PHD BY PUBLICATION

Topology optimisation of structures exposed to large (non-linear) deformations

Christensen, Jesper

Award date:
2015

Awarding institution:
Coventry University

Topology Optimisation of Structures Exposed to Large (non-linear) Deformations

Jesper Christensen

September 2015

A thesis submitted in partial fulfilment of Coventry University's requirements for the Degree of Doctor of Philosophy

This page is intentionally left blank.

Acknowledgements

I would like to thank my academic advisors Dr. Christophe Bastien and Professor Mike V. Blundell for their commitment, support, patience, expertise and interest in my work; for which I am very grateful.

I would like to thank my wife Rachael, my parents Børge and Jane, my brother Lars, his wife Anja as well as my parents-in-law Kim and Jane for all of their support, patience and understanding before and during the development of my work.

I would like to express my gratitude to the Faculty of Engineering and Computing at Coventry University and the many members of staff who have taken an interest and help support my work.

Finally I would like to thank all contributors to the Low Carbon Vehicle Technology Project as well as the Towards Affordable Recyclable Future Low Carbon Vehicle (TARF-LCV) for providing info and background information and opportunities to promote and develop my research activities.

This page is intentionally left blank.

Table of Contents

This page is intentionally left blank.

Critical Review Document

This page is intentionally left blank.

Acronyms

BC	Boundary Condition
BEETS	Bi-directional Evolutionary Entropy Tabu search Simulated annealing
BESO	Bi-directional Evolutionary Structural Optimisation
BIW	Body In White
ESLM	Equivalent Static Load Method
FE / FEM / FEA	Finite Element / FE Model /FE Analysis
HEV	Hybrid Electric Vehicle
SIMP	Solid Isotropic Material with Penalisation
VDM	Variable Density Method

Table of Contents

1. Introduction

The purpose of this document is to describe the context for the overall PhD portfolio submitted, which has been developed over a 5 year period. Before the details are discussed it is important to list the proposed portfolio title as well as the aims and objectives of the research:

Proposed title for the PhD by Portfolio

Topology optimisation of structures exposed to large (non-linear) deformations.

Research aim

- To investigate if topology optimisation can be used for the development of mechanical structures exposed to large (non-linear) deformations.

Research objectives

1. Analyse and critically evaluate the potential for using state of the art commercially available Finite Element software (and associated topology optimisation algorithms) for topology optimisation of structures exposed to large-deformations.
2. Based on 1 (where feasible) suggest, develop and critically appraise opportunities, methodologies and tools for enhancing the accuracy and precision of current state of the art topology optimisation algorithms for non-linear applications.
3. Based on the outcomes of 1 and 2 define / refine and integrate a topology optimisation algorithm / methodology with enhanced levels of accuracy for structures exposed to large (non-linear) deformations.
4. Critically analyse and assess the outcomes of the tool developed in 3 to competing algorithms and “sound engineering judgement” using case-studies and objectively evaluate the potential for further development/refinement of the proposed algorithm / methodology.

2. Autobiographical Context for the Portfolio of Evidence

The PhD candidate of this Portfolio commenced his Bachelor of Science (BSc) studies in Industrial Engineering at Aalborg University, Denmark, in 2004 and was awarded the BSc in June of 2007. Following this, the candidate enrolled on a Master of Science (MSc) degree at Aalborg University in September of 2007 and graduating in January 2010 specialising in Design of Mechanical Systems. During the last 18 months of study the candidate attended 4 modules at Coventry University (CU), which reinforced links with the automotive industry, culminating with a 6 month placement at Jaguar Land Rover (JLR) where the candidate investigated the effects of seat foam material properties upon automotive crash safety. Following this the candidate completed his individual thesis (6 month full time) investigating the potential for modelling airbag tear seams through plate theory. The combination of these 3 events coupled with previous courses of optimization principles, Finite Element (FE) modelling and Structured Programming (SP) inspired a deep interest in the development and application of structural optimisation algorithms to non-linear (large deformation) problems. In this context the applicant successfully applied for a Research Associate (RA) role at Coventry University 2 months after graduating. The full-time RA role was dedicated to the Low Carbon Vehicle Technology Project (LCVTP), (LCVTP, nd) a multi-partner project funded by the European Regional Development Fund (ERDF), Advantage West Midlands (AWM) and a number of industrial partners including Tata Motors European Technology Centre and JLR. The candidate worked full time on this project for approximately 12 months specifically focussing on lightweighting of automotive crash structures. The outcome of this work, documented by contribution 1 to this portfolio see section 1, could in essence be considered to be the full time equivalent of the first year of a “conventional” PhD candidate. Based on the achievements of the LCVTP work the candidate was promoted to a permanent lecturing position at CU, with the title of “lecturer in Stress Analysis. Although this meant that the candidate would no longer be able to continue his research on a full time basis the development of the PhD portfolio did continue as the teaching activities were in relevant and associated fields; this ultimately led to the publications of contributions 2, 3 and 4 of this portfolio in October 2011, May 2013 and December 2013 respectively. Due to an increase in and variation of responsibilities of the candidate the work of the portfolio did continue at varying intensity, as the candidate was firstly promoted to the role of Senior Lecturer in Stress Analysis and subsequently to Senior Research Fellow. Although the candidate has maintained and developed a continuous portfolio of research relevant to this PhD portfolio throughout his career there has also been emphasis

on other areas of research and development; a natural consequence of being an early career academic. The candidate has however made specific targeted efforts to maintain focus on his true passion; the development and implementation of structural optimisation algorithms. The most noteworthy achievements being presentations in the USA, Germany, at the House of Commons in Westminster and a publishing contract with Elsevier, scheduled for publication in December 2015.

3. Chronological Description of Portfolio Development

As indicated in section 2, the portfolio has been developed over a period of approximately 5 years. The portfolio contains a total of 6 contributions:

1. One html document containing models and reports from an ERDF / AWM / Industrial funded project entitled LCVTP.
2. One double-blind peer reviewed journal paper entitled "*Effects of roof crush loading scenario upon body in white using topology optimisation"* published in the International Journal of Crashworthiness in October 2011 (Christensen et. al., 2011).
3. One double-blind peer reviewed journal paper entitled "*Buckling Considerations and Cross-*
   Sectional Geometry Development for Topology Optimised Body In White" published in the International Journal of Crashworthiness in May 2013 (Christensen et. al., 2013).
4. One double-blind peer reviewed journal paper entitled "*Feasibility of ESLM for BIW Roof Structure Development and Optimisation*" published in the Journal of Mathematical Research and Applications in December 2013 (Christensen et. al., 2013a).
5. One monologue presenting the definition, programming and initial case studies of a non-linear FE based topology optimisation algorithm entitled Bi-directional Evolutionary Entropy Tabu search Simulated annealing (BEETS).
6. One software program, BEETS, for non-linear FE based topology optimisation as documented by contribution 5.

The following sub-sections will detail the chronological description of the 6 contributions to the portfolio.

3.1 Contribution 1

The starting point was the LCVTP project; wherein the work was associated with Work-Package (WP) 7: lightweighting. The LCVTP project was extensive and contained a total of 15 WPs under simultaneous development; with an overall project objective of mapping out technical directions for future low carbon vehicles. Although the project did contain a demonstrator vehicle; a converted Jaguar XJ, the focus of the project was predominately on the development of techniques, tools and methodologies enabling future low carbon vehicles. This was to be achieved through a wealth of activities the most relevant ones in this context are materials, aerodynamic performance, powertrain selection and development in addition to lightweight

design (WP 7). Due to the overall purpose of the project it was necessary to monitor and consider the project advances in the fields of material development, aerodynamic performance and powertrain selection as all of these would of course have an effect upon the lightweight design. As WP 7 was scheduled to run concurrently with the other aforementioned WPs it was evident that some fundamental assumptions had to be made in order to commence WP7. The ultimate scope of the PhD candidates' work within WP7 was to:

"Define an optimisation strategy / methodology for developing lightweight automotive vehicle structures capable of meeting current (2010) and future crash performance standards"

Based on discussions with relevant project partners and WP owners the following list of assumptions was made:

1. The material properties were assumed to be isotropic.
2. Two competing drive train scenarios were considered:
    a. A fully electric vehicle.
    b. A range extended electric vehicle.
3. A case study would be based around the Tata Beacon concept vehicle.

The work rapidly started to focus on topology optimisation technologies as it would be an ideal starting point for creating a truly lightweight concept vehicle. The initial idea was to investigate and critically analyse the opportunities for utilising existing (2010) state-of-the-art Finite Element Analysis (FEA) and optimisation principles to create a lightweight vehicle based upon an exterior styling envelope, i.e. a draft CAD design. The ideology was that implementing topology optimisation at such an early stage of the design cycle would enable the engineer to determine "ideal" locations for key electric powertrain components with an overall view of minimising the mass of the vehicle. The output of the work, as detailed in contribution 1, provided a draft methodology for this process including a substantial sensitivity study aimed at determining the robustness of the approach. The work also included a number of extended studies for example investigating manufacturing feasibility and full crash structure development; although these have not been submitted for consideration in this PhD portfolio.

3.2 Contribution 2

Contribution 2 is a direct extension of the LCVTP work documented by contribution 1, and followed approximately 6 months after the LCVTP program terminated. The work of contribution 2 also emphasises on sensitivity studies; it did however focus on a different

aspect of vehicle crash structures; namely the roof structure. There were several reasons for this. Firstly, the roof crush is a very severe scenario dominated by buckling and structural instability. At the time there was also a lot of focus upon roof crush modelling and test procedures in general, including several publications which proposed changes to the Federal Motor Vehicle Safety Standards (FMVSS) standard for roof crush; FMVSS 216. The work reviewed the potential impacts of the proposed changes to the FMVSS 216 standard upon hybrid vehicle roof design, using topology optimization, and found the most severe scenario, i.e. combination of pitch and roll angles, matching that proposed in other papers investigating the physical test procedures, for example (Parent et. al., 2010).

3.3 Contribution 3

Contribution 3 was developed as a direct and natural extension of contribution 2, and was published approximately 18 months after contribution 2. The reason for the natural link between contributions 2 and 3 is that although contribution 2 answered questions relating to the probable effects of the proposed changes to FMVSS 216 it also raised some severe shortcomings of the approach, as shall be further discussed in sections 4.3 and 4.4. The shortcomings of the approach were know in advance, and the work leading up to the publication of contribution 3 was aimed at proving that “taken for granted” assumptions about the approach were not true in this context.

3.4 Contribution 4

The background work for Contribution 4 is based on some of the ideas developed during the LCVTP project, and was initiated at around the same time as contribution 3, this is the reason why contribution 4 was published approximately six months after contribution 3. Contribution 4 investigates a supposedly significantly different approach to non-linear topology optimisation; namely the use of equivalent static loading, as shall be further discussed in section 4.5.

3.5 Contributions 5 and 6

The underlying concepts leading to contributions 5 and 6 were initially developed during the LCVTP project and were slowly matured from that point onwards; i.e. from 2011. The further development of the ideas and principles were not fully addressed until the start of 2014, which included revisiting a number of previous papers (included in contribution 5), as well as

reviewing contemporary developments of topology optimisation algorithms from 2011 onwards.

As indicated in the preceding sections the contributions in this portfolio are thereby numbered chronologically in the order they were completed. It is worth noting that contributions 5 and 6 were developed simultaneously.

4. Evaluative Description of Originality

The purpose of this section is to review and evaluate the originality of each individual contribution to the proposed PhD portfolio. With the research aims and objectives listed in section 1 in mind, the next sub-section will explain the basis for which the contributions to the portfolio have been selected.

4.1 Basis for Selection of Outputs for Portfolio of Evidence

The outputs have been carefully chosen to represent the overall evolution of the work firstly showing that "taken for granted" assumptions about linear static optimisation are not entirely supported by contemporary evidence, contributions 1, 2 and 4. This is followed by the development of additional tools for improving the optimisation outcomes, contribution 3. Finally, primarily based on the first 4 contributions a new and novel optimisation algorithm has been developed utilising aspects of multiple algorithms in addition to mathematical principles from other disciplines and finally integrated into a software tool, contribution 6. Contribution 5 contains the documentation for the software, as well as a series of case-studies critically assessing the software and overall methodology.

Complementary to the work directly included in the portfolio an additional 4 journal papers and 6 conference papers have been published with the candidate as the main author. Additionally the candidate has given national and international presentations based on this work including the Set for Britain (SFB) 2013 competition in the House of Commons and to the German Association for Computational Mechanics in Dresden, Germany. Furthermore, the candidate has signed a publishing contract with Elsevier publishers for a text book entitled "Non-linear Optimization of Vehicle Safety Structures" as a co-author. Finally, the work has been extended through at least 5 individual Master's thesis', and more recently it has led to a spin-off PhD on Hybrid Optimisation partly funded by Coventry University and MIRA (£150k) in addition to being the backbone of the recently EPSRC/TSB awarded Virtual Exhaust Prototype (VExPro) project with a total project value of £780k, for which the candidate is the Principal Investigator (PI).

4.2 Contribution 1

The originality in terms of contribution to knowledge for contribution 1 is to be found in the application of the modelling and the depth to which the investigations are conducted. A general assumption is that linear topology optimisation techniques cannot be used for

automotive crash structure development. The work undertaken in contribution 1 challenges this conception by critically reviewing the fundamental FE modelling techniques with a particular emphasis on model setup including mesh size and Boundary Conditions (BC) review. At the time very little detailed and in depth material was published about the use of topology optimisation for crash structure development. One of the most relevant and detailed investigations was the Future Steel Vehicle project (FSV, 2011) which demonstrated the use of topology optimisation as a tool in the design process. In terms of topology optimisation application and analysis the level of detail and depth of work is however considerably higher in contribution 1 than that of the FSV project. In particular contribution 1 provided original contributions to knowledge by determining the following:

- Commercially available linear static topology optimisation can advantageously be used for development of automotive crash structures based upon exterior styling envelopes.
- A specific FE modelling methodology was developed to ensure the dynamic nature of front, rear, side impact, pole and roof crush accident scenarios can be appropriately considered.
- The importance of the use of Inertia Relief (IR) boundary conditions as opposed to Single Point Constraints (SPC) was thoroughly investigated.
- Sensitivity studies were completed in order to map and critically evaluate the general sensitivity of topology optimisation results to mass distribution and load application angles.
- A general list of requirements for topology optimisation algorithms dealing with (non-linear) structures exposed to large deformations and dynamic loading was defined, and current state-of-the-art algorithms were reviewed and their suitability for further development was identified.
- A topology optimisation algorithm capable of conducting optimisation of non-linear structures was proposed. This method is based on the principles of linear static topology optimisation, thus “inheriting” the majority of advantages thereof. The proposed method shares some commonality with the Equivalent Static Load Method (ESLM) which has subsequently been introduced into multiple commercial FE suites such as Altair HyperWorks. This methodology is however not the one implemented in contributions 5 and 6 as shall be subsequently explained.

4.3 Contribution 2

The originality in terms of contribution to knowledge for contribution 2 is to be found in the unique application focusing on roof crush modelling including the associated post-processing and identification of roof topology trends as a function of variations in the applied load angles. Although the paper utilises an unconventional approach, i.e. linear static topology optimisation, to assess the effects of various roof crush scenarios the determination of the worst case scenario corresponds to that identified by other papers (focusing on physical roof crush experimentation and accurate FE modelling). The paper did however also state that the mass of the roof structure does not change significantly as a function of load application angles; contradictory to the majority of other published material. Although this unconventional method was used contribution 2 was the first paper to demonstrate the potential use of this methodology in this particular context. The paper also highlighted some shortcomings of the approach taken; predominately the inability to incorporate buckling and localised crushing effects as parameters in the topology optimisation. The paper does make suggestions as to how this may be corrected; e.g. by the adaptation of homogenisation based optimisation, as also discussed in contribution 1.

4.4 Contribution 3

There are a number of original contributions to knowledge in contribution 3. Firstly the paper overall demonstrates that the proposed methodology, i.e. use of topology, shape and size optimisation can indeed be used to design a structure capable of meeting the structural performance criteria of a crash scenario. Although the resulting topologies are significantly different than “conventional” automotive roof structures it should be noted that the vehicle layout is significantly different, i.e. a Hybrid Electric Vehicle (HEV), see contribution 1, and that the roof crush loading conditions are also significantly different as discussed in contribution 2. The paper also demonstrates how buckling can be incorporated as a parameter in the optimisation process by using size optimisation subsequent to the topology optimisation. This is done by estimating local second moment of area values assuming a tubular cross-section. Following the size optimisation a “conventional” vehicle roof structure cross-section is used as the starting point of a shape optimisation step with an aim of “matching” the second moments of area found during the size optimisation step in order to produce a more feasible cross-section subject to maximum exterior dimension constraints.

4.5 Contribution 4

The overall original contribution to knowledge of contribution 4 can be defined as a critical and detailed review of the Equivalent Static Load Method (ESLM) to vehicle crash structure topology and highly non-linear / dynamic problems in general. Since the time when contribution 1 was developed a number of "new" optimisation algorithms had been implemented into commercial FE software; the most prominent one being ESLM. As there are many parallels between ESLM and the methodology applied in contributions 1, 2 and 3 it was a relevant and important step to review the ESLM approach. Contribution 4 firstly compared the theoretical aspects of ESLM to the "conventional" Variable Density Method (VDM) and Solid Isotropic Material with Penalisation (SIMP) interpolation scheme, including potential optimisation parameters. Subsequently a large case study using identical FE models to those used in contributions 1, 2 and 3, was completed in order to benchmark the ESLM approach against the linear static VDM SIMP optimisation algorithm of HyperWorks. It was found that the ESLM approach had significant stability issues with the non-linear / large deformation nature of the models. A parameter study was therefore undertaken in an attempt to improve the stability of the models. This was somewhat successful; however a significant number of the ESLM models retained significant stability issues. At this point it was double checked that the level of non-linearity and dynamics of the models did not violate the stipulations for ESLM; via manuals and direct contact with the software vendors. When comparing the "successful" ESLM models to those using the VDM SIMP optimisation algorithm it was deemed that the substantial increase in CPU time associated with the ESLM approach did not provide a comparable increase of topology refinement justifying the substantial increase in CPU time and requirements.

4.6 Contributions 5 and 6

Based on the knowledge obtained in contributions 1 through 4 the aim of contributions 5 and 6 was to develop an optimisation algorithm specifically for structures exposed to large (non-linear) deformations and dynamic loading. The original contribution to knowledge of contribution 5 is the definition, development, implementation and initial assessment of a topology optimisation algorithm capable of appropriately catering for (non-linear) deformations and highly dynamic loading. The original contribution to knowledge in contribution 6 is the software tool itself.

