads vector. For the redundancy constraint, at least one additional degree of indeterminacy is required. From Freund (Freund, 2004), the internal force f_k is given as a function of the displacements.

$$f_k = -\frac{E_k}{l_k^2} t_k \mathbf{A_k^T u} \Leftrightarrow \mathbf{f} = -\mathbf{B^{-1}A^T u}, \tag{3.2}$$

where

$$\mathbf{B} = \begin{pmatrix} \frac{l_1^2 t_1}{E_1} & & & \\ & \frac{l_2^2 t_2}{E_2} & & \\ & & \ddots & \\ & & & \frac{l_{N_e}^2 t_{N_e}}{E_{N_e}} \end{pmatrix}.$$

Rearranging equation (3.2) by multiplying with $\mathbf{B}$ on both sides results in

$$\mathbf{Bf} + \mathbf{A^T u} = 0 \tag{3.3}$$

[1] In this thesis we always assume that there are enough fixities for stability

Combining equations (3.1),(3.3)

$$\mathbf{F} = \mathbf{Af} = -\mathbf{AB^{-1}A^{T}u} = \mathbf{Ku}, \tag{3.4}$$

where $\mathbf{K}$ the stiffness matrix.

Now to express equilibrium for any substructure of the truss system with one element removed is equivalent to set any of the columns of matrix $\mathbf{A}$ to be zero. The matrix equation (3.1) of the reduced structure will have solutions $\mathbf{f}_i$. To generate general reduced matrices the following *removal matrix* is defined.

Definition 7. Let the $N_e \times N_e$ removal matrix $\mathbf{R}_i$, defined as

$$\mathbf{R}_i = \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 1 \end{pmatrix},$$

where there is a 0 only in the i^{th} entry of the diagonal and everywhere outside the diagonal.

Equilibrium at the reduced substructure without element i can then be written as

$$\mathbf{AR}_i\mathbf{f}_i = \mathbf{F} \tag{3.5}$$

From (3.4), the stiffness of the full structure was derived to be $\mathbf{K} = \mathbf{AB^{-1}A^{T}}$. In a similar manner, any substructure with element i removed has a stiffness

$$\mathbf{K}_i = \mathbf{AR}_i\mathbf{B^{-1}R}_i\mathbf{A}^T \tag{3.6}$$

3.2 Redundancy-accounting Formulation

This section develops the final linear formulation of the 1^{st} order redundancy problem. Since, this development is not straightforward, most of the remaining Chapter is dedicated to the development of the Redundancy-accounting formulation. To construct the optimization, three new key theorems are proved, and combined together they make the problem tractable. The first theorem that is developed, identifies the relationship between the internal forces of the redundant structure with the internal forces of its reduced counterpart. In fact we show that for 1^{st} order redundancy, that relation is linear. In the second theorem, we show that the system of equilibrium in the reduced substructures always has a solution if and only if the original structure is indeterminate. In simple words, this means that when formulating the redundancy optimization problem, the number of redundant constraints is always less than the number of unknown internal force vectors. Next, this section provides a third essential theorem, which shows that the ideal objective function of the 1^{st} order redundancy optimization is the load path of the substructure when the most critical element of the original structure is missing. The most critical element is defined as the element with the highest internal force that passes through it. After showing these three important results in detail, we present the linear 1^{st} order redundancy optimization algorithm.

With the redundancy constraint derived to be $\mathbf{K}_i\mathbf{u}_i = \mathbf{F}\ \forall i \in [1, ...N_e]]$, the traditional strain energy topology optimization formulation (Sigmund, 2001) takes the form

$$\begin{aligned} \min_{t} \quad & c(t) = \mathbf{U^T K U} = \tfrac{1}{2}\mathbf{F^T u} \\ \text{s.t.} \quad & \mathbf{K U}(t) = \mathbf{F} \\ & \mathbf{K}_i(t)\mathbf{u}_i = \mathbf{F} \\ & \mathbf{M}t \leq d \\ & t \geq 0, \qquad\qquad u \in \mathbb{R}^N, t \in \mathbb{R}^m, \end{aligned} \tag{3.7}$$

Although this formulation has multiple constraints that need to be calculated for all reduced substructures, the objective function is vague in the sense that optimizing

the strain energy of the initial structure does not guarantee global optimality. To put this in context, the strain energy of the structure could be minimized and yet removing one element could result in a huge change of the strain energy. *A better objective function would aim to minimize the strain energy of all substructures.*

The multiple drawbacks of this formulation drive the introduction of plasticity. In the plastic layout optimization formulation, only equation (3.5) is required. An additional constraint in the plastic case, is that the internal forces of the members in the reduced structures should lie within the limit stresses:

$$-\sigma_C a_i \leq f_{i_{red}} \leq \sigma_T a_i. \tag{3.8}$$

By introducing slack variables for the structure and substructures, the optimization has the potential to be converted to a linear program. Let us define slack variable vectors $\mathbf{s_i^+}, \mathbf{s_i^-}$ that turn inequalities (3.8) to

$$f_{i_{red}} + 2\frac{\sigma_0}{\sigma_C} s_{i_{red}}^- = \sigma_T a_i, \tag{3.9a}$$

$$-f_{i_{red}} + 2\frac{\sigma_0}{\sigma_T} s_{i_{red}}^+ = \sigma_C a_i, \tag{3.9b}$$

where σ_0 the current stress of each element. Now there is a clear relationship between $s_{i_{red}}^-$ and s_i^- and is given by

$$s_{i_{red}}^- = s_i^- + \frac{\sigma_C}{2\sigma_0}(f_i - f_{i_{red}}) \tag{3.10a}$$

$$s_{i_{red}}^+ = s_i^+ + \frac{\sigma_C}{2\sigma_0}(f_{i_{red}} - f_i) \tag{3.10b}$$

Thus, with $\mathbf{L}$ the $m \times 1$ element lengths vector, the plastic layout linear program

accounting for specific redundancy constraints is

$$\begin{aligned} \min_{\mathbf{s}^+,\mathbf{s}^-} \quad & V = \mathbf{l}^\mathbf{T}\left(\frac{\mathbf{s}^+}{\sigma_T} + \frac{\mathbf{s}^-}{\sigma_C}\right) \\ \text{s.t.} \quad & \mathbf{A}^\mathbf{T}(\mathbf{s}^+ - \mathbf{s}^-) = \mathbf{F} \\ & \mathbf{R_i}\mathbf{A}^\mathbf{T}(\mathbf{s}^+_\mathbf{red} - \mathbf{s}^-_\mathbf{red}) = \mathbf{F} \\ & s_i^+, s_i^- \geq 0 \end{aligned} \tag{3.11}$$

It is worth noting that in this formulation, $\mathbf{s_{red}}$ cannot be one of the objective variables since they are a function of $\mathbf{s}^+, \mathbf{s}^-$ from (3.10).

The current issue, is that internal forces of the reduced structure $f_{i_{red}}$ and of the original structure f_i are related, and the optimization cannot account for the redundancy constraint unless the equation is expressed solely in terms of f. In other words, for a constraint to be accounted in an optimization, it has to be a direct function of the objective variables, something that is not evident in this case. Hence, in order to compute the optimization accurately, a closed form function is required that relates the internal forces of the original structure and those of the substructure with element i reduced.

3.2.1 Structure-Substructure Internal Force Relation

Below the analytical equation is provided that relates the internal forces of the original structure with $\mathbf{f_i}$, the forces of the reduced structure.

From equation (3.5), the internal force vector of the reduced structure is given by the equation

$$\mathbf{A}\mathbf{R_i}\mathbf{f_i} = \mathbf{F}.$$

At the same time, by equilibrium of the initial structure, $\mathbf{Af} = \mathbf{F}$. Combining the above equations,

$$\mathbf{Af} = \mathbf{AR_if_i} = \mathbf{F} \Leftrightarrow \mathbf{A}(\mathbf{f} - \mathbf{R_if_i}) = 0 \Leftrightarrow \mathbf{f} - \mathbf{R_if_i} \in \ker A \tag{3.12}$$

In other words, $\mathbf{f} - \mathbf{R_if_i}$ has to be within the kernel of A. With this information

in mind, we can now proceed to find the relationship between $\mathbf{f_i}$ and $\mathbf{f}$. Before proceeding, the following useful lemma regarding linear independence of full rank matrices is presented:

Lemma 1. Let $c_1, c_2, .., c_m$, the columns of the force equilibrium matrix $\mathbf{A}$. For the original truss system to be in equilibrium, $n \times m$ matrix A^2 should have rank $n \leq m$. If $n = m$ then the structure is determinate and any substructure will directly be unstable. For $n < m$, by definition of nullity and rank, there exists $\vec{v} = \begin{pmatrix} \kappa_1 \\ \vdots \\ \kappa_m \end{pmatrix}$, such that

$$\mathbf{A}\vec{v} = 0 \Leftrightarrow \kappa_1 c_1 + \cdots + \kappa_m c_m = 0. \tag{3.13}$$

The proof of this lemma is trivial and can be found in any introductory Linear Algebra Textbook. The author recommends G. Strang's Linear Algebra textbook (Strang, 2006). With the above Lemma in mind, we can expand the internal force vectors and find their relative relationship. Let $\mathbf{f} = \begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix}$ and $\mathbf{f_i} = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}$ the original and substructure internal force vectors respectively. The expression $\mathbf{A}(\mathbf{f} - \mathbf{R_i f_i}) = 0$ is then written as

$$c_1(a_1 - x_1) + c_2(a_2 - x_2) + \cdots + c_i a_i + \cdots + c_m(a_m - x_m) = 0. \tag{3.14}$$

If $a_i = 0$, then $\mathbf{f} = \mathbf{f}_i$ and the solution is trivial. For $a_i \neq 0$, multiplying (3.14) by $\frac{\kappa_i}{a_i}$ gives

$$\frac{\kappa_i}{a_i}(a_1 - x_1)c_1 + \cdots + \kappa_i c_i + \cdots + \frac{\kappa_i}{a_i}(a_m - x_m)c_m = 0. \tag{3.15}$$

From equations (3.13),(3.15) there $\exists$ some κ_j $(i \neq j)$, such that

$$\frac{\kappa_i}{a_i}(a_j - x_j) = \kappa_j. \tag{3.16}$$

[2]For the sake of simplicity. the fixities and two-dimensional simplification of the nodes are neglected, as we assume enough fixities are provided to guarantee stability.

Solving (3.16) in terms of x_j gives

$$x_j = a_j - \frac{a_i}{\kappa_i}\kappa_j, \tag{3.17}$$

and since this holds $\forall\ j$, the relationship for $\mathbf{f_i}$ is given by

$$\mathbf{f_i} = \mathbf{f} - \frac{a_i}{\kappa_i}\vec{v} = \mathbf{f} - \frac{\vec{e_i}^T\mathbf{f}}{\vec{e_i}^T\vec{v}}\vec{v} = \left(\mathbf{I}_m - \frac{\vec{v}\vec{e_i}^T}{\vec{e_i}^T\vec{v}}\right)\cdot\mathbf{f}. \tag{3.18}$$

We just showed that there is a linear relationship between the internal forces of any redundant structure and any stable substructure, no matter the material properties of the system. This novel equation can be restated in a theorem form as follows:

Theorem 1. Let a truss network Ξ with n nodes and m branches, in equilibrium with the external $n \times 1$ force vector $\mathbf{F}$ and the respective internal $m \times 1$ force vector $\mathbf{f}$. If removing any member i from the system under the identical loading conditions and the truss system remains in equilibrium under internal force vector $\mathbf{f_i}$, then the relationship between the internal forces of the reduced structure and the actual structure, is given by:

$$\mathbf{f_i} = \left(\mathbf{I}_m - \frac{\vec{v}\vec{e_i}^T}{\vec{e_i}^T\vec{v}}\right)\cdot\mathbf{f}, \tag{3.19}$$

where $\mathbf{I}_m$ the $m \times m$ identity matrix, $\vec{e_i}$ the i^{th} $m \times 1$ basis vector, and $\vec{v}$ any nonzero basis of $\ker A$.

It is worth restating the assumptions that this theorem holds:

- First and foremost, the linear relationship holds if and only if the substructure is stable
- The theorem corresponds to internal forces of truss systems only i.e. no self-weight of the bar members is considered and loads are strictly applied on the nodes
- The above theorem does **not** require any plasticity assumptions – it is also material invariant

For level 1 redundant structure, the above theorem showed that the internal forces of the original structure and any substructure are linearly related. Hence, all equilibrium constraints can be expressed as a linear function of the known internal force vector variable**f**. This allows us to keep considering the optimization as a linear program. Now that the relationship between the internal forces of the substructure and structure has been defined, the refined redundancy optimization constraint can be expressed as a function of the known variable **f** only.

3.2.2 Refined Redundancy Optimization

Since the redundancy constraint can now be expressed in terms of **f** only, and it has to hold for each element i, the final constraint is the set of all substructure equilibrium equations for all i and translates to

$$\begin{pmatrix} \mathbf{AR}_i(\mathbf{I_m} - \frac{\vec{v}\vec{e_1}^T}{\vec{e_1}^T\vec{v}}) \\ \vdots \\ \mathbf{AR}_i(\mathbf{I_m} - \frac{\vec{v}\vec{e_m}^T}{\vec{e_m}^T\vec{v}}) \end{pmatrix} \mathbf{f} = \begin{pmatrix} \mathbf{F} \\ \vdots \\ \mathbf{F} \end{pmatrix}. \tag{3.20}$$

So, the new plastic layout optimization formulation takes the form:

$$
\begin{aligned}
&\min_{\mathbf{a}} \quad V = \mathbf{L^T a} \\
&\text{s.t.} \quad \begin{pmatrix} \mathbf{AR}_i(\mathbf{I_m} - \frac{\vec{v}\vec{e}_1^T}{\vec{e}_1^T\vec{v}}) \\ \vdots \\ \mathbf{AR}_i(\mathbf{I_m} - \frac{\vec{v}\vec{e}_m^T}{\vec{e}_m^T\vec{v}}) \end{pmatrix} \mathbf{f} = \begin{pmatrix} \mathbf{F} \\ \vdots \\ \mathbf{F} \end{pmatrix} \\
&\qquad\qquad -\sigma_C a_i \leq f_i \leq \sigma_T a_i, \ \forall i = 1, 2...N_e,
\end{aligned} \tag{3.21}
$$

where **a** the cross-sectional area vector. Clearly this is an overdetermined optimization problem, as there are m unknowns and $m \times n$ equations. To prove the existence of solutions, we need to show that at most m equations are linearly independent, and hence the system remains indeterminate.

Linear Independence of Constraints

To tackle the issue of an over-determined system, we need to show some form of linear dependence among the constraints. The proposed approach is a typical application of the matrix rank theorem. In general, for any matrix $\mathbf{P}$ with x rows and y columns:

$$\text{row rank}(\mathbf{P}) \leq x, \tag{3.22a}$$

$$\text{column rank}(\mathbf{P}) \leq y, \tag{3.22b}$$

$$\text{row rank}(\mathbf{P}) = \text{column rank}(\mathbf{P}). \tag{3.22c}$$

By definition of matrix $\mathbf{A}$, $m \geq n$. The $n \times m$ matrix $\mathbf{AR}_i(\mathbf{I_m} - \frac{\vec{v}\vec{e}_i^T}{\vec{e}_i^T\vec{v}})$ has rank at most n. By stacking all n matrices, the $mn \times m$ matrix (3.20) is constructed. This matrix can have at most rank m by the rank theorem. Since each of its elements $\mathbf{AR}_i(\mathbf{I_m} - \frac{\vec{v}\vec{e}_i^T}{\vec{e}_i^T\vec{v}})$ have ranks at most n and this holds for all the rows of matrix (3.20), the whole matrix can have rank at most n. Thus the system of equations remains

indeterminate.

In other words, we showed that at most n equations of the constraints can be linearly independent. Thus, the system has a unique[3] solution. This is an essential result towards the formulation of the optimization and we restate it below as a theorem:

Theorem 2. Let m matrix equations of the form $\mathbf{AR}_i(\mathbf{I_m} - \frac{\vec{v}\vec{e}_i^T}{\vec{e}_i^T\vec{v}})\mathbf{f} = \mathbf{F}$. Then at most m equations are linearly independent.

For simplicity on the notation, the linearly independent elements of matrix (3.20) will from now on be referred as as

$$\text{linearly independent rows of} \begin{pmatrix} \mathbf{AR}_i(\mathbf{I_m} - \frac{\vec{v}\vec{e}_1^T}{\vec{e}_1^T\vec{v}}) \\ \vdots \\ \mathbf{AR}_i(\mathbf{I_m} - \frac{\vec{v}\vec{e}_m^T}{\vec{e}_m^T\vec{v}}) \end{pmatrix} = \perp\!\!\!\perp \mathbf{AR}_i(\mathbf{I_m} - \frac{\vec{v}\vec{e}_i^T}{\vec{e}_i^T\vec{v}}), \qquad (3.23)$$

where the operator $\perp\!\!\!\perp$ symbolizes the linear independent rows of the matrix.

With the constraints of the 1^{st} order redundancy optimization finalized, the next step to convert it to a linear program is to add non-negativity constraints to the variables.

Nonnegativity of Objective Variables

Besides the linearity of the objective function and the constraints, in a linear program non-negativity of the objective variables is required. To achieve that, the *trick* lies in the binary nature of stressed trusses. A member is either in tension or compression. Hence the nonnegative slack tension/compression variables are introduced (Zegard et al., 2014). Let $\mathbf{f}^+$ the nonzero $m \times 1$ vector of internal forces that are purely in tension. That is for any element that is in tension $\mathbf{f}^+$ has their value while it has 0 for any non-stressed or compressed elements. Similarly let $\mathbf{f}^-$ the nonzero

[3]Uniqueness of the optimal solution will be shown later.

$m \times 1$ vector that includes all internal forces in compression, while having zero entries for tensed or non-stressed elements. Hence $\mathbf{f} = \mathbf{f}^+ - \mathbf{f}^-$. Replacing the internal force vector $\mathbf{f}$ with the introduced nonzero slack objective variables, all mathematical requirements for a linear program are satisfied. Nevertheless, mathematical accuracy does not imply correctness of the solution. In the next subsection, we identify the correct objective function to achieve the optimal solution of the redundancy problem.

3.2.3 Optimal Objective Function

In traditional plasticity layout optimization, the optimization is performed with respect to the area, i.e. the algorithm minimizes the cross-sectional area of elements until the constraints are all active. In other words, plasticity optimization requires all elements to be fully stressed (active constraints). This is the case, because if a member was not fully stressed, more material could be removed till it becomes fully stressed. This is a typical requirement for pure volumetric optimization which does not hold in the case of redundancy. The area vector is expressed as

$$\mathbf{a} = (\sigma_{tension}\ \sigma_{compression}) \begin{pmatrix} \mathbf{f}^+ \\ \mathbf{f}^- \end{pmatrix} = \sigma_{tension}\mathbf{f}^+ + \sigma_{compression}\mathbf{f}^-. \tag{3.24}$$

Without loss of generality let $\sigma_{tension} = \sigma_{compression} = 1$. Then $\mathbf{a} = \mathbf{f}$[4]. Hence, in the plastic layout optimization, minimizing for area is equivalent to optimizing for the internal forces. However, in the case of redundancy optimization, this would be a faulty assumption. By definition of redundancy, removing one element results in a substructure that yet has the capacity to maintain the original applied load. If the original structure had been fully stressed, then clearly the substructure would not be able to accommodate for the original load. This means that the fully stressed assumption for all elements of $\mathbf{a}$ cannot be used in any redundancy accounting optimization. What is more, from the previous section, it is clear that keeping the objective function as the load path $\vec{l}^T \cdot \mathbf{f}$ optimizes only the initial force structure i.e. finding non-optimal

[4]This equality is clearly in magnitude and not units.

solutions as soon as an element of the primary system is removed. The correct objective function should include the vector that optimizes for the load path created by the largest possible forces felt on each element. Below, we show mathematically this result.

Define the $m \times m$ matrix that contains all substructure internal force vectors

$$P = \begin{pmatrix} | & | & | & | & | \\ \mathbf{f}_1 & ... & \mathbf{f}_i & ... & \mathbf{f}_m \\ | & | & | & | & | \end{pmatrix} \tag{3.25}$$

From (3.25) we need to create a vector out of the absolute maximum value of each row of the matrix. In addition, we would require to keep $\mathbf{f}$ as isolated from the maximum function as possible, in order to maintain the linearity of this problem. Mathematically, the desired vector $\mathbf{S}$ is given by

$$\mathbf{S} = \max_j(P_{ij}) \ \forall i \in 1, ..., m. \tag{3.26}$$

Deriving $\mathbf{S}$ is not straightforward since $\mathbf{f}$ is deeply embedded in P. Note that (3.25) is also equivalent to

$$P = \begin{pmatrix} | & | & | & | & | \\ K_1\mathbf{f} & ... & K_i\mathbf{f} & ... & K_m\mathbf{f} \\ | & | & | & | & | \end{pmatrix} = \begin{pmatrix} \vec{e_1}^T K_1\mathbf{f} & \cdots & \vec{e_1}^T K_i\mathbf{f} & \cdots & \vec{e_1}^T K_m\mathbf{f} \\ \vec{e_2}^T K_1\mathbf{f} & \cdots & \vec{e_2}^T K_i\mathbf{f} & \cdots & \vec{e_2}^T K_m\mathbf{f} \\ \vdots & \cdots & \ddots & \cdots & \vdots \\ \vec{e_m}^T K_1\mathbf{f} & \cdots & \vec{e_m}^T K_i\mathbf{f} & \cdots & \vec{e_m}^T K_m\mathbf{f} \end{pmatrix}, \tag{3.27}$$

where $\vec{e_i}$ the i^{th} basis and K_i such that $\mathbf{f}_i = K_i\mathbf{f}$. This $m \times m$ matrix was derived in section 3.2.1 to be

$$K_i = I_m - \frac{\vec{v} \cdot \vec{e_i}^T}{\vec{e_i}^T \cdot \vec{v}}. \tag{3.28}$$

Combining (3.26) with (3.27), each row j of vector $\mathbf{S}$ equals

$$\mathbf{S}_j = \max_i \vec{e_j}^T K_i\mathbf{f} \tag{3.29}$$

Placing (3.28) in (3.29) we get

$$\vec{e}_j^T K_i \mathbf{f} = \vec{e}_j^T (I_m - \frac{\vec{v} \cdot \vec{e}_i^T}{\vec{e}_i^T \cdot \vec{v}})\mathbf{f} = \vec{e}_j^T \mathbf{f} - \frac{\vec{e}_j^T \vec{v}}{\vec{e}_i^T \vec{v}} \vec{e}_i^T \mathbf{f} \tag{3.30}$$

So for $j = 1$ we have

$$\mathbf{S}_1 = \max_i (\vec{e}_1^T \mathbf{f} - \frac{v_1}{v_i} \vec{e}_i^T \mathbf{f}) = f_1 - v_1 \max_i \frac{f_i}{v_i} = f_1 + v_1 \min_i (-\frac{f_i}{v_i}). \tag{3.31}$$

The value $\frac{f_i}{v_i}$ is a ratio of the internal force of element i over a constant. Minimizing the negative of this value is equivalent to say that we require to find the most critical element in the truss system (with the largest. internal force). Note that since $\vec{e}_1^T \mathbf{f}$ is invariant of i we can take it out of the maximum being treated as a constant. Performing this for each row, we get the desired vector $\mathbf{S}$ to be

$$\mathbf{S} = \begin{pmatrix} f_1 + v_1 \min_i(-\frac{f_i}{v_i}) \\ f_2 + v_2 \min_i(-\frac{f_i}{v_i}) \\ \vdots \\ f_m + v_m \min_i(-\frac{f_i}{v_i}) \end{pmatrix} = \mathbf{f} + \min_i (-\frac{f_i}{v_i}) \vec{v}. \tag{3.32}$$

With $\mathbf{S}$ determined, the updated load path objective function then takes the form

$$\begin{aligned} \min_{\mathbf{f}} g(\mathbf{f}) = \min_{\mathbf{f}} \mathbf{L}^T \cdot \mathbf{S} = \min_{\mathbf{f}} \mathbf{L}^T \cdot (\mathbf{f} + \min_i (-\frac{f_i}{v_i}) \vec{v}) = \\ = \min_{\mathbf{f}} (\mathbf{L}^T \mathbf{f} + \min_{i \in [1,\ldots m]} (-\frac{f_i}{v_i}) \mathbf{L}^T \vec{v}) = \min_{\mathbf{f},i} (\mathbf{L}^T \mathbf{f} - \frac{f_i}{v_i} \mathbf{L}^T \vec{v}) = \\ = \min_{\mathbf{f},i} \mathbf{L}^T (I_m - \frac{\vec{v}^T \vec{e}_i^T}{\vec{e}_i^T \vec{v}}) \mathbf{f} = \min_{\mathbf{f},i} \mathbf{L}^T K_i \mathbf{f} = \min_{\mathbf{f},i} \mathbf{L}^T \mathbf{f}_i. \end{aligned} \tag{3.33}$$

The result is very intriguing, as we just proved that instead of taking the maximum value of each row of matrix P, by the nature of the matrices, we only need to identify one column vector $\mathbf{f}_i$ for which $(-\frac{f_i}{v_i})$ is minimized. This means that minimizing the load path of a redundant structure is equivalent to minimizing the load path of the most crucial substructure, that is acquired by removing the highest stressed element of the original structure. To restate this important result as a theorem:

Theorem 3. The objective function of the problem is equivalent to

$$\min g(\mathbf{f}) = \min_{\mathbf{f},i} \mathbf{L}^T \mathbf{f_i}, \tag{3.34}$$

where $\mathbf{f}_i$ the internal force vector of the structure, when element i is removed.