As previously mentioned at this stage there had been significant developments in terms of topology optimisation availability in commercial FE codes since contribution 1 was developed. In addition to this the “original” literature review included in contribution 1 was conducted in a slightly more “narrow and industrial” context due to the environment in which it was completed than may be anticipated in a “true” research environment. As a consequence of these two facts it was decided to thoroughly revise and extend the literature review in contribution 5. The purpose was to propose and define an optimisation algorithm capable of appropriately catering for (non-linear) deformations and highly dynamic loading using Finite Element Analysis (FEA), whilst meeting the “non-linear topology optimisation criteria” list set forth in contribution 1.

The outcome was that Bi-directional Evolutionary Structural Optimisation (BESO) was the most suitable candidate for this purpose. The algorithm does however have some severe limitations; which were addressed through the implementation of principles from other algorithms and mathematical principles relating to statistics and uncertainty. The ultimate outcome of this extended literature review was the definition of a flow chart for an optimisation algorithm entitled Bi-directional Evolutionary Entropy Tabu search Simulated annealing (BEETS). The BEETS algorithm was subsequently programmed into a PowerShell script, contribution 6, enabling it to utilise commercial FE software for structural analysis. At present this is limited to LS-Dyna keyword 971 input decks.

Finally a series of case studies were completed in order to test the validity, relevance and extended potential of the algorithm for non-linear and dynamic structural optimisation problems. At this stage it should be highlighted that the purpose of this work was not to fully validate the BEETS algorithm; rather to initially investigate its potential for further development. It is concluded that the BEETS algorithm does demonstrate potential for further development, although a few aspects of the algorithm must be additionally reviewed before any extended development is undertaken.

5. Evaluative Review of Contribution to Subject Area

In the authors opinion the portfolio has successfully contributed to the subject area in a variety of different ways and with varying impact. The variation in the type of impact and the level of impact is not surprising given the varying nature of the work, from application and modelling techniques to theoretical developments and software implementation. The specific impact of contribution 1 is difficult to directly measure; this is predominately because the majority of work completed was bound by confidentiality in terms of practical applications (direct case studies relating to the Tata Beacon). The impact of the work could however be indirectly measured in terms of invited presentations and queries relating to the work. In this context a number of invited presentations to senior industrial experts were completed. These included invited presentations of the work and methodology to Craig Bonham, Chief Engineer Body & Exteriors at Tata Motors European Technical Centre (TMETC) in February 2011, Tayeb Zeguer, CAE Manager at Jaguar Land Rover in March 2011, and a presentation to Damian Dry, Principle Engineer Body & Exterior BIW Feasibility, SAIC in December 2011. The work also led to invited presentations at the 4th colloquium for the German Association of Computational Mechanics (GACM) in Dresden, Germany, in August 2011 as well as an invited presentation at the Altair Americas HyperWorks user Technology Conference (HTC) in Detroit, MI, USA in May 2012 including an associated video interview.

Contributions 2 and 3 were both published in the International Journal of Crashworthiness which has an impact factor of 0.88. Although this may seem a relatively low impact factor the journal itself is very relevant to the work undertaken. Unfortunately it has not been possible to review the number of official citations of these articles based upon the Taylor and Francis publishers' data. According to (Taylor & Francis, 2015) contribution 2 has had 138 views and contribution 3 has had 124 views (Taylor & Francis, 2015a), indicating a moderate amount of interest in the work. The moderate amount of interest in these particular publications was expected due to the controversy of the applied principles therein.

Contribution 4 was published in the Journal of Mathematical Research and Applications (JMRA), the impact factor of which is unknown. JMRA is an open access journal which under normal circumstances charges a fee for publication; it should be noted that the fee was waived for contribution 4, i.e. no fee was paid for this publication. It has not been possible to obtain neither the number of citations nor the number of views for this particular article. Although the direct impact and reach of contributions 2, 3 and 4 are not well documented it should be

noted that these 3 papers was used as examples of work in connection with applying for a book publishing contract with Elsevier publishers. The book entitled “Non-linear Optimization of Vehicle Safety Structures” is scheduled for publication in December 2015 (Elsevier, 2015), and will include aspects of the work presented in contributions all contributions to this PhD portfolio. The approval process for the book was subject to reviews of 4 external subjects experts who are unknown by name, but were listed with the following titles:

- Reviewer 1: Professor XXX, Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI USA.
- Reviewer 2: Dr XXX, School of Mechanical Engineering, University of Leeds.
- Reviewer 3: XXX, Professor of Solid Mechanics, University of Leeds.
- Reviewer 4: Professor XXX, Technical University of Denmark.

As previously stated the publishing contract was successfully awarded with the book currently undergoing final reviews as of May 2015.

Although the original contribution to knowledge of contributions 5 and 6 were clear to see the impact of these is low at present. The primary reason for this is simply that this work has not yet been published. There are a number of reasons behind this, primarily because the author intends to further develop and test the methodology before any proposed publications are submitted for review including validation studies. The first instance of public dissemination of the work included in contributions 5 and 6 will be the aforementioned text book.

6. Synthesis and Evaluation of Links between Outputs and Portfolio

All 6 outputs are linked as a continuous chronological evolvement of research into non-linear topology optimisation, as indicated through previous sections. The below paragraphs will further explain the direct links between all 6 contributions.

- Contribution 1, a HTML file, which documents the outcomes of the candidates' involvement in the ERDF funded LCVTP project. This includes a novel and substantial investigation into the advantages, disadvantages, pitfalls and limitations of applying linear static topology optimisation techniques for developing structures exposed to large deformations, in this case automotive crash-structures. This includes aspects such as sensitivity to variations of input parameters, e.g. load application angles, material characteristics, boundary conditions etc. The work is based around Hybrid Electric Vehicle (HEV) architectures and was initially based upon approximately 332 FE models. The contribution also contains critical appraisal reports of the findings in addition to a literature review which ultimately outlines suggestions for further development of an algorithm for non-linear topology optimisation.
- Contribution 2 is a published double-blind peer reviewed journal paper, which utilises the lessons learnt in contribution 1 to investigate the potential effects of proposed changes to simulation and testing standards (FMVSS 216) of automotive roof crush scenarios upon Body In White (BIW) / automotive roof structure design for HEVs. The paper not only focuses upon the variation of load cases but also draws conclusions on general "topology trends" in addition to critically appraising the suitability of the method applied.
- Contribution 3 is also a published double-blind peer reviewed journal paper published in the same journal as contribution 2. Contribution 3 is in fact a direct continuation of contribution 2, as it picks up some of the areas for further research, analysis and development identified in contribution 2. More specifically contribution 3 demonstrates the feasibility / accuracy of the topologies obtained in 2 and proves, for the first time, that the resulting structures are in fact able to meet the structural requirements set forth in 2, thus proving that "taken for granted" assumptions about the limitations of linear static optimisation can be disproven; knowing the limitations is

essential to the successful application of linear static topology optimisation to non-linear problems.

- Contribution 4 is a published double-blind peer reviewed journal paper critically assessing and comparing the techniques developed in contributions 1, 2 and 3 to the competing (but very similar) ESLM which at the time had been released in commercial (FE) software. The paper highlights the issues with ESLM, and thoroughly discusses why the limitations of linear static optimisation also applies to ESLM, thus concluding that this approach is not a "stronger candidate" for non-linear topology optimisation when compared to the methodology developed and refined in contributions 1, 2 and 3.
- Contribution 5 is a monologue containing three major parts. The first of which is a detailed literature review of optimisation algorithms and critically reviewing recent developments thereof. The overall aim of this part was to define a novel new optimisation algorithm fulfilling the requirements set forth based on contributions 1 - 4; with an ultimate aim of resolving the majority of limitations associated with the methodology used in contributions 1 - 4. The second part of the monologue defines and explains how the optimisation algorithm from the first part was integrated into software, contribution 6. The final part of the monologue contains a series of case-studies comparing results obtained using the developed software / algorithm to competing algorithms in order to critically assess the potential of the algorithm for non-linear topology optimisation.
- Contribution 6 is the novel and new algorithm embedded in software documented by contribution 5.

7. Critical Reflection of Candidate's Development as a Research Practitioner

At the early stages of the development of the PhD portfolio the candidate was highly influenced by his previous work as an engine fitter as well as his undergraduate and postgraduate engineering studies. This meant a strong predisposal towards quantitative research methods, which is also prominent throughout the majority of contributions included in the portfolio. Contribution 1 is a prime example of this where emphasis is on the percentage mass reduction of each model. Due to the exploratory nature of the research and the associated difficulties with conclusively demonstrating the benefits, relevance and indeed accuracy of the methods developed and applied, i.e. adaptation of linear static optimisation and the initial development of the BEETS method, has however "forced" the candidate to include aspects of qualitative research in combination with the quantitative approach. Although outside the "comfort zone" of the candidate, who prefers mathematical proof, this was necessary when considering time and capacity limitations. However even without these limitations the required intricacy of FE based topology optimisation algorithms for structures exposed to large (non-linear) deformations and dynamic loading, for example through "the curse of dimensionality", see contribution 1, suggests that a fully qualitative based proof of the accuracy, validity and relevance of such an algorithm may be impossible to obtain. The first instance where this became apparent to the candidate was through the latter stages of developing contribution 1, forcing the candidate to "change" to a qualitative approach in order to further the research; the outcome of this being the "list of requirements" for a non-linear topology optimisation algorithm originally listed in contribution 1 and repeated in contribution 5.

This mixed approach of qualitative and quantitative research methods and argumentation was adapted through the remainder of the portfolio development. The publication process of contribution 2 proved to be another significant landmark in developing the candidates mixed research approach; as one of the blind peer reviewers was very critical of the approach; citing the "unconventional" nature of the methodology as the main reason for recommending that the article not be published without a major rewrite. The author did however dispute this recommendation using a series of quantitative and above all qualitative measures. Eventually the editor chose to ignore the recommendation of that particular peer reviewer based on the evidence and argumentation set forth by the author, and thus the journal paper was published. As previously stated this was a significant turning point; reinforcing the potential powerful

argumentations and evidence that can be found via qualitative research methods even in a mechanical engineering context.

Contributions 3 and 4 were primarily based on quantitative research methods; whereas contributions 5 and 6 contained a mixed approach of qualitative and quantitative research methods.

In summary the development of the portfolio has significantly changed the candidates' perspective of the importance, efficiency and potential benefits of using a mixed approach of qualitative and quantitative research methods; however due to his mechanical engineering background the candidates' preference remains to apply quantitative research methods where feasible.

8. Contributions of Others

The first 4 contributions in this portfolio all have named co-authors; the following will describe the specific contributions of those individuals:

- **Christophe Bastien** and **Mike Blundell**. Both Christophe and Mike have acted as line managers of the PhD candidate, and has supported the candidate in an identical way to that of PhD supervisors for a conventional PhD candidate. For example by reviewing ideas, results, overall research direction and internal review of papers prior to submission for external publication.

- **Pierre Andre Batt**. Pierre Andre was a student intern at Coventry University from January to June of 2012. Pierre Andre worked under the supervision of Jesper Christensen and Christophe Bastien developing a shape optimisation tool featured in section 3 of contribution 3. It should however be noted that the actual application featured in contribution 3 was completed by the candidate, including writing the entire journal paper.

Contributions 5 and 6 were entirely and exclusively developed by the candidate Jesper Christensen.

9. Conclusions and Future Work

In the authors opinion this portfolio covers a substantial amount of work equivalent to that of a conventional PhD. The work contains a mixture of in-depth critical reviews, derivations, development, programming and verification / validation work both from theoretical as well as practical viewpoints. All 6 contributions have been carefully selected to clearly and concisely demonstrate the natural progression of the candidate as a research practitioner as well as the gradual and "natural" development of a number of significant contributions to knowledge. The work contained within the portfolio of evidence constitutes a continuous, novel and coherent contribution to the development of non-linear topology optimisation as well as non-linear optimisation in general; specifically the work within the portfolio addresses the following areas of "*significant and original contribution to knowledge*" as stipulated by the University regulations (CU, nd):

- Develop a new model, paradigm or conceptual framework and test it in application. (**Contributions 5 and 6)**
- Successfully challenge an existing model or paradigm and show how it can be improved or why it should be discarded (in certain circumstances).
  (Contributions 1 - 4)
- Show that "taken for granted" truths or assumptions are not substantiated by contemporary evidence.
  (Contributions 1 - 3)
- Develop an existing methodology, form of enquiry or tool set for data collection, analysis, display or interpretation and show how its use in application proved to be superior in some circumstances compared to other tools etc.
  (Contributions 1 - 3 and 5 - 6)
- Show limitations and errors in existing dominant methodologies, forms of enquiry or use of existing tools or analytical methods and the consequences for interpretation of previous structures.
  (Contributions 1 - 4)

- Add progressively to understanding of an issue, part of a field of a complex problem (e.g. multidisciplinary one), social or natural phenomenon or professional practice by a series of linked in-depth studies or experiments. **(Contributions 1-5)**

In the authors view the work included in the portfolio can in general be divided into two main aspects:

A1. Development of advanced application methodologies and critical appraisal of linear static optimisation methods, contributions 1 - 4.

A2. Definition, development, implementation, programming, application and critical evaluation of a non-linear dynamic topology optimisation algorithm, contributions 5 and 6.

Using these "divisions of work" definitions as well as the areas of "*significant and original contribution to knowledge*" stipulated above general conclusions of the work can be drawn; starting with A1.

In the authors opinion the work included in A1 has successfully *challenged* the existing "model" for FE based topology optimisation; the VDM SIMP approach. Contributions 1, 2 and 3 "*demonstrated how this approach could be improved"* by developing a specific methodology for appropriate creation of models to represent dynamic and non-linear load crash scenarios using a linear static solver. This included detailed review and critical appraisal of aspects such as BCs as well as creation and definition of equivalent static loads. This methodology was applied and refined using a large number of case studies focusing on lightweight Hybrid Electrical Vehicle (HEV) Body In White (BIW) / crash structure design and optimisation. Contributions 2 and 3 added further to this by demonstrating that the "final design" was actually capable of meeting the structural performance criteria stipulated for at the topology optimisation stage, thus showing that "*taken for granted*" assumptions about linear static optimisation were not substantiated for this particular application.

In the authors experience the "*taken for granted assumptions*" about linear static optimisation varies significantly throughout the engineering community. Automotive engineers are generally highly sceptic about this approach when applied to structures exposed to large deformations and dynamic loading. This attitude does however seem to have changed somewhat with the introduction of ESLM. However, as shown in contribution 4, and also revised in contribution 5 the introduction of ESLM retains some of the significant issues relating to linear static FE based

optimisation; predominately the continued use of Young's modulus to dictate the material stiffness, regardless of stress and strain levels, most likely leading to an underestimation of the structural compliance when exposed to large deformations. Contribution 4 also demonstrated some of the "practical" pitfalls, i.e. *limitations* of the approach, including excessive CPU requirements with no significant improvement in terms of accuracy, precision or confidence of results thus concluding that the ESLM approach contributes very little in terms of advancing state-of-the-art topology optimisation algorithms for structures exposed to large (non-linear) deformations and dynamic loading, concluding that *this approach should be discarded*.

Aspect A2 utilises the knowledge obtained through A1, in particular with respect to generating a *"list of requirements"* for a topology optimisation algorithm for structures exposed to large (non-linear) deformations and dynamic loading, as included in contributions 1 and refined in contribution 5. This list was subsequently used in contributions 5 and 6 to develop a "*new model or conceptual framework*" which was *initially tested* through fundamental case studies. The new optimisation algorithm was denoted BEETS; it was concluded that the cases studies using the algorithm demonstrate potential for further development although there are a few "teething" problems partially related to the algorithm and partially related to the method of implementation which must be investigated and resolved before additional development of the algorithm is undertaken.

The author recognises that the portfolio contains no physical experimentation results which may prove critical for further development, validation and correlation of any algorithm developed. It is important to emphasise that any optimisation algorithm will only be as accurate as the information that is fed into it. FE based optimisation algorithms will therefore only be as accurate as the FE analysis that provides the input information.

It is the authors' opinion that the portfolio of evidence, i.e. the 6 contributions has "*progressively added to the understanding of the complex issue of topology optimisation of mechanical structures exposed to large (non-linear) deformations and dynamic loading*" thus fulfilling this criteria for a significant and original contribution to knowledge.

As specified in section 4 the 6 contributions to this portfolio of evidence was carefully selected from a significantly larger bulk of work, which in combination with the work included in the portfolio has led to a number of additional achievements and activities including:

a) A signed publishing contract with Elsevier; to be published in December 2015.

b) A £780k TSB/EPSRC funded VExPro project, partnered with Unipart (EPSRC, 2015).

c) A fully funded PhD student working on Hybrid Optimisation in collaboration with MIRA.

d) A self-funded PhD student working on Holistic Optimisation.

e) Further integration into the CU “Transport” Faculty Research Centre.

Point e) above relates to the fact that the candidate has been selected to be part of a recently formed Faculty Research Centre (FRC) at Coventry University focusing on the overall theme of Transport and Mobility. It is the candidates’ intention to seek further funding for the continued development and critical evaluation / validation of the BEETS algorithm, in order to better assess its potential for topology optimisation of mechanical structures exposed to large (non-linear) deformations and dynamic loading.

References

Christensen, J., Bastien, C., Blundell, M. V. (2011) "Effects of Roof Crush Loading Scenarios Upon Body in White using Topology Optimisation". International Journal of Crashworthiness 17 (1), 29-38.

Christensen, J., Bastien, C., Blundell, M. V., Batt, P.A. (2013) "Buckling Considerations and Cross-Sectional Geometry Development for Topology Optimised Body In White" International Journal of Crashworthiness 18 (4), 319-330.

Christensen, J., Bastien, C., Blundell, M. V. (2013a) "The Feasibility of ESLM for BIW Roof Structure Development and Optimisation". Journal of Mathematical Research and Applications 1 (2), 34-47.

CU (nd) "Coventry University, PhD by Portfolio information pack" (online) available from: <https://students.coventry.ac.uk/ResearchDegrees/Documents/Forms/AllItems.aspx?RootFolder=https%3a%2f%2fstudents%2ecoventry%2eac%2euk%2fResearchDegrees%2fDocuments%2fNEW%20QRS%20Site%202013%2fPhD%20by%20Portfolio&FolderCTID=0x0120004C094EE44CFFFB459B2AE9924F5DEB33 >(20 May 2015)

Elsevier (2015) "Elsevier store" (online) available from: < http://store.elsevier.com/Non-linear-Optimization-of-Vehicle-Safety-Structures/Jesper-Christensen/isbn-9780124172975/> (19 May 2015)

EPSRC (2015) "Engineering and Physical Sciences Research Council" (online) available from: < http://gow.epsrc.ac.uk/NGBOViewGrant.aspx?GrantRef=EP/M506771/1> (20 May 2015)

Low Carbon Vehicle Technology Project - LCVTP (nd) "Low Carbon Vehicle Technology Project" (online) available from: < http://www2.warwick.ac.uk/fac/sci/wmg/research/lcvtp/> (18 May 2015)

Parent, D.P., Kerrigan, J.R. and Crandall, J.R. (2010) "Comprehensive computational rollover sensitivity study part 1: Influence of vehicle pre-crash parameters on crash kinematics and roof crush, ICRASH 2010 Conference, Washington, DC, USA, 22–24 September 2010.

Taylor & Francis (2015) "International Journal of Crashworthiness" (online) available from: < http://www.tandfonline.com/doi/abs/10.1080/13588265.2011.625640> (19 May 2015)

Taylor & Francis (2015a) “International Journal of Crashworthiness”
(online) available from:
< http://www.tandfonline.com/doi/abs/10.1080/13588265.2013.792442> (19 May 2015)

This page is intentionally left blank.

CONTRIBUTION 1

HTML document
Applied Research Documentation from the Low Carbon Vehicle Technology Project

The printed is the non-linear topology optimisation report. All reports, associated models and full documentation of the contribution can be found on the enclosed CD

This page is intentionally left blank.

CONTRIBUTION 2

Double blind peer review journal paper published in the International Journal of Crashworthiness on October 13th 2011

[*Format has been edited to comply with Coventry University regulations*]

This page is intentionally left blank.

International Journal of Crashworthiness
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tcrs20

Effects of roof crush loading scenario upon body in white using topology optimisation

J. Christensen [a], C. Bastien [a] & M. V. Blundell [a]
[a] Faculty of Engineering and Computing, Coventry University, Coventry, UK

Available online: 13 Oct 2011

To cite this article: J. Christensen, C. Bastien & M. V. Blundell (2011): Effects of roof crush loading scenario upon body in
white using topology optimisation, International Journal of Crashworthiness,
DOI:10.1080/13588265.2011.625640
To link to this article: http://dx.doi.org/10.1080/13588265.2011.625640

International Journal of Crashworthiness
iFirst 2011, 1–10

Christensen, J. , Bastien, C. and Blundell, M.V. (2012) Effects of roof crush loading scenario upon body in white using topology optimisation. *International Journal of Crashworthiness*, volume 17 (1): 29-38.
http://dx.doi.org/10.1080/13588265.2011.625640

This page is intentionally left blank.

CONTRIBUTION 3

Double blind peer review journal paper published in the International Journal of Crashworthiness on May 1st 2013

[*Format has been edited to comply with Coventry University regulations*]

This page is intentionally left blank.

International Journal of Crashworthiness
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tcrs20

Buckling considerations and cross-sectional geometry development for topology optimised body in white

J. Christensen [a] , C. Bastien [a] , M. V. Blundell [a] & P. A. Batt [a]
[a] Faculty of Engineering and Computing , Coventry University , Coventry , UK

Published online: 01 May 2013

To cite this article: J. Christensen , C. Bastien , M. V. Blundell & P. A. Batt (2013): Buckling considerations and cros sectional
geometry development for topology optimised body in white, International Journal of Crashworthiness,
DOI:10.1080/13588265.2013.792442
To link to this article: http://dx.doi.org/10.1080/13588265.2013.792442

Christensen, J. , Bastien, C. , Blundell, M.V. and Batt, P.A. (2013) Buckling considerations and cross-sectional geometry development for topology optimised body in white. International Journal of Crashworthiness, volume 18 (4): 319-330.
http://dx.doi.org/10.1080/13588265.2013.792442

CONTRIBUTION 4

Double blind peer review journal paper published in the Journal of Mathematical Research and Applications (JMRA) on Dec 23rd 2013

[*Format has been edited to comply with Coventry University regulations*]

Christensen, J. , Bastien, C. , Blundell, M.V. (2013) The Feasibility of ESLM for BIW Roof Structure Development and Optimisation. Journal of Mathematical Research and Applications, volume 1 (2): 34-47.
http://www.academicpub.org/jmra/paperInfo.aspx?PaperID=14490

CONTRIBUTION 5

Monologue

Completed December 2014

This page is intentionally left blank.

Acronyms

α	Lagrangian Multiplier, element sensitivity number
ACO	Ant Colony Optimisation
ANS	Average Neighbourhood Strain energy
AR / ARMAX	Admissions Ratio / AR Maximum
β	Lagrangian Multiplier
BC	Boundary Condition
BEETS	Bi-directional Evolutionary Entropy Tabu search Simulated Annealing
CA / CAA	Cellular AutomatA
CAN	Cellular AutomatoN
CNE	Current Number of non-void Elements
COOLIT	Cooling Interval
COP / DCOP / SCOP	Combinatorial Optimisation Problem / Deterministic COP/ Stochastic COP
COOLUSE	Boolean operator to activate / deactivate cooling function
D	Design variable
Δt	Time step size
DV	Design Volume
e	Base of natural logarithm
E	Young's Modulus / Element strain energy
ER	Evolutionary Ratio
ESE	Element Strain Energy
ESLM	Equivalent Static Load Method
ESN	Element Sensitivity Number
ESO / AESO / BESO	Evolutionary Structural Optimisation / Additive ESO / Bi-directional ESO
f	Generic function
F	Externally applied force
FE / FEM / FEA	Finite Element / FE Model /FE Analysis
FL	Fuzzy Logic
G	Constraint
HCA	Hybrid Cellular Automata
L	Function

log / $\log_2$	Natural logarithm (base 10) / base 2
Max_iter	Maximum Number of Iterations
MEP	Maximum Entropy Principle
MEPUSE	Boolean operator to activate / deactivate the use of MEP
MCS	Monte Carlo Simulation
MNEA	Maximum Number of Added Elements
NEA	Number of Elements Added
NESTL	Number of Elements Stored in the Tabu List
NITS	Number of Iterations per Tabu Search
NN	Neural Network
NN_{els}	Number of Neighbouring Elements
ω	Weighting function
P	Probability function
PSO / BPSO	Particle Swarm Optimisation / Binary PSO
Q / q	Local / global quantity or constraint
r	Radius / distance between element centres
RADIUS	Parameter to define distance for neighbourhood influence
RMS	Root Mean Square
S	Entropy
SA / QSA	Simulated Annealing / Quasi-SA
SCH	Stochastic Hill Climbing
SFCOOL	Scale Factor for cooling function
SIMP	Solid Isotropic Material with Penalisation
t	Time (s)
T	Temperature
TABU_offset	Function variable to offset the start of TS
TABU_LIST	Boolean operator to activate / deactivate the Tabu list
TERMTOL	Termination tolerance
TLF	Tabu List Fraction
TS / TSI / TSF	Tabu Search / TS Iteration / TS Fraction
TV / TVF	Target Volume / TV Fraction
V	Volume
VE	Volume of individual Element

VDM	Variable Density Method
VR	Variation Ratio
VRE	Volume of Removed Elements
VTS	Variations per Tabu Search
VTSE	Volume to Total Strain Energy ratio

List of Figures

List of Tables

Table of Contents

1. Introduction

This monologue which constitutes Contribution 5 to the PhD portfolio aims to:

1. Review potential approaches for non-linear (material and geometry) topology optimisation of structures exposed to large deformations.
2. Present the theoretical aspects as well as documenting the development of the optimisation algorithm embedded within Contribution 6 to this PhD portfolio.
3. Present and discuss a series of case studies using the software which constitutes Contribution 6 to this PhD portfolio.

The ultimate aim of chapter 2 is to define a strategy for the continued research into and definition of a non-linear topology optimisation algorithm for structures exposed to large deformations. The first chapter will therefore provide the overarching algorithm which will form the basis for the final, i.e. the sixth, contribution to this PhD portfolio, the optimisation software.

Chapter 3 of this monologue will document the initial development and implementation of the algorithm from chapter 1 into the software, i.e. Contribution 6, including flowcharts, parameters, variables, user inputs, initial verification / validation of the software etc.

Chapter 4 of this monologue will present a series of topology optimisation case studies used to compare the embedded algorithms to competing algorithms, including commercial software where applicable. The ultimate aim of chapter 4 is not to fully validate the software (contribution 6), but rather to analyse the performance of the algorithm and assess the feasibility for further development work.

2. Review of topology optimisation algorithms

Contribution 1 to this PhD portfolio contains a report entitled “Non-Linear Finite Element based Topology Optimisation for Low Carbon Vehicle Technology Project (LCVTP)”, which reviews the potential for the use of the Solid Isotropic Material with Penalisation (SIMP) interpolation scheme in combination with the Variable Density Method (VDM) as well Evolutionary Structural Optimisation (ESO) algorithms for non-linear optimisation. This report was completed in November 2011. As there has been some considerable developments in terms of commercially available (Finite Element based) optimisation software since then it was deemed appropriate to revisit and extend the literature review at this stage.

Optimisation algorithms may be categorised in a number of different ways. In this monologue the focus will be aimed at two main categories: mathematical and heuristic / meta-heuristic algorithms as shall be subsequently presented. Moreover, optimisation algorithms may also be divided into approximation algorithms and exact algorithms, (Vazirani, 2001). Exact algorithms work on the principle that all permutations of a given optimisation problem be computed; the optimisation problem thus reduces to a “simple” selection problem. However, as discussed in Contribution 1, the curse of dimensionality rapidly eradicates the feasibility of employing exact algorithms for most topology optimisation applications. (Hosny, 2010) states:

“*The term approximation algorithm is often used to refer to an optimization algorithm which provides a solution that is guaranteed to be within a certain distance from the optimum solution every time it is run, with provable runtime bounds*”

All algorithms discussed in this monologue may therefore generally be considered to belong to the approximate algorithm family, although certain limitations will apply, as shall be discussed throughout the remainder of this chapter.

2.1 Mathematical Algorithms

From a “practical” engineering viewpoint mathematical based algorithms are “ideal” for structural optimisation, as these generally have a clear and systematic structure, enabling a relatively straightforward implementation into numerical analysis software such as commercial FE programs. Gradient based methods such as the Optimality Criteria Method (OCM), Moving Asymptotes (MA) and Sequential Linear Programming (SLP) are most often employed. From a scientific viewpoint the perhaps most pivotal feature of mathematical algorithms is that they (generally) allow it to be proven (within certain limitations) that the optimisation is converging

towards a (global) optimum solution, as opposed to a (local) “random” solution, provided a unique global optimum solution exists. This “proof” is of course dependent upon a number of parameters governing the specific optimisation problem. As mentioned above, Contribution 1 of this PhD portfolio contains a literature review of mathematical based algorithms including Homogenisation Based Optimisation (HBO). As concluded in Contribution 1, the use of approaches such as HBO has extremely limited applications within engineering in general, as it is not possible to define a continuous function to describe e.g. the stress state throughout a component. The opportunities of utilising such an approach thus addresses a much wider aspect of analytical and numerical modelling, and would (if solved) make the use of Finite Element obsolete for stress analysis. This approach will therefore not be further discussed.

Contribution 1 also contains detailed discussions relating to the VDM method and the SIMP interpolation scheme, which are perhaps the most widely used and accepted approaches to topology optimisation of linear elastic and isotropic problems. Indeed many adaptations and variations of the SIMP interpolation scheme has been used for a wide variety of applications outside of the linear elastic and / or isotropic remits of the original formulation. (Sorensen and Lund, 2012) did for example attempt to utilise the scheme for optimisation of composite ply lay-up. The “Non-Linear Finite Element based Topology Optimisation for Low Carbon Vehicle Technology Project (LCVTP)” document also contains suggestions as to how the SIMP scheme could be “modified” to cater for (minor) levels of material non-linearity, however the general issues related to SIMP as highlighted throughout the document are still applicable. It can be justified that the overall topology optimisation study of Contribution 1 utilised a “manual” version of the Equivalent Static Load Method (ESLM) originally proposed by (Park, 2011). Since Contribution 1 was completed in November 2011, the ESLM method has become available in commercial software such as Altair HyperWorks. In light of this Contribution 4 contains a more detailed comparison of the underpinning differences between the “single” linear static topology optimisation algorithm and the ESLM approach, this includes a detailed case study. Contribution 4 concurs with the conclusions of Contribution 1, in concluding that the SIMP interpolation scheme is not suited for large deformation non-linear topology optimisation. In light of this information and the infeasibility of formulating continuous functions, e.g. stress functions, for most mechanical engineering structural analysis / optimisation problems the purely mathematical algorithms are dismissed as a potential approach for a non-linear topology optimisation algorithm in connection with this PhD portfolio. Instead the focus will now turn to heuristic algorithms.

2.2 Heuristic & Meta-Heuristic Algorithms

If the mathematical optimisation algorithms presented in section 2.1 are considered capable of providing the "exact" (optimum) solution to a given optimisation problem, heuristic algorithms may (in the extreme) be considered to take the most "practical" approach to optimisation. Pure heuristic algorithms are most often experience based and utilise a rational and above all practical approach to optimisation and problem solving in general.

The scenario of a structural engineer reviewing and modifying an initial design based upon personal experience and numerical analysis results may thus be defined as a heuristic "algorithm". Hence pure heuristic algorithms have inherent uncertainties with e.g. repeatability (imagine that the structural engineer is replaced) and relevance of solution (how can it be justified that the solution is indeed an optimum?). Heuristic algorithms thus generally have many undesirable factors of uncertainty associated with them.

Meta-heuristic algorithms are an adaption of heuristic algorithms and overall address many of the common issues associated with heuristic algorithms. Meta-heuristic algorithms most often adapt ideologies from other sciences such as biology and evolution, and combine these with physical experimentation results as well as experience. The following sub-sections will introduce some of the competing meta-heuristic algorithms.

2.2.1 Hybrid Cellular Automata

Hybrid Cellular Automata (HCA) is the optimisation algorithm incorporated in LSTC's commercially available LS-TaSC optimisation software capable of handling large deformation non-linear problems, such as vehicle crashworthiness (LSTC, 2011). According to (LSTC, 2011 a) the algorithm is a heuristic one. However, according to (LSTC, 2011) the algorithm was first derived by (Tovar, 2004) which clearly utilises bone biology to define the Hybrid Cellular Automata algorithm, arguably making it a meta-heuristic algorithm. The HCA algorithm is an adaptation of the Cellular Automata (CAA) model originally developed by Stanislaw Ulman and John Von Neuman in the 1940s, (originally Cellular AutmatoN (CAN). The general principle behind CAA is as follows. Imagine a series of "individual cells" which may have finite states, e.g. "1" or "0" as is the basic principle of FE based topology optimisation. Each individual cell is "related" to a set of cells denoted its "neighbourhood". An initial state of each cell is set (at t=0), a new overall state (at t = 1) is then achieved using a fixed rule, for example the outcome of an FE analysis, the new state of each cell (at t = 1) is then obtained based upon its current state (at t = 0) and "the responses" from its neighbourhood. The neighbourhood may be

defined in slightly different ways; one of the most often used neighbourhoods is the Von Neumann neighbourhood (Barati, 2014).

Comparing the above description to that of the Evolutionary Structural Optimisation (ESO) algorithms presented in Contribution 1 the parallels are very distinct; as the ESO algorithms use a "snapshot" of the current state, i.e. FE analysis results to choose "initial states" and thereafter use "neighbourhoods" to determine the state of each element (cell).

The basic topology optimisation problem in LS-TaSC is solved using a uniform internal energy density as the objective function, (Roux, 2011). The topology optimisation problem in LS-TaSC is formulated as equation (2.1) (Roux, 2011a):

$$\left.\begin{aligned} &\min_{x} \sum_{i=}^{N}\sum_{j=1}^{L}\left(w_j U_j\left(x_i\right)-U_j^{*}\right) \\ &subject\ to:\ \sum_{i=1}^{N}\rho\left(x_i\right)V_i \le M^{*} \\ &C_j^{l} \le C_j \le C_j^{u},\ j=1,2...,J \\ &x_{\min} \le x_i \le 1.0 \end{aligned}\right\} \tag{2.1}$$

In (2.1) w is a weighting factor, U is the internal energy density of element I, V_i is the element volume, U^{*} is the internal energy density set point and C_j is constraint j with lover (l) and upper (u) subscript boundaries, (Roux, 2011). The change in the individual design variable I at time t and the updated design variable at t+1 are defined in expression (2.2) (Roux, 2011a):

$$\begin{aligned} \Delta x_i^{t} &= \frac{K\left(U_i^{t}-U^{*}\right)}{U^{*}} \\ x_i^{t+1} &= x_i^{t}+\Delta x_i^{t} \end{aligned} \tag{2.2}$$

In (2.2) K is a scale factor. According to (Roux, 2011), LS-TaSC is controlled by 3 user-defined parameters: Mass Fraction (MF), Convergence Tolerance Parameter (GTP) and Proximity Tolerance Parameter (PTL). LS-TaSC utilises a "full design volume" for a starting point, as does, see Contribution 1 "Non-Linear Finite Element based Topology Optimisation for Low Carbon Vehicle Technology Project (LCVTP)" Figure 27.

In Contribution 1 "Non-Linear Finite Element based Topology Optimisation for Low Carbon Vehicle Technology Project (LCVTP)" the objective function and constraints for the Variable

Density Method were formulated as equation (9) and (11). These are repeated in (2.3) and (2.4) for convenience:

$$\begin{gathered} \text{minimise } F(\rho) \\ \textit{subject to}: \\ g_1(\rho) \le Val_1 \quad ; \quad g_2(\rho) = Val_2 \quad ; \quad g_3(\rho) > Val_3 \end{gathered} \tag{2.3}$$

$$F(\rho) = U(\rho) = \frac{1}{2}\int\left(\{\varepsilon\}^T \cdot \left[E(\rho)\right] \cdot \{\varepsilon\}\right) dV \tag{2.4}$$

Comparing (2.3) and (2.4) to (2.1) it is clear that there are distinct similarities between the two "algorithms". The main difference lies in the fact that (2.1) utilises the strain energy density while (2.4) uses the (total) linear elastic strain energy. It should be noted that (2.4) is an example of how VDM works in general, the actual implementation will vary from software to software. (Altair, 2009) does however support that the above explanation is indeed the way that topology optimisation is implemented into Altair HyperWorks' optimisation module Optistruct. (Altair, 2009) also confirms that Optistruct utilises a gradient based approach to perform the actual optimisation, as discussed in Contribution 1. Despite the similarities discussed above HCA and LS-TaSC does not utilise a gradient based approach, (LSTC, 2011 a):

"This method updates the density of elements based on information from its neighbors. No gradient information was required".

This means that LS-TaSC / HCA performs the optimisation based on "current" information, and is not influenced by "time-history", i.e. previous iterations are not considered as part of the optimisation process. In other words, the optimisation process has no "memory" as e.g. Optistruct does (Altair, 2009). The above statement thus reveals a distinct difference between CA and HCA, as the latter does not use "time-history", i.e. gradient information, whereas the former (CA) does.