3.2.4 Level 1 Redundancy Linear Formulation

A final adjustment that can be performed to maintain the optimization linear and simpler, is to place the two slack variables $\mathbf{f}^+, \mathbf{f}^-$ in one $2m \times 1$ variable vector $\vec{f} = \binom{\mathbf{f}^+}{\mathbf{f}^-}$. Then the objective function is converted to

$$\mathbf{L}^T \mathbf{f_i} = (\mathbf{L}^T \ \mathbf{L}^T) \begin{pmatrix} \mathbf{K_i} & 0 \\ 0 & \mathbf{K_i} \end{pmatrix} \vec{f}, \tag{3.35}$$

where $\mathbf{K_i}$ the matrix coefficient such that $\mathbf{f}_i = \mathbf{K_i f}$. Likewise, the constraints become

$$\perp \mathbf{AR_i}(\mathbf{I_m} - \frac{\vec{v}\vec{e}_i^T}{\vec{e}_i^T \vec{v}})\mathbf{f} = \begin{pmatrix} \mathbf{AR_iK_i} & -\mathbf{AR_iK_i} \end{pmatrix} \vec{f} = \mathbf{F}. \tag{3.36}$$

Hence, the level 1 redundancy linear program has the final form:

$$\boxed{\begin{array}{lll} \min\limits_{\vec{f},i} & \left|\mathbf{L}^T\mathbf{K_i} \quad \mathbf{L}^T\mathbf{K_i}\right|\vec{f} & \\ \text{s.t.} & \perp \left(\mathbf{AR_iK_i} \quad -\mathbf{AR_iK_i}\right)\vec{f} & = \mathbf{F} \\ & \vec{f} \geq 0 \ \forall \ i \in [1, 2..m] & \end{array}} \tag{3.37}$$

This is a typical linear optimization problem with equality constraints, and can be directly solved with either simplex method or an interior point algorithm.
To summarize, in this section, we proved the following key findings. First, we showed that the internal forces of a "damaged"/reduced substructure are related with those of the original one. In addition we showed that to optimize the volume of the redundant structure, it is sufficient to find the least load path of the most critical substructure. Combining these theorems, we were able to formulate the volumetric redundancy

optimization as a linear program.

3.3 Computational Implementation

As mentioned in the previous chapter, the foundations of the computational solutions were drawn from the GRAND framework. Specifically the mesh generation and the geometry matrix construction use Zegard's GRAND code in Matlab (Zegard et al., 2014). The main motivation behind this choice is that Zegard's ground structure generation development accounts for co-linear trusses (up to a user-defined sensitivity) and also allows the control of the level of connectivity i.e. how many neighbors of neighbors are connected to each other. Hence, Zegard's and Paulino's impactful work prove to be an ideal foundation for the developed algorithms. Following the Ground Structure development, the algorithm needs to complete several computationally heavy processes in order to derive the optimal redundant solution, such as calculating the nullspace of the connectivity matrix, storing the set of $\mathbf{K_i}$'s, and iterating over the redundant elements to find the optimal ones.

In this section, a few key parts of the code are provided, as well as an expository of the optimization algorithm and the visualization of the results.

3.3.1 Preliminaries

GRAND (Zegard et al., 2014) can input any collection of nodes, connectivity levels and geometries, and constructs the geometry matrix $\mathbf{A}$, calculates the $m \times 1$ elements length vector $\mathbf{L}$ as well as the external force vector $\mathbf{F}$. Then through linear programming, it minimizes the load path $\mathbf{L}^T\vec{f}$, under the equilibrium constraint $\mathbf{A}\vec{f} = \mathbf{F}$. To solve the optimal problem linearly, Zegard implements the trick of doubling the size of $\vec{f}$ under virtual tension and compression elements with non negative entries. All the following described calculations were performed in Matlab. The redundancy optimization algorithm requires to calculate a nonzero elements vector $\vec{v}$ that belongs to the nullspace of A. Since A needs to have full rank to be solvable, the nullspace will be nonzero and will hence always have basis with nonzero elements. To guarantee

that the used vector $\vec{v}$ has no zero entries, the sum of all basis vectors of the nullspace is taken, as the magnitude of the vector's entries are invariant of the result.

With $\vec{v}$ defined, all $\mathbf{K_i}$ matrices can be calculated. After the coefficients of the equations of $\mathbf{AR_iK_i}\vec{f} = F$ have been stored, only the linearly independent ones need to be kept. To do so, a custom function removes all independent rows of these matrices while saving the indices of the remaining rows to find the corresponding entries of $\mathbf{F}$. The resulting coefficients and respective external force entries are then used in the linear optimization.

With the above preliminary calculations completed, the linear optimization procedure can be initiated.

3.3.2 Linear Programming

After the preliminary calculation, the redundancy optimization can take place. In subsection 3.2.3 it was shown that the redundancy objective function is equivalent to minimizing the load paths of the most crucial substructure. However, the most crucial substructure is not known and the only element removals that have direct impact on the optimum are the ones that exist in the volumetric optimal. Hence, the algorithmic implementation of redundancy optimization is as follows:

- Find the volumetric optimal through traditional topology optimization and save the indices of the nonzero force elements
- Loop through these elements and run the optimization of their reduced substructures with the total force vector as the objective variable – Matlab's *linprog* command was used for this step
- Save the maximum force of each entry as the optimal force and update the optimal force vector

When the loop terminates, the optimal area is calculated as described in the next section. The complete code with the loop of optimizations can be found in Appendix B.

3.3.3 Drawing the solution

Since the fully stressed assumption does not hold for the redundancy optimization, after finding each $\vec{f_i}$ we cannot simply say $\mathbf{A} = \sigma \vec{f}$. Instead, the way the thickness is calculated, is by assuming that the maximum thickness of each element of $\vec{f_i}$ is acquired when that element is fully stressed. That way we can calculate $\mathbf{A}$ = maximum of each row of$\vec{f_i}$, $\forall i$. After $\mathbf{A}$ is calculated it can be drawn.

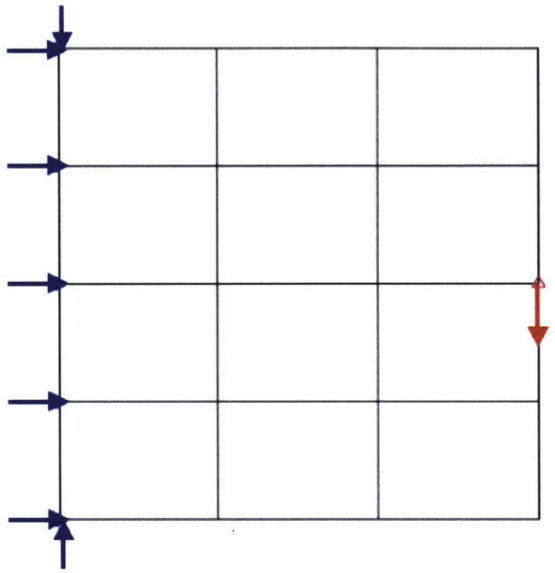

Figure 3-2: A 4×3 orthogonal cantilever type ground structure with five candidate fixities and one point load on the right

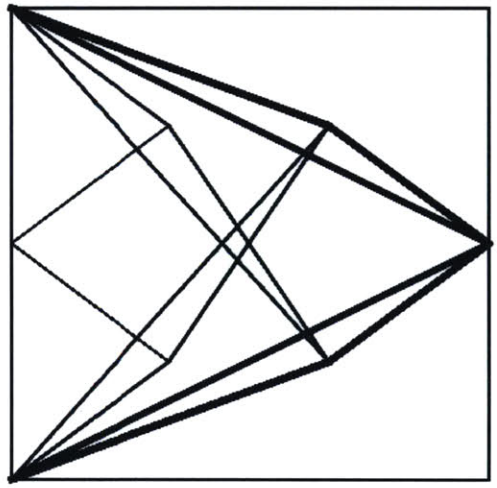

Figure 3-3: Optimal level 1 redundant least volume combination – as expected it resembles a Michell truss

Figure 3-2 shows an initial user constructed 3×4 box grid with fixities on the left side and a point load on the right. The initial ground structure connects all intersection points of this grid with all other points up to a level 6 connectivity. Figure 3-3 is the level 1 redundant optimal volume structure. The thickness of each

line represents the element's cross-sectional area. Given a length of 10 units on each side of the ground structure and a unit load applied, the least volume cantilever for the particular grid is 43 cubic units, which is below the double of the volumetric optimal for the same grid (25 cu). More visualizations and comparison of the volume savings of the redundant structures are provided in Chapter 5.

3.4 Discussion

This section provides brief qualitative explanations and extensions on certain topics that were previously described mainly from a mathematical viewpoint. In addition, alternative optimization approaches such as integer optimization are encountered along discussions on their performance and potential recommendations to make the implemented algorithms more efficient.

3.4.1 A Closer Look in the Structure–Substructure Relation

One of the most crucial relations derived in this thesis is the closed form linear relationship between the internal forces of a redundant structure, and the forces of any of its substructures. First of all, it is not intuitive that there is even a relationship between the internal forces of the two structures given all the potential redistributions of forces, and secondly it is remarkable that for one element the relationship is in fact linear. This relationship holds if and only if the substructure we are dealing with is stable (not just in equilibrium). In addition, we are assuming the truss system receives loads only on its nodes and the elements have no self-weight. The structure-substructure relationship in invariant of the material's system and it holds for both plastic and elastic structures.

On the next Chapter a proof for the i level redundancy structure substructure relationship is provided, where it shows that the degree of the relationship is equal to the level of the desired redundancy. To recite relationship (4.3) here, if $\mathbf{f}$ the internal force vector of the original network and $\mathbf{f_i}$ the internal force vector of the reduced

network (clearly the i^{th} entry of $\mathbf{f_i}$ is zero), then the relationship is given by

$$\mathbf{f_i} = \left(\mathbf{I}_m - \frac{\vec{v}\vec{e}_i^T}{\vec{e}_i^T\vec{v}}\right) \cdot \mathbf{f}. \tag{3.38}$$

It is mentioned that any arbitrary $\vec{v}$ can be used in this relationship, meaning any nonzero element vector of the nullspace. The reason we demand nonzero elements is because we impose the restriction of reducing any element i and hence divide by the i^{th} entry of $\vec{v}$. The careful reader might notice that if a nonzero $\vec{v}$ is used, then any $\alpha\vec{v}$ could also be used, for any $\alpha \neq 0$. This is because if $\mathbf{A}\vec{v} = 0 \Leftrightarrow \mathbf{A}(\alpha\vec{v}) = 0$. It is straightforward to show that for any $\vec{v_1} = \alpha\vec{v}$ for arbitrary nonzero α, the coefficient $\mathbf{K_i}$ remains unique. Mathematically,

$$\mathbf{K_i} = \mathbf{I}_m - \frac{\vec{v_1}\vec{e}_i^T}{\vec{e}_i^T\vec{v_1}} = \mathbf{I}_m - \frac{\alpha\vec{v}\vec{e}_i^T}{\alpha\vec{e}_i^T\vec{v}} = \mathbf{I}_m - \frac{\vec{v}\vec{e}_i^T}{\vec{e}_i^T\vec{v}}. \tag{3.39}$$

Hence the choice of $\vec{v}$ is invariant of the result. Yet, a more general question that can rise, is *how does the coefficient* $\mathbf{K_i}$ *particularly look*?

A matrix look on $\mathbf{K_i}$

Here we show $\mathbf{K_i}$ in matrix form, in an attempt to better demonstrate how the forces of structure and substructure are related.

Let the original $m \times 1$ internal force vector $\mathbf{f} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}$ and any nonzero vector $\vec{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{pmatrix}$ of the nullspace of $n \times m$ geometry matrix $\mathbf{A}$. Then the linear coefficient

$\mathbf{K_i}$ for which $\mathbf{f_i} = \mathbf{K_i f}$ can be written

$$\mathbf{K_i} = \mathbf{I_m} - \frac{1}{v_i}\begin{pmatrix} 0\cdots v_1 \cdots 0 \\ 0\cdots v_2 \cdots 0 \\ \vdots\cdots\vdots\cdots\vdots \\ 0\cdots v_i \cdots 0 \\ \vdots\cdots\vdots\cdots\vdots \\ 0\cdots v_m \cdots 0 \end{pmatrix} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix} - \begin{pmatrix} 0\cdots \frac{v_1}{v_i} \cdots 0 \\ 0\cdots \frac{v_2}{v_i} \cdots 0 \\ \vdots\cdots\vdots\cdots\vdots \\ 0\cdots 1 \cdots 0 \\ \vdots\cdots\vdots\cdots\vdots \\ 0\cdots \frac{v_m}{v_i} \cdots 0 \end{pmatrix} \tag{3.40}$$

and putting this together gives the form of the linear coefficient to be

$$\mathbf{K_i} = \begin{pmatrix} 1 & 0 & \cdots & -\frac{v_1}{v_i} & \cdots & 0 & \cdots \\ & 1 & \cdots & -\frac{v_2}{v_i} & \cdots & 0 & \vdots \\ & & \ddots & \vdots & & & \\ & & & 0 & & & \\ & & & -\frac{v_{i+1}}{v_i} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & -\frac{v_m}{v_i} & \cdots & & 1 \end{pmatrix}. \tag{3.41}$$

So the internal force f_{i_p} of the p^{th} element when element i is removed from the structure is given by equation

$$f_{i_p} = f_p - \frac{v_p}{v_i} f_i, \tag{3.42}$$

where f_p the internal force of the p^{th} element of the original structure. Hence, based on what signs $\frac{v_p}{v_i}$ and f_i are, the reduced structure internal force change.

3.4.2 Expository on the Objective Function

The selection of the appropriate objective function on the linear program is of critical importance. There are two major flaws why the traditional load path function $\mathbf{L}^T\vec{f}$ would have not worked.

- The load path function is appropriate for volume minimization only under the

fully stressed assumption

- The initial load $\vec{f}$ would optimize only the part of the structure that is fully stressed without any removals

The second point is key to the solution. If the optimization would attempt to minimize the internal forces of the original structure, then this would not be too different from traditional volumetric optimization. This is the case because the initial structure has several layers of determinate structures, and hence only one of them might be stressed. Hence, to achieve a total minimization of the worst case load paths, the appropriate objective function as mentioned in Section 3.2.3 is $\min_{\mathbf{f},i} \mathbf{L}^T\mathbf{f_i}$. This result shows that optimizing for the worst case load path is equivalent to optimizing for the maximum potential force of each element under all substructures. This result is crucial in the progress of the solution, as it manages to linearly extract the unknown variable vector $\vec{f}$ from the maximum potential load paths of the substructures. In simple words, this allows us to run a traditional topology optimization and identify the critical elements. Removing these elements one by one and maintaining the largest forces for each element for every optimization outputs the desired vector we are minimizing for. The proof in 3.2.3 significantly simplifies the optimization process.

3.4.3 Uniqueness of Solution

Problem 3.37 can be directly solved using either simplex or floating point algorithms. Am intriguing next question one can ask, is whether the solution is unique and under what conditions. This is a broad Linear Programming concern and the answer is highly dependent on the ground structure's topology, the $\mathbf{A}$ geometry matrix and the overall symmetry of the structure. There are several algorithms that can be applied in a case by case basis to test uniqueness, and the author recommends relevant literature (Appa, 2002), (Rozvany, 2011). The simplest check that can be applied on specific examples, follows the steps below:

- Perform the optimization and acquire the optimal load path, say V_{min}.

- Add the equation $\left|\mathbf{L}^T\mathbf{K_i} \quad -\mathbf{L}^T\mathbf{K_i}\right|\vec{f} = V_{min}$ as a new constraint and solve simultaneously the problems $\min_{\vec{f}} \vec{e}_i^T\vec{f}$ and $\max_{\vec{f}} \vec{e}_i^T\vec{f}$ for all elements i. The additional constraint implies that we are searching a different optimal solution (same function value). But the new function to maximize/minimize, namely f_i, means that we are looking for an optimal solution with maximal/minimal f_i value.

- Check whether the optimal minimum and maximum f are equal to each other $\forall\, i$. If that is the case, then the solution is unique and there is no other configuration that is optimal. Alternatively, there can be other optimal configurations – material distributions.

The issue of multiple solutions/configurations in topology optimization has been a fairly unexplored field. Kutylowski has shown that for certain families of normal and uniform grid ground structures, the density topology optimization can generate non-unique optimal values that in general belong to the same shape of families and with similar densities (Kutylowski, 2002). However, little is known for nonuniform grids (not to mention additional constraints such as redundancy) in this promising niche field of research. Rozvany provides a set of conjectures regarding the uniqueness of solutions based on the symmetry of a truss structure (Rozvany, 2011). More specifically, Rozvany provides two important conjectures regarding uniqueness and symmetry of optimal solutions.

Conjecture 1. In a topology optimization problem with feasible continuous solutions there exists either one or an infinite number of optimal designs with the same least volume.

Conjecture 2. For a symmetric topology optimization problem with feasible continuous solutions, at least one optimal design and the internal forces in it are symmetric.

Both of the above conjectures have been shown for several examples, however there has not been a general mathematical proof yet. We believe that uniqueness is a more frequent phenomenon in the redundancy topology optimization compared

to traditional topology optimization, given the additional limitations imposed in the structural freedom and the increased lack of symmetry. In any case, configuration uniqueness for both redundancy and traditional topology optimization need to be researched further from a theoretical topology group perspective, in order to provide more general statements.

3.4.4 Controlling the Number of Elements with Integer Programming

After experimentation with the optimization algorithm, one can easily notice that the more refined that initial mesh grid, the linear redundancy optimization attempts to reduce the criticality of all trusses by adding several thinner ones. Hence the number of elements and varying thicknesses dramatically increases after an increased enough density, setting the process impractical and uneconomical. Thus after a point, it might prove more insightful to either optimize for the number of elements used, or maintain a restriction on the potential options the area of each element can take. A partial solution in that issue can be given by integer programming.

Mixed Integer Linear Programming

Mixed Integer Linear Programs (MILP) allow some of the objective variables to be integer by relaxing the constraints. In MILP the user is able to restrict the solution to binary or multiple variables. This would set the actual manufacturing of a truss system much more practical, requiring only a set of three or four different cross sections. In the specific case of redundancy the optimization would take the following form:

$$\begin{aligned} \min_{\vec{f},i} \quad & \left|\mathbf{L}^T\mathbf{K_i} \quad \mathbf{L}^T\mathbf{K_i}\right|\vec{f} \\ \text{s.t.} \quad & \perp\!\!\!\perp \left(\mathbf{AR_iK_i} \quad -\mathbf{AR_iK_i}\right)\vec{f} \;\leq\; \mathbf{F} \\ & \vec{f} \geq 0 \;\forall\, i \in [1, 2..m],\; \vec{f} \in \mathbb{Z} \end{aligned} \tag{3.43}$$

The only drawback is that some elements might be overestimated and the design

load limit is relaxed as the above constraint shows. Nevertheless, given the desired applicability, refinement and allowable design space, the integer programming formulation could potentially be a more desirable solution. The way one could control the number of elements used in the optimization through linear programming, is by controlling the range of allowed solutions. For instance, if we have two allowable sets from which the objective variable can draw from, say $[0, 1, 2, 3, 4, ...]$ and $[0, 2, 4, 6, 8...]$, then the optimal solution from the second set would contain far less elements than the solution from the first set. This is because the distance between the cross-sectional area of accepted elements is larger and hence less elements can take multiple values. In general, increasing the distance between accepted element values decreases the number of elements in the MILP solutions (and at the same time it decreases the material efficiency of the shapes).

The computational implementation is more expensive than continuous linear optimization, given that the constraint is relaxed and the code has to iterate further in order to approximate the integer solution. To put the solutions in more perspective, Figure 3-4 shows the linear integer program the for a 4×3 box grid of the same dimensions as the optimal redundant structure in Figure 3-3. The topology of the structure is dramatically different and the structure only uses 5 different cross section for all of its elements.

The total volume of the structure is 55 cubic units, which is almost 30% more material than the continuous redundant optimal. Thus one can accommodate less variability of the used elements by sacrificing additional material. In any case, with a well formulated continuous linear program, integer programming is a direct and feasible extension of the problem, providing alternative solutions.