From an overall perspective this means that HCA / LS-TaSC is in essence very similar to evolutionary algorithms which are discussed in the "Non-Linear Finite Element based Topology Optimisation for Low Carbon Vehicle Technology Project (LCVTP)" report contained within Contribution 1. This also suggests that LS-TaSC "shares" some of the inherent problems of evolutionary algorithms with respect to proven convergence towards a truly optimum solution

as also discussed in Contribution 1. As previously discussed this is one of the major advantages of mathematical based algorithms, however when large levels of non-linearity is introduced the "predictive" (linear) nature of the systems and the associated benefits diminish as highlighted in Contribution 4. In light of the above discussions it was chosen to abandon the HCA and ESO based algorithms. Furthermore, the lack of "time-history" influence, e.g. gradient based information" is a potential area of concern, as it is very likely that both the HCA and ESO algorithms will rapidly move towards a specific solution (topology). For relatively "simple" topology optimisation problems, i.e. small design volumes, this may in fact be advantageous, but for larger more complex problems it indicates that neither algorithm is able to "diversify" its search pattern and avoid local extremum / optimum points.

The general advantages and disadvantages of the HCA / CA based algorithms are listed in Table 1.

Table 1, advantages and disadvantages of CA / HCA algorithms.

Advantages	Disadvantages
Strong, direct and relatively simple implementation into FEA.	No time-history data considered.
Local and global constraint control.	No convergence justification.
Direct link between objective function and FEA.	Relatively high computational cost.
Neighbourhood influence.	Low dispersion of search within potential solution space, high probability of "converging" towards local optimum points.

In addition to the continuous comparison between CA and BESO throughout this section it is convenient to review the most recent developments of the BESO based algorithm in isolation. This is because BESO was found to have the highest potential for non-linear topology optimisation of the three "conventional" evolutionary based algorithms as discussed in "Non-Linear Finite Element based Topology Optimisation for Low Carbon Vehicle Technology Project (LCVTP)", Contribution 1.

2.2.2 Bi-directional Evolutionary Structural Optimisation

An extensive search for publications and original research into the most recent developments of BESO, AESO and ESO for non-linear topology optimisation has returned very little documentation. ESO based optimisation can in general be classified as belonging to the "greater family" of optimisation algorithms known as Genetic Algorithms (GA). Expanding the literature review to include GA at this stage did provide additional publications but none directly related to non-linear topology optimisation. The most recent paper, (Huang and Xie, 2008), from the "fathers" of BESO, Y.M. Xie and X. Huang directly focusing on non-linear topology optimisation is from 2008, (Researchgate, 2014) , indicating that the development of BESO for non-linear topology optimisation has not been widely pursued recently.

Given the general similarities between GA and BESO and that the literature review did not disclose any publications from other authors specifically relating to the development of GA or BESO for non-linear topology optimisation post 2008 it is presumed that (Huang and Xie, 2008) contains the most recent version of BESO for non-linear topology optimisation; the following subsection will therefore utilise the information of this publication, as this is representative of GA algorithms in general.

The paper focuses on both geometrically as well as material non-linearity under displacement loading, and focuses on the robustness of the algorithm. Please note that robust optimisation will be further discussed in section 2.3.1. Based on information in (Huang and Xie, 2008) an overall flowchart for the revised BESO algorithm has been created, Figure 1.

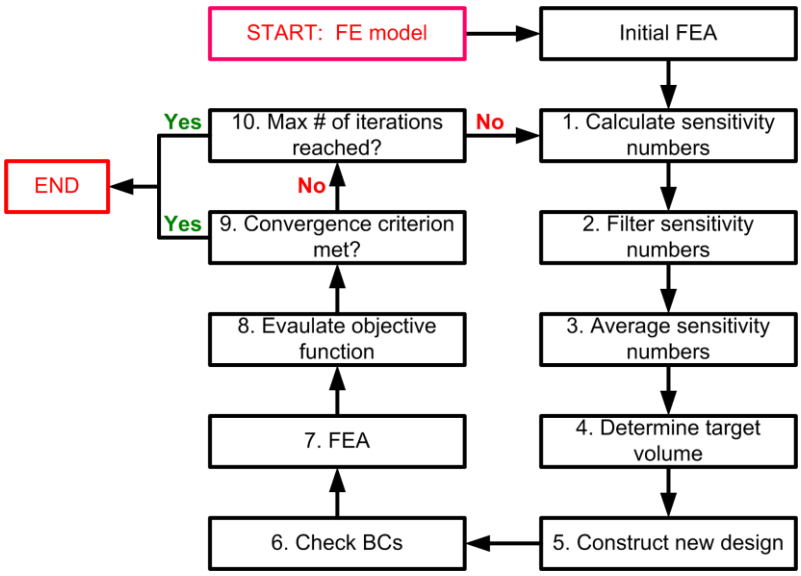


Figure 1, BESO flow diagram.

The paper highlights three main aspects which have been improved in the "updated" BESO algorithm as:

a) Filtering scheme (2).
b) Improved sensitivity analysis (1, 3).
c) A new procedure for adding and removing elements (5).

The brackets above relate to the algorithm steps illustrated in Figure 1. It can be noted that the "filtering scheme step" falls in between the two parts of the "sensitivity analysis", and indeed the two are intertwined. The algorithm in (Huang and Xie, 2008) aims to maximise the structural stiffness of the structure with a constraint based on maximum allowed displacements. The objective function thus seeks to maximise the external work which is set equal to the total strain energy under the caveat of a quasi-static load condition. While this condition may not be of major concern to certain topology optimisation problems it is certainly an aspect which must be thoroughly considered if dealing with e.g. high velocity crashworthiness scenarios. This potential concern could possibly be circumnavigated by introducing a very low timestep in the FE analysis; this would however have a significant effect upon the overall CPU run time.

The revised sensitivity analysis and filtering scheme aims to determine the influence upon the individual (FE) element upon the response of the objective function, i.e. the sensitivity. Overall, this is achieved by observing the (strain) energy levels within the individual element and the method used to summarise the total strain energy. Equation 15 in (Huang and Xie, 2008) denotes the variation of the objective function as a function of the individual element *x:*

$$\Delta f\left(x\right)=\lim_{n\to\infty}\sum_{i=1}^{n}\left(E_i^e-E_{i-1}^e\right)=-E_n^e \tag{2.5}$$

In (2.5) *E* denotes the strain energy, *e* denotes that the definition is carried out on a single element, *i* denotes the specific interval on the prescribe design displacement, and *n* denotes the total number of intervals on the design displacement. E_n^e is thus the total (numeric) strain energy of the removed element. There are two important aspect to denote in relationship to the above; firstly the *i* index can be considered to be the equivalent of the timestep *t* in an FE analysis as the two are coincident. With this in mind the second and more general significance of (2.5) becomes clearer: the algorithm now utilises time-history information; i.e. it utilises the

value of the previous time-step to assess the impact upon the objective function. Although this information is perhaps of limited use at present it may be worth analysing the potential of expanding this "time-history" further in order to enhance the credibility of the algorithm. (Huang and Xie, 2008a) states:

"The Eq. (15) means that the decrease of the total external work due to removing one element is equal to the total strain energy of the element in its final deformed state and irrelevant to the size of displacement intervals"

There are two main implications of the above statement. Firstly, the above statement is the basis for the claim in Huang and Xie, 2008) that the algorithm is mesh independent. In the authors opinion this statement is on the face of it quite bold, but in terms of technical significance it is of minor importance as all FEA per definition is mesh dependent. This means that despite the above statement the "input information" is arguably mesh-dependent; questioning the overall importance of the "mesh independence" statement.

The second main implication of the above citation lies within the intricate details in the wording, more specifically the term "final deformed state". This implies that ultimately only the final deformed state is of relevance and the intermediate steps are not. The validity of this statement can be justified in combination with the previous caveat of quasi-static loading, but arguably warrants further investigation in connection with optimisation of e.g. crash structures. A more holistic approach to this topic is of course related to the overall aim of any specific topology optimisation study. For example the design of a front end crash structure (where acceleration levels during the impact must be taken into consideration) is significantly different to a support bracket where only the final deformation is of relevance. In other words the significance of the dynamic nature of any given problem must be carefully considered. Returning to equation (2.5) (Huang and Xie, 2008a) defined the sensitivity of the (existing) element i as:

$$\alpha_i = E_n^i \tag{2.6}$$

Equation (2.6) thus confirms the above statement that the sensitivity α_i of element *i* is only dependent upon the last state n, i.e. the final deformed state. (Huang and Xie, 2008a) also states:

"*Note that the sensitivity number is a direct measure of the variation of the objective function due to element removal or addition. To maximize the total external work, the evolutionary process will be conducted by removing elements with the smallest sensitivity numbers and adding the elements with the highest ones. Mathematically such a procedure is known as the "hill-climb" method or the "steepest descent" algorithm.*"

The above citation thus indicates that the BESO algorithm fundamentally uses the Hill Climbing (HC) optimisation technique which will be subsequently presented in sub-section 2.2.5. This also reinforces the statement relating to the lack of "diversification" of the BESO optimisation algorithm significantly increasing the likelihood of converging towards local optimum solutions as previously discussed in section 2.2.1.

The focus of attention is now turned to the "filtering scheme" improvement in (Huang and Xie, 2008), i.e. step 2, Figure 1. As per equation (2.6) the sensitivity number for existing elements are extracted directly from the FEA. This is of course not possible for void elements, but it may be necessary to add elements to the model, e.g. in highly stressed areas. At this point it is convenient to expand the explanation of what constitutes a void element. It is important to remember that BESO optimisation works from a full Design Volume (DV) and subsequently voids (deletes) elements in each iteration as illustrated between Figure 2 A) and Figure 2 B). Finally void elements may be added as illustrated in Figure 2 C).

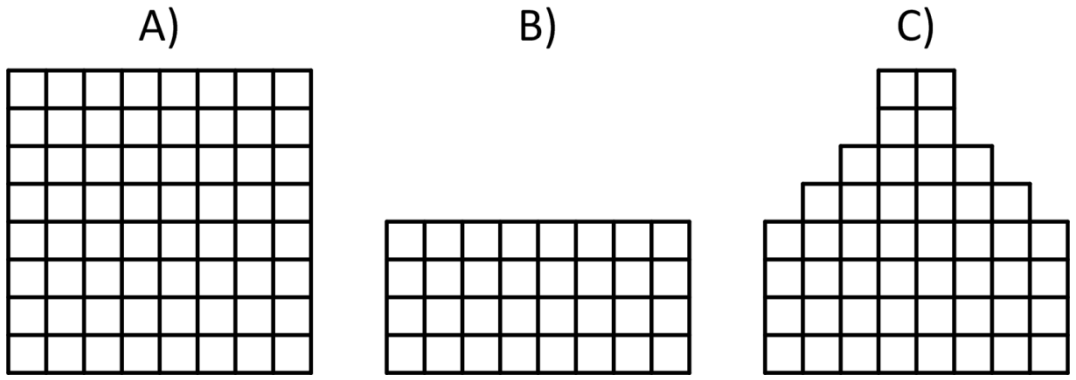


Figure 2, A) full design volume B) elements void (deleted) C) elements added

Two things should be noted when observing Figure 2; firstly that the elements added in C) were originally part of the full DV in A), hence the nodal coordinates are pre-defined. Secondly it should be noted that B) and C) in Figure 2 may occur within the same iteration step, i.e. elements may be both voided and added. The above may seem identical to the VDM / SIMP

approach, it is however important to remember that the VDM approach does not delete elements from the FE model between iterations, it merely reduces Young's Modulus. In BESO the elements are completely removed from the stiffness matrix of the FE model, however the nodal coordinates and element definitions remain in the memory of the optimisation software enabling the transition between Figure 2 B) and C); in other words the element is marked as void. It should also be noted that it is possible to initially define void elements. As an example the structure included in the FE analysis at iteration 0 is Figure 2 C), but the full DV of the optimisation is as illustrated in Figure 2 A).

The mathematical description for adding elements may be obtained by redefining the sensitivity numbers and simultaneously introducing the filtering scheme. (Huang and Xie, 2008b) therefore defines the sensitivity number for adding elements as:

$$\alpha_i = \frac{\sum_{j=1}^{N} \omega\left(r_{ij}\right)\alpha_j}{\sum_{j=1}^{N} \omega\left(r_{ij}\right)} \tag{2.7}$$

In equation (2.7) *N* is the total number of elements in the FE mesh, $\omega\left(r_{ij}\right)$ is a weighting function defined as equation (2.8), (Huang and Xie, 2008b).

$$\omega\left(r_{ij}\right) = \frac{\exp\left(-\left(\frac{\left(r_{ij}\right)^2}{2\left(\frac{r}{3}\right)^3}\right)\right)}{2\pi\left(\frac{r}{3}\right)} \quad \left| \begin{array}{l} \left\{i \in N \middle| r_{ij} \le r\right\} \\ j = 1, 2, \ldots, N \end{array} \right. \tag{2.8}$$

In equation (2.8) r_{ij} is the direct distance between the centres of elements *i* and *j; r* is a user selected filtering radius. Equations (2.7) and (2.8) thus underpin the discussions of section 2.2.1 which highlighted some of the similarities between CA / HCA and BESO; in this case the influences of the "neighbourhood". In relation to equations (2.7) and (2.8) (Huang and Xie, 2008b) state:

"The above sensitivity filter serves two purposes: (1) to extrapolate sensitivity number within the full design domain and (2) to smooth the sensitivity number in the neighbourhood of each element and prevent the problem of checkerboard patterns and mesh-dependencies in topology optimization."

With the above in mind the focus of attention can now turn to the 3rd improvement of the BESO algorithm proposed in (Huang and Xie, 2008); the new procedure for adding or removing element, step 5, Figure 1. The intricate details of the algorithm are of less relevance in the current context; what is important is the overall use of 0 (void) and 1 (full) element values. The revised procedure ranks and treats all elements in the model, i.e. both 0 and 1 elements, together. The main reason why this is significant is that Boundary Conditions (BCs) are not initially considered at this step (5, Figure 1), when determining the elements to void or add. This could for example lead to computational instability or even a significant change in the overall physics of the optimisation problem if e.g. an external load or constraint is removed. In an attempt to avoid this scenario step 6, Figure 1, is introduced. In this context (Huang and Xie, 2008c) states:

"*Once such a problem occurs, using a finer mesh or replacing the removed elements with "soft" elements to preserve the boundary or load can effectively circumnavigate problem.*"

(Huang and Xie, 2008) do not further specify the definition of a "soft" element, but it can be deduced that this is an element with lower stiffness or indeed lower volumetric mass density. Parallels can thus be drawn between the BESO algorithm and the VDM / SIMP algorithm discussed in section 2.1.

The final sections of (Huang and Xie, 2008) contain topology optimisation examples from both linear and non-linear models using BESO. The conclusions contain two separate statements of particular interest, (Huang and Xie, 2008d):

"Due to partial unloading of material, the defined optimization problem for nonlinear material is ill-posed and the solution may not always convergent because oscillation between designs of two different deformation modes."

Convergence is always a key aspect of any topology optimisation algorithm; the above statement reveals a very undesirable potential pitfall of utilising BESO in connection with non-linear material behaviour. Questions can be raised as to why this problem seemingly has not been further addressed since 2008; regardless of what the answer to these are it is certain that

if BESO is to be further refined for topology optimisation of non-linear structures the above must be a key area of investigation.

The second main finding according from (Huang and Xie, 2008d) is:

"*The examples illustrate substantial differences in the topologies of the linear and nonlinear optimal designs. Unlike the linear design which is independent of the design displacement, the nonlinear optimal design is highly dependent on the specified design displacement. The results show that the nonlinear design at its design displacement is always better than the linear design in terms of the total energy, which is the objective function*"

This is a very interesting statement which underpins the previous statement relating to the significance of the "final deformed shape"; but the above also clearly indicates that the final deformed shape is heavily dependent upon the previous stages of the deformation, which is hardly surprising from a mechanical engineering viewpoint. It does however also emphasise that the need for analysis into the direct importance and influence of the "intermediate deformation steps" upon the results of the topology optimisation must be further explored. In other words, should the intermediate energy levels, E_i, in equation (2.5) be further implemented into the optimisation algorithm? This will be further addressed in section 4.2.6.
The final remark to be noticed in the above citation is that using the BESO algorithm and test cases in (Huang and Xie, 2008) the nonlinear topology optimisation results always outperform the linear counterparts. If this is indeed the case it is suggested that applying non-linear topology optimisation algorithms to linear topology optimisation problems will lead to more optimal solutions. It is understood that this potential advantage comes at the cost of increased CPU time.
The review of the BESO algorithms is thereby complete. Despite a number of significant issues relating to the application and results from the BESO algorithm it also raises a number of potential benefits and general areas of interest which should be considered whichever non-linear topology optimisation algorithm is chosen going forward. The main advantages and disadvantages of the BESO algorithm are summarised in Table 2.

.

Table 2, advantages and disadvantages of BESO algorithm.

Advantages	Disadvantages
Strong, direct and relatively simple implementation into FEA.	Quasi-static loading assumption.
Local and global constraint control.	Non-convergent and potential oscillation between “equal” design solutions.
Direct link between objective function and FEA.	No direct consideration of “intermediate” deformation modes / levels.
Neighbourhood influence.	Relatively high computational cost.
Inclusion of gradient information (time-history)	Significant issues with high levels of non-linear material behaviour.
	Low dispersion of search within potential solution space, high probability of “converging” towards local optimum points.

All discussions throughout Contributions 1-4 of this PhD portfolio, section 2.1, sub-sections 2.2.1 and 2.2.2 of this monologue have predominately been focused on algorithms which are currently being used for structural optimisation on a large scale, e.g. in commercial FE software. All previous discussions, models and results have clearly indicated that none of these approaches clearly meets all the requirements of a topology optimisation algorithm for structures exposed to large deformations. The overarching requirements can in general be listed as:

1. Proof of convergence towards an optimum solution.
2. Reasonable overall computing requirements (CPU time, memory etc.).
3. Appropriate level of ability to cater for the natural complexity of the load scenario such as:
    a. Time dependent loading.
    b. Non-linear system response.
    c. Variations in material characteristics as a function of deformation.
    d. Overall system response versus localised effects.
    e. Relevance and feasibility of results to “real world” engineering.

The above list of "requirements" is by no means exhaustive and is in general open to interpretation and discussion. It is however clear that all individual algorithms discussed so far have severe limitations with one or multiple aspects of the above list. Furthermore it is unlikely that any one algorithm will be able to "fully" meet all requirements; ironically this problem can in itself be considered to be a multi-objective optimisation problem. With this in mind this review will now focus on optimisation techniques and algorithms in general. This means that the majority of algorithms presented will not previously have been applied to topology optimisation of structures exposed to large deformations using a rigorous and well documented basis. The potential application of the algorithms and the feasibility to meet the above list of requirements will be discussed throughout the following sections.