3.4.5 Algorithm Performance

The algorithm has not yet reached optimal performance yet and there are several steps that could potentially be added to make it more efficient and less computationally expensive. In principle, the Ground Structure approach is by default highly memory consuming. For instance saving the sparse geometry matrix of a 50×50

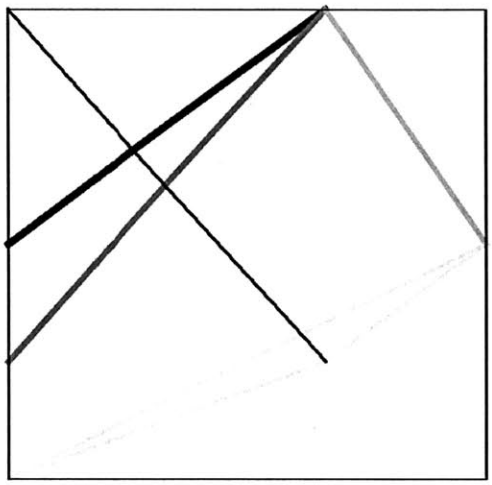

Figure 3-4: Integer optimal level 1 redundant structure with 7 elements of 5 different cross sections

orthogonal ground structure with Level 6 connectivity takes about 8.6GB of memory. Calculating the nullspace of such a matrix using the default Matlab *null()* function is very time consuming. Then m different K_i matrices that span from $\mathbf{A}$ need to be stored during the loop of optimization. In fact, the linear optimization process is the fastest computational part of the code. To accelarate the speed of the matrix, custom made functions for calculating the nullspace of sparse matrices have been implemented that save considerable time, however for larger than 10×10 orthogonal grids, the required memory exceeds standard PC RAM memory. In the future, we hope to implement more sophisticated storing techniques to make the code more efficient and user-friendly.

3.4.6 Chapter Summary

In this Chapter the level 1 redundancy optimization – i.e. the least volume structure, for which any element can be removed while maintaining equilibrium – was developed as a linear program. The general algorithm for any initial Ground Structure has been fully developed for any input geometry, including normal grids or nonlin-

ear highly customizable geometries. There are several advantages that make level 1 redundancy optimization distinct and much more efficient compared to higher lever redundancies. One major reason is the fact that the substructure internal forces are linearly related with the original structure's forces. In addition, the different set of elements that can be removed are at most m, while there are factorially more combinations in the case of higher redundancy. In the next section, we show how the nonlinear problems of higher redundancy are formulated and solved.

Chapter 4

Higher Order Redundancy Topology Optimization

Following the results and formulation of the redundancy optimization for 1 element, we can provide a generalization of this formulation for $i \geq 2$ elements. In other words, we can formulate the problem of finding the least volume redundant structure, for which *any* i number of elements can be removed, with the remaining structure still stable. To achieve this formulation, and proceed with computational results, the key is again to identify the relationship of the internal forces of any substructure with i number of elements removed, and the internal forces of the original full structure. As we prove in this section, the relationship between structure and substructure internal forces are not linear anymore for $i \geq 2$ elements, but instead they have a degree equal to the number of elements that are removed. First, we show this relationship, and then we formulate the general $i \geq 2$ level redundancy optimization. We show that such Equality Constraint Nonlinear Problems (ECNP) are directly solvable using Lagrange Multipliers. Finally the case of 2 elements is shown in closed form and taking advantage of symmetry, some insight in their solutions is provided.

4.1 General $i \geq 2$ level redundant substructure – structure relationship

In order to express the optimization problem in terms of the original internal force variables, a relationship between reduced structure and original structure internal forces needs to be established.

4.1.1 Maximum Number of Element Removals

First of all, a question of interest, is whether we can determine the number of elements that can be removed, given the number of nodes and connectivities. Since the reduction matrix, as defined in the previous Chapter, simply reduces some columns of the geometry matrix A to zero, the largest reduction of columns we can perform is up to the point where rank m = rank n. Thus, a broad upper limit for the set of elements that can be directly established is that at most rank $\mathbf{m}$ – rank $\mathbf{n}$ elements can be removed, in a structure for which it has full rank (stability requirement). This number is also known as Maxwell number(Maxwell, 1869), and it is directly related to the geometry matrix i.e the number of nodes and topology of the structure.

A major constraint in the problem formulation, is that we allow the removal of any i number of elements – *not* specific ones. Thus, neglecting symmetry, the number of potential substructures that need to be checked for a structure with m elements and i removals is

$$^{m}C_{i} = \frac{m!}{i!(m-i)!}. \tag{4.1}$$

For instance, the second order redundancy optimization would require $^{m}C_{2}$ stable substructures. Hence the number of substructures that have to be checked for stability increases geometrically. The relationship between substructure and structure internal forces also becomes more complex and nonlinear. The reduction matrix in the general

case is defined as

$$\mathbf{R}_{el_1,el_2..,el_i} = \begin{pmatrix} & & & i \text{ zeros on the diagonal} & & & \\ 1 & & & & & & \\ & 0 & & & & & \\ & & \ddots & & & & \\ & & & 0 & & & \\ & & & & \ddots & & \\ & & & & & 1 & \end{pmatrix}$$

There are $\theta = {}^{m}C_i$ such $m \times m$ matrices, and that creates θ different equations of the form

$$\mathbf{AR}_{\mathbf{el_1},\mathbf{el_2}..,\mathbf{el_i}}\mathbf{f}_{\mathbf{el_1},\mathbf{el_2}..,\mathbf{el_i}} = \mathbf{F} \quad \forall el_1, el_2.., el_i \in \theta \tag{4.2}$$

4.1.2 Substructure Expression

On this section, the generalization of theorem 1 from Section 3.2.1 relating the internal forces of a full structure and the forces of the $2 \leq i \leq (\text{rank } m - \text{rank } n)$ reduced substructure is provided. In particular, we show that the relationship between the original force vector $\mathbf{f}$ and the reduced structure force vector $\underbrace{\mathbf{f}_{el_1,el_2,...el_i}}_{i \text{ elements removed}}$, has the same degree as i, the number of removed elements.

Theorem 4. Let truss network Ξ with n nodes and m branches, in equilibrium with the external $n \times 1$ force vector $\mathbf{F}$ and the respective internal $m \times 1$ force vector $\mathbf{f}$. If by removing $i \geq 2$ $el_1, el_2.., el_i$ trusses from the system under the identical loading conditions, the truss system maintains in equilibrium under internal loading $\mathbf{f}_{el_1 el_2 ... el_i}$, then the relationship between the internal forces of the reduced structure and the actual structure, is given by:

$$\mathbf{f}_{\mathbf{el_1},\mathbf{el_2}..,\mathbf{el_i}} = \mathbf{f} - \prod_{r=1}^{i} \left(\frac{\vec{e}_{el_r}^T \mathbf{f}}{\vec{e}_{el_r}^T \vec{v}}\right) \cdot \vec{v}, \tag{4.3}$$

where $\vec{e}_{el_r}$ the el_r^{th} $m \times 1$ basis vector, and $\vec{v}$ any nonzero basis of $\ker A$.

Proof. The following procedure closely follows the derivation in Section 3.2.1. From Lemma 1, $\exists \vec{v} = \begin{pmatrix} \kappa_1 \\ \vdots \\ \kappa_m \end{pmatrix}$, such that $\mathbf{A}\vec{v} = 0 \Leftrightarrow \kappa_1 c_1 + \cdots + \kappa_m c_m = 0$, where c_i the columns of geometry matrix $\mathbf{A}$. Let $\mathbf{f} = \begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix}$ and $\mathbf{f_{el_1 el_2 \ldots el_i}} = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}$ the original and i^{th} order substructure internal force vectors respectively. The expression $\mathbf{A}(\mathbf{f} - \mathbf{R_{el_1,el_2,\ldots el_i}} \mathbf{f_{el_1,el_2,\ldots el_i}}) = 0$ is derived directly by Equation (4.2) and equilibrium of the original structure. This is equivalent to

$$c_1(a_1 - x_1) + c_2(a_2 - x_2) + \cdots + c_{el_r} a_{el_r} + \cdots + c_m(a_m - x_m) = 0 \tag{4.4}$$

If any of the $a_r = 0$ then the problem is equivalent to the reduced $i - 1^{th}$ order substructure problem Without loss of generality, we can assume no a_{el_r}'s are zero. For $a_{el_r} \neq 0 \; \forall r = 1, ..i$, multiplying (4.4) by $\prod_{r=1}^{i} \frac{\kappa_{el_r}}{a_{el_r}}$ gives

$$\prod_{r=1}^{i} \frac{\kappa_{el_r}}{a_{el_r}}(a_1 - x_1)c_1 + \cdots + \prod_{r=1..,p-1,p+1,..i} \frac{\kappa_{el_r}}{a_{el_r}} \kappa_{el_p} c_{el_p} + \cdots + \prod_{r=1}^{i} \frac{\kappa_r}{a_r}(a_m - x_m)c_m = 0 \tag{4.5}$$

From equations (3.13),(4.5) there $\exists$ some κ_j $(j \neq el_1, el_2, ...el_i)$, such that

$$\prod_{r=1}^{i} \frac{\kappa_r}{a_r}(a_j - x_j) = \kappa_j. \tag{4.6}$$

Solving (4.6) in terms of x_j gives

$$x_j = a_j - \prod_{r=1}^{i} \frac{a_{el_r}}{\kappa_{el_r}} \kappa_j, \tag{4.7}$$

and since this holds $\forall \; j$ (besides the removed elements) the relationship for the reduced

structure's internal forces is

$$\mathbf{f}_{\mathbf{el_1},\mathbf{el_2},\ldots\mathbf{el_i}} = \mathbf{f} - \prod_{r=1}^{i} \frac{a_{el_r}}{\kappa_{el_r}} \vec{v} = \mathbf{f} - \prod_{r=1}^{i} \left(\frac{\vec{e}_{el_r}^T \mathbf{f}}{\vec{e}_{el_r}^T \vec{v}} \vec{v}\right) \qquad \square \quad (4.8)$$

From Equation (4.8), one can see that the relationship between $\mathbf{f}_{\mathbf{el_1},\mathbf{el_2},\ldots\mathbf{el_i}}$ and $\mathbf{f}$ is an i^{th} degree vector polynomial.

4.2 Higher Order Formulation

With the higher order structure substructure relationship formulated, we can now proceed to formulate the optimization problem, which closely resembles the case of first degree removal but it is not linear. Note that the non-negativity trick used in the first order removal is not necessary, since the optimization is nonlinear in the first place. Hence there is no need for the virtual structure doubling method implemented in Subsection 3.2.2. Let $\mathcal{J}$ the set of the $\binom{m}{i}$ permutations of the ground structure's elements. The general i^{th} order optimization takes the form

$$\boxed{\begin{array}{ll} \min\limits_{\mathbf{f},i} & \left|\mathbf{L}^T\left(\mathbf{f} - \prod_{r=1}^{i} \left(\frac{\vec{e}_{el_r}^T \mathbf{f}}{\vec{e}_{el_r}^T \vec{v}} \vec{v}\right)\right)\right| \\ \text{s.t.} & \perp\!\!\!\perp \mathbf{AR}_{\mathbf{el_1},\ldots\mathbf{el_i}}\left(\mathbf{f} - \prod_{r=1}^{i} \left(\frac{\vec{e}_{el_r}^T \mathbf{f}}{\vec{e}_{el_r}^T \vec{v}} \vec{v}\right)\right) = \mathbf{F} \quad \forall\, i \in \mathcal{J} \end{array}} \quad (4.9)$$

Note that in this case, the $\perp\!\!\!\perp$ operator finds the linearly independent rows of all possible ${}^mC_i$ matrices of the form $R_{el_1,..el_i}$, which directly shows the magnitude of such a process.

4.2.1 Analytical Solution

Even though the preliminaries of setting up such an optimization problem are very time consuming (identifying linearly independent rows from ${}^mC_i$ $n \times m$ matrices, i^{th} degree polynomial), the particular nonlinear optimization problem falls under the family of *Nonlinear Equality Constrained Problems* (NECP), which if continuously

differentiable, can be directly solved using *Lagrange Multipliers*. More on NECPs and Lagrange multipliers can be found in any graduate nonlinear programming textbook. The author recommends Bertseka's Nonlinear Programming book (Bertsekas, 2005). The order of redundancy and consequently the order of the nonlinear optimization problem, is actually irrelevant to the solution method as long as $i \geq 2$. For this reason, the derivation is provided below for $i = 2$, and similarly it can be extended for any $i > 2$.

4.3 The case of 2 elements

The case of 2 elements is particularly special, because under symmetrical ground structures and loadings, it can be considered as a natural extension of the first order redundancy optimization, as seen later in the section. The relationship between the internal forces of the structure by removing two elements, say i, j, is given by (4.3):

$$\mathbf{f_{ij}} = \mathbf{f} - \frac{\mathbf{f}^T \vec{e}_i \vec{e}_j^T \mathbf{f}}{\vec{v}^T \vec{e}_i \vec{e}_j^T \vec{v}} \cdot \vec{v}. \tag{4.10}$$

The complete formulation for the $i = 2$ problem is provided by (4.9) and resolved below:

$$\begin{aligned} \min_{\mathbf{f},i,j} \quad & \left| \mathbf{L}^T \left(\mathbf{f} - \frac{\vec{e}_i^T \mathbf{f} \vec{e}_j^T \mathbf{f}}{\vec{e}_{e_i}^T \vec{v} \vec{e}_{e_j}^T \vec{v}} \vec{v} \right) \right| \\ \text{s.t.} \quad & \perp\!\!\!\perp \mathbf{AR_{el_1,..el_i}} \left(\mathbf{f} - \prod_{r=1}^{i} \left(\frac{\vec{e}_{el_r}^T \mathbf{f}}{\vec{e}_{el_r}^T \vec{v}} \vec{v} \right) \right) = \mathbf{F} \quad \forall\, i, j \in 1, ...m \end{aligned} \tag{4.11}$$

This remains an NECP, and it can be directly solved in close form with Lagrange Multipliers. Assuming that the objective function and the constraint are respectively

$$\text{objective function} = \mathbf{L}^T \mathbf{f_{ij}} \tag{4.12}$$

$$h(\mathbf{f}) = \mathbf{AR_{ij}} \mathbf{f_{ij}} - \mathbf{F} \quad \forall i, j \in 1, ..m, \tag{4.13}$$

the general form of Lagrange system for this problem would take the following form

$$\nabla_{\mathbf{f}}\mathcal{L}(\mathbf{f},\lambda) = \nabla_{\mathbf{f}}(\mathbf{L}^T) + \nabla\lambda^T(\mathbf{A}\mathbf{R}_{\mathbf{ij}}\mathbf{f}_{\mathbf{ij}} - \mathbf{F}) = 0 \tag{4.14a}$$

$$\nabla_{\lambda}\mathcal{L}(\mathbf{f},\lambda) = h(\mathbf{f}) = 0, \tag{4.14b}$$

where the $n \times 1$ vector λ is the unknown Lagrange coefficient. Since the mathematical solution involves matrix derivatives, some important matrix derivation theorems are required. Before proceeding with the extensive calculation of the Lagrange Method, it is worth looking into the derivative dimensions as a frame of reference. In general, the following theorem holds regarding matrix derivatives:

Theorem 5. Let continuous infinite differentiable mapping $f : \mathbb{R}^{m\times n} \longrightarrow \mathbb{R}^{p\times q}$. The dimension of the derivative map of f with respect to matrix x of size $(m \times n)$ is

$$\frac{df}{dx} : \mathbb{R}^{m\times n} \longrightarrow \mathbb{R}^{pq\times mn} \tag{4.15}$$

The proof of this theorem is omitted due to its irrelevance on a Civil Engineering graduate thesis. A detailed proof can be found in several graduate nonlinear algebra textbooks. A recommended one is by Kagiwada et al(Kagiwada et al., 1988).

The load path objective function has dimensions (1×1) (scalar value), while the internal force vector has dimensions $(m \times 1)$, so by theorem 5, the derivative of $\mathcal{L}$ with respect to $\mathbf{f}$ will have dimensions $(1 \times m)$. This is a good reference point throughout the calculations that are to follow.

What is more, the following matrix derivatives are useful

$$\frac{dg(\vec{x})}{d\vec{x}} = \frac{dg(y)}{dy} \cdot \frac{dy(\vec{x})}{d\vec{x}}. \tag{4.16a}$$

$$\frac{dA\vec{x}}{d\vec{x}} = A. \tag{4.16b}$$

$$\frac{d\vec{x}^T A\vec{x}}{d\vec{x}} = \vec{x}^T(A + A^T). \tag{4.16c}$$

Plugging equation (4.10) in $\mathbf{L}^T\mathbf{f}_{\mathbf{ij}}$ and in (4.13), we can solve the system of equa-

tions (4.14a) & (4.14b) in terms of $\mathbf{f}$

$$\nabla_{\mathbf{f}}\mathcal{L}(\mathbf{f},\lambda) = \nabla_{\mathbf{f}}\big[\mathbf{L}^T(\mathbf{f} - \frac{\mathbf{f}^T\vec{e}_i\vec{e}_j^T\mathbf{f}}{\vec{v}^T\vec{e}_i\vec{e}_j^T\vec{v}}\cdot\vec{v})\big] + \nabla_{\mathbf{f}}\big[\lambda\cdot\big(\mathbf{AR_{ij}}(\mathbf{f} - \frac{\mathbf{f}^T\vec{e}_i\vec{e}_j^T\mathbf{f}}{\vec{v}^T\vec{e}_i\vec{e}_j^T\vec{v}}\cdot\vec{v}) - \mathbf{F}\big)\big] =$$
$$\frac{d(\mathbf{L}^T\mathbf{f})}{d\mathbf{f}} - \frac{d\mathbf{f}^T\vec{e}_i\vec{e}_j\mathbf{f}}{d\mathbf{f}}\cdot\frac{\mathbf{L}^T\vec{v}}{\vec{v}^T\vec{e}_i\vec{e}_j^T\vec{f}} + \frac{d(\lambda^T\mathbf{AR_{ij}}\mathbf{f})}{d\mathbf{f}} - \frac{d\mathbf{f}^T\vec{e}_i\vec{e}_j\mathbf{f}}{d\mathbf{f}}\cdot\frac{\lambda^T\mathbf{AR_{ij}}\vec{v}}{\vec{v}^T\vec{e}_i\vec{e}_j^T\vec{v}} =$$
$$\mathbf{L}^T - \frac{\mathbf{L}^T\vec{v}}{\vec{v}^T\vec{e}_i\vec{e}_j^T\vec{v}}\cdot\mathbf{f}^T(\vec{e}_i\vec{e}_j^T + \vec{e}_j\vec{e}_i^T) + \lambda^T\mathbf{AR_{ij}} - \frac{\lambda^T\mathbf{AR_{ij}}\vec{v}}{\vec{v}^T\vec{e}_i\vec{e}_j^T\vec{v}}\cdot\mathbf{f}^T(\vec{e}_i\vec{e}_j^T + \vec{e}_j\vec{e}_i^T) = 0 \tag{4.17}$$

Now this can be solved in terms of $\mathbf{f}$ with λ as a parameter. Reordering (4.17) gives

$$\frac{\mathbf{L}^T\vec{v} + \lambda^T\mathbf{AR_{ij}}\vec{v}}{\vec{v}^T\vec{e}_i\vec{e}_j^T\vec{v}}\cdot\mathbf{f}^T(\vec{e}_i\vec{e}_j^T + \vec{e}_j\vec{e}_i^T) = \mathbf{L}^T + \lambda^T\mathbf{AR_{ij}} \Leftrightarrow$$
$$\mathbf{f}^T(\vec{e}_i\vec{e}_j^T + \vec{e}_j\vec{e}_i^T) = \frac{\vec{v}^T\vec{e}_i\vec{e}_j^T\vec{v}}{\mathbf{L}^T\vec{v} + \lambda^T\mathbf{AR_{ij}}\vec{v}}\cdot(\mathbf{L}^T + \lambda^T\mathbf{AR_{ij}}). \tag{4.18}$$

Equation (4.14b) is directly equivalent to the stability constraint. Hence the system of equations that needs to be solved in order to deduct the optimal $\mathbf{f}$ is

$$\begin{cases} \mathbf{f}^T(\vec{e}_i\vec{e}_j^T + \vec{e}_j\vec{e}_i^T) = \frac{\vec{v}^T\vec{e}_i\vec{e}_j^T\vec{v}}{\mathbf{L}^T\vec{v} + \lambda^T\mathbf{AR_{ij}}\vec{v}}\cdot(\mathbf{L}^T + \lambda^T\mathbf{AR_{ij}}) \\ \mathbf{AR_{ij}}(\mathbf{f} - \frac{\mathbf{f}^T\vec{e}_i\vec{e}_j^T\mathbf{f}}{\vec{v}^T\vec{e}_i\vec{e}_j^T\vec{v}}\cdot\vec{v}) = \mathbf{F} \end{cases} \tag{4.19}$$

4.3.1 Second Order System Solution Attempt

To solve system (4.19), note that the second equation only has $\mathbf{f}$ as an unknown. Hence, some elements of $\mathbf{f}$ are solved and then plugged in the first equation to find λ. Using the same notation as in Proof 4.1.2 for the columns of $\mathbf{A}$, the elements of $\vec{v}$ and $\mathbf{f}$, we have

$$\mathbf{AR_{ij}}(\mathbf{f} - \frac{\mathbf{f}^T\vec{e}_i\vec{e}_j^T\mathbf{f}}{\vec{v}^T\vec{e}_i\vec{e}_j^T\vec{v}}\cdot\vec{v}) = c_1f_1 + \cdots + c_mf_m - \frac{f_if_j}{\kappa_i\kappa_j}(c_1\kappa_1 + \cdots + c_m\kappa_m). \tag{4.20}$$

Note that the left hand side of the above equation is equal to $\mathbf{F}$, but also the original equation $\mathbf{Af} = \mathbf{F}$ holds. In addition $\vec{v}$ was defined so that $\mathbf{A}\vec{v} = 0$. Hence the above equation can be written as

$$\begin{aligned}
c_1 f_1 + \cdots + c_m f_m - \frac{f_i f_j}{\kappa_i \kappa_j}(c_1\kappa_1 + \cdots + c_m\kappa_m) = \mathbf{F} \Leftrightarrow \\
(\sum_{r=1}^{m} c_r f_r - \mathbf{F}) - c_i f_i - c_j f_j - \frac{f_i f_j}{\kappa_i \kappa_j}(\sum_{r=1}^{m} c_r\kappa_r - c_i\kappa_i - c_j\kappa_j) = \mathbf{0} \Leftrightarrow \\
(\mathbf{Af} - \mathbf{F} - c_i f_i - c_j f_j) - \frac{f_i f_j}{\kappa_i \kappa_j}(\mathbf{A}\vec{v} - c_i\kappa_i - c_j\kappa_j) = \mathbf{0} \Leftrightarrow \\
-c_i f_i - c_j f_j + \frac{f_i f_j}{\kappa_i \kappa_j}(c_i\kappa_i + c_j\kappa_j) = \mathbf{0} \Leftrightarrow \\
c_i f_i - c_i \frac{f_i f_j}{\kappa_j} + c_j f_j \frac{f_i f_j}{\kappa_i} = \mathbf{0} \Leftrightarrow \\
\begin{pmatrix} \vdots \\ c_i \\ \vdots \end{pmatrix} f_i(1 - \frac{f_j}{\kappa_j}) + \begin{pmatrix} \vdots \\ c_j \\ \vdots \end{pmatrix} f_j(1 - \frac{f_i}{\kappa_i}) = \mathbf{0}. \qquad (4.21)
\end{aligned}$$

From a first look, there are n equations and 2 unknowns, the i^{th} and j^{th} entries of the original force vector $\mathbf{f}$. However, looking in the geometry matrix $\mathbf{A}$, note that each column represents one element, and hence, there are only two nonzero entries in each column. That means that there are exactly two equations for c_i and two equations for c_j. These entries can share at most one row[1] and hence there are either four or three equations. For the system not to be overdetermined, the elements that can be removed have to have at least two or one linearly dependent equation, in the cases of four or three distinct equations respectively. The equation is then solvable (determinate or indeterminate) for the i^{th} and j^{th} entries of $\mathbf{f}$, and the first equation of system (4.19) can solve for the rest of the elements of $\mathbf{f}$.

[1]Since in the geometry matrix each row represents a node, if both rows where the same then both columns would be equal and correspond to the same element.

4.3.2 Symmetry of second order redundancy

Even though the solutions of the above problem can be very computationally expensive due to the number of cases that the algorithm requires to check for stability, for specific initial ground structures, second order redundancy, can be acquired by implementing *symmetry* on the first level redundant optimas. Specifically, the authors provide and justify the following conjecture

Conjecture 3. If the ground structure is symmetric in terms of the axis that is defined by the total loading (hence also under symmetric loading), then an optimal second order redundant is the mirror projection of a first order redundant assymmetric optimal with respect to the axis of symmetry.

This conjecture has mainly risen from the fact that on symmetric ground structures under symmetric loading, half of the first order redundant optimal structure has the required second degree redundancy. That is adding the symmetric element to the symmetric side missing the particular one from the other half of the first order redundant structure is the least volume addition to transform from first to second degree redundancy. For instance, Figure 4-1 is a symmetric 4×4 ground structure for a simply supported beam with a loading in the axis of symmetry. Figure 4-2 depicts the first order redundant optimal.

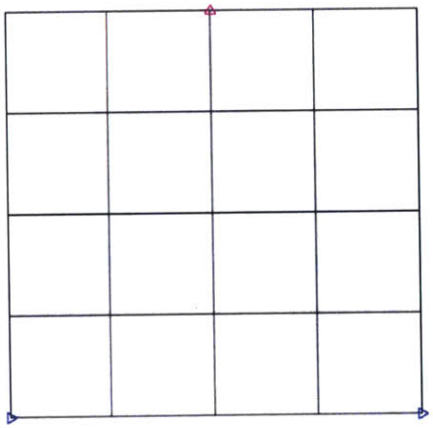

Figure 4-1: A 4 × 4 simply supported beam type ground structure with two fixities and one point load on the axis of symmetry

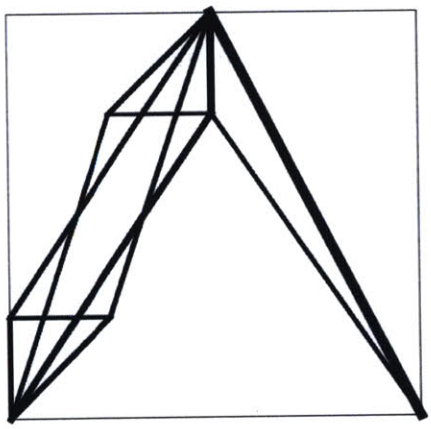

Figure 4-2: Optimal level 1 redundant least volume simply supported beam – each side implements an optimal mechanism of achieving redundancy

One can notice that on each side of Figure 4-2, the algorithm has optimized the structure to allow the removal of each one element. By mirroring the structure with respect to the axis of symmetry, we obtain a structure that exactly any two elements can be removed and the structure would still be stable (Fig. 4-3). This is the best way to reach second order redundancy from the first order redundant structure.

In this simple mirroring way we can acquire second order redundant, possibly

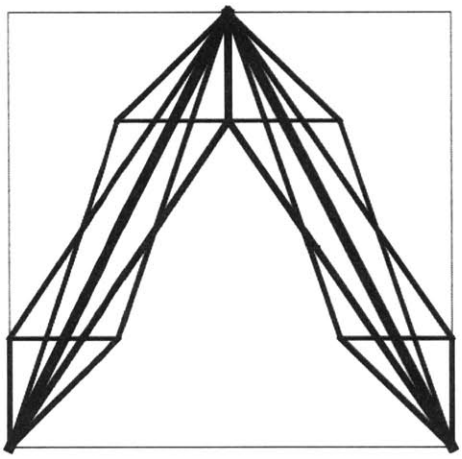

Figure 4-3: The second order redundant optimal as a result of mirroring the first order structure

optimal, if not optimal structures without using almost no additional computational cost from the linear approach. This holds for all symmetric GS with symmetric loading. Even though this might seem like a method that can be extended for higher order cases, it actually cannot be generalized for higher orders of redundancy, partly due to the nonlinear relationship of the substructures of higher degrees of redundancy. This is explained further in the next section.

4.4 Relative Reduction Relations of Optimal Redundant Structures

One question that might arise from the previous section is whether one can achieve and i^{th} level of redundancy by optimizing the $(i-1)^{th}$ redundant structure for an additional redundancy level. However, for this to be the case, there would have to exist a general linear relationship between the i^{th} and $(i-1)^{th}$ redundant substructures. In this section, we show that this unfortunately is not the case. This is shown for a second to first order order of redundancy and can easily be extended to higher orders by induction.

4.4.1 Relation between Second and First Order Redundancy

To identify the relation between the first and second order redundant structures, we need to express the relation between the internal forces of their substructures. Theoretically, if these forces were to be linear, then we could virtually achieve any order of redundancy with linear programming.

Let $\mathbf{f}_i, \mathbf{f}_j$ and $\mathbf{f}_{ij}$ the internal force vectors of the reduced substructures when element i, j and both i, j are removed respectively. The question is to identify the relationship between these three. First from the previous section we showed that $\mathbf{f}_i = \mathbf{K_i f}$ where $\mathbf{K_i} = \mathbf{I}_m - \frac{\vec{v}\vec{e}_i^T}{\vec{e}_i^T\vec{v}}$. This equation can be rewritten as

$$\begin{aligned}
\mathbf{f}_i &= \mathbf{f} - \frac{\mathbf{f}^T\vec{e}_i}{\vec{v}^T\vec{e}_i}\vec{v} \Leftrightarrow \\
\mathbf{f} - \mathbf{f}_i &= \frac{\mathbf{f}^T\vec{e}_i}{\vec{v}^T\vec{e}_i}\vec{v} \Leftrightarrow \\
\vec{v}^T(\mathbf{f} - \mathbf{f}_i) &= \frac{\mathbf{f}^T\vec{e}_i}{\vec{v}^T\vec{e}_i}\vec{v}^T\vec{v}.
\end{aligned} \tag{4.22}$$

Rewriting $\mathbf{f}_{ij}$ and plugging in (4.4.1) for i and j gives

$$\begin{aligned}
\mathbf{f}_{ij} &= \mathbf{f} - \frac{\mathbf{f}^T\vec{e}_i\mathbf{f}^T\vec{e}_j}{\vec{v}^T\vec{e}_i\vec{v}^T\vec{e}_j}\vec{v} \Leftrightarrow \\
\mathbf{f}_{ij} &= \mathbf{f} - \frac{\mathbf{f}^T\vec{e}_j}{\vec{v}^T\vec{e}_j}(\mathbf{f} - \mathbf{f}_i) \Leftrightarrow \\
\mathbf{f}_{ij} &= \mathbf{f} - \frac{\vec{v}^T}{\vec{v}^T\vec{v}}(\mathbf{f} - \mathbf{f}_i)(\mathbf{f} - \mathbf{f}_j)\vec{v}
\end{aligned} \tag{4.23}$$

Hence, the relation between the internal forces of the structure with elements i,j removed can be given with respect to the internal forces of the reduced structures with i and j elements removed respectively and the original structure's forces, by the above equation. Notice that this equation is symmetric for i and j confirming that $\mathbf{f}_{ij} = \mathbf{f}_{ji}$, and is nonlinear with respect to $\mathbf{f}_i$ and $\mathbf{f}_j$ due to the second term. This relation confirms that second order redundancy cannot be achieved by performing the same linear programming on the first order redundant structure.

Now by induction it is easy to generalize this equation for i^{th} order redundancy

directly to first order reduced elements. Specifically if $\mathbf{f}_{el_1,\ldots el_i}$ the internal force vector of the i^{th} level reduced substructure, and $\mathbf{f}_{el_r}$ the internal force vector of the first order substructure with r element removed, then the relationship between these vectors is given by

$$\boxed{\mathbf{f}_{el_1,\ldots el_i} = \mathbf{f} - \frac{1}{(\vec{v}^T\vec{v})^{i-1}}\vec{v}^T(\mathbf{f}-\mathbf{f}_{el_1})\cdots\vec{v}^T(\mathbf{f}-\mathbf{f}_{el_{i-1}})(\mathbf{f}-\mathbf{f}_{el_i})\vec{v}.} \tag{4.24}$$

4.5 Discussion

This chapter extended the algorithm used to identify first order optimal redundant structures to higher orders. We showed that with the current approach the computational cost of higher orders of redundancy increases geometrically and hence the proposed approach is likely not the optimal path to the higher redundancy problem. In this section, we discuss further the current optimization and recommend potential extensions that could provide a faster solution to the higher order optimization problem.

4.5.1 State of the Art – Pure Topology Optimization

Since the relation between a higher order substructure and the original structure is polynomial with a degree same as the order of the substructure, there is little one can do to improve the approach with a Ground Structure. Especially given the fact that with the Ground Structure approach, the algorithm builds the geometry matrix for every single connection – something extremely computationally costly. This is particularly useful if the user input has a limited allowable design space, such as forbidden zones. Refined Ground Structures perform both Topology and Geometry Optimization simultaneously. In the case where the user only requires a more abstract optimal value, topology and geometry optimization could be decoupled. If a theorem could show that the total volumetric optimization is equivalent to a series of consecutive topology and geometry optimizations, then the process could be much more computationally efficient.

In this scenario, the user would input the number of allowable nodes and fixities, and in response the algorithm could first identify the acceptable connections. After the connectivities are finalized, a geometry optimization would occur by moving around the nodes in the design space. Even though this approach significantly reduces the computational cost of calculating the geometry matrix and the kernel of an entire Ground Structure, little is known in the literature of Topology Optimization, whether a series of consecutive topology and geometry optimizations result in the global optimal.

4.5.2 On Symmetry beyond Second Order Redundancy

In Section 4.3.2, a conjecture for a second order redundancy in symmetric Ground Structures under symmetric loading was provided. This approach is likely to be extended for larger number of elements in a similar manner. That is on symmetric GS, each element is either a part of 1 or more individual stable substructures. Hence, in such structures we require to add not just elements but layers of substructures that are able to take the loads in case of failure of one element. That means that in several cases, we do not desire to use one main element for several substructures, as its removal would neutralize all dependent substructures. Figure 4-4 shows a 4×4 transmission tower GS with rightward wind loading and vertical gravity loading. Each element is used by only one substructure, in this optimal redundant configuration. To increase a level of redundancy, the symmetric loadings are required. However, since the loading is not symmetric, two projections of the redundant dominant side need to be taken (Figure 4-5). This two-way symmetry then increases redundancy not just to second but third degree, as we indirectly added another two levels of substructures that successfully transfer the loads.

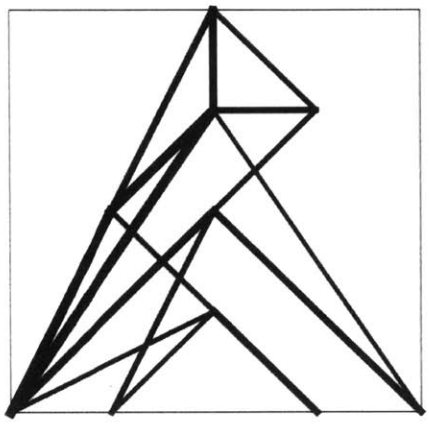

Figure 4-4: A 4×4 transmission tower with level 1 redundancy

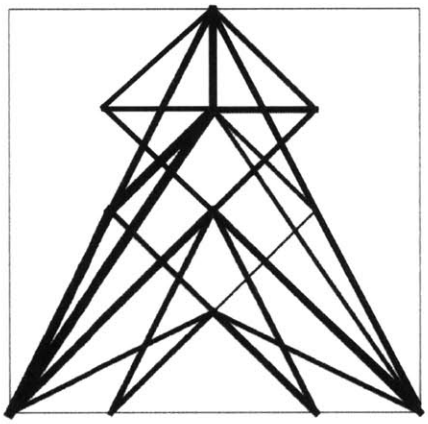

Figure 4-5: Optimal level 3 redundant least volume transmission tower resulting from symmetry

Hence, in specific cases of structures with several axes of symmetry, the redundant optimal can be extended to higher even orders (twice the number of axes of symmetry) of redundancy.

4.5.3 Integer optimization

As discussed in the previous Chapter, after a refined enough Ground Structure, the optimization algorithm would likely add multiple elements of smaller cross sec-

tional area with different values. This sets the solution uneconomical especially for higher orders of redundancy. Thus the higher level redundancy optimization can be formulated as an integer program by relaxing the stability constraint. This results in a very computationally expensive process, however it allows the user to limit the allowable cross sectional areas to limited numbers. The integer program takes the following form

$$\begin{array}{ll} \min\limits_{\mathbf{f}\in\mathbb{Z},i} & \left|\mathbf{L}^T\left(\mathbf{f}-\prod_{r=1}^{i}\left(\frac{\vec{e}_{el_r}^T\mathbf{f}}{\vec{e}_{el_r}^T\vec{v}}\vec{v}\right)\right)\right| \\ \text{s.t.} & \perp\!\!\!\perp \mathbf{AR}_{\mathbf{el_1},..\mathbf{el_i}}\left(\mathbf{f}-\prod_{r=1}^{i}\left(\frac{\vec{e}_{el_r}^T\mathbf{f}}{\vec{e}_{el_r}^T\vec{v}}\vec{v}\right)\right) \leq \mathbf{F} \quad \forall\, i\in\mathcal{J}. \end{array} \tag{4.25}$$

Chapter 5

Analysis of Results

In this Chapter, several visual results and examples of the optimal redundant structures under different loadings and Ground Structures are showcased. The Chapter focuses on presenting the breadth of applications and solutions of Redundant Optimal Structures but also highlights some interesting insights about how the densities of different Ground Structures perform compared with each other as well as the purely volumetrically optimal solution. The Chapter is structured in terms of Ground Structure linearity. First, the analysis is performed on normal orthogonal grid examples, specifically a Cantilever beam, a Simply Supported Beam and a transmission tower example. Then some nonlinear examples are used that implement Talischi's Polymesher (Talischi et al., 2012) non-regular mesh generator, inspired by *Voronoi diagrams*. The Chapter concludes with a discussion on the infinity of redundant solutions and some comparisons of the redundant optimal with the pure volumetric optimal.

5.1 Orthogonal Grid Examples

Using the GRAND (Zegard et al., 2014) platform and plastic layout optimization with the refined constraints, several shapes with free form boundary conditions that are redundant have been tested. Due to the fact, that the redundancy results require knowledge of the kernel of the geometry matrix and every connectivity matrix of

each potential substructure needs to be stored, the algorithm is computationally expensive. That is, the more refined the initial grid, the more time the algorithm takes to compute, but yet the more geometry independent the result is. However, when the grid is refined to the extent of including more than two orders of magnitude blocks, then the computational time increases polynomially resulting to the range of a day to compute the optimal results. Hence the maximum orthogonal Ground Structures that have been tested do not have more than 100 fixed nodes (and their corresponding fixities).

In the GRAND script, the user has the freedom to vary the number of subdivisions of the Ground Structure in the x and y directions, as well as the length of each direction. The redundancy optimization algorithm maintains this freedom.

5.1.1 Cantilever Beam

A cantilever beam is defined as a fixed horizontal beam with a vertical unit load applied on its edge. Its truss approximation is a structure with at least two fixities on the vertical direction and a certain span on the horizontal direction. Figure 5-1 shows an example of a truss cantilever ground structure with horizontal length of 20 units and vertical of 5 units, while it is discretized in 10 blocks horizontally and 5 blocks vertically. This results in 6 fixities on the vertical direction (blue triangles), and one point load (shown in red) in the middle of the cantilever edge.

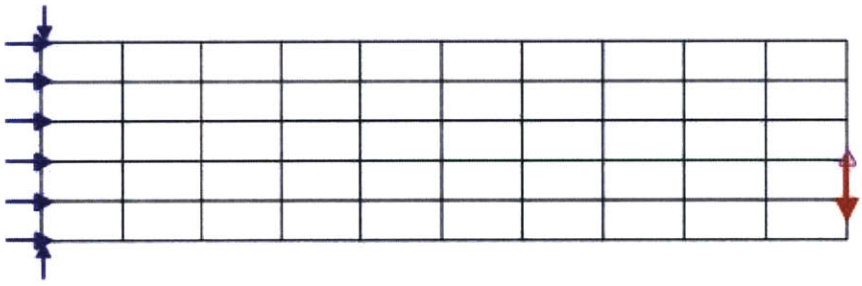

Figure 5-1: Initial cantilever 10×5 grid GS with horizontal and vertical lengths 20 and 5 respectively

The user is able to vary all these values, and in the analysis we cover both the way the volume changes as the refinement changes, but also the ratio of span to width

with volume under the identical discretization. When the redundancy algorithm is used, a huge variety of shapes and formations are acquired, based on the refinement and the span to width ratio of the cantilever. Figures 5-2 and 5-3 are an example of a very short span to width ratio Ground Structure and redundant Optimal respectively, while Figures 5-4 and 5-5 show an example of a span to width ratio under the identical discretization. One can easily note that even though the discretization is exactly the

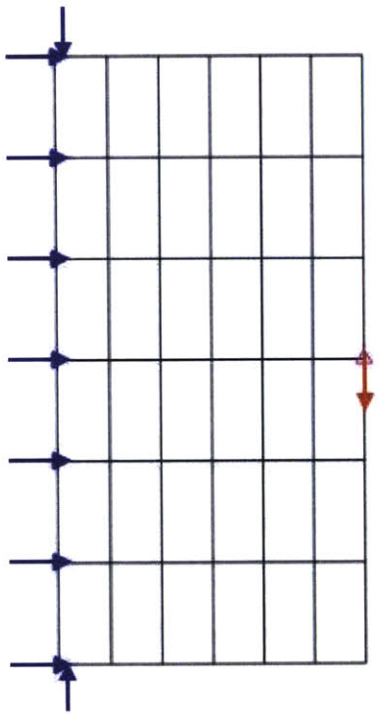

Figure 5-2: 6×6 Ground Structure with Span to Width ratio of $\frac{1}{2}$

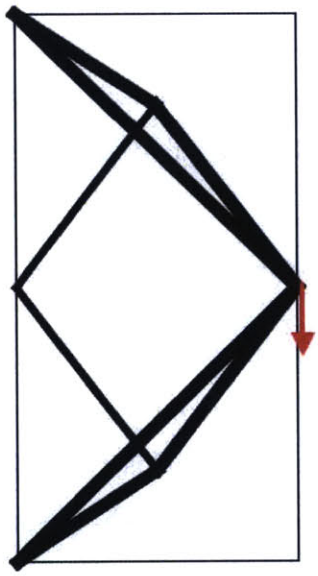

Figure 5-3: 1^{st} order Red. Optimal for a $\frac{1}{2}$ S-W ratio and 6×6 density

same (6×6) for both cases, the shape is highly different. Increasing the span would result in larger load paths and additionally more elements, but at the same time a very short span with very large width would not allow the structure to reach a value close to the optimal as it would be hindered by the limited spanning. However, in redundancy optimization, the algorithm benefits a short span structure volumetrically, as that would mean more direct load paths and less elements.

However, comparing span to width ratio with volume does not make so much sense, given the fact that we are essentially comparing different ground structures. A

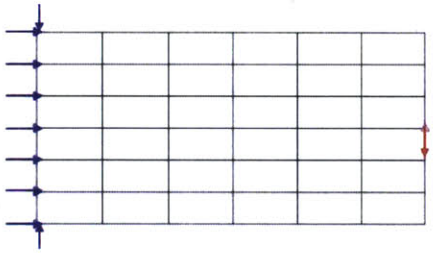

Figure 5-4: 6×6 Ground Structure with Span to Width ratio of 2

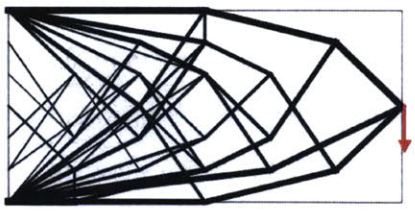

Figure 5-5: 1^{st} order Red. Optimal for a 2 S-W ratio and 6×6 density

more interesting comparison is that of the density of the grid for an identical ground structure with respect to volumetric performance.

Cantilever Ground Structure Design Space

For any Ground Structure Sigmund (Sigmund, 2001) has shown that the higher the mesh density of the optimization, the more geometrically independent the optimization is, and hence the closer it is to the optimal value. In the traditional volume-only Ground Structure approach, the mesh density is very close to the optimal for a 10×10 grid. Figure 5-6 shows the *design space* of a 10×10 length Cantilever, under different discretizations from a 1×1 to a 10×10 grid.

From the above figure, one can see that in general the design space tends to slope downwards as the refinement increases. For both small horizontal and vertical refinements, the solutions are far from the optimal as the do not allow for sufficient spanning. Especially on the horizontal grid refinement, not allowing multiple nodes to span sufficiently distorts the cantilever shape. Another interesting point on the Optimal Redundant Design Space is that the reason that a lot of *noise* appears on the volume of the grids has to do with the symmetry of the Ground Structure. When the vertical refinement is an odd number, it means that the Ground Structure is not symmetric and there is no middle point for the load to be applied on. Hence, the redundant optimal for an asymmetric structure is slightly above the symmetric

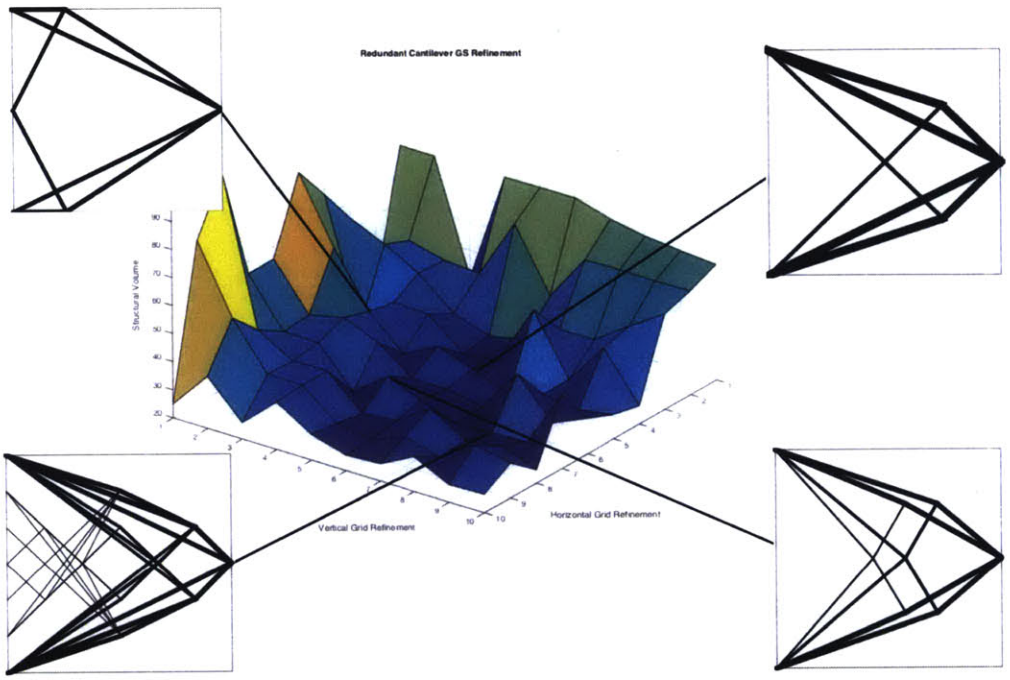

Figure 5-6: Ground Structure Design Space for a square cantilever

optimal, resulting in the *wavy* behavior of the design space.