2.2.3 Combinatory Optimisation Problems

A Combinatorial Optimisation Problem (COP) can in general be perceived as an optimisation problem containing only discrete variables. Finite Element based topology optimisation falls into this category as each individual element should either be included or excluded from the structure, this could e.g. be done using the Boolean operators 1 (included) or 0 (excluded). As presented in Contribution 1, the introduction of the Variable Density Method (VDM) turns the problem into a continuous one. A further general requirement of COPs is that the solution space should contain a finite number of entries, i.e. discrete or alternatively a countable infinite set (Schrijver, 2003).

Due to the previously discussed curse of dimensionality it is clear that the reference to a "finite solution space" above rapidly becomes violated with respect to topology optimisation, but this may not necessarily be the case for the "countable infinite set". It is clear that most "real world" topology optimisation problems will have an infinite number of solutions, and so the "reduction" to a "countable number of infinite solutions" will most likely influence the results of such an algorithm. With this in mind, the study into COPs and the applicability to non-linear topology optimisation may continue.

COPs may further be divided into two main categories; Deterministic Combinatorial Optimisation Problems (DCOP) or Stochastic Combinatorial Optimisation Problems (SCOP). The difference between the two types of problems is easily understood through the following example. Consider the optimisation problem of creating packed lunches for a set upper sale price with maximum nutrition. The packed lunch may consist of eggs, meat and bread. Knowing the cost as well as the nutritional value of each item the DCOP can be solved relatively easily by creating an objective function of nutrition and a single constraint relating to

the cost. Imagine now that the “packed lunch business” will be selling lunches over the coming school year. No single supplier is however able to commit to a set price or even a consistent product. This means that both the price as well as the nutritional value of each item will vary. The optimisation problem is now more complex, as a certain level of “randomness” or “uncertainty” has been introduced; the nature of the problem has become stochastic, and the overall optimisation problem is now a Stochastic Combinatory Optimisation Problem (SCOP). Leaving the world of “packed lunches” and returning to the world of structural optimisation it is obvious that there are direct parallels between the above example and that of linear vs. non-linear topology optimisation. Although there are several practical differences the essence of the DCOP problem is (in its basic form) similar to that of linear optimisation; it is possible with great accuracy to “predict” the outcome of the optimisation and it can be justified that an optimum solution has been obtained, e.g. via the use of gradient based optimisation. The essence of the SCOP problem also holds parallels to the non-linear topology optimisation problem; it is very difficult if not impossible to predict the variations in the response of the system. Consider for example the task of consistently and repeatedly predicting the changes of structural response of a FE crash model when adding or deleting elements, this is the equivalent of adding uncertainty into the system. The stochastic nature of this system is most likely not completely random which indicates that it may be possible to introduce probabilities into the optimisation algorithm. With this in mind it is now useful to review algorithms which in general focus on SCOP; for simplicity it is however more convenient to introduce these in terms of DCOP and subsequently address the additional complications with respect to SCOP. (Bianchi et. al., 2008) completed a significant survey of meta-heuristic algorithms. This survey forms the basis of the explanations presented below.

2.2.4 Ant Colony Optimisation

Ant Colony Optimisation (ACO) is as the name suggests based on the natural behaviour of ant colonies and their individual worker ants. When ants forage they naturally seem to find a ”logical” and “effective” route between their nest and the food source; in other words they seem to determine an optimum route; this observed behaviour is the basis of ACO, (Bianchi et. al., 2008). Imagine two ants walking from the nest to a food source using two different routes. As the ants walk they release pheromones which naturally decay over time. The ant which (randomly) selected the short route will commence the return leg of the journey quicker than the other ant, thus reinforcing the pheromone trace on the shorter route. Other ants will instinctively follow the stronger pheromone path, reinforcing and even adding to this.

According to (Bianchi et. al., 2008) the ACO algorithm contains three major steps which constitute the central optimisation loop of the algorithm:

1. **Construct Ants Solutions**. This is the procedure for which "ants" incrementally and stochastically construct paths, i.e. solutions in a wider optimisation context.
2. **Evaporate Pheromone**. This is the process in which pheromone for certain solutions are decreased using "local" information, therefore this step is also often referred to as local update. This step is pivotal to ensure that the ACO algorithm does not prematurely converge to a single solution.
3. **Deamon Actions.** This step refers to decisions made based on global information relating to the optimisation problem. Note the difference between local in step 2 and global in step 3. Analogue to step 2, step 3 is also often referred to as global update.

The three steps highlighted above are repeated until the optimisation problem has converged or is otherwise terminated via a pre-specified termination condition.

Even though the above description is general and perhaps somewhat abstract in relation to non-linear topology optimisation and the much higher level of details in sub-section 2.2.2 the basic ideology of the ACO algorithm can be understood based on the above. Based on the above it is clear that significant development and analysis work is required before ACO could be applied to the non-linear topology optimisation problems of concern. It is also clear that it is unlikely for the ACO algorithm to be a "stand-alone solution" to the non-linear topology optimisation problem. It is perhaps more appropriate to see the ACO algorithm as something which could be merged into other structural optimisation algorithms, for example as an "evolution" of BESO, i.e.an algorithm which could be combined with that presented in section 2.2.2. Based on this relatively short review of ACO it is not feasible to detail a specific list of advantages and disadvantages of ACO for non-linear topology optimisation, quite simply because additional analysis and development work is required. It should however be noted that ACO has the potential to be combined with e.g. BESO in order to overcome the previously stated concerns with respect to the diversity of the search area for BESO algorithms.

2.2.5 Stochastic Hill Climbing

Stochastic Hill Climbing (SHC) is a variation of the more general optimisation search technique denoted Hill Climbing (HC). As presented in sub-section 2.2.2 HC is the fundamental search

method used in the BESO optimisation algorithm. HC is generally considered to belong to the "local search" techniques, the basic principles of HC is as follows. A starting solution is required, i.e. an initial guess for a viable solution is required; in the case of BESO this constitutes an initial FE analysis. Based on this initial guess, variations to the current solution are introduced based on some pre-defined rules and the newly "updated" solution is then analysed. The performance, i.e. objective function, of the "updated" solution is then compared to the "original" solution, if the former out-performs the latter it will be accepted as the "current" solution, and the iterative process will repeat. If the "original" solution out-performs the "updated" solution the former will remain the "current" solution and further variations of the "original" solution will be made. It is of course necessary to introduce termination criteria, e.g. maximum number of iterations. As discussed in discussed in sub-section 2.2.2, the specific implementation of the "pre-defined rules" constitutes the filtering and averaging of the element sensitivity numbers, i.e. steps 2 and 3, Figure 1. As highlighted in Table 2, questions about the "low dispersion of search within the solution space" for this type of algorithm. This statement is supported by (Bianchi et. al., 2008a):

"*Such a simple and very local general local search behaves quite poorly in practice, particularly because when a local optimum is found, the algorithm stops improving, and combinatorial problems often have a local optima whose objective values are much worse than that of the global optimum.*"

SCH attempts to reduce this problem by not only "obeying" the pre-defined rules for variation, but also introduce a certain level of "randomness", hence stochastic HC, by including larger (global) variations not governed by the local variation laws. The ideology behind this is an attempt to avoid converging towards local optimum points. It should however be noted that this "randomness" will still be governed by some rules in order to enable the implementation into numerical software. A very crude example of SHC adaptation of the BESO algorithm would be to "at random" select a pre-defined number of elements or perhaps percentage of the design volume to remove per iteration.

Although the SHC adaptation introduces a potential method for avoiding local optimum solutions it also has the unfortunate likelihood of introducing numerical instability into the FE model, let alone the optimisation problem in general. A perhaps more refined and systematic variation of HC is denoted Tabu Search (TS).

2.2.6 Tabu Search

As mentioned above TS is in essence a variation of HC. The ideology of TS is very much the same as SHC; to avoid premature convergence towards a local optimum solution. There are 3 general TS concepts, (Bianchi et. al., 2008):

1. **Best Improvement.** Consider a cluster of solutions, which are variations of each other as was the case with the HC above. Imagine a simple minimisation problem, where the current solution AA provides the objective value 5. Variations of S1, i.e. the "neighbours" AB and AC provide the values 6 and 7 respectively. AB is then chosen as the solution to replace AA because it is the "best neighbour", despite the fact that AA is overall a better solution. The reason behind this is simply to avoid local optimums by expanding the search area inside the solution space.
2. **Tabu Lists.** Following on from point 1 AB is now the current solution, the search is repeated. The neighbours of AB are AA and AC. Under the rules of "best improvement" AA will once again be chosen as the new solution quite simply because it is the best neighbour to AB. This is of course detrimental to the efficiency of the algorithm, as the solution will now simply enter into a cycle of swapping between AA and AB. This general phenomenon was also observed in Huang and Xie, 2008) where the convergence of the BESO algorithm was hindered by the oscillation between two separate solutions, as discussed in sub-section 2.2.2. TS attempts to avoid this by storing information about each solution, i.e. AA, AB and AC in a so-called Tabu list. Due to the scale of information involved and limits of memory available it is unfeasible to store each solution in its entirety, and therefore the "changes" from one solution to another is typically stored. By utilising a Tabu list moves to previous solutions can be prohibited.
3. **Aspiration Criteria.** The introduction of Tabu lists may therefore reduce or even remove the likelihood of oscillation between two solutions; it may however also prevent solutions which have not yet been explored. In extreme events this may lead to better local or even the global optimum solution being discarded from the search space. This may occur as it may not be possible to distinguish a previous solution from an "unexplored" one due to the fact that not all details of the individual solutions are stored in the Tabu lists. Therefore an "aspiration criteria" is introduced**,** a typical example of which is to stipulate that any new solution found must be better than the best solution found from the start of the algorithm, (Bianchi et. al., 2008).

Based on the above, the introduction of TS into the BESO algorithm holds the potential to reduce some of the disadvantages of the algorithm, namely the dispersion of the search as well as the oscillation between competing solutions.

2.2.7 Simulated Annealing

Simulated Annealing (SA) is an optimisation method which has received a lot of recent attention and general development within a wide range of different optimisation applications, e.g. (Bureerat and Limtragoool, 2008), (Garcia-Lopez et. al., 2011) and (Barati, 2014). The principle of SA based optimisation was first presented by (Kirkpatrick et. al., 1983) in order to physically model the temperature controlled annealing process. In other words, the SA optimisation principle has its roots in statistical mechanics a commonality it shares with entropy, which shall be presented in section 2.3.3. The SA algorithm may be classed as a "random search method" (Garcia-Lopez et. al., 2011), indicating that the fundamental algorithm has similarities with e.g. ACO, SHC and TS as previously presented. Analogue to e.g. SHC SA commences with a "random solution", i.e. an initial "guess" of a solution. SA then attempts to gradually obtain a more optimum solution by searching for alternative solutions within the neighbourhood of the current solution. This approach is equivalent to the controlled cooling required for the physical annealing process. The first step is also analogue to many other optimisation algorithms including SHC and TS. The method in which alternative solutions may be accepted as the "current" solution is however what sets SA apart from e.g. SHC and TS. There are two paths in which this can happen, (Garcia-Lopez et. al., 2011):

1. **Improved fitness value.** In this rule a new solution "B" is accepted to replace the current solution "A" if the objective function value of B is less than A (for a minimisation problem).
2. **Reduced fitness value.** In this rule a new solution "B" is accepted to replace the current solution "A" if the objective function value of B is larger than A (for a minimisation problem), given a certain probability.

The rule of point 1 above is clear and intuitive, but if used in isolation may lead to premature convergence towards a local optimum as extensively discussed throughout this monologue. The rule in point 2 above is perhaps counterintuitive. Firstly it is important to recognise that this rule is introduced to avoid premature convergence as per point 1 above. The probability

mentioned in point 2 above is where the direct link to the annealing process lies, as the probability $P(t)$ is calculated using equation (2), (Garcia-Lopez et. al., 2011a):

$$P(t) = \exp\left(-T^{-1}\Delta F_t\right) \tag{2.9}$$

In equation (2.9) ΔF_t is the difference in objective function value between the two steps (A and B using the above notation).T is the current "temperature", i.e. scaling parameter which is calculated as equation (3), (Garcia-Lopez et. al., 2011a):

$$T(t) = \alpha^j T_0 \quad \Big| \quad 0<\alpha<1 \tag{2.10}$$

In equation (2.10) α is referred to as the cooling parameter, *j* is the number of steps completed since the start of the algorithm, and T_0 is the initial temperature value.

As presented above, the fundamental principles of SA are relatively simple and easy to understand. The inspiration from the physical annealing process is clear, as are the rules introduced in attempt to circumnavigate the likelihood of premature convergence towards an inferior local optimum point.

As stated at the start of this section, several recent applications of SA in combination with other optimisation principles have recently been published. These will be briefly discussed below.

(Bureerat and Limtragoool, 2008) proposed to use SA in combination with Multi Resolution Design Variables (MRDV) in an attempt to reduce overall CPU time and checkerboard effects. The fundamental principle was to introduce a "translation function" altering the number of design variables between the actual FE analysis and the optimisation problem. This enables the number of elements in the FE analysis to be larger than the number of design variables in the optimisation problem reducing the computational effort of this step. (Bureerat and Limtragoool, 2008) conducted this for linear topology optimisation problems, and concluded:

"From the numerical test, it can be concluded that the use of MRDV results in the enhanced performance of the SA optimisation method. The optimum results obtained using SA are said to be comparable to those obtained from using the classic gradient-based approach."

(Bureerat and Limtragoool, 2008a)

The general principle of this paper is interesting, however based on the above the results are questionable both in terms of engineering feasibility and in terms of optimisation algorithm efficiency. Furthermore, the task of defining a robust translation function for non-linear analysis must be labelled as extremely difficult at best. Despite the above the general principle remains intriguing and the potential benefits most relevant to non-linear optimisation in general. The ideology is therefore worth further exploration once a suitable non-linear topology optimisation algorithm has been identified and developed, this does however fall outside the scope of this PhD.

(Barati, 2014) investigated the opportunities for utilising a combination of CA and Quasi Simulated Annealing (QSA) to solve a multi-objective optimisation problem relating to a nuclear reactor. Overall there is no difference between SA and QSA in terms of the definition of the two rules previously presented. The difference lies in where the rules are applied. SA applies the rules on a global level, where QSA applied the rules on a local level. Using the above example SA would apply the rules using the objective function, whereas QSA would apply the rules on the element level. This is where the CA method comes into play as it is used to define the neighbourhood influence of the individual element as described in sub-section 2.2.1. (Barati, 2014) did not use this approach for a “conventional” structural optimisation problem, the emphasis was on a fuel element for a nuclear reactor. The overarching task was to estimate the performance of the proposed algorithm in direct comparison with previously published work on Genetic Algorithms (GA), see sub-section 2.2.2 and Neural Networks (NN) see sub-section 2.2.9. The results obtained by (Barati, 2014) are not decisive, and merely refers to the potential and future work as opposed to drawing any detailed conclusions. The work is therefore interesting and may certainly serve as inspiration for future research directions, but does not provide significant detailed information for future work.

(Garcia-Lopez et. al., 2011) investigated the effects of combining SA with SIMP for linear topology optimisation using “classic” optimisation examples such as cantilever beams with an overall aim of reducing checkerboard effects. The methodology developed is referred to as SA-SIMP. Although the SIMP methodology is not of primary interest due to the previous work conducted throughout this PhD portfolio, the overall strategy of the SA-SIMP approach is intriguing. The process of SA-SIMP defined in (Garcia-Lopez et. al., 2011) commences with an initial solution obtained using “conventional” SIMP in connection with FEA. Following this the algorithm contains three major aspects:

1. **Selection Method.** This is the method in which "intermediate element densities" of the SIMP scheme are selected to be "upgraded" to 1 or "downgraded" to (nearly) 0 in the fixing method step. (Garcia-Lopez et. al., 2011) suggests four different ways of doing this: random selection, probability based selection, random cluster selection or probability based cluster selection. The relative difference between the first two and the latter two is in essence whether or not the selection is carried out on an individual element basis or on a cluster of elements (using a neighbourhood approach).
2. **Fixing Method.** As mentioned this is the step in which elements are "upgraded" or downgraded". (Garcia-Lopez et. al., 2011) suggests two different ways of doing this: random assignment or rounding assignment.
3. **Look-Ahead Strategy.** Following steps 1 and 2 described above a single "conventional" SIMP optimisation iteration is completed. The results of the SIMP run is then included within the SA algorithm selection basis as described at the start of this sub-section.

The results of (Garcia-Lopez et. al., 2011) found that the checkerboard effect was indeed reduced when using SA-SIMP when compared to gradient based SIMP. Furthermore the results indicated that using minimum compliance as the objective the results of SA-SIMP were consistently less compliant than the SIMP based counterparts. Even though this algorithm and associated results are not directly applicable to non-linear topology optimisation the overall methodology is certainly of interest for future work.
Based on the information within this sub-section the use of SA or a variation thereof could be combined with other optimisation techniques such as BESO and holds great potential to reduce some of the disadvantages of the BESO algorithm, namely the dispersion of the search as well as the oscillation between competing solutions.

2.2.8 Particle Swarm Optimisation

Particle Swarm Optimisation (PSO) is another "nature inspired" meta-heuristic optimisation algorithm. It was first proposed by (Kennedy and Eberhart, 1995) as a solution methodology to continuous non-linear problems. The fundamentals of PSO are based upon the social behaviour of animals hunting for food, such as a school of fish or a flock of birds. The original intent was:

"To graphically simulate the graceful but unpredictable choreography of a bird flock."

(Kennedy and Eberhart, 1995a)

PSO is a population based algorithm, in the initial formulation the population (or swarm) consisted of a number of "collision proof-birds" denoted as agents or particles, (Kennedy and Eberhart, 1995). As a starting point consider a simple 2D coordinate system wherein each particle (bird) is given a random initial (x,y) location. Imagine now that each bird represents a potential solution to an optimisation problem. For every iteration each individual particle is given the x and y velocity of its nearest neighbour, and thus the system of solutions is updated. This simple rule did however quickly result in the following observation:

"Unfortunately, the flock quickly settled on a unanimous, unchanging direction."
(Kennedy and Eberhart, 1995a)

(Kennedy and Eberhart, 1995) therefore introduced a stochastic variable which was initially labelled "*craziness*", which randomly changed x and y velocities of random individual particles in each iteration. Since the original introduction in 1995 there have been many minor adaptations and refinements of PSO, but the fundamental principles remain as described above. One of the more recent adaptations and implementations of PSO in connection with structural optimisation is published in (Luh, et. al., 2011), who introduced an adaptation denoted Binary PSO (BPSO). This adaptation predominately consisted of introducing an alternative type of particles, i.e. changing the rules for how particles "interact". The paper focused upon linear multi-modal topology optimisation problems, and concluded:

"The results show that the modified BPSO algorithm and part of the structures show very well defined truss-like members. Moreover, parts of the derived structures have the same weight but quite different topologies." (Luh, et. al., 2011a)

Based on the information above PSO also seemingly offer an extended search area when compared to the standalone BESO algorithm. When compared to e.g. SA, CA and TS it is worth noting that PSO computes multiple solutions (particles) per iteration, whereas the former 3 only compute 1 solution per iteration of the optimisation loop.