Even though Figure 5-6 defends the assumption that more refinement implies less volume, from a practical perspective, due to the nature of redundancy optimization, increasing the refinement would result in hundreds of tiny elements that would be very costly to actually produce. Figure 5-7 shows the volume of an optimal structure with square grid refinement with respect to the number of elements, for the pure redundancy optimization. Combining these two figures, the user is able to compare and contrast the extent of volume savings to individual element production. This problem could also be formulated as a bi-objective optimization problem towards minimizing volume as well as number of elements or number of different elements types. This could be promising future work in the field.

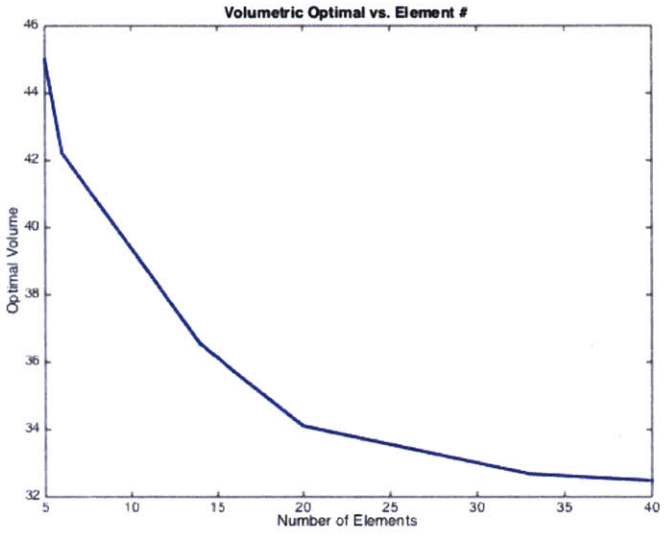

Figure 5-7: Number of elements of redundant optimal square cantilever for different grid refinements versus volume

5.1.2 Simply Supported Beam

Another common example that is presented is the orthogonal grid with two point fixities on the corners and a point load applied in the middle of the section. Figure 5-8 shows an example of a simply supported beam ground structure with horizontal length of 20 units and vertical of 5 units, while it is discretized in 10 blocks horizontally and 5 blocks vertically.

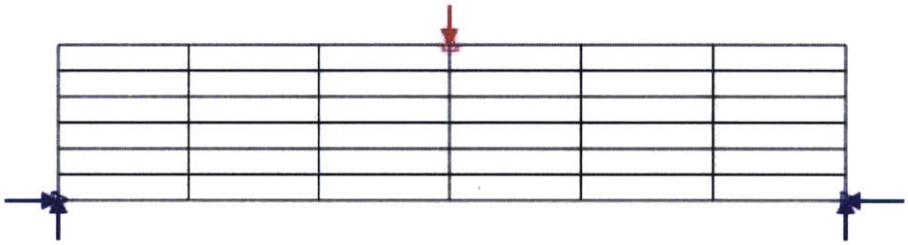

Figure 5-8: 10×5 grid GS with horizontal and vertical lengths 20 and 5 respectively

Again, length and mesh refinement in both directions are variable, and the results vary significantly for identical shapes and different refinements, and vice versa (Fig. 5-10 & 5-12)

From Figures 5-9 and 5-11 one can see that even though the initial refinement and

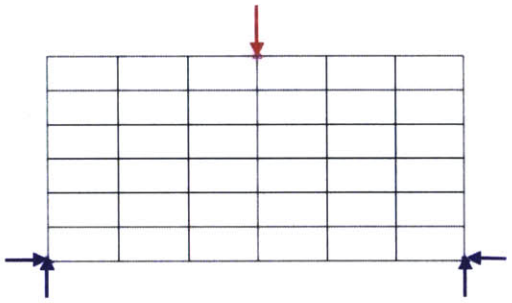

Figure 5-9: 6×6 midspan loaded beam GS with S-W ratio of $\frac{1}{2}$

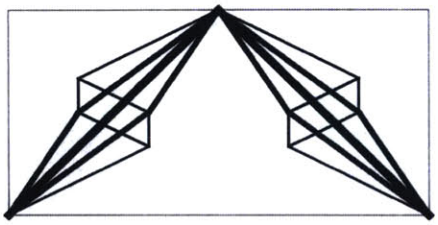

Figure 5-10: SSB Redundant Optimal for a $\frac{1}{2}$ span to width ratio and 6×6 density

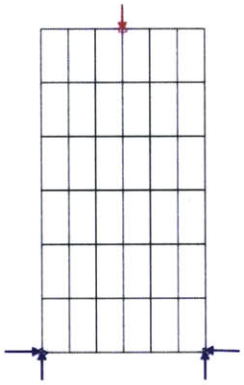

Figure 5-11: SSB 6×6 Ground Structure with Span to Width ratio of 2

Figure 5-12: SSB Redundant Optimal for a 2 S-W ratio and 6×6 density

load location is exactly the same, the optimal redundant trusses vary significantly.

SSB Ground Structure Design Space

Like in the Cantilever case, varying the mesh grid density of the Simply Supported Beam Ground Structure density for identical load – fixities location and dimensions, gives the design space of optimal redundant solutions as a function of the volume. Figure 5-13 shows the *design space* of a 10×10 length Cantilever, under different discretizations from a 1×1 to a 8×8 grid.

In the case of the Simply Supported beam, one can see that after the design space is refined further on the horizontal direction for larger than 2, the volume fluctuates

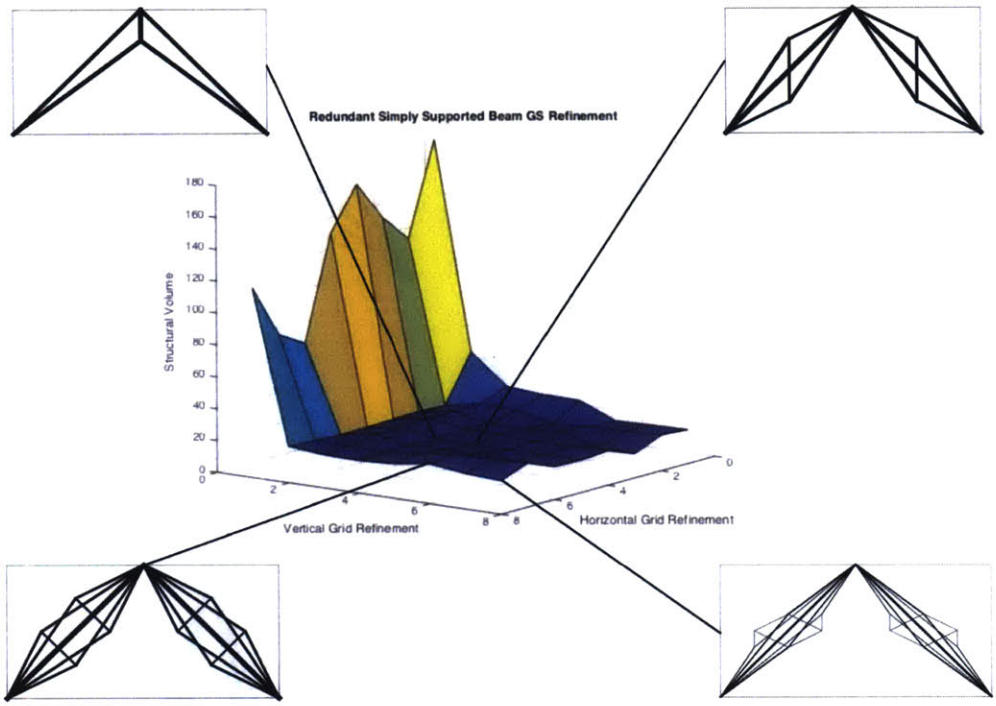

Figure 5-13: Design Space for a Simply Supported Beam

around the same value. This is the case because in a Simply Supported Beam, two layers of stable optimal structures (which would give the desired level of redundancy) are acquired early on in the refinement (for a horizontal density ≥ 2) and hence further refinements *relieves* material from heavy elements adding more elements but without a significant effect. This fluctuation is a typical example of symmetric structures with infinite solutions, i.e. there is a family of solutions with infinitely different shapes (but approximately similar as one can see from Fig. 5-13) but with the same volume. This example raises the problem of **nodal redundancy** i.e. the removal of entire nodes. One can easily notice from Figure 5-13, that even though the refinement of elements increases, certain nodes remain critical in the structure. Nodal removal is equivalent to failure of several elements and usually has a much more significant damage compared to element removal. This is another aspect of redundancy that is not covered in this thesis.

5.1.3 Free Standing Cantilever with Self-Weight

A very applicable example that is worthy of attention is a transmission tower design. A transmission tower is a cantilever truss that receives discretized lateral wind loads throughout its height and has the self-weight applied downwards on the top middle of the structure. For the Ground Structure, we assume that the point load is always applied in the top middle of the orthogonal grid, and its magnitude equals the height of the grid itself. The horizontal loads are applied on the middle of the orthogonal grid and to resemble a uniform distribution, they are taken to be unit loads and face from left to right of the grid. Figure 5-14 shows an example of a free standing cantilever ground structure with horizontal length of 20 units and vertical of 10 units, while it is discretized in 6 blocks horizontally and 6 blocks vertically.

As expected, varying the height and width of the tower under the same discretization affects the redundant optimal shape.(Fig. 5-16 & 5-18)

Transmission Tower Ground Structure Design Space

As before, part of the design space of optimal redundant solutions can be given by varying the horizontal and vertical grid refinement. Figure 5-6 shows the *design space* of a 10×20 length Transmission Tower, under different discretizations from a 1×1 to a 8×8 grid.

Figure 5-19 highlights a very interesting result. The vertical grid refinement induces a much more drastic change than the horizontal grid. In fact the horizontal grid refinement seems to have very little effect towards the volumetric optimal. On the contrary, the vertical refinement allows more points to discretize the distributed loads and send them through smaller capacity elements directly to the base base fixities with the optimal load path. In the horizontal direction, most loads tend to accumulate in the two corners hence intermediate fixities have no significant effect. This structure is one of the cases where the number of elements increases drastically the closer we get to the redundant volumetric optimal. Thus, transmission towers might be a promising case study to combine with integer programming, finding a Pareto

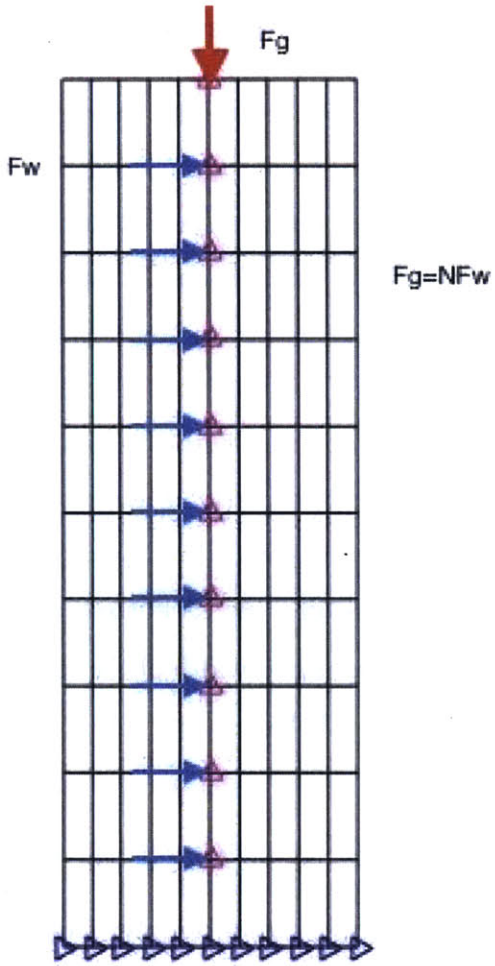

Figure 5-14: 10×10 grid GS for a Transmission Tower with height 30 and width 10 respectively

front between total volume and number of different cross sectional elements.

5.1.4 Bridge Structure

The bridge in a truss form can be considered as a spanning structure that has to transfer a point load (could make an assumption for more) to the two fixities on the sides. Figure 5-20 shows an example of a 10×20 Bridge Ground Structure with allowable width 20 and height 10.

We vary the height and width of the bridge under the same discretization, and

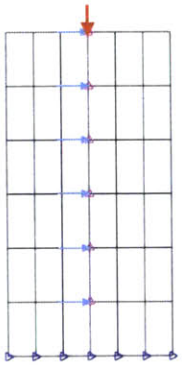

Figure 5-15: 6×6 transmission tower GS with Height-Width ratio of 2

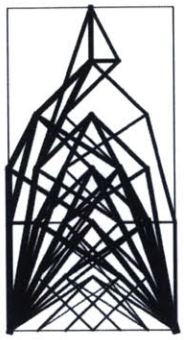

Figure 5-16: Trans. Tower Red. Optimal for a $\frac{1}{2}$ H-W ratio and 6×6 density

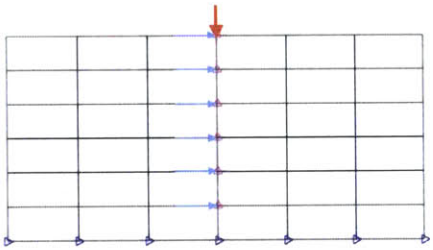

Figure 5-17: SSB 6×6 Ground Structure with Span to Width ratio of 2

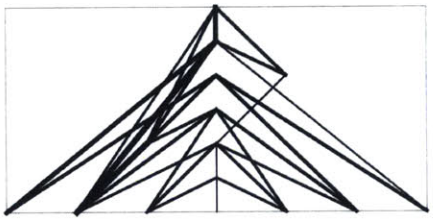

Figure 5-18: Transmission Tower Optimal for a 2 S-W ratio and 6×6 density

the redundant optimal changes accordingly (Fig. 5-16 & 5-18).

Bridge Ground Structure Design Space

The volume varies by allowing the bridge density to change. However this change is quite nonuniform. Figure 5-25 shows the *design space* of a 20×10 length bridge, under different discretizations from a 1×1 to a 8×8 grid.

From Figure 5-25, one can see that increasing the horizontal grid refinement of the Bridge GS to anything larger than 3, drops the volume down to approximately the same value (with the exception of a few outliers). This is the case because a refined enough horizontal direction, allows for several light horizontal bridge substructures to carry the middle load to the fixities. The shape of different topologies with almost identical volumes is another example of families of shapes with an *optimal* behavior.

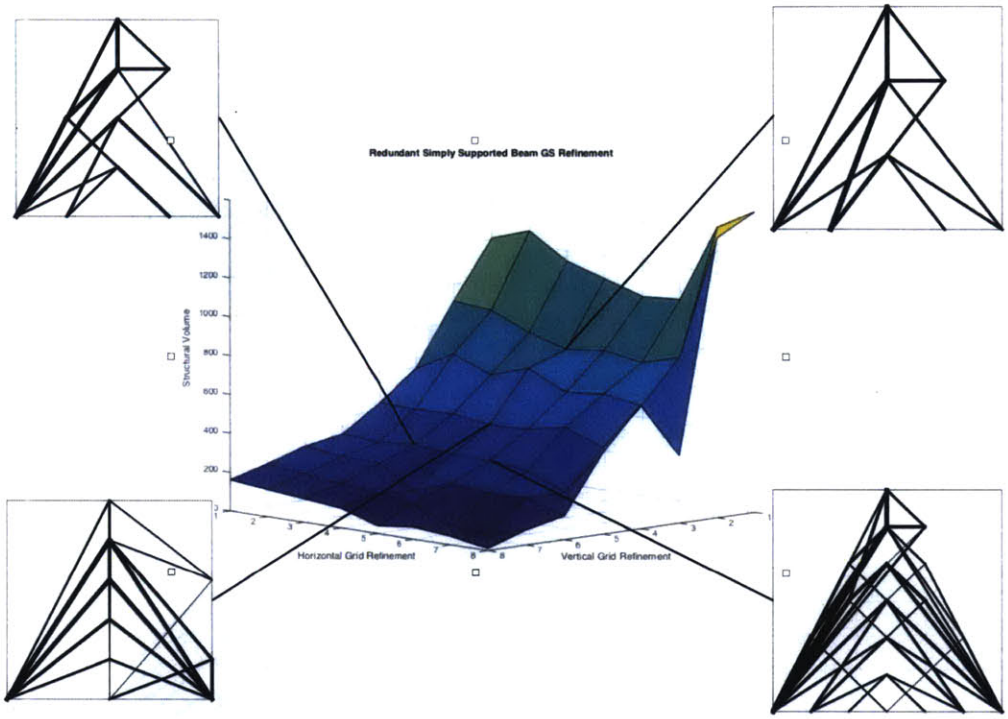

Figure 5-19: Design Space for a Transmission Tower

5.2 Free-form Ground Structures

In the previous section, several examples of linear orthogonal grids along their respective design spaces were sampled. Nevertheless, the developed algorithm can input any possible mesh with any number of discretized loads and fixities. The most convenient way to showcase its power, is to implement the Polymesher generator, a code developed in Matlab by Talischi et al (Talischi et al., 2012) that allows the computationally rapid generation of polyhonal grids from virtually any initial structure. Below, two examples of irregular Ground Structures are shown, one for the regular Mitchel Cantilever with circular base, and one for

5.2.1 Michell Cantilever

A Michell Cantilever (Michell, 1904) with a circular base is created by limiting the spanning space of the space to be a semicircle. Note that by nature of the Polymesher algorithm the only controllable refinement is the number of equal-area polygons the

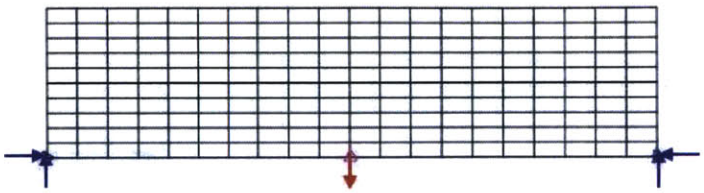

Figure 5-20: 10×20 grid GS for a Bridge with height 10 and width 20 respectively

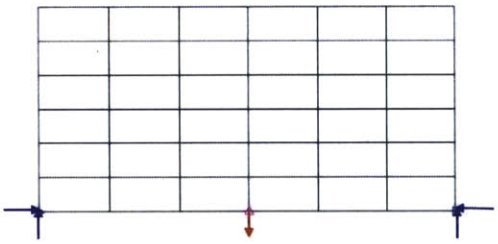

Figure 5-21: 6×6 Bridge Ground Structure with Width-Height ratio of 2

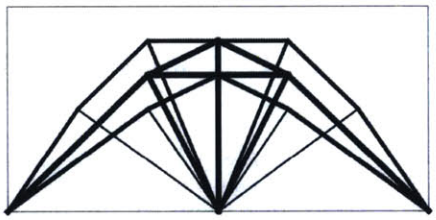

Figure 5-22: Redundant Optimal Bridge for a 2 W-H ratio and 6×6 density

space can be divided in, hence a three dimensional space is not applicable. Figure 5-26 is a 20-polygon Michell Ground Structure with two fixities. Note that the number and location of fixities can change (more fixities equally distributed around the circle is an option).

Figure 5-27 is the redundant optimal solution of the Michell Ground Structure in Figure 5-26. The redundant optimal interestingly resembles overlaid typical Michell trusses acting together.

Below some additional examples of Michell Cantilever with different number of area refinements. Refining number of areas and changing the number of fixities can result in significantly different and more efficient shapes. For instance, Figure 5-28 uses less than 30% of the material as in Fig 5-27 as well as less elements, due to the fact that more fixities allow additional more efficient load paths.

The extension of fixities and polygon separation can be taken further (Fig. 5-29), till the semicircle is continuously covered by fixities and the equal area polygons

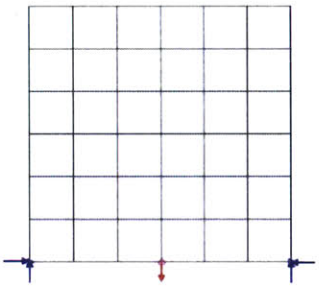

Figure 5-23: 6×6 Bridge Ground Structure with W-H ratio of 1

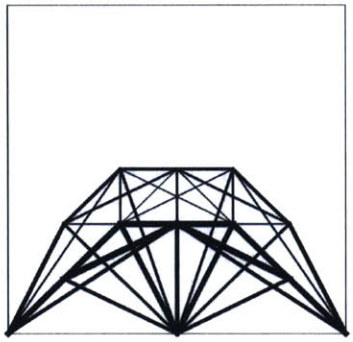

Figure 5-24: Bridge Redundant Optimal for a 1 W-H ratio and 6 × 6 density

approximate points. The only limiting factor for such nonlinear GS is the computational time and memory, as it becomes even more costly compared to orthogonal grids. Nevertheless, a more efficient algorithm is definitely feasible given the early stage of development of the redundancy optimization algorithm.

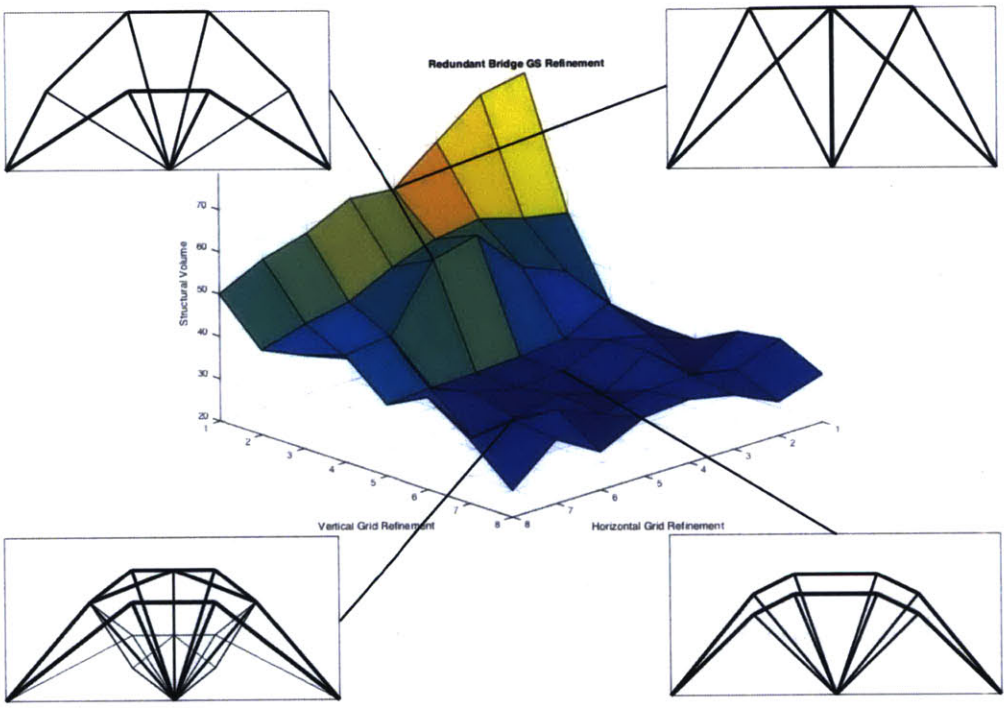

Figure 5-25: Mesh Grid Refinement Design Space for a Bridge

5.2.2 Serpentine Beam

Another interesting nonlinear example is the Serpentine beam. The Serpentine beam is like a curved cantilever and is also obtained by implementing Voronoi segregation by Polymesher. Similarly to Zegard's volumetric optimal Serpentine (Zegard et al., 2014), the result remains a curved Michell-like structure as shown in Figure 5-30.

Again, this shape can be further refined to achieve additional material efficiency with the cost of additional elements and more computational cost. Several other examples of nonlinear Ground Structures are taken from Taschini's Polymesher (Talischi et al., 2012) source code and their redundant optimal shapes are presented in Appendix A.

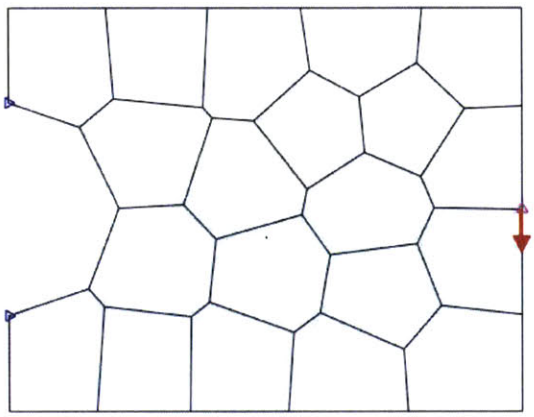

Figure 5-26: A Michell Ground Structure with 20 equal polygons and 2 fixities

5.3 Discussion

In the previous sections, the optimal redundant value and shape was identified for several initial configurations of GS, mesh refinements and dimensions. In this section, these values are introduced again and discussed from the point of performance with respect to the pure volumetric optimal, but also with each other in the sense of infinite solutions.

5.3.1 Comparison to Pure Volumetric Optimization

For first order redundancy optimization, a convenient rule of thumb on the performance of a structure for a particular refinement is its relative performance with the identical Ground Structure of the pure volume optimal. That is, if the volumetric optimal has an optimal volume, say V, then the first order redundant optimal should be less than $2V$ to be meaningful[1]. The same principle can be theoretically applied in higher orders of redundancy, i.e. if the 5^{th} order redundant structure is greater than $5V$, five layers of the volumetric optimal would provide the desired level of redundancy (even if structural depth of the system would likely be undesirable).

[1]This is the case since if the first order redundant optimal is heavier, the user could theoretically achieve equivalent redundancy by having two duplicate layers of the pure volumetric optimal

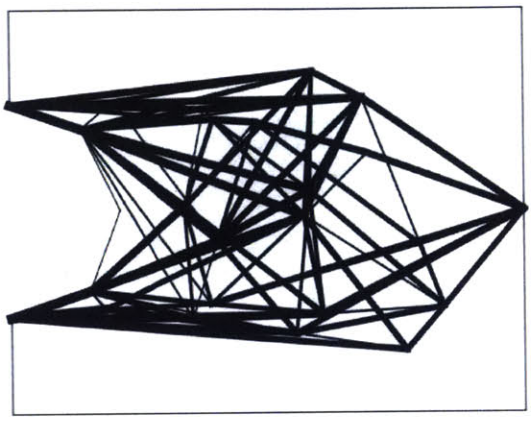

Figure 5-27: The Michell-like redundant optimal solution for the given GS

Below, for each case, some comparisons are made for the redundant optimal of certain configurations and grid refinements compared to the volumetric optimal, both in terms of volume as well as number of elements.

10×10 Square Orthogonal Cantilever GS			
Grid Refinement (Ver.$\times$Hor.)	Pure Volumetric Optimal (a)	1^{st} order Redundancy Opt. (b)	Material Savings $(\frac{2a-b}{2a})\%$
2×2	25.0	65.0	-30
4×2	25.0	56.7	-13.4
2×4	25.0	46.4	7.2
4×4	25.0	44.4	11.2
4×6	24.49	43.9	10.4
6×6	24.52	38.9	20.7
∞	~ 23.9	~ 30	37.2

Table 5.1: Volumetric Comparison of square cantilever with Redundant Optimal

Table 5.1 shows how the volume of the redundant optimal solution compares to the pure volumetric optimal. One can see that as the refinement increases, the change is much more drastic on the redundant optimal than the volume-only optimal. In

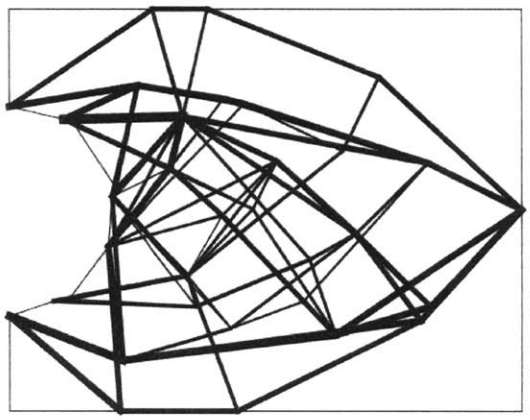

Figure 5-28: Redundant Michell truss for 4 fixities and 24 polygons

addition, as the refinement increases the saving on the material from the redundant optimal is more and more evident. In fact, already when a 6×6 refinement is achieved, there is about 24.3% savings on the material of the redundant optimal compared to doubling the volume of the single-volume optimal to reach the same redundancy. This number increases as the grid refinement increases. The volumetric optimal asymptotically reaches a volume slightly less than 24 while the highest computational refinements achieved from the redundant optimization[2] reach a value of about 30. This means that the redundant optimization can achieve material savings of more than 37% for the Cantilever Structure.

Below two similar tables are provided for a bridge design and a transmission tower along the percentage savings of each refinement.

[2]Improving the computational cost further would allow for higher order refinements and hence even more efficiency

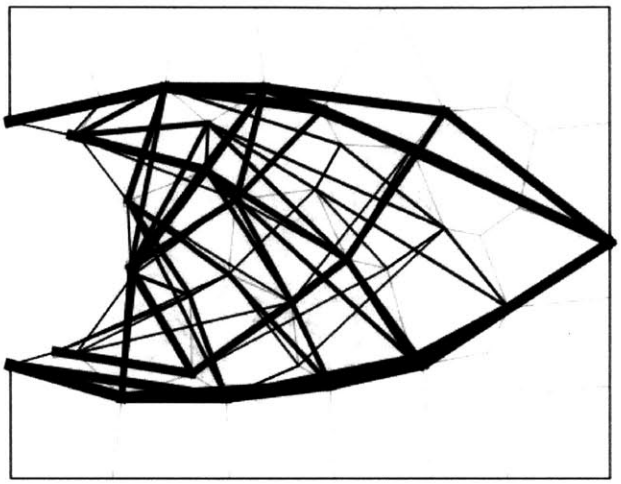

Figure 5-29: Redundant Michell truss for 4 fixities and 24 polygons

10×5 Orthogonal Bridge GS			
Grid Refinement (Ver.×Hor.)	Pure Volumetric Optimal (a)	1^{st} order Redundancy Opt. (b)	Material Savings $(\frac{2a-b}{2a})\%$
2×2	15.0	27.5	8.3
4×2	15.0	26.4	12
2×4	14.2	26.1	8.1
4×4	14.16	25.28	11.1
4×6	13.42	24.1	10.2
6×6	13.33	21.4	19.7
∞	12	~ 20	16.7

Table 5.2: Volumetric Comparison of Bridge Structure with Redundant Optimal

Bridge savings from the redundant optimal can reach up to 20% less material compared to double the volumetric optimal, as shown in Table 5.2. From Table 5.3 one can see that the Transmission Tower savings can be vary significant fluctuating at about 40% of twice the volumetric optimal – a very considerable material savings to set this kind of optimization very applicable in the case of Transmission Tower

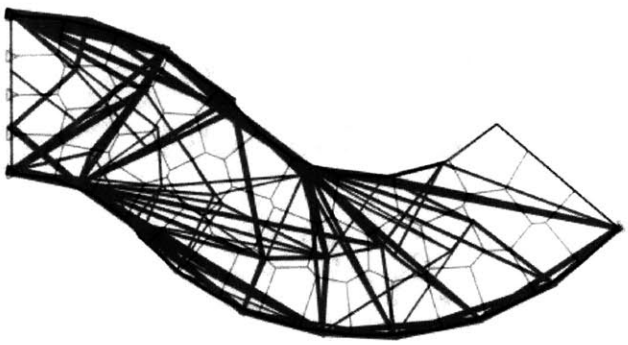

Figure 5-30: Redundant Optimal Serpentine beam for 4 fixities and 20 polygons

design.

5×10 Orthogonal Transmission Tower GS			
Grid Refinement (Ver.×Hor.)	Pure Volumetric Optimal (a)	1^{st} order Redundancy Opt. (b)	Material Savings $(\frac{2a-b}{2a})\%$
2×2	35.0	45.0	42.8
4×2	34.5	43.9	36.4
2×4	34.8	44.5	36.1
4×4	32.4	43.8	32.4
4×6	32.1	43	33
6×6	31.7	42.4	33.1
∞	30	~ 40	33.3

Table 5.3: Volumetric Comparison of Transmission Tower with Redundant Optimal

5.3.2 Infinite Solutions

Several times in the above sections (5-13,5-25), the redundant volumetric optimal seemed to fluctuate around certain families of shapes. Even for a fixed grid and

geometry of a GS, there are cases where the volume stays the same as well as the topology, but some elements interchangeably and continuously exchange magnitude of cross sectional areas setting them as *simultaneous optimals*. These optimal values are sometimes similar to what was discussed in Section 3.4.3 and infinite solutions. Rozvany studied this phenomenon of uniqueness, for pure volumetric optimization as discussed in 3.4.3, and this same phenomenon is encountered in redundant optimization. In fact, since redundant optimization is adding more elements compared to its purely volumetric counterpart, the effect of equivalent optimal values might be even more frequent. This effect can appear in two ways: either through an identical topology and continuous redistribution of material among the common elements, or through changes in topology that result in the same effect.

To showcase this phenomenon, we present some equivalent solutions for a square cantilever under an identical 4×4 refinement.

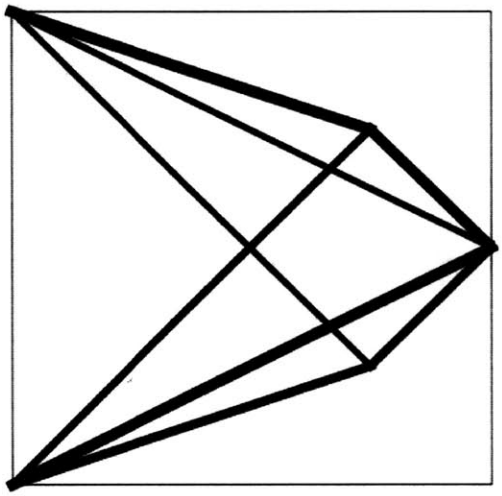

Figure 5-31: 4×4 cantilever optimal with 35 volume and 2^{nd} order redundancy

Figure 5-31 is the first design one gets from the redundant algorithm. To the reader's surprise, this structure topologically has a second order redundancy, allowing the removal of any two elements[3]. This algorithm can switch the cross sectional areas

[3]This is only topologically true because by definition the removal of any one element would result in a fully stressed structure

of the elements resulting in equivalent total volume and different element sizes. Since this can be performed continuously, one dimension of infinity is obtained from this variation. More interestingly, the second dimension of multiple solutions is acquired by constraining the spanning of some elements and inducing the same algorithm. Note that by prohibiting the spanning of one critical element of the optimal cantilever in Fig. 5-31, results in optimal configurations of exactly the same volume (35) and different topologies, as shown in Figures 5-32 and 5-33. Even though these shapes topologically have only one level of redundancy, they have exactly the same overall material as the first cantilever in FIg. 5-31.

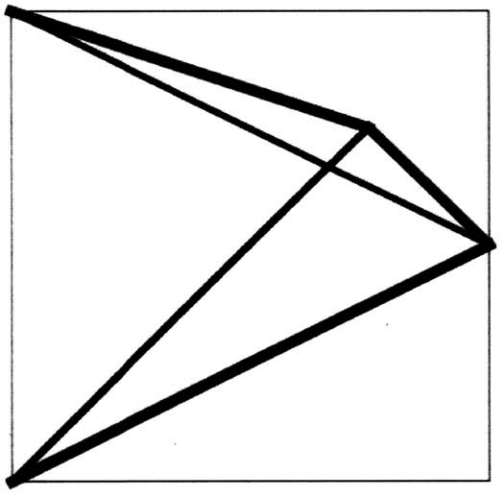

Figure 5-32: 4×4 cantilever redundant optimal with 35 volume and single order optimality

The multitude of solutions might be attributed to the biaxial symmetry of the Ground Structure and the applied load. Studying the relationship between Ground Structure topologies and symmetries with uniqueness of solutions is a very promising field with no research performed whatsoever in the case of redundant optimal structures.

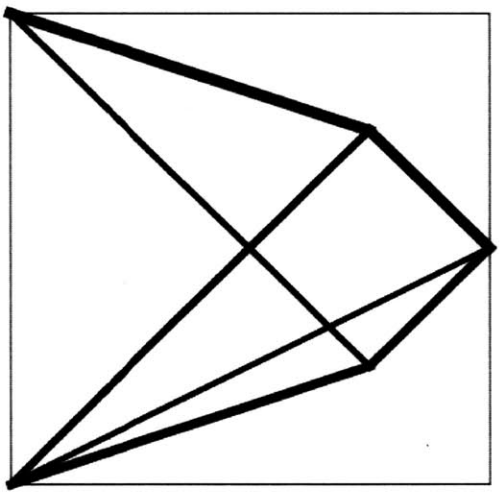

Figure 5-33: Alternative 4×4 cantilever redundant optimal also with 35 volume

Chapter 6

Conclusion

In this Chapter, a brief summary of the key contributions of this thesis is presented as well as the potential impact of the developed algorithm. Finally the chapter concludes with future research and potential extensions of this algorithm. Some concluding remarks restate why redundancy optimization is a vital field in the future of buildings.

6.1 Summary of Contributions

This thesis developed a volumetric topology optimization algorithm under redundancy constraints. Specifically, the algorithm produced the minimum volume structure that remains stable even after any element has been removed. This algorithm could potentially alleviate practicing engineer's concerns on the structural integrity of optimized structures, by embedding redundancy within the optimization.

In addition to the proposed algorithm, several original results were shown in this thesis with both practical and theoretical impact. First of all, the general closed form equation was shown between a redundant structure and any stable substructure missing i elements. This formula gives us a better understanding of the relationship between damaged stable structures and their original counterparts, allowing us to practically understand when redundancy works effectively. What is more, this result shows the nature of load paths and why partial failure sometimes propagates globally.

The algorithm was implemented in Matlab not just for continuous linear programming, but also for mixed integer programming. This way, we were able to compare the total volume of a truss with the number of elements in the system. Several applications of this algorithm were tested with different loadings and boundary conditions. We gathered these cases and plotted the design spaces of several optimal redundant structures with respect to their refinement. Plotting the design space allowed us to see families of optimal shapes but also how the optimal volume changes as the refinement of a ground structure changes. Finally, another key contribution provided in this thesis was showing that the 1^{st} order redundant optimal shapes have volume that is less than twice the volume of the pure volumetric optimal structure (Figure 6-1). This means that redundant optimal shapes can effectively combine safety with material efficiency.

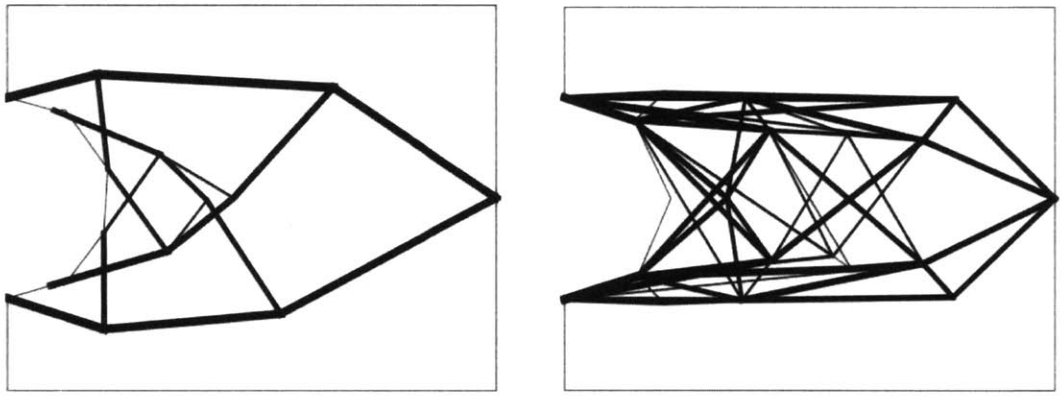

Figure 6-1: Redundant Michell (right) with 20% less material than twice the volume of the pure volumetric Michell truss (left).

6.2 Potential Impact

The developed algorithm allows us to generate a wide variety of shapes under different boundary conditions and Ground Structure refinements. However, from a practical perspective, the question that needs to be answered is whether the proposed structures can be built, and how feasible/costly is their constructability. There is no

binary answer to this question. Certain designs are actually very easy to construct and switching to the redundant optimal might take minimal effort and through optimization they can even save material compared to the conventional solution (Figure 6-2 vs. Figure 6-3).

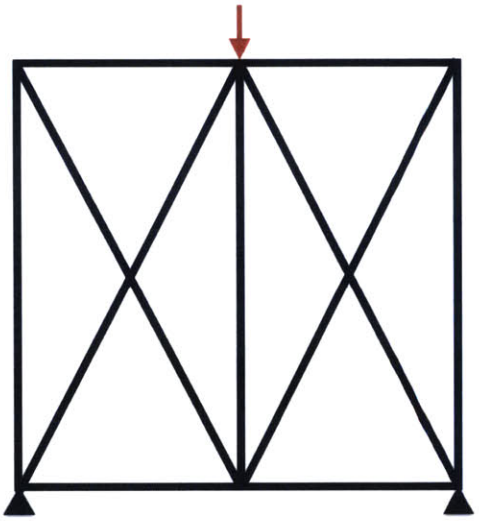

Figure 6-2: A conventional cantilever truss design for a midspan point load

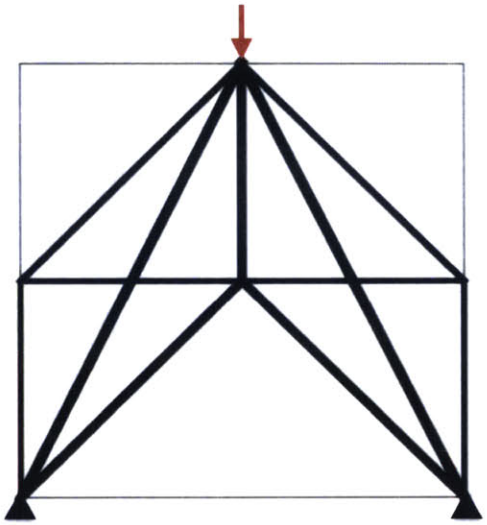

Figure 6-3: Solution output with 2^{nd} order redundancy + 45% less material

On the other hand, there can be outputs of the algorithm (especially for very large mesh refinement) that might not be very constructible (Figure 6-4). Nevertheless, the shape of the redundant optimal can drive the designer's inspiration of where the forces desire to go and how the alternative load paths work. This way, designers can incorporate these crucial notions in their designs early on.

6.3 Future Work – Alternative Approaches

There are several natural extensions this algorithm can take, but before proceeding to these, it should be noted that this is a primitive version of the computational procedure, and hence the computation is yet inefficient (given its large processing time). The immediate next steps for this project would be to optimize the code for

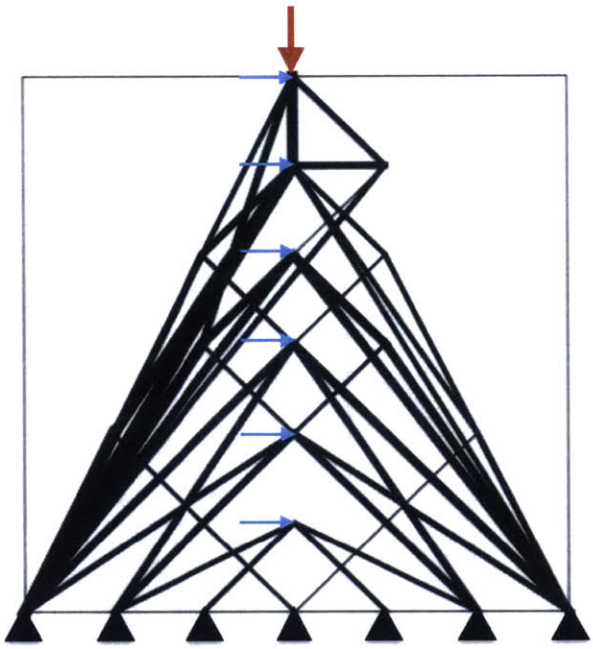

Figure 6-4: Algorithm output for 6×6 refined transmission tower

its computational time by implementing Matlab storage structures[1] and exploiting symmetry of the initial structures, to compute all the desired connectivity matrices faster. After polishing the code and making it more user-friendly, this code could be bundled as a software tool or an add-on, where the user could control the level of redundancy, the boundary conditions and the applied loads, in order to visualize the corresponding optimal solution. Zegard recently extended the GRAND script in 3 dimensions (Zegard et al., 2015), setting the benchmark for the natural extension of redundancy optimization very feasible (given that GRAND3 uses the same equations as GRAND). Following the $3D$ generalization of the redundant optimization, additional in depth work can be performed in Mixed-Integer Programming, allowing more viable solutions. Figure 5-13 raised the issue of nodal redundancy. Even though the member redundancy of the optimal simply supported beam was satisfactory, the nodal redundancy was minimal, meaning that the removal of just one node could result in global failure, even if we were able to remove 2 or 3 elements. Accounting for

[1]This process has already occurred for a large part of the current code

nodal redundancy is a generalization of element redundancy since removing a node implies the removal of all respective spanning elements. The first steps in developing a nodal redundancy optimization algorithm would closely follow the steps of element redundancy. For instance, starting off with a $n \times n$ node removal matrix $\mathbf{R}_j$ and multiplying it with A would give us the equilibrium equation of the substructure missing node j. Similarly, finding the relationship between the internal forces of the node-missing substructure and those of the original structure would allow us to proceed with the formulation of the problem. Extending this thesis' methodology could potentially provide a solution for the nodal redundancy problem and subsequently for the combined element & node redundancy problem.

Finally, even though redundancy is a good metric for additional safety, the future of structural optimization should account for both redundancy and robustness simultaneously. Hence, the general algorithm should be able to handle sets of load cases with given uncertainties instead of a single load case. All these steps are excellent paths for future development of the algorithm.

6.4 Concluding Remarks

Structural optimization has emerged from the need for more sustainable and energy-efficient structures. However, without reliability embedded inside the optimization process, it is unlikely for practitioners to inherit such practices. This thesis provides a *safer* approach to volumetric topology optimization. Through a novel mathematical algorithm we can optimize designs for ultimate material efficiency while adding layers of safety through member redundancy. This thesis covers the first algorithm that tackles least volume non-probabilistic redundant truss systems and it paves the beginning of promising answers on the broad research field of Redundant Topology Optimization.