2.2.9 Neural Networks

Neural Networks (NN) also sometimes referred to as Artificial Neural Networks (ANN) is yet another nature inspired methodology, which is based upon the study of the central nervous system, and in particular the brain. The concept of NN was first devised in the early 1940s by W.S. McCulloch and Walter Pitts who:

"…tried to understand how the brain could produce highly complex patterns by using many basic cells that are connected together. These basic brain cells are called neurons…"

(Marsalli, 2006)

The original principle was not denoted NN, but threshold logic. As indicated above, NN was not initially meant as an optimisation tool, rather a modelling tool for complex non-linear problems, and has recently been adapted for structural optimisation on a small scale, e.g. (Lin and Lin, 2005). The central theme of NN is a pattern recognition methodology, in keeping with the original research question stated above. Just like the brain NN are capable of learning systems (patterns) which enables them to make predictions about the "future state" of the system under observation (Alavala, 2008). In mathematical terms a NN may be perceived as a single function $f\left(x_i\right)$ which can be used to evaluate the system based on input parameters (design variables) x_i . In order to "define" $f\left(x_i\right)$ the NN needs to be trained or learn. The learning mode of a NN is generally referred to as "machine learning" (or pattern recognition) and is commonly divided into 3 main categories, (Alavala, 2008):

1. **Supervised learning.** In this mode a set of example pairs y, z and the results $f\left(y\right)$ and $f\left(z\right)$ are known.
2. **Unsupervised learning.** In this mode input data is given along with an objective function, the result are however not available.
3. **Reinforcement learning.** In this more, input data is normally not given but established via e.g. another function.

In the ideal situation supervised learning would always be applied. However, as this data is most often not available one of the other learning modes must be utilised. In relation to non-linear (structural) topology optimisation it is perhaps most convenient to use reinforced learning mode. This is simply because the NN can be trained using other optimisation techniques, such as ESO, PSO or SA, (Alavala, 2008). NN should therefore not be considered as a stand-alone optimisation solution, but rather as a potential "additional refinement". The principle of NN is certainly captivating from an overall algorithm efficiency viewpoint, however it is obvious that there are significant obstacles to overcome if a NN is to be "trained" for non-linear topology optimisation. It is also worth noting that NN algorithms are widely available in numerical software packages, such as MatLab.

With this sub-section the overall literature review of optimisation algorithms is complete. It is understood that this is by no means an exhaustive review, however the general field of optimisation algorithms is so extensive that it would be impossible to cover every single algorithm in detail. The above review can therefore be considered to be a "selective" review of relevant and potential algorithms for non-linear topology optimisation. The next section of this chapter will focus upon other general aspects which should be considered during the development of the proposed non-linear topology optimisation algorithm.

2.3 General Principles

The purpose of this section is to present and discuss general principles and parameters which should be taken into account during the subsequent definition and development of the non-linear topology optimisation algorithm in chapter 3.

2.3.1 Robust Optimisation

Throughout Contribution 1 the term "robustness" was used to describe the "stability" or "sturdiness" of an algorithm, in other words its ability to avoid "error" termination for example due to numerical instability during FE analysis between iterations. The term "robust optimisation" does however have a very specific definition which is beneficial to introduce at this point:

"Robust design optimization consists of finding structural solutions such that the variability of the response – i.e., performance measure – is relatively insensitive to the uncertainty of the material properties and the demands. Most approaches to structural robustness evaluate the structural response in terms of displacement; however, others use the first and second moments of the response function."

(Garcia-Lopez et. al., 2013)

In other words, the principle of robust optimisation relates to an optimisation problem which considers the "noise factors" leading to uncertainty of the results. These are most often not parameters which can be controlled, and may include aspects such as the numerical inaccuracy of FE modelling. Robust optimisation is a field which has seen a lot of recent development, such as (Garcia-Lopez et. al., 2013a) and (Chen, 2012), no specific developments for non-linear optimisation have been found. Furthermore, the implementation of this principle will not be a top priority in the initial stages of the algorithm development and implementation. It may however be beneficial to consider potential avenues for implementation in the later stages of

algorithm refinement in an attempt to enhance the overall algorithm viability as well as feasibility of results.

2.3.2 Fuzzy Logic

In relation to the above introduction of robust optimisation and the repeated discussions relating to numerical instability in connection with topology optimisation and FE analysis in general, it may be beneficial to consider the principles of Fuzzy Logic (FL). FL may be considered as a form of "many-valued logic". Consider the simple question: "Is it Thursday today"? The obvious two answers to that question are "yes" or "no". In FL other (fuzzy) answers such as "maybe" are also allowed. In the context of the above example the advantages of FL are perhaps not clear, nevertheless the fundamental principle of FL is. If a "conventional" topology optimisation problem, such as a cantilever beam is now considered. To solve such a problem could be defined as "asking" a simple question on a single element basis: "should this element be a part of the structure"? The obvious two answers are again "yes" or "no", or "1" or "0" using Boolean operators. This strict regime may lead to numerical instability (of FE models in particular); and is one of the reasons why the Variable Density Method (VDM) is used for topology optimisation, where relative element densities of e.g. 0.2 are allowed. A further example is the introduction of "soft" elements discussed in (Huang and Xie, 2008) see sub-section 2.2.2 for further details. In section 2.1.3 where COPs were introduced one of the pretences was that the answers should be discrete, which initially rules out the concepts of FL. In addition to this one of the main drawbacks of VDM/SIMP is the checkerboard effect; an inherent side-effect of introducing FL (VDM). Despite these shortcomings it may be necessary to introduce a certain level of FL into the optimisation algorithm to ensure numeric stability.

2.3.3 Monte Carlo Simulation, Probability and Entropy

In sub-section 2.3.1 the definitions of robustness was discussed. This included the difficulty in predicting the change of results as a function of varying the design variables; in other words foreseeing the change in the system response (objective function) due to adding or removing elements from the FE model. A commonly used method of solving systems containing a large number of variables numerically is denoted the Monte Carlo Method (MCM). MCM is widely applied from engineering to financial forecasts, stock rates and manufacturing strategies. Many different versions of MCM or Monte Carlo Simulation (MCS) exist and are readily

available within commercial software packages such as Microsoft Excel and MatLab. According to (MatLab, 2014), MCS can be defined as:

"Monte Carlo simulation is a method for exploring the sensitivity of a complex system by varying parameters within statistical constraints."

In its basic form MCS is therefore not an explicit optimisation tool, rather a tool for prediction of sensitivity (robustness) of a particular system. MCS could therefore be an interesting "additional tool" for exploration of the robustness of results obtained from the non-linear topology optimisation algorithm, or alternatively be incorporated into the optimisation algorithm; this will however not be extensively investigated in this monologue.

The above citation also contains the phrase "statistical constraints" in connection with MCS. It may be possible to include these within the actual optimisation algorithm. This could for example be implemented in connection with the "intermediate element energy levels" discussed in sub-section 2.2.2; as will be further elaborated in chapter 3. Before this is done it is however convenient to introduce some statistical principles. The reason for introducing these is the desire to reduce the level of "randomness" in the optimisation by applying a more robust selection method for parameter variation, e.g. element deletion or addition in connection with BESO. In sub-section 2.2.2 it was stated that only the final deformed shape (element energy level) is used to inform the optimisation process. Expanding this to include intermediate levels may result in a more robust and accurate optimisation process, as shall be further investigated in chapters 0 and 4. The methodology used to implement these is however pivotal to the overall success of the algorithm, as a poor choice of implementation may add additional uncertainty, numerical instability and perhaps even cause divergence of the optimisation algorithm in general, an example of which could be an element subjected to high levels of hourglass energy in isolated time-steps. It has therefore been chosen to implement the principles of entropy from statistical mechanics into the optimisation algorithm.

Different types of entropy exist; throughout this monologue the term will refer to Shannon entropy unless otherwise specified. The phrase Shannon Entropy relates to the fact that Claude Shannon developed and expanded Jaynes' theory on entropy throughout several years. The term entropy is perhaps somewhat misleading as in this case it does not directly relate to the thermodynamical entropy property. In this context entropy should be interpreted as an

expression for the relative level of uncertainty of a system under consideration, this will subsequently be further expanded.

The principle of entropy (in the statistical sense) was first introduced by (Jaynes, 1957) where the aim was to use probability to determine the state (thermodynamical entropy) of a physical system. Jaynes' ultimate goal was to predict the final state of a system based upon sample data (from a much larger population) whilst avoiding making any "assumptions" about the system state not supported by the sample data; this was done using the entropy as an expression for the level of uncertainty. The idea was that maximising the uncertainty would minimise the "unfounded assumptions", hence Jaynes' method was entitled the Maximum Entropy Method (MEM) aka the Maximum Entropy Principle (MEP). A number of Maximum Entropy (ME) models have been derived such as Gibbs and Boltzmann distributions, Markov Random Fields, logistic regression, exponential families and Boltzmann machines, all serving different types of problems. A layman's description of MEP can be found in (Penfield, 2003a):

"*The Principle of Maximum Entropy is a technique that can be used to estimate input probabilities more generally. The results is consistent with known constraints expressed in terms of averages, or expected values, of one or more quantities, but is otherwise unbiased*"

Furthermore (Penfield, 2003a) also states:

"*Information theory, particularly the definition of information in terms of probability distributions, provides a quantitative measure of ignorance (or uncertainty, or entropy) that can be maximized mathematically to find the probability distribution that is maximally unbiased*"

The above two citations captures the essence of MEP in the context of which it will subsequently be used in connection with non-linear topology optimisation, as shall be explained throughout chapter 3. The derivations and explanations throughout this section generally follow those of (Jaynes, 1957) and (Penfield, 2003) although the notation has been adapted to avoid confusion. According to (Penfield, 2003) a simple form of entropy can be expressed as equation (2.11) (equation 9.2 (Penfield, 2003b)):

$$S = \sum_i \left[p(D_i) \log_2 \left(\frac{1}{p(D_i)} \right) \right] \tag{2.11}$$

In equation (2.11) S is the entropy, i.e. uncertainty and $p(D_i)$ is the probability distribution of for the individual states, i.e. design variables D_i of the system under consideration. As the natural logarithm with base 2 is used the entropy is measured in bits (Penfield, 2003). Also please note that as a consequence of utilising the natural logarithm in (2.11) the resulting probabilities assume a normal distribution. It is possible to use other distributions, but for the remainder of this monologue a normal distribution will be used. As per the rules of probability theory the sum of probabilities must equal 1, equation (2.12).

$$\sum_i p(D_i) = 1 \tag{2.12}$$

In order to aid the understanding of the principle of entropy S as well as detailing how S can be found in practice it is convenient to introduce a relevant example. This will serve two main purposes; firstly to explain the fundamental mathematics behind calculating the entropy, but will also indirectly highlights the strategy for the subsequent implementation into a non-linear topology optimisation algorithm.

Consider a simple FE model consisting of 4 elements exposed to predefined boundary conditions, the task is to obtain the lightest possible structure. Firstly four design variables, D_i are defined, in this example these will be interpreted as Boolean operators defining whether an element should be included (1) or excluded (0) from the structure, analogue to the VDM methodology. Initially there is no reason for penalising one element as opposed to another, hence the initial design variable values are all set to 1; expression (2.13):

$$D_1 = D_2 = D_3 = D_4 = 1 \tag{2.13}$$

At this stage it makes no sense to “select” the elements, as any such selection will be completely objective and heavily biased, but what about from an MEP viewpoint? As there it is impossible to distinguish between the four elements they should be assigned an equal probability, i.e. $p(D_i) = 0.25$, which leads to a non-biased solution with the highest possible entropy, which can be calculated using equation (2.11). This scenario / type of solution is generally referred to as Bernoulli’s principle of insufficient reason, and is of course not a relevant engineering solution. Additional information is therefore required enabling the determination of a better probability distribution in the sense of a lower entropy, i.e. uncertainty and a feasible structural solution. The MEP is a direct solution to this scenario and

may in general be considered to be an extension of Bernoulli's principle of insufficient reason (Jaynes, 1957).

The additional information should enhance the ability to distinguish the individual design variables from each other. In other words by introducing an overall (global) quantity Q for which there are local values associated (with each design variable). Consider for example the total strain energy Q of the model with the individual element strain energy values $q(D_i)$. A global constraint can thus be set up, equation (9.3) (Penfield, 2003b):

$$Q = \sum_i p(D_i) q(D_i) \tag{2.14}$$

Continuing the above example an FE analysis is now conducted, as a result the strain energy levels $q(D_i)$ are now known; expression (2.15):

$$\begin{aligned} q(D_1) &= 0.1 \\ q(D_2) &= 0.2 \\ q(D_3) &= 0.3 \\ q(D_4) &= 0.6 \end{aligned} \tag{2.15}$$

An average strain energy level Q of the entire model is now chosen:

$$Q = 0.3 \tag{2.16}$$

Based on the information in (2.15) and (2.16) which of the four elements should now be "selected" to remain in the model, and indeed how many should be selected for the structure with the lowest mass? It is obvious that based on the information in (2.15) there is still no unique solution to the constraint in (2.16). The constraint could for example be met by only selecting element 4 or alternatively by selecting elements 1, 2 and 3. With the given information there is no objective justification for choosing one solution over the other, so which should be chosen? According to the MEP the solution with the highest entropy should be chosen as to minimise any bias.

The above scenario is arguably a somewhat "idealised" scenario, firstly the constraint of 0.3 can "conveniently" be obtained by the element combinations as listed above, but what if the constraint was 0.34 or the $q(D_i)$ were "less neat" than listed in (2.15)? Although the

constraint of (2.16) can be manually set it is clear that more “versatility” needs to be implemented into the MEP. By introducing the Boolean operators above it was (indirectly) assumed that the elements could only be fully included or alternatively fully excluded from the structure. For calculation and FE stability purposes this may be undesirable. It may therefore be convenient to introduce continuous variables to define the inclusion or exclusion of the individual design variables; in other words allowing fuzzy logic into the solution of the system, analogue to the VDM method. The primary potential negative effect of introducing fuzzy logic is of course the checkerboard effect. The discussion of whether or not this should be included will be continued throughout the remainder of this monologue. Setting this discussion aside for the time being, maintaining the constraint in (2.16) and the element strain energies of (2.15) the problem can now be formulated using (2.12) and (2.14) - (2.16) two expressions for the system can now be defined in (2.17):

$$\begin{aligned} &p(D_1)+p(D_2)+p(D_3)+p(D_4)=1 \\ &0.1p(D_1)+0.2p(D_2)+0.3p(D_3)+0.6p(D_4)=0.3 \end{aligned} \tag{2.17}$$

The system in (2.17) is an indeterminate system as there are 2 equations and 4 unknowns. The amount of entropy S associated with (2.17) can be obtained using equation(2.11) leading to equation (2.18):

$$S=\begin{cases} p(D_1)\log_2\left(\dfrac{1}{p(D_1)}\right)+p(D_2)\log_2\left(\dfrac{1}{p(D_2)}\right)+ \\ p(D_3)\log_2\left(\dfrac{1}{p(D_3)}\right)+p(D_4)\log_2\left(\dfrac{1}{p(D_4)}\right) \end{cases} \tag{2.18}$$

For simpler problems (with e.g. 3 elements) it is possible to solve the above analytically by expressing one of the variables in terms of the other two. In general it is however more convenient to solve the system numerically by the utilisation of Lagrangian multipliers, this can be done using equation (10.12) (Penfield, 2003c):

$$L=S-(\alpha-\log_2 e)\left(\sum_i p(D_i)-1\right)-\beta\left(\sum_i q(D_i)\,p(D_i)-Q\right) \tag{2.19}$$

In (2.19) e is the base of the natural logarithm, α is a Lagrangian multiplier measured in bits and β is a Lagrangian multiplier measured in bits per unit of G. In the above example β is therefore measured in bits per Joule. The task at hand is to maximise L thereby maximising the entropy S. This can be done by differentiating (2.19) with respect to $p(D_i)$ and assuming S, α and β to be constants the result is equation (2.20):

$$p(D_i) = 2^{-\alpha} 2^{-\beta q(D_i)} \tag{2.20}$$

According to (Penfield, 2003), the entropy can be calculated without determining the probability distributions $p(D_i)$ as equation (10.18), (Penfield, 2003d):

$$S = \alpha + \beta Q \tag{2.21}$$

Equation (2.21) can therefore be used to determine the level of uncertainty of a given problem, a number which could potentially be used as a parameter in the overall topology optimisation algorithm as shall be discussed later. In order to get to this stage it is necessary to determine the values of α and β, the former, i.e. α can be calculated as defined by equation (2.22); (10.19) (Penfield, 2003d):

$$\alpha = \log_2 \left(\sum_i 2^{-\beta q(D_i)} \right) \tag{2.22}$$

(Penfield, 2003) states that in general the highest computational efforts associated with MEP relates to solving equation (2.22). There are however some additional caveats which should be noted:

1. If the system at hand contains a moderate number of states (potential solutions) and only one constraint solving equation (2.22) is not difficult. At this stage it should be noted that the constraint referred to does not necessarily equal the “structural” constraints, e.g. a maximum displacement constraint applied at the topology optimisation level, as shall be further discussed in chapter 3.

2. The level of difficulty rises considerably as the number of constraints increase. If there is very large number of solutions the solution cannot be obtained using equation (2.22)

, although the overall solution may still be obtained using Lagrangian multipliers (Penfield, 2003).

The two points above are critical to ensure that the MEP is applied in an appropriate context, and will be further addressed throughout chapters 0 and 4. Assuming that the conditions of point 1 above are met equation (2.22) is valid to determine α. In order to determine α it is necessary to know β which can be found using equation (10.22) (Penfield, 2003e); equation (2.23) is a slightly modified version of equation (10.22).

$$f(\beta) = \sum_i \left[\left(q(D_i) - Q \right) 2^{-\beta\left(q(D_i)-Q\right)} \right] \tag{2.23}$$

Equation (2.23) is the fundamental equation to be solved; with one unknown: β. According to (Penfield, 2003) L (2.19) and thereby S (2.18) are maximised when equation (2.24) is true:

$$f(\beta) = 0 \tag{2.24}$$

The overall MEP thereby "reduces" to solving equation (2.23) subject to equation (2.24). Please note that (2.23) does not contain any probability distributions, it is merely a function of the "sample values" $q(D_i)$ and the single constraint G. Solving equation (2.23) numerically under these conditions is relatively straightforward and may be conducted using a number of techniques including the gradient-based Newton-Raphson method.