Bibliography

P. Caston *The amazing Mathematical Bridge.* 5^{th} International Congress on Construction History, 2012

D.N. Ghista *Structural optimization with probability of failure constraint.* NASA TECHNICAL NOTE Vol. 3777, December 1966

A. G. Michell *The limits of economy of material in framed-structures.* Phil. Mag. S. Vol.68, pp.589-597, 1904

T. Zegard, G. Paulino *GRAND3 – Ground structure based topology optimization for arbitrary 3D domains using MATLAB.* Structural Multidisciplinary Optimization Vol.52, pp.1161-1184, 2015

D. Bertsekas *Nonlinear Programming.* Athena Scientific 2^{nd} Edition, 2005

O. Sigmund *A 99 line topology optimization code written in Matlab.* Structural Multidisciplinary Optimization Vol. 21, pp. 120 -127, 2001

P.P. Guilani, M Sharifi, S.T.A. Niaki, A. Zaretalab *Redundancy Allocation Problem of a System with Three-state Components: A Genetic Algorithm.* International Journal of Engineering Vol.27 No.11 pp.1663-1672, 2014

O. Sigmund, J. Petersson *Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima.* Journal of Structural Optimization Vol.16, pp.68-75, 1998

W. S. Hemp *Optimum Structures.* Oxford Engineering Science Series, 1973

R. M. Freund *Truss Design and Convex Optimization.* MIT Course notes, 2004

G. Strang *Linear Algebra and its Applications.* Thomson Brooks/Cole, Fourth Edition, 2006

Kanno Y., Ben-Haim Y. *Redundancy and Robustness, or When Is Redundancy Redundant?.* JOURNAL OF STRUCTURAL ENGINEERING ĺ' ASCE, Vol. 137 pp. 935-945, September 2011

Luo Y., Zhou M., Yu Wang M., Deng Z.*Reliability based topology optimization for continuum structures with local failure constraints.* COMPUTERS AND STRUCTURES Vol. 143, pp. 73-84, 2014

Okasha N., Frangopol D.M. *Lifetime-oriented multi-objective optimization of structural maintenance considering system reliability, redundancy and life cycle cost using GA.* STRUCTURAL SAFETY Vol. 31, pp. 460-474, 2009

Pereira J.T., Fancello E.A., Barcellos C.S. *Topology optimization of continuum structures with material failure constraints.* STRUCT MULTIDISC OPTIM Vol. 26, pp. 50-66, 2004

Safari J. *Multi-objective reliability optimization of series-parallel systems with a choice of redundancy strategies.* RELIABILITY ENGINEERING AND SYSTEM SAFETY Vol. 108, pp. 10-20, 2012

Save M., Guerlement G., D. Lamblin*On the safety of optimized structures.* STRUCTURAL OPTIMIZATION Vol. 1, pp. 113-116, 1989

Schafer B.W., Bajpai P. *Stability degradation and redundancy in damaged structures.* ENGINEERING STRUCTURES Vol. 27, pp. 1642-1651, 2005

Zegard T., Paulino G.H.*GRAND – Ground structure based topology optimization for arbitrary 2D domains using MATLAB.* Structural Multidisciplinary Optimization Vol. 50, pp. 861-882, 2014

J. Maxwell *On reciprocal diagrams in space and their relation to Airy's function of stress.* Proceedings of the London Mathematical Society Vol.2, pg.58-60, 1869

G. Appa *On the uniqueness of solutions to linear programs.* Journal of the Operational Research Society Vol.53, pp. 1127-1132, 2002

R. Kutylowski *On nonunique solutions in topology optimization.* Structural Multidisciplinary Optimization Vol.23, pp. 398-403, 2002

G. Rozvany *On symmetry and non-uniqueness in exact topology optimization.* Structural Multidisciplinary Optimization Vol.43, pp. 297 - 317, 2011

H. Kagiwada, R. Kalaba, N. Rosakhoo, K. Spingarn *Numerical Derivatives and Nonlinear Analysis.* SIAM Review Vol. 30, No.2 pg. 327-329, June 1988

A. Ben-tal, A. Nemirovski *Robust Truss Topology Design via Semidefinite Programming.* SIAM Journal of Optimization Vol.7-No.4, pp. 991-1016, 1997

D.P. Mohr, I. Stein, T. Matzies, C.A. Knapek *Robust Topology Optimization of Truss with regard to Volume.* arXiv:1109.3782v2 [math.OC], 30 April 2012

P.Pandey, S. Barai *Structural Sensitivity as a Measure of Redundancy* ASCE Journal of Structural Engineering Vol.123 Issue 3, pp. 360-364, 1997

D.M. Frangopol, J.P. Curley *Effects of Damage and Redundancy on Structural Reliability* ASCE Journal of Structural Engineering Vol.113 Issue 7, pp.1533-1549, 1987

C. Talischi, G. Paulino, A. Pereira, I.F.M. Menezes *PolyMesher: a general-purpose mesh generator for polygonal elements written in Matlab.* Structural Multidisciplinary Optimization Vol.45, pp. 309-328, 2012

R. Faturechi, E. Miller-Hooks *Measuring the Performance of Transportation Infrastructure Systems in Disasters: A Comprehensive Review.* Journal of Infrastructure Systems Vol.21, March 2015

T. Sokol, G.N. Rozvany *New analytical benchmarks for topology optimization and their implications. Part I: bi-symmetric trusses with two point loads between supports.* Structural Multidisciplinary Optimization Vol.46, pp. 477-486, 2012

A. Tyas, A. Pichugin, M. Gilbert *Optimum structure to carry a uniform load between pinned supports: exact analytical solution.* Proceedings of the Royal Society A Vol.467, 1101-1120, October 2010

Rippmann M. and Block P. *Funicular Shell Design Exploration.* Proceedings of the 33rd Annual Conference of the ACADIA, Waterloo/Buffalo/Nottingham, Canada, September 2013

L. An, Z. Wang, G. Wang, Z. Li *Design Optimization of Base Widths of Transmission Tower Using Mode-Pursuing Sampling Global Optimization Method.* International Conference on Computer Application and System Modeling (ICCASM 2010)

T. Hagishita, M. Ohsaki *Topology optimization of trusses by growing ground structure method.* Structural and Multidisciplinary Optimization Vol.47, pp. 377-393, 2009

Appendix A

Algorithm Shape Outputs

In this Appendix, several shapes are presented for different ground structures and different refinements, to showcase the breadth and variety of redundant shapes that exist for each boundary condition.

A.1 Cantilever

A.2 Bridge

A.3 Simply Supported Beam

A.4 Transmission Tower

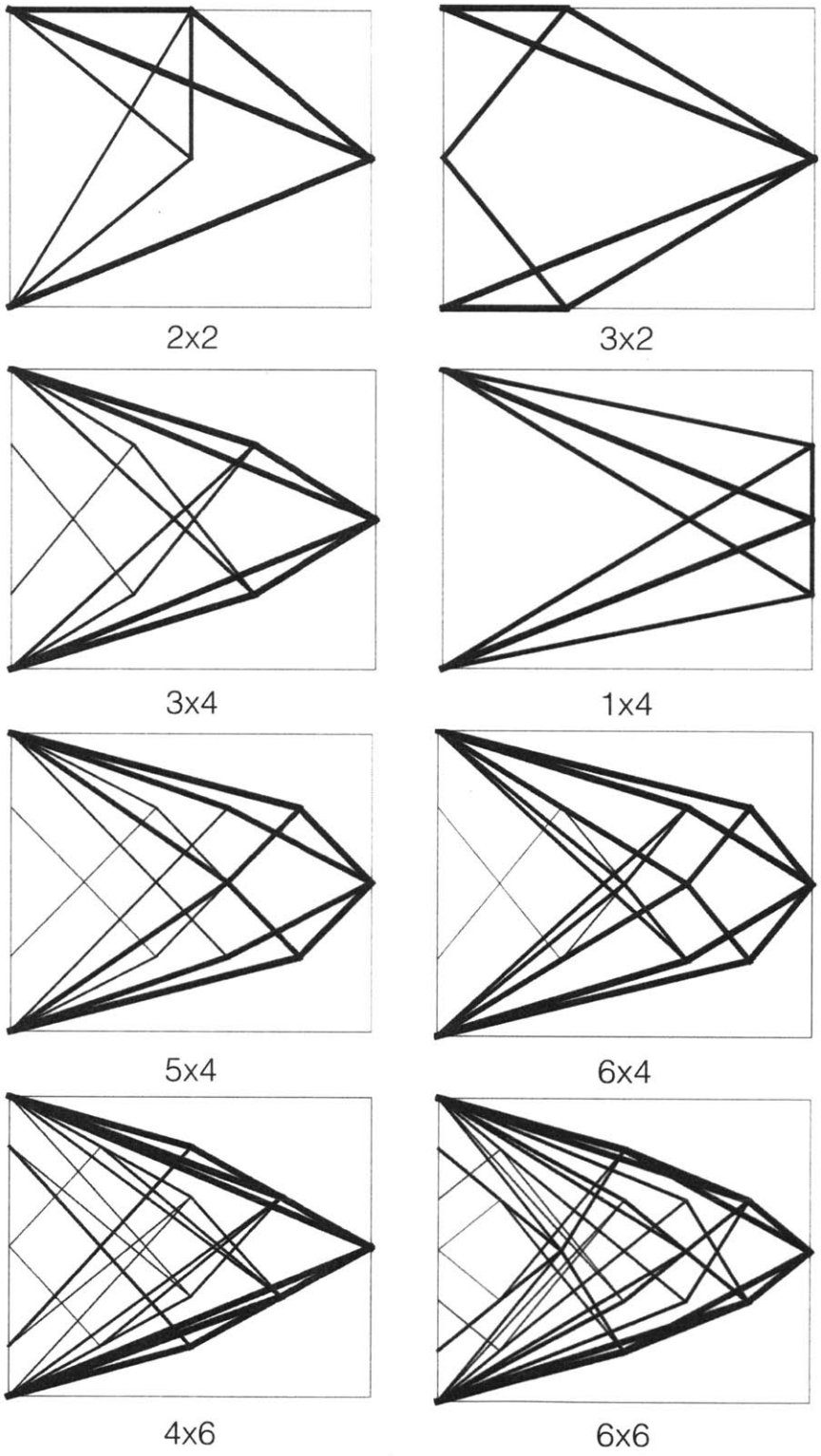

Figure A-1: Optimal redundant cantilever shapes w.r.t mesh refinement

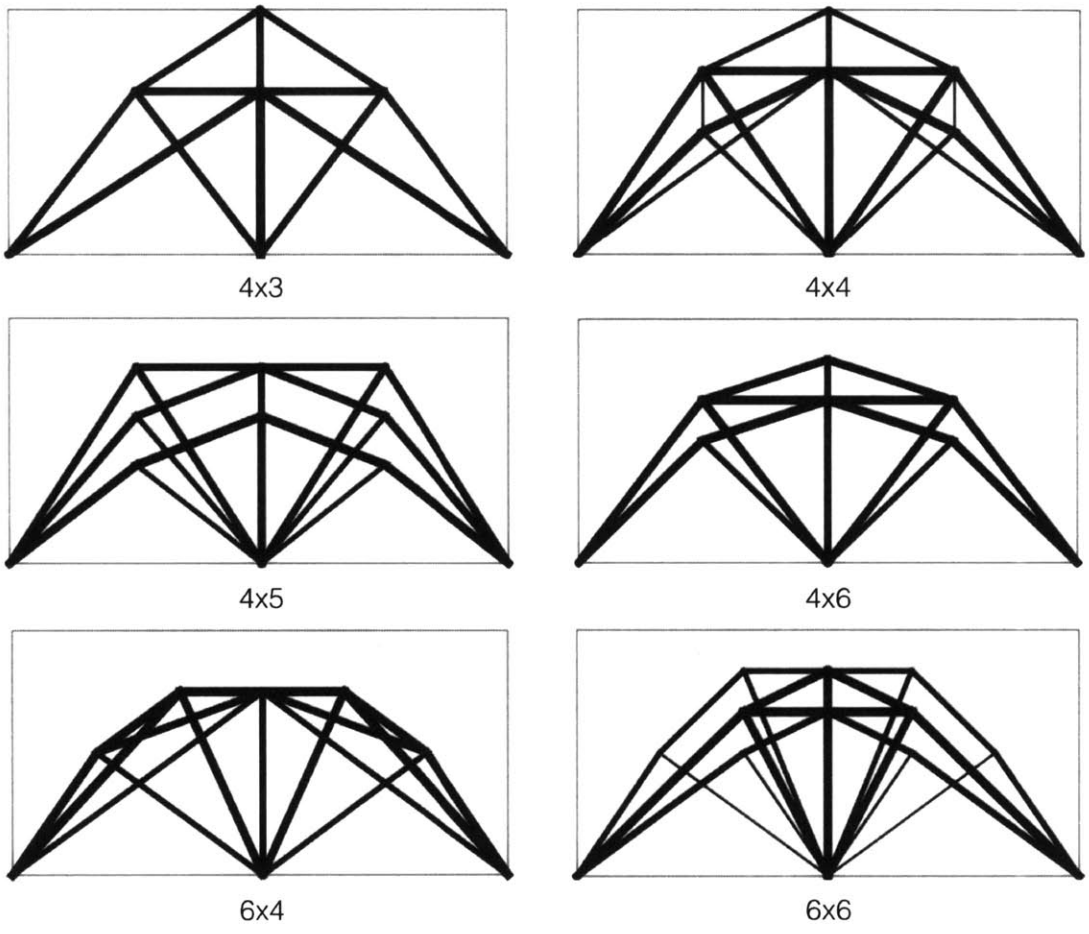

Figure A-2: Optimal redundant bridge shapes w.r.t mesh refinement

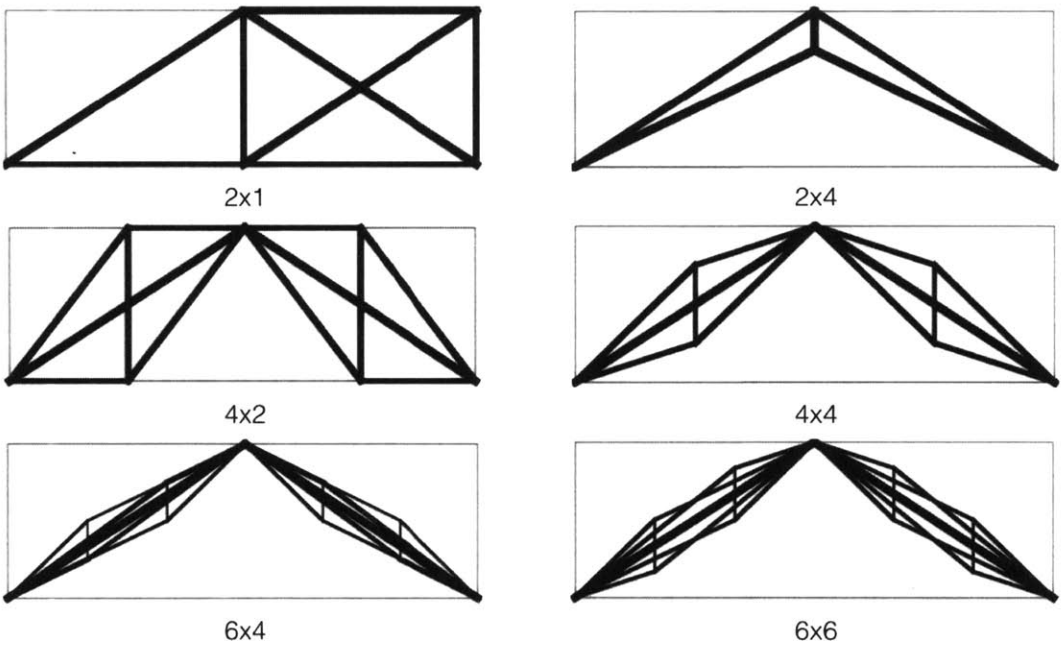

Figure A-3: Optimal redundant bridge shapes w.r.t mesh refinement

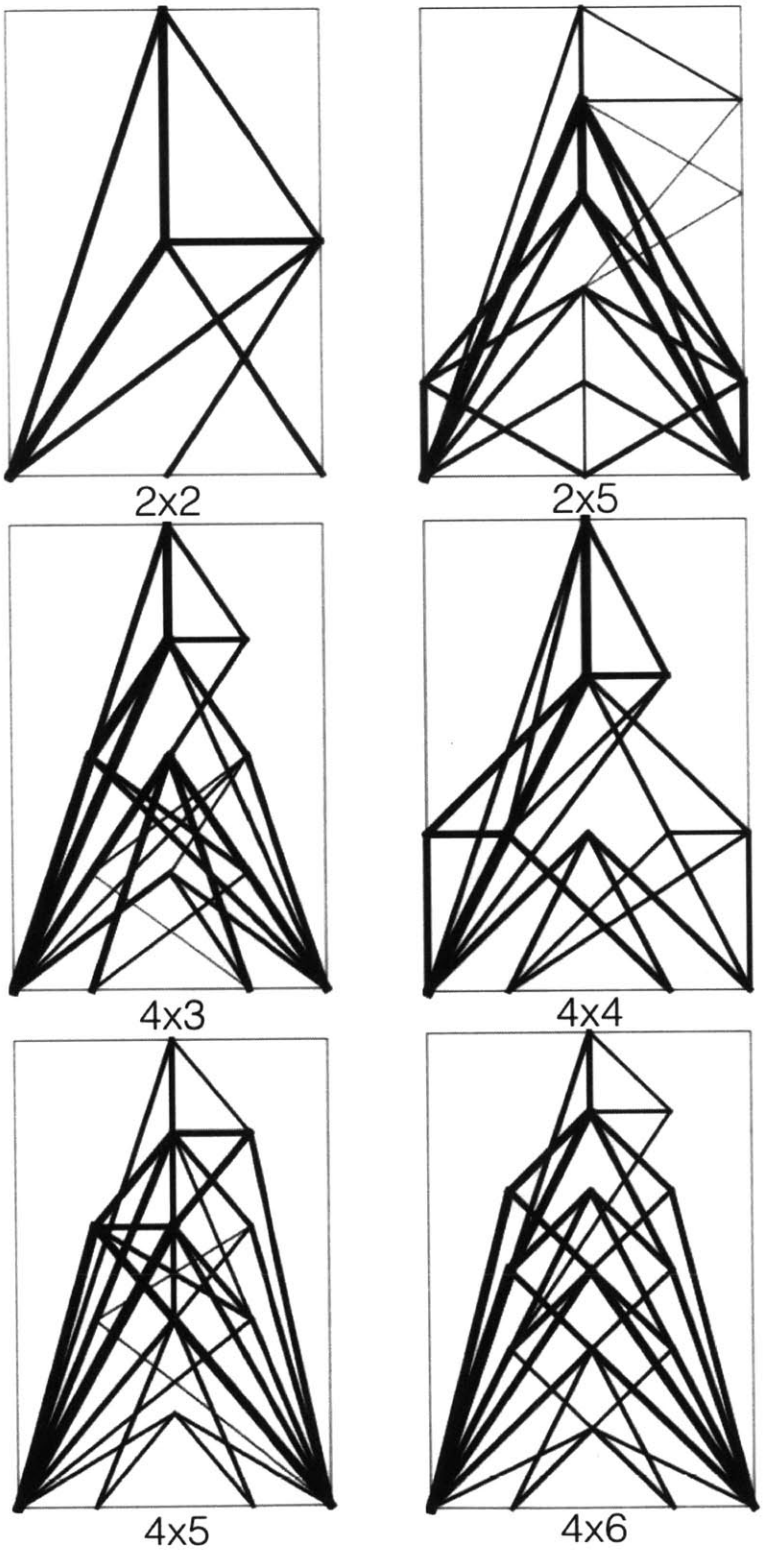

Figure A-4: Optimal redundant transmission towers w.r.t mesh refinement

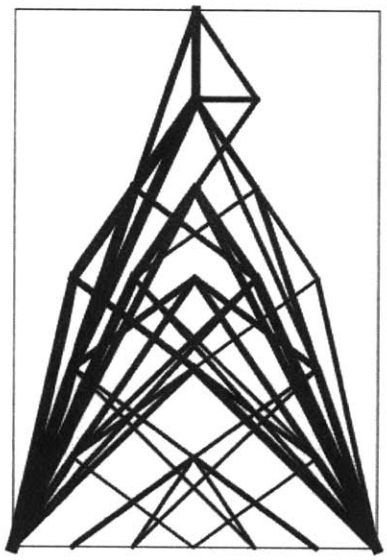

Figure A-5: Simply supported beam with height 30, width 20 & refinement 6×6

A.5 Free Form Examples

A.5.1 L-shape Beam

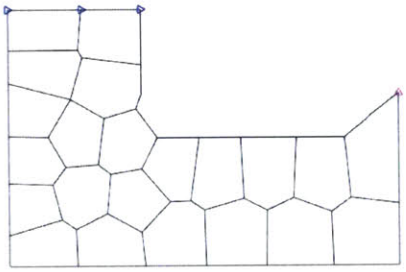

Figure A-6: Lshape polygon mesh for 22 polygons

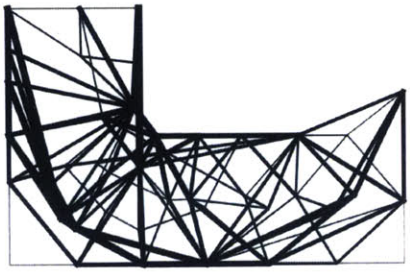

Figure A-7: Optimal L-shape for 22 polygon density

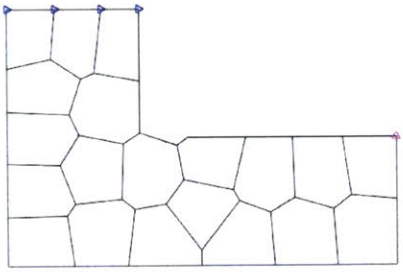

Figure A-8: Lshape polygon mesh for 26 polygons

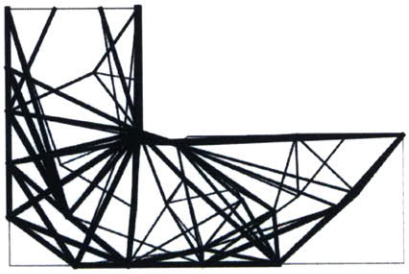

Figure A-9: Optimal L-shape for 26 polygon density

A.5.2 Ring

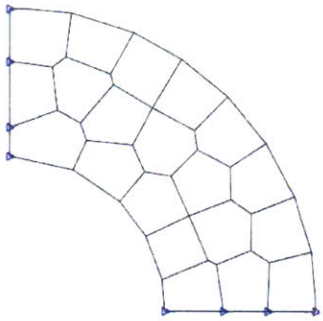

Figure A-10: Lshape base mesh for 20 polygons

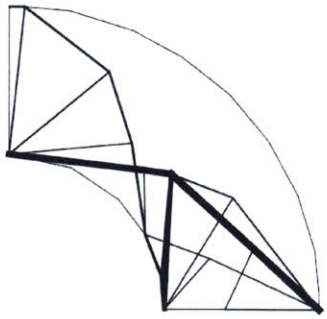

Figure A-11: Optimal L-shape for 20 polygon density

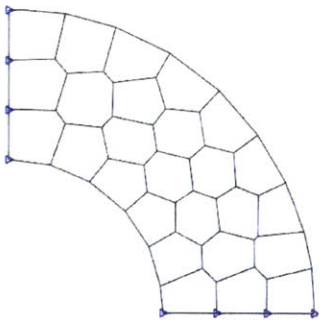

Figure A-12: L-shape base mesh for 26 polygons

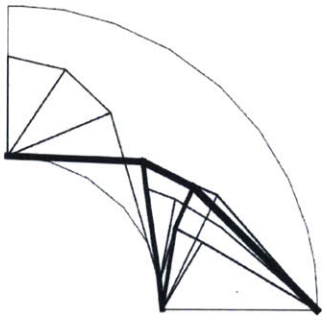

Figure A-13: Optimal L-shape for 26 polygon density

A.5.3 Redundant Wrench

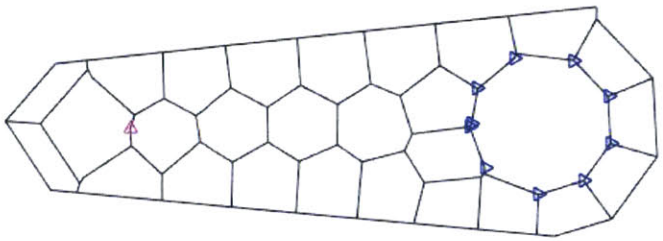

Figure A-14: Wrench base mesh with 20 polygons

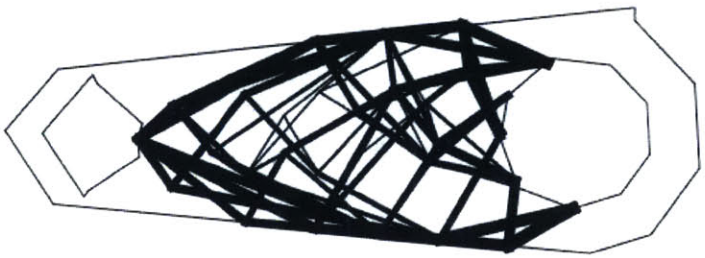

Figure A-15: Optimal Wrench for 20 polygon density

A.5.4 Serpentine

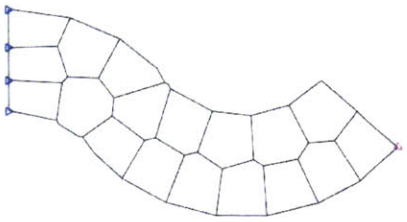

Figure A-16: Base serpentine mesh for 20 polygons

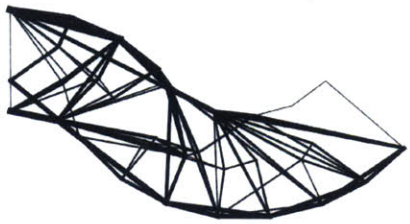

Figure A-17: Optimal serpentine for 20 polygon density.