Returning to the 4 element FE model example introduced at the start of this section it is now possible to calculate the entropy and individual probabilities associated with each element. Appendix A contains the MatLab code used to calculate this example based upon the MEP principles and equations presented in this section in combination with a Newton-Raphson iterative solver. The results can be found in Table 3.

.

Table 3, results from first numerical FE example.

Parameter	Value	Unit
α	2	Bits
β	5.0376e-8	Bits per Joule
Entropy (S)	2	Bits
$p(D_1)$	0.25	
$p(D_2)$	0.25	
$p(D_3)$	0.25	
$p(D_4)$	0.25	

When reviewing the results in table 3 the numerical values of α and β are of little interest, the primary interest lies in the actual entropy S and the probability values $p(D_i)$. The numerical value of the entropy on its own can perhaps be somewhat cumbersome to interpret. Remembering that the entropy is a measure of uncertainty the value becomes easier to comprehend; the higher the value the higher the level of uncertainty. The probability values $p(D_i)$ denote the likelihood of the individual design variable D_i being included in the system in order to meet the single constraint Q; equation (2.16) in this example. The example thus suggests that based on the input information there is no reason (probability) to use one element over another; this is indeed the least biased solution. The solution is indeed identical to the one initially obtained by calculating average values. The MEP does however provide a repetitive and systematic approach to obtaining such as a solution. Furthermore, the solution is provided per design variable (element) easing the implementation into topology optimisation. As previously discussed the above example is rather "idealistic" and could indeed be solved manually. Complications of this was previously discussed, such as changing the average strain energy value Q. In order to investigate the effects of this Q is now updated as defined in equation (2.25).

$$Q = 0.34 \tag{2.25}$$

With the change in equation (2.25) it is now much more difficult to manually assign probabilities to the individual design parameters, avoiding any particular bias. Using the MEP approach, the updated system was solved, the results of which are listed in Table 4.

Table 4, results from second numerical FE example.

Parameter	Value	Unit
α	2.2762	Bits
β	-0.88794	Bits per Joule
Entropy (S)	1.9743	Bits
$p(D_1)$	0.2195	
$p(D_2)$	0.2335	
$p(D_3)$	0.2483	
$p(D_4)$	0.2987	

Comparing the results in Table 4 to those of table 3 it can be seen that the entropy has decreased in Table 4, indicating that there is less uncertainty in the result. This claim is supported by the probability values which now distinguish between all four design variables. The probabilities $p(D_1)$, $p(D_2)$ and $p(D_3)$ have decreased, whilst $p(D_4)$ has increased. The results thus suggest that the "first" element to be removed should be D_1, and the "last" element to be removed should be D_4. This is in accordance with "engineering intuition" which would remove the element containing the least deformation energy in order to maximise the efficiency of the structure (in terms of energy absorption per unit mass). To draw a general conclusion based on this observation is however not as straightforward as it may seem. To understand why the average strain energy value is now changed according to equation (2.26).

$$Q = 0.1 \tag{2.26}$$

The results from the change in (2.26) are listed in Table 5

Table 5, results from third numerical FE example.

Parameter	Value	Unit
α	-0.12196	Bits
β	9.0767	Bits per Joule
Entropy (S)	0.89547	Bits
$p(D_1)$	0.5376	
$p(D_2)$	0.2866	
$p(D_3)$	0.1527	
$p(D_4)$	0.0231	

Firstly observing the entropy in Table 5 reveals that the level of uncertainty has considerably reduced when compared to the first two examples. According to Table 5 the "element ranking order" in which the elements should be chosen for inclusion in the FE model to meet the single constraint in (2.26) should now be: D_1, D_2, D_3 and finally D_4. From a purely probabilistic viewpoint this makes perfect sense; from a purely structural engineering viewpoint this is counter intuitive as removing the element with the largest strain energy will inadvertently have an effect on the remaining elements. Hence the implementation of MEP in topology optimisation is not necessarily a case of choosing the elements with the highest probability values, as shall be further discussed in chapters 0 and 4. For the next example the average strain energy value is now defined as equation (2.27) .

$$Q = 0.6 \tag{2.27}$$

The results from the updated constraint value are listed in Table 6.

Table 6, results from the fourth numerical FE example.

Parameter	Value	Unit
α	46.1277	Bits
β	-76.8795	Bits per Joule
Entropy (S)	1.6534e-7	Bits
$p(D_1)$	0.0000	
$p(D_2)$	0.0000	
$p(D_3)$	0.000	
$p(D_4)$	1.0000	

The entropy in Table 6 is in effect 0 bits, indicating that there is no uncertainty in this particular example; only one element should be used; D_4. Even though it must be concluded that this is the "correct" solution according to the MEP it may not necessarily be a useful solution in a structural optimisation context as shall be further discussed in chapter 3.

With this the initial presentation of MEP complete the starting point of chapter 3 is to initially define the non-linear topology optimisation algorithm.

3. Definition and Implementation of Non-linear Topology Optimisation Algorithm

The purpose of this chapter is to firstly justify and outline the topology optimisation algorithm embedded within the software in Contribution 6 to this PhD portfolio, and secondly to document how the software was created. The first step is therefore to overall compare and rank the optimisation techniques and principles which were presented in the previous chapter.

3.1 Algorithm definition

The task of defining a topology optimisation algorithm can in effect be divided into two main sections; firstly to choose/define the overarching principle of the algorithm, and secondly to add/change "minor" details relating to specific attributes of the algorithm, e.g. determine how neighbouring elements affect each other. The first part of this section will therefore be dedicated to defining the overarching algorithm. As per the discussions in chapter 2 these were initially grouped into two categories; mathematical and meta-heuristic. The severe limitations of the pure mathematical algorithms have been continuously discussed throughout all contributions to this PhD; these algorithms are therefore dismissed as a potential backbone for the optimisation algorithm. The second major group of algorithms discussed in chapter 2 were the meta-heuristic algorithms; which have already been used in commercial software such as LSTCs LS-TaSC. The fact that is has been used in commercial software does not necessarily substantiate the suitability of the algorithm as has been shown for the ESLM method in Contribution 4. Many specific meta-heuristic algorithms were presented and discussed in section 2.2; remembering that the overall "trade-mark" of this type of algorithm is the combination of physical theories and practicality in terms of computational efficiency. At this point it is convenient to remember the list of requirements set forth in sub-section 2.2.2, which included the need for "reasonable overall computing requirements". The potential trade-off between precision and algorithm efficiency (speed) makes the meta-heuristic algorithms attractive and is the main reason why the overall methodology of the optimisation algorithm will be based in meta-heuristics. As presented in chapter 2 there are already (at least) two reasonably well established meta-heuristics based non-linear topology optimisation algorithms; HCA and ESO including variations thereof, such as CAA and BESO. As these algorithms have been analysed and developed over a number of years they could provide a good starting point for further algorithm development; this is of course subject to scrutiny and careful assessment of the advantages as well as the disadvantages of the individual algorithm

in addition to feasibility of improving the algorithm. Starting with HCA as presented in sub-section 2.2.1 one of the main drawbacks of this algorithm is it lack of time-history data; the optimisation is in essence "isolated" iterations (with no data directly shared between optimisation iterations) in addition to this the optimisation is based upon the final deformed shape of the component or system being optimised. In the authors opinion these are significant shortcomings of the algorithm as the time-data history between iterations is important for algorithm stability and control as shall be further discussed in section 4.2.6. In addition to this it is suggested that considering the intermediate stages of deformation is pivotal to obtaining an appropriate level of accuracy for optimisation scenarios of structures exposed to large levels of non-linear deformation. In the authors opinion the principle of HCA is therefore fundamentally flawed in terms of suitability for non-linear topology optimisation.

This claim is substantiated by (Singh, 2014) and (Chuang, 2012) who stated:

"Several issues and limitations were discovered during the release of LS-TaSC. One main concern is the HCA algorithm accuracy and robustness during optimization process. It has been seen through published examples that numerous iterations are required in order to achieve better visible topology optimization results. The non-periodic oscillatory behavior also found on the objective function (total internal energy density) history is an issue. These raise the concern of the robustness of HCA algorithm and eventually lead to computation inefficiency when applying to the large scale industry application problems." (Chuang, 2012a)

(Chuang, 2012) also includes other case studies of ESLM and a method denoted Inertia Relief Method (IRM) which is very similar to that presented in Contributions 1, 2 and 3 of this PhD Portfolio. Based on the above statements it was chosen to abandon the HCA algorithm as a starting point for the non-linear topology optimisation algorithm. This leaves only one type of "developed" meta-heuristic algorithm; the evolutionary based algorithms. As previously presented this includes three main derivations; the Bi-directional ESO (BESO) was determined to be the most suitable one.

Throughout sub-sections 2.2.1 and 2.2.2 some of the similarities between HCA and BESO were highlighted in addition to the differences. To summarise, the BESO algorithms did resolve one of the fundamental issues of HCA, the gradient information (time-history) link between iterations. There were however some disadvantages repeated between both algorithms, such as the low dispersion of search within the potential solution space, the relatively high

computational cost and the lack of consideration for intermediate deformation modes. Despite these BESO was considered a good starting point for a non-linear topology optimisation algorithm, as it was deemed feasible to resolve or reduce many of the disadvantages by implementing various principles and techniques.

Acknowledging that it might not be possible to fully resolve all disadvantages of BESO the importance of each disadvantage to the overall success of the revised algorithm was prioritised, as well as the initial thoughts on how to improve the specific points are listed in Table 7.

Table 7, importance of resolving BESO disadvantages.

"Original" BESO advantage from Table 2	**Importance of resolving disadvantage**	**Initial thoughts for improvement**
Quasi-static loading assumption.	Medium	The quasi-static load assumption is overall consistent to that of explicit FEA, an adjustment in timestep size may be sufficient.
Non-convergent and potential oscillation between "equal" design solutions.	High	Merging BESO with other optimisation techniques, e.g. SA or TS.
No direct consideration of "intermediate" deformation modes / levels.	High	Introduction of statistical measures such as entropy to enable the inclusion of intermediate values in a systematic and repeatable manner.
Low dispersion of search within potential solution space, high probability of "converging" towards local optimum points.	High	Merging BESO with other optimisation techniques, e.g. SA or TS.

"Original" BESO advantage from Table 2	**Importance of resolving disadvantage**	**Initial thoughts for improvement**
Relatively high computational cost.	Low	The accuracy of the algorithm is the main priority; it may be possible to reduce the optimisation iterations to FEA ratio by the use of e.g. meta-modelling or an outer entropy based loop.
Significant issues with high levels of non-linear material behaviour.	Medium	The sensitivity to high levels of non-linear material behaviour must be reanalysed following the algorithm moderations; it is believed that the measure introduced to reduce the oscillation between solutions will also reduce this issue.

Figure 3 illustrates the updated BESO flowchart of Figure 1, demonstrating in which steps the proposed changes will be incorporated.

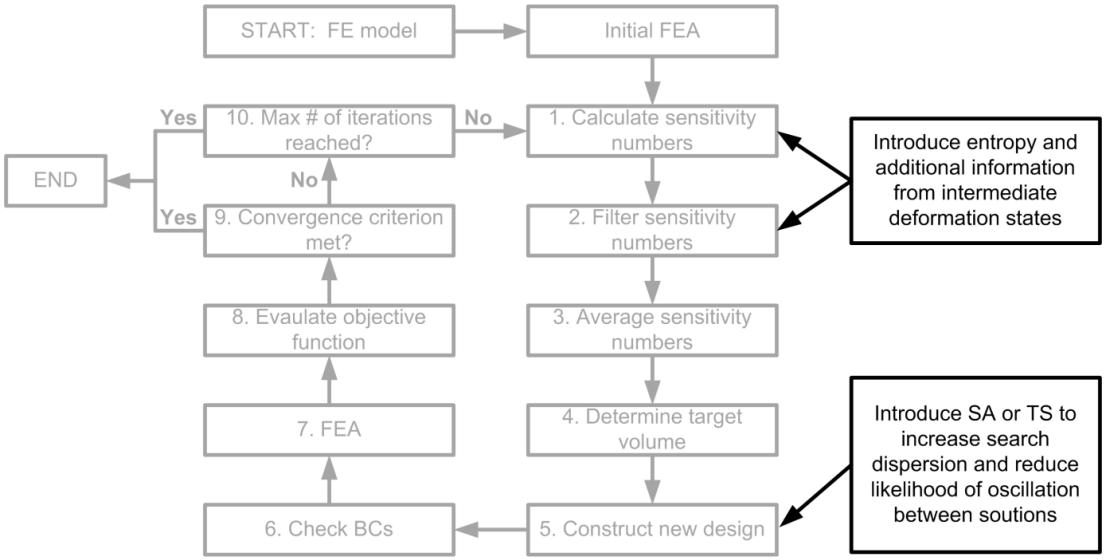


Figure 3, proposed algorithm flowchart.

As indicated in Table 7 and Figure 3 the only two optimisation techniques chosen are SA and TS. The primary reason for this is that these two approaches are believed to be the only two simultaneously "offering" a solution to the "low dispersion" and the "oscillation" disadvantages listed in Table 7. Many of the algorithms presented in section 2.2 offer potential solutions to the low dispersion disadvantage, but not to the oscillation disadvantage; hence all other algorithms presented have been discarded. Please note that if the ranking presented in Table 7 was revised other algorithms may once again be taken into consideration. Another aspect which may not be immediately clear from the above is that at this stage FL has not been introduced into the optimisation algorithm, hence the optimisation problem remains discrete, as the element inclusion or exclusion may be defined using the Boolean 0 1 operator. The following sub-sections will present and discuss the changes to the individual algorithm steps outlined in Figure 3. It is important to note that the following presentations and discussions will only consider full DV, i.e. no initial void elements are defined.

3.1.1 Calculate Sensitivity Numbers

As presented in sub-section 2.2.2 there are two different ways in which the element sensitivity numbers are calculated. For "existing" elements the sensitivity number is directly obtained from the FEA data, as defined in equation (2.6). For void elements, i.e. elements which could potentially be added the sensitivity numbers are calculated as defined in equation (2.7); in the authors opinion there is no need to adapt the sensitivity calculation for void elements. As discussed throughout chapter 2 it is however suggested that the consideration of intermediate deformation states (energy) is key to enhancing the overall algorithm robustness and accuracy, meaning that equation (2.6) should be revised. There is of course many different ways to do this, the simplest would be to utilise a mean or median value for each element, although this

might lead to numerical instability if the strain energy as a function of time is somewhat exponential for example.

Other statistical principles such as the Root Mean Square (RMS) value could also be an option. In the authors opinion there are two main factors which should be taking into account when defining how the element sensitivity numbers should be calculated:

1. The calculation should be as objective as at all possible, i.e. unbiased.
2. The discretisation of the overall physical system should be considered; i.e. FEA is merely an approximation of the "real world structure"; strain energy levels may be high in individual elements but considerably lower in the overall "local area" of the structure. The real world structure does not experience the "peak" element levels, but rather the overall local area level.

Point 1 above thus suggests using an unbiased selection procedure to statistically represent the state of the structure; an "ideal" choice would be the MEP as presented in sub-section 2.3.3.
Point 2 suggests that the element sensitivity should not solely be determined by the individual element, but also be influenced by its immediate neighbours. As presented throughout chapter 2 there are many different ways of establishing neighbourhood influence such as the Von Neuman neighbourhood. The BESO algorithm does however already contain a neighbourhood weighting function, equation (2.8), which is deemed more suitable as it also takes the distance between element centres into account, thus also considering the mesh size. The intended changes to this step therefor consist of two main aspects; firstly to introduce entropy as presented in sub-section 2.3.3 and secondly to introduce the weighting function of equation (2.8). Figure 4 illustrates the intricate steps of the step 1 of the overall algorithm in Figure 3.

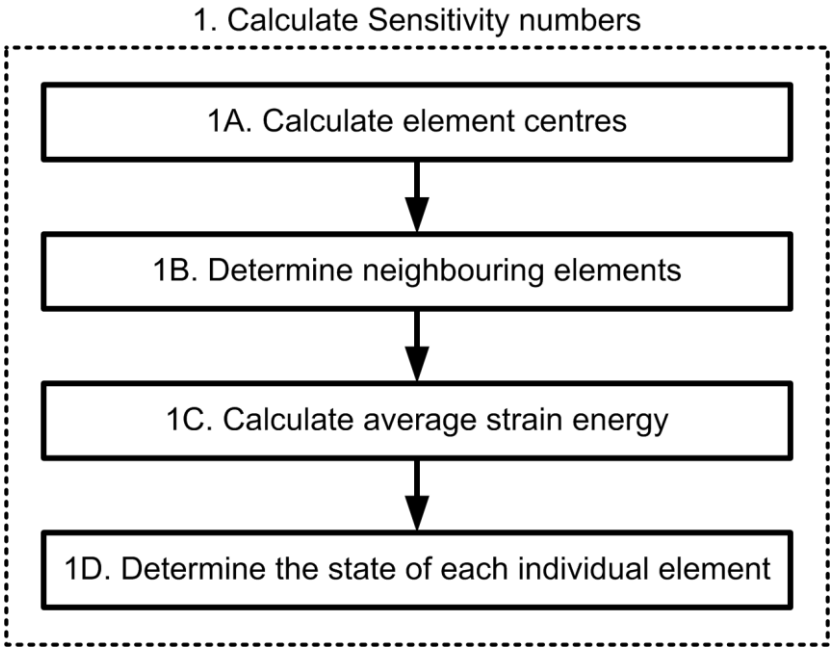


Figure 4, detailed flowchart of step 1.

Steps 1A and 1B of Figure 4 are relatively straightforward, they simply consist of determining the element centres as a function of nodal coordinates and element type, in addition to determining neighbouring centres using the calculated element centre coordinates and the user defined radius r_{ij} previously introduced in connection with equation (2.8). Figure 5 illustrates how varying r_{ij} values will determine the size of the "neighbourhood" and indirectly influence the subsequent average strain value calculations.

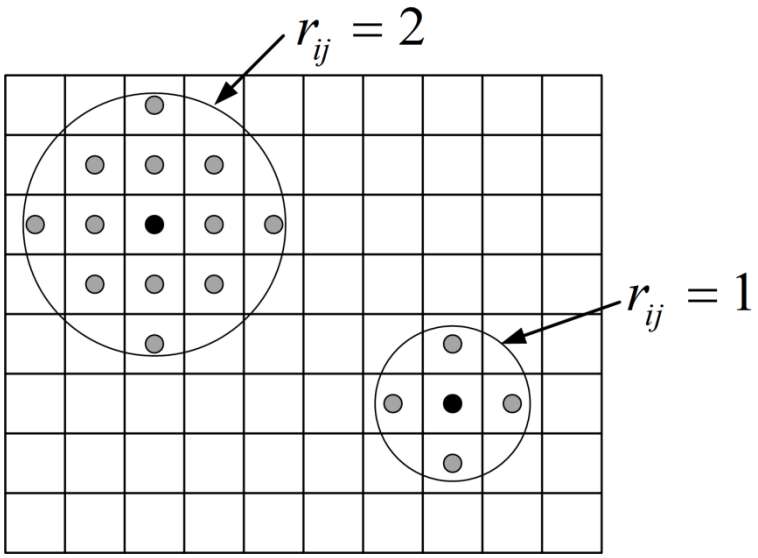


Figure 5, illustration of neighbourhood influence and strain energy calculation.