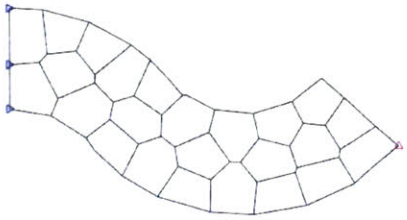

Figure A-18: Base serpentine mesh for 24 polygons

Figure A-19: Optimal serpentine for 24 polygon density

A.5.5 Redundant Hook

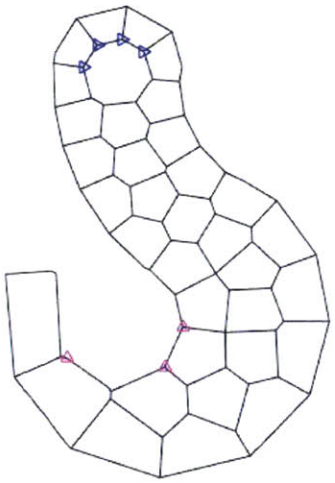

Figure A-20: Hook base mesh with 20 polygons

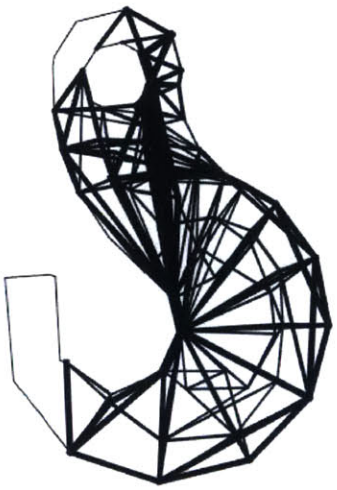

Figure A-21: Optimal Hook for 20 polygon density

Other examples that were tested from Paulino's Polymesher(Talischi et al., 2012) environment and not presented in this thesis are: Flower Domain, Suspension Domain, Michell Domain and Triangular Domains. Note these domains can still be tested by the redundant optimization algorithm attached in Appendix B

Appendix B

Matlab Code

This appendix has attached the code for the redundancy topology optimization. Note that for the sake of keeping this section brief, we have not attached the Ground Structure generation and all parts of GRAND's code that were not modified or minimally modified. In addition, we did not attach the linear integer programming file as the only thing that changes from the first file is the command *linprog* and the relaxing of the constraint (from equality to inequality). The reader is encouraged to look into the code themselves and try some examples of their own. In this Appendix, the code for orthogonal domain generation is attached, the Level 1 Redundancy Code, the meta code that loops through the optimizations for the Design Space creation, and the higher level redundancy file along its objective function file.

B.1 Orthogonal Structural Domains

morekeywords

```
function [NODE,ELEM,SUPP,LOAD]=StructDomain(Nx,Ny,Lx,Ly,ProblemID)
% Generate structured-orthogonal domains
[X,Y] = meshgrid(linspace(0,Lx,Nx+1),linspace(0,Ly,Ny+1));
NODE = [reshape(X,numel(X),1) reshape(Y,numel(Y),1)];
k = 0; ELEM = cell(Nx*Ny,1);
for j=1:Ny, for i=1:Nx
        k = k+1;
        n1 = (i-1)*(Ny+1)+j; n2 = i*(Ny+1)+j;
        ELEM{k} = [n1 n2 n2+1 n1+1];
end, end

if (nargin==4 || isempty(ProblemID)), ProblemID = 1; end
switch ProblemID
```

```
    case {'Cantilever','cantilever',1} % Cantilever construction...
    %of supports and loads
        SUPP = [(1:Ny+1)' ones(Ny+1,2)];
        LOAD = [Nx*(Ny+1)+round((Ny+1)/2) 0 -1];
    case {'MBB','Mbb','mbb',2} % Construction of Moment Beam
        SUPP = [Nx*(Ny+1)+1 NaN 1;
                (1:Ny+1)' ones(Ny+1,1) nan(Ny+1,1)];
        LOAD = [Ny+1 0 -0.5];
    case {'Bridge','bridge',3} %Bridge with midspan point load...
    %(can be generalized for distributed loads)
        SUPP = [      1       1 1;
                Nx*(Ny+1)+1 1 1];
        LOAD = [(Ny+1)*round(Nx/2)+1 0 -1];
    case {'SSB','ssb',4} % Simply Supported Beam construction
        SUPP = [      1       1 1;
                Nx*(Ny+1)+1 1 1];
        LOAD = [(Ny+1)*(round(Nx/2)+1) 0 -1];
    case {'Trans','trans',5} % Transmission Tower Construction
        SUPP = [(1:Ny+1:Nx*Ny+Nx+1)' ones(Nx+1,1) ones(Nx+1,1)];
        LOAD = [((Ny+1)*round(Nx/2)+2:(Ny+1)*(round(Nx/2)+1))'...
          (2:Ny+1)' nan(Ny,1); (Ny+1)*(round(Nx/2)+1) 0 Nx];
      otherwise % The reader can test out any other load case
        SUPP = []; LOAD = [];
        disp('-INFO- Structured domain generated with no loads/BC')
end
```

B.2 1^{st} Order Redundancy Code

morekeywords

```
%Level 1 Redundancy
%% ===== MESH GENERATION LOADS/BCS ======================
kappa = 1.0; ColTol = 0.999999;
Cutoff = 0.002; Ng = 50;

 a = 35; % Input either horizontal mesh refinement or number...
 % of polygons in mesh generation
 b = 20;% Input vertical mesh refinement (Note that one of the...
 % options should only be uncommented
 % --- OPTION 1: POLYMESHER MESH GENERATION ---------
addpath('./PolyMesher')
[NODE,ELEM,SUPP,LOAD] = PolyMesher(@HookDomain,a,b);
Lvl = 5; RestrictDomain = @RestrictHook;
rmpath('./PolyMesher')

% % % --- OPTION 2: STRUCTURED-ORTHOGONAL MESH GENERATION ---------
 [NODE,ELEM,SUPP,LOAD] = StructDomain(a,b,20,30,'Cantilever');
 % any of the defined cases created in previous section go here
 Lvl = 6; RestrictDomain = [];
 Nx,Ny,Lx,Ly

% --- OPTION 3: LOAD EXTERNALLY GENERATED MESH ----------
 load MeshHook
 Lvl = 4; RestrictDomain = @RestrictHook;

 load MeshSerpentine
 Lvl = 5; RestrictDomain = @RestrictSerpentine;

```

```
load MeshMichell
Lvl = 6; RestrictDomain = @RestrictMichell;

load MeshFlower
Lvl = 4; RestrictDomain = @RestrictFlower;

 load MeshLshape
 Lvl = 6; RestrictDomain = @RestrictLShape;

PlotPolyMesh(NODE,ELEM,SUPP,LOAD);
[BARS] = GenerateGS(NODE,ELEM,Lvl,RestrictDomain,ColTol);
Nn = size(NODE,1);
Ne = length(ELEM);
Nb = size(BARS,1)
[BC] = GetSupports(SUPP);
[BT,L] = GetMatrixBT(NODE,BARS,BC,Nn,Nb);
[F] = GetVectorF(LOAD,BC,Nn);
B=BT; % geometry matrix
fprintf('Mesh: Elements d, Nodes d, Bars d, Level d\n',...
Ne,Nn,Nb,Lvl);
BTBT = [BT -BT];
LL = [L; kappa*L];
clear BT L

V = null(full(BT)); % find the kernel of the geometry matrix

V=licols(V); % find linearly independent columns;
Vi=sum(V,2); % sum the columns to guarantee that none of the...
% entries are zero

whos('V');
whos('BTBT');

Const = zeros(size(BT,1),Nb,Nb); % initialize 3D array of ARiKi's
later = zeros(Nb,Nb,Nb); % 3D array of RiKi's
%%
r = 1; % number of desired elements to be removed -- ...
%for Lv 1 Redundancy keep this 1
remov = nchoosek(1:1:Nb,r);
Si = repmat(eye(Nb),[1 1 size(remov,1)]);
for i = 1:size(remov,1) %Create all the Ri's...
% and store in 3D structure
    for j = 1:size(remov,2)
        Si(remov(i,j),remov(i,j),i)=0;
    end
end


ei = eye(Nb);
for i=1:Nb %fill in the 3D arrays
    later(:,:,i)=(ei-(1/Vi(i,1))*Vi*ei(:,i)');
    Const(:,:,i)=BT*Si(:,:,i)*(ei-(1/Vi(i,1))*Vi*ei(:,i)');
end

Const2 = reshape_(Const);
Fcat=repmat(F,[Nb 1]);
[Const3,idx] = licols(Const2',1e-10);
Const4 = Const3';
Fcat2 = Fcat(idx,:);
MConst = [Const4 -Const4]; % duplicated Ki matrix

```

```
[Sin,volin,exitflagin] = linprog(LL,[],[],BTBT,F,zeros(2*Nb,1));
%solve pure volumetric optimization problem
Sin = (reshape(Sin,numel(Sin)/2,2));
Aopt = Sin(:,1) + kappa*Sin(:,2);

[row,col] = find(Sin>1e-5); % store critical elements
finarea = zeros(Nb,size(row,1)); % initialize final area
looping = row(1:ceil(end/2))';
% cut the critical elements by half due to symmetry
 for el = looping

diagKi = [later(:,:,el) zeros(size(later,1)); ....
zeros(size(later,1)) later(:,:,el)];
tic, [S,vol,exitflag] = linprog(abs(LL'*(diagKi)),[],[],...
sparse(full(MConst)),sparse(full(Fcat2)),...
zeros(2*Nb,1)); % redundancy optimization

S = reshape(S,numel(S)/2,2); % internal forces
A = S(:,1) + S(:,2); % reshaped internal forces
finarea(:,el) = A;
end

 total = max(abs(finarea), [], 2);
 %pick the maximum force of each element in every case
 finvol =   L'*total %redundant load path

PlotGroundStructureBW(NODE,BARS,total,Cutoff,Ng)
PlotBoundary(ELEM,NODE) %plot the solution
```

B.3 Design Space Exploration

This section contains the design space creation of the 1^{st} order redundancy optimization in order to allocate the memory better, since no plotting is required (only the final volume of each redundant optimal structure). Two files are included below, RedundantVolume.m and MetaDS.m:

morekeywords

```
function [finvol] = RedundantVolume(ah,bh)

% storing each orthogonal domain's volume
kappa = 1.0; ColTol = 0.999999;
Cutoff = 0.002; Ng = 50;

sto = zeros(ah,bh); % initialize matrix with volumes

[NODE,ELEM,SUPP,LOAD] = StructDomain(ah,bh,10,5,'Bridge');
Lvl = 6; RestrictDomain = [];

[BARS] = GenerateGS(NODE,ELEM,Lvl,RestrictDomain,ColTol);
Nn = size(NODE,1);
Ne = length(ELEM);
Nb = size(BARS,1)
[BC] = GetSupports(SUPP);
[BT,L] = GetMatrixBT(NODE,BARS,BC,Nn,Nb);
```

```
[F] = GetVectorF(LOAD,BC,Nn);
B=BT;
BTBT = [BT -BT];
LL = [L; kappa*L];
clear BT L

V = null(full(BT));

V=licols(V); % find linearly independent columns;
Vi=sum(V,2);


r = 1; % number of desired elements to be removed
remov = nchoosek(1:1:Nb,r); % list of 1 to nchoosek(Nb,r)

Si = struct();
% implement Matlab structures for better memory allocation
for i = 1:size(remov,1)
    b = ones(Nb,1);
    re = remov(i); % obtain row of numbers to remove
    b(re) = 0;
    a = diag(b);
    a = sparse(a);
    Si = setfield(Si,strcat('a',num2str(i)),a);
end
clear i;
SiNames = fieldnames(Si);
%%
ei = eye(Nb);
names = {};
for i=1:Nb
    names{end+1} = strcat('a',num2str(i));
end
for i =  1:Nb
    SL.(names{i}) = [];
    SC.(names{i}) = [];
end
for loopIndex = 1:Nb
    dummy_lat = (ei-(1/Vi(loopIndex,1))*Vi*ei(:,loopIndex)');
    SL.(names{loopIndex}) = sparse(dummy_lat);
end
%%
ei = eye(Nb);
for loopIndex = 1:Nb
    getSi = Si.(SiNames{loopIndex});
    dummy_con = BT*getSi*(ei-(1/Vi(loopIndex,1))*...
    Vi*ei(:,loopIndex)');
    SC.(names{loopIndex}) = sparse(dummy_con);
end
%%
conNames = fieldnames(SC);
Fcat=repmat(F,[Nb 1]);
conNames = fieldnames(SC);
Const2 = SC.(conNames{1});
los = size(Const2);
S = spalloc(los(1)*Nb,los(2),los(2)*los(1));
for i = 1:los(1):los(1)*Nb

    if i == 1
        j = 1;
    else
```

```
79          j = round(i/los(1)) + 1;
80      end
81      Const2 = SC.(conNames{j});
82      S(i:i+los(1)-1,:) = Const2;
83  end
84  %%
85   filename = strcat('sparseS',...
86   num2str(ah),num2str(bh)); % save matrix file
87   save(filename,'S');
88  %%
89  S = load('S1010.mat'); % load file in case of recovering
90  %%
91  Fcat=repmat(F,[Nb 1]);
92  transpS = S';
93  size(transpS,2)
94  for i = 1:10000:size(transpS,2)
95      if size(transpS,2) > (i+9999)
96          rel = full(transpS(:,i:i+9999));
97      else
98          rel = full(transpS(:,i:end));
99      end
100     [transpConst,idx] = licols(rel,1e-10);
101     if i == 1
102         MtranspConst = [transpConst];
103         Minx = [idx];
104     else
105         MtranspConst = [MtranspConst, transpConst];
106         Minx = [Minx,((i-1)+idx)];
107     end
108     clear rel transpConst idx
109 end
110 %%
111 [transpConstAll,idxAll] = licols(MtranspConst,1e-10);
112 b = Minx(idxAll);
113 ConstAll = transpConstAll';
114 Fcat2 = Fcat(b,:);
115 FinalConst = [ConstAll -ConstAll];
116 %%
117 [Sin,volin,exitflagin] = linprog(LL,[],[],BTBT,F,zeros(2*Nb,1));
118 Sin = (reshape(Sin,numel(Sin)/2,2));
119 Aopt = Sin(:,1) + kappa*Sin(:,2);
120 [row,col] = find(Sin>1e-5);
121 finarea = zeros(Nb,size(row,1));
122  looping = row(1:ceil(end/2))';
123 for el = looping
124     lat_i = full(SL.(conNames{el}));
125     diagKi = [lat_i zeros(size(lat_i,1));...
126     zeros(size(lat_i,1)) lat_i];
127     tic, [S,vol,exitflag] = linprog(abs(LL'*(diagKi)),[],[],...
128     sparse(full(FinalConst)),sparse(full(Fcat2)),zeros(2*Nb,1));
129     %redundancy optimization for each critical element
130     S = reshape(S,numel(S)/2,2);
131     A = S(:,1) + S(:,2);
132     finarea(:,el) = A;
133 end
134 %%
135 total = max(abs(finarea), [], 2);
136 finvol = L'*total;
137 clear NODE ELEM SUPP LOAD BARS Nn Ne Nb BC BT L F Si SL SC names S;
138 end
```

metaDS.m

morekeywords

```
maxah = 10; % Maximum Horizontal Grid Refinement
maxbh = 10; % Maximum Vertical Grid Refinement
sto = zeros(maxah,maxbh);
for ah = 1:maxah
    for bh = 1:1:maxbh
        finvol = RedundantVolume(ah,bh);
        sto(ah,bh) = finvol;
        filename = strcat('only','Finvol',num2str(ah),'by',...
        num2str(bh));
        save(filename,'finvol');
    end
end
surf(sto)
title('Redundant Bridge GS Refinement')
xlabel('Horizontal Grid Refinement') % x-axis label
ylabel('Vertical Grid Refinement') % y-axis label
zlabel('Structural Volume')

%% In case of loading the matrix values later use the code below
 t1 = load('total1by1.mat');
 t2 = load('total2by2.mat');
 t3 = load('total3by3.mat');
 t4 = load('total4by4.mat');
 t5 = load('total5by5.mat');
 t6 = load('total6by6.mat');
 t7 = load('total7by7.mat');
 t8 = load('total8by8.mat');
 t1 = t1.total;
 t2=t2.total;
 t3=t3.total;
 t4=t4.total;
 t5=t5.total;
 t6=t6.total;
 t7=t7.total;
 t8=t8.total;
 % Plot # of elements vs. volume
 n(1) = numel(find(t1 > 1e-5));
 n(2) = numel(find(t2 > 1e-5));
 n(3) = numel(find(t3 > 1e-5));
 n(4) = numel(find(t4 > 1e-5));
 n(5) = numel(find(t5 > 1e-5));
 n(6) = numel(find(t6 > 1e-5));
 n(7) = numel(find(t7 > 1e-5));
 n(8) = numel(find(t8 > 1e-5));
 y = diag(sto(1:ah,1:bh));
 figure(1)
 plot(n,y,'b-','linewidth',2)
 title('Volumetric Optimal vs. Element #')
 xlabel('Number of Elements') % x-axis label
 ylabel('Optimal Volume') % y-axis label
```

B.4 Higher Order Redundancy Code

In this section the nonlinear redundancy optimization is shown. The main file implements *fmincon* in Matlab to solve the nonlinear equality constraint problem and the secondary file includes the objective function along the corresponding gradient.

HigherRedundancy.m

morekeywords

```
% Higher Redundancy
kappa = 1.0; ColTol = 0.999999;
Cutoff = 0.002; Ng = 50;
global Si Vi F remov B L first;
% define global variables used in objective function file
[NODE,ELEM,SUPP,LOAD] = StructDomain(2,2,10,10,'Cantilever');
Lvl = 6; RestrictDomain = [];
PlotPolyMesh(NODE,ELEM,SUPP,LOAD)
[BARS] = GenerateGS(NODE,ELEM,Lvl,RestrictDomain,ColTol);
Nn = size(NODE,1);
Ne = length(ELEM);
Nb = size(BARS,1);
[BC] = GetSupports(SUPP);
[BT,L] = GetMatrixBT(NODE,BARS,BC,Nn,Nb);
[F] = GetVectorF(LOAD,BC,Nn);
B=BT;
BTBT = [BT -BT];
LL = [L; kappa*L];
clear BT L
V = null(full(BT));
V=licols(V); % find linearly independent columns;
Vi=sum(V,2);
Const = zeros(size(BT,1),Nb,Nb);
later = zeros(Nb,Nb,Nb);
%%
r = 2; %number of desired elements to be removed (Lvl 2 redundancy)
remov = nchoosek(1:1:Nb,r);
Si = repmat(eye(Nb),[1 1 size(remov,1)]);
for i = 1:size(remov,1)
    for j = 1:size(remov,2)
        Si(remov(i,j),remov(i,j),i)=0;
    end
end
[Sin,volin,exitflagin] = ...
linprog(LL,[],[],BTBT,F,zeros(2*Nb,1));
% original linear program to identify critical elements
Sin = (reshape(Sin,numel(Sin)/2,2));
Aopt = Sin(:,1) + kappa*Sin(:,2);
 [row,col] = find(Sin>1e-5);
 finarea = zeros(Nb,size(row,1));
%%
for first = row'
x0 = ones(Nb,1);
options = optimoptions('fmincon','GradObj','on');
[S,fval,exitflag] = fmincon(@objfivec,x0,[],[],[],[],-inf,inf,...
@eqconstraint,options);
A = abs(S);
finarea(:,first) = A;
end
```

```
%%
   total = max(abs(finarea), [], 2);
   finvol =    L'*total;
    PlotGroundStructure(NODE,BARS,total,Cutoff,Ng)
    PlotBoundary(ELEM,NODE) % plot solution
```

Objective function objfivec.m

```
morekeywords
   function [fun] = objfivec(var)%, gradf]
%the objective function for the nonlinear problem
% var the internal force variable
global Si Vi remov L first; %global variables
,Si,L,v,remov
 Si = getGlobalSi;
 L = getGlobalL;
 v = getGlobalVi;
 remov = getGlobalremov;
 e = eye(size(Si,1));
%% objective function
fun = abs(L'*(var - 1/(e(:,remov(first,1))'*Vi*...
e(:,remov(first,2))'*Vi)*(e(:,remov(first,1))'*...
var*e(:,remov(first,2))'*var)*Vi));
% gradient
 gradf = L' - L'*Vi*var'*(e(:,remov(first,1))*...
 e(:,remov(first,2))'+e(:,remov(first,2))*e(:,remov(first,1))');
end
```