The first change of the BESO algorithm is incorporated into step 1C, which involves calculating the average strain energy of each "neighbourhood". Please note that it is possible to use other quantities than strain energy; however it was deemed that the strain energy is a good representation for the "condition" of the element, and will therefore be used throughout the remaining derivations and discussions. If it was subsequently desired to use a different quantity this would only constitute in minor changes to the software in contribution 6.

The average strain energy is calculated using the data from 1A and 1B in addition to the strain data obtained via FEA, the average strain data calculations are therefore indirectly a function of the user defined r_{ij} value as indicated in Figure 5. As previously discussed the influence of each neighbouring element will be weighted as a function of distance, i.e. $\omega\left(r_{ij}\right)$ as defined in equation (2.8). The calculation of Average Neighbourhood Strain energy ANS_i to be used in connection with the determination of the sensitivity number of the existing element i can therefore be defined as equation (2.28).

$$ANS_i = \frac{\sum_{t=\Delta t_{start}}^{\Delta t_{end}} \left(E_i(t)\right)}{N\Delta t} + \sum_{j=1}^{NN_{els}} \left[\frac{\sum_{t=\Delta t_{start}}^{\Delta t_{end}} \left(E_j(t)\right)}{N\Delta t} \cdot \omega_{ij} \right] \tag{2.28}$$

In (2.28) NN_{els} is the number of neighbourhood elements, Δt is the timestep from FEA, $E_j(t)$ is the strain energy of element j at time t and $N\Delta t$ is the total number of time steps included in the calculation. Please note that equation (2.28) allows for the number of time steps to be included to be user defined, per default this value will be set to include all time steps from the FEA, the effects of this will be subsequently investigated in chapter 4. With the average neighbourhood strain energy ANS_i determined it is now possible to determine the "probable" state of element i using MEP. This will be conducted analogue to the Lagrangian multiplier single constraint example of sub-section 2.3.3. Using equation(2.28) equation (2.23) can now be updated to equation (2.29), where $ESE_i(t)$ is the strain energy of element i at time t.

$$f(\beta) = \sum_{t=\Delta t_{start}}^{\Delta t_{end}} \left[\left(ESE_i(t) - ANS_i \right) \cdot 2^{-\beta\left(ESE(t)-ANS_i\right)} \right] \tag{2.29}$$

Using equation (2.29) probabilities for representative value of the element strain energy can now be calculated which incorporates neighbourhood as well as time history information, i.e. intermediate deformation stages. The value with the highest probability will be chosen as the representative strain energy for the specific element.

This means that steps 1 and 2 of the original BESO algorithm flowchart of Figure 1 are incorporated into a single step; using the flowchart in Figure 4 for "existing" elements and the "original" BESO sensitivity calculation of equations (2.7) and (2.8) for void elements.

3.1.2 Average Sensitivity Numbers and Determine Target Volume

Steps 3 and 4 of the original BESO algorithm in Figure 1, i.e. i.e. "average sensitivity numbers" and "determine target volume" remains unchanged at this stage. It is however necessary to describe how these are implemented as this was not fully presented in sub-section 2.2.2.
Step 3 in Figure 1 "average sensitivity numbers" may initially seem obsolete following the significant changes introdued in how the sensitivity numbers are calculated as definied in sub-section 3.1.1. However the averaging of step 3 serves a different purpose than that introduced in step 1. In step 1 the "averaging" was based on values obtaine from timesteps from a single FE analysis of a defined structure within one optimisation iteration. The "averaging" of step 3 refers to sensitivity values between separate optimisation iterations, i.e. between "different

structures". The average Element Sensitivity Number (ESN) is therefore a function of element number i as well as optimisation iteration number j, equation (2.30):

$$ESN_i = \frac{ESN_i(j-1) + ESN_i(j)}{2} \tag{2.30}$$

Equation (2.30) thus introduces a time-history link between optimisation iteration loops. Step 4 in Figure 1 "determine target volume" inadvertently decides the number of elements to void and / or add to the structure for the next optimisation iteration. The first step is simply to calculate the individual volume of each solid element or alternatively the area of each shell element; this can be straightfowardly completed using nodal coordinates. The volume or surface area V_j of the structure for iteration j can thus be obtained by summation of the individual elemental volumes / areas. The volume is controlled by two user defined parameters: Target Volume Fraction (TVF) and Evolutionary Ratio (ER). TVF simply dictates the target fraction of the DV remaining after topology optimisation. For example a TVF value of 0.1 means that 10% of the origininal DV or area at iteration 0, $V(0)$ should remain at the end of the optimisation process. Thus the rule in expression (2.31) must be followed.

$$V_{end} \simeq TVF \cdot V_0 \tag{2.31}$$

In most cases it will ofcourse not be possible to strictly adhere to expression (2.31), hence an acceptable residual value should be defined. Expression (2.31) and the residual thus also represent the "convergence criterion" of step 9, Figure 1. Provided that the convergence criterion has not been met the Target Volume TV_{j+1} for the next iteration is calculated using V_j and ER as defined in equation (2.32).

$$TV_{j+1} = V_j \cdot (1 \pm ER) \tag{2.32}$$

The $\pm$ in (2.32) depends on whether V_j is larger or smaller than V_{end}. ER does in essence determine the "aggressiveness" of the algorithm; a relatively large number will allow a relatively large fraction of the DV to be removed or added in a single iteration, whilst a relatively small number will have the opposite effect. The ER value is therefore a key factor in determining the effectiveness of the algorithm, as well as ensuring the robustness of it. As presented above, the ER parameter is a constant once initially defined by the user. It is

however possible to investigate the effects of making this parameter a variable, for example by using the probability function $p(B)$ principle from SA as defined in equation (2.9). This will however not be further explored for the time being. The new design, i.e. the structure at $j+1$ can therefore now be determined using the ESN and TV parameters.

3.1.3 Construct New Design

Following the revised calculation of the ESN and TV parameters as presented in the previous sections the new design for the next iteration can now be created. This is a multistep process, step 5 in Figure 3, the overall flowchart of this specific step is illustrated in Figure 6.

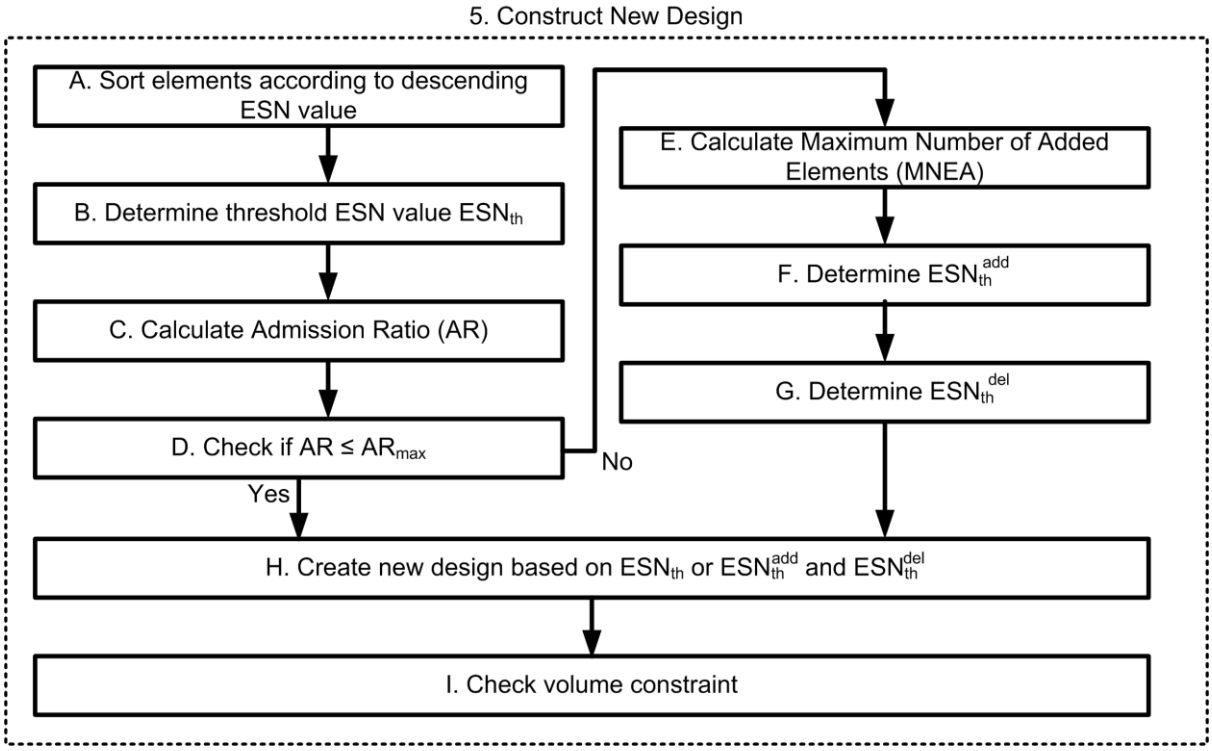


Figure 6, detailed flowchart of step 5.

At this point it is important to remember that in the revised BESO algorithm presented in sub-section 2.2.2 existing and void elements are evaluated simultaneously. Step 5A may therefore be considered to be a matrix with one row per element in the DV, with the following header: element number i, ESN_i, element volume and current element status (existing or void). In step 5A this matrix is sorted according to descending ESN_i number, regardless of element status. In Step 5B ESN_{th} is determined by summarising the individual Element Volume EV_i according to the descending ESN_i values until expression (2.33) becomes true.

$$\sum EV_i \geq TV_{j+1} \tag{2.33}$$

ESN_{th} is then obtained by setting it equal to the ESN_i value of the element which makes expression (2.33) true. In step 5C an Admissions Ratio $\left(AR_j\right)$ is calculated as defined in equation (2.34).

$$AR_j = \frac{\mathrm{NEA_j}}{\mathrm{CNE_j}} \tag{2.34}$$

In (2.34) NEA_j is the Number of Elements Added and CNE_j is the Current Number of non-void Elements in the model. NEA_j can be determined by counting the number of elements with a current void status for which expression (2.35) is true.

$$ESN_i > ESN_{th} \tag{2.35}$$

Next a maximum Admission Ratio $\left(AR_{\max}\right)$ is introduced; this is a user defined valued which indirectly dictates the maximum number of elements allowed to be added in any single iteration j. The primary reasons for introducing this factor are numerical stability and overall algorithm robustness. Step 5D is a simple comparison of AR to $AR_{\max}$ thereby checking whether or not too many elements have been added in the current optimisation iteration. If this is not the case the process will proceed to step 5E; if this is the case the process will proceed to step 5H.

The loop consisting of steps 5E, 5F and 5G in Figure 6 are as previously mentioned introduced to enhance algorithm robustness; these steps become active if too many elements are added in a single iteration. The first step in this loop, 5E, is simply to determine the Maximum Number of Added Elements $MNEA_j$, and is calculated as defined in equation (2.36).

$$MNEA_j = AR_{\max} \cdot CNE_j \tag{2.36}$$

$MNEA_j$ can now be used to determine a revised threshold value for added elements ESN_{th}^{add} in step 5F. This value found by ranking the void elements only according to descending ESN_i value and choosing the entry equal to $MNEA_j + 1$. The total Volume of the Added Elements

VAE_j can then be calculated by summation of the void elements for which expression (2.37) is true.

$$ESN_i > ESN_{th}^{add} \tag{2.37}$$

With the ESN_{th}^{add} value determined a corresponding value must be found for the deletion (void) of elements, step 5G, this is carried out according to the previously defined TV_{j+1} value ensuring that the Volume of Removed Elements VRE_j is as defined in equation (2.38).

$$VRE_j = V_j + VAE_j - TV_{j+1} \tag{2.38}$$

Once VRE_j has been found a revised threshold value for deleted elements ESN_{th}^{del} can be found by sorting all non-void elements according to descending ESN_i values and subsequently summation of the EV_i values until expression (2.39) is satisfied.

$$\sum EV_i \geq VRE_j \tag{2.39}$$

ESN_{th}^{del} is thus equal to the ESN_i value of the element which fulfils (2.39).

The next step of the flowchart in Figure 6 is 5H "create new design". The specifics of this step will either be based using ESN_{th} or ESN_{th}^{add} and ESN_{th}^{del} depending on the outcome of step 5D. Despite this the "input to step 5H is in essence the same; a list of elements which will compose the structure in iteration $j+1$, hence the following explanation will only utilise the ESN_{th} notation as the differences between the two possibilities are minute.

In the original BESO algorithm presented in sub-section 2.2.2 this step is relatively trivial as it merely consists of "reassigning" 0 or 1 values to the individual elements thus creating an updated FE model for the following iteration. However as identified in sub-section 2.2.2 and summarised in Table 2 one of the weaknesses of the original algorithm is its lack of ability to disperse the search area within the solution space and oscillation between two potential solutions. Step 5H presents an obvious opportunity for implementing SA or TS to alleviate these tendencies as previously discussed. The introduction of either of these will most likely have various effects upon the effectiveness and indeed robustness of the algorithm. SA does

perhaps represent the most “diverse” of the two proposed changes, as it is classed as a “random search method” as presented in sub-section 2.2.7. the potential positive effects of this is that it ensures a significantly improved dispersion of search within the potential solution space; this may also be one of the negative aspects of the methodology, as “inferior” solutions may be accepted as a function of the “temperature” potentially causing divergence of the optimisation as opposed to convergence. In addition SA offers no immediate solution to the oscillation phenomenon previously discussed. TS on the other hand does offer a relatively straightforward way of preventing oscillation by using the Tabu Lists and Aspiration Criteria discussed in 2.2.6. On the other hand TS does perhaps not offer as wide a search pattern as can be expected from an SA algorithm. This may be a disadvantage but the effects of this are presently unknown. One of the base assumptions throughout this PhD portfolio has been the notion that the level of accurate structural performance can only be obtained using FEA. Hence it follows that FEA provides good estimations of the current performance of the structure and indirectly provide good indications of load paths, regardless of the unpredictive nature of the overall problem. Based on this as well as weighting the criteria listed in Table 7 it has been chosen to implement TS as an addition at stage 5. As stated in sub-section 2.2.6 TS is in essence a variation of HC, the “original” solution methodology of BESO which may indicate a smoother transition and implementation of TS.

The revised algorithm will therefore utilise aspects of **B**i-directional **E**volutionary optimisation, **E**ntropy, **T**abu search and **S**imulated annealing methodologies and will therefore be denoted BEETS.

3.1.4 Tabu Search Implementation

The focus is now returned to the specific task of implementing TS in step 5H, Figure 6. This can be done in many different ways, however emphasis has been added to ensure that the number of iterations and thereby FE analyses does not become excessive, significantly increasing computing requirements. It was therefore decided to implement TS with parameters which can subsequently be adjusted by the user these are Number of Iterations per Tabu Search (NITS), Variations per Tabu Search (VTS) as well as Variation Ratio (VR). Figure 7 illustrates a flowchart providing an overview of the Tabu Search Iteration implementation into the BEETS algorithm.

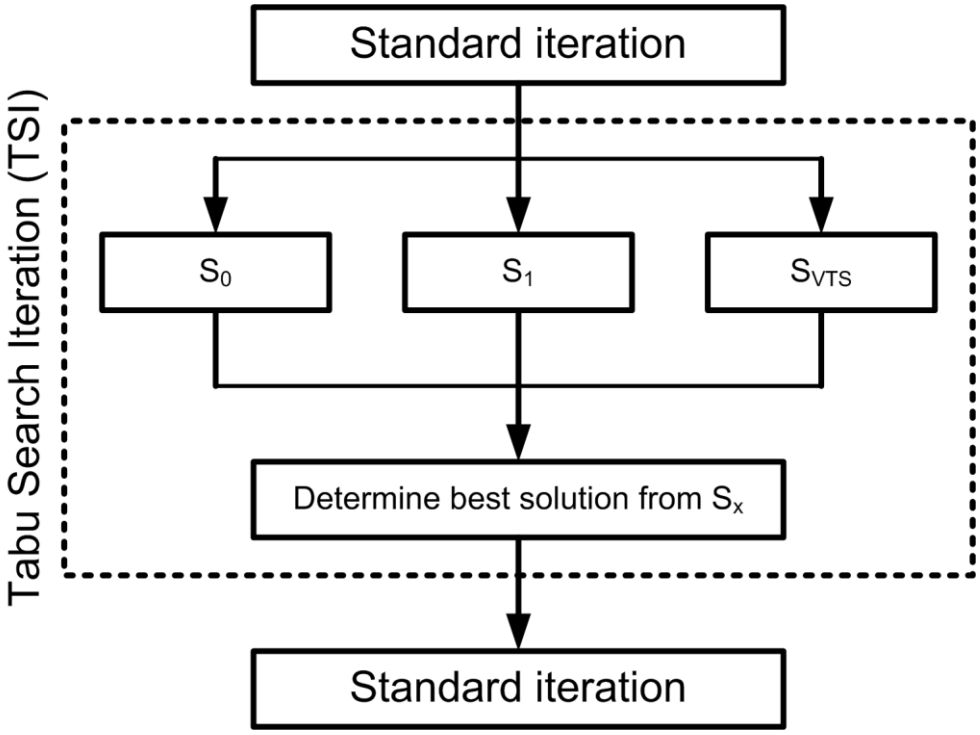


Figure 7, Tabu Search iteration implementation.

In Figure 7 a "standard iteration" is analogue to a normal optimisation iteration, i.e. all steps included in Figure 3 with the modifications discussed above. The NITS parameter decides how many "standard iterations" are to occur per Tabu Search Iteration (TSI). For example if NITS equals 2 every other optimisation iteration will be a TSI, if NITS equals 1 all iterations will be TSI. The TSI loop is in essence identical to the standard iteration up until step 5H. In step 5H for a TSI a solution S_0 is created based on ESN_i as explained above. However a fixed number of solutions equal to VTS with "random variations is also created, i.e. S_1, …, S_{VTS}. These "random" variations will simply be obtained using a random number generator to identify non void element numbers whose element status will subsequently be set to 1 subject to on their current element, i.e. S_0 status.

The Number of Elements to be Changed (NEC) will be calculated as a fraction of the current number of non-void elements (CNE) using the VR parameter as defined in equation (2.40).

$$\mathrm{NEC_j}=\mathrm{VR}\cdot\mathrm{CNE_j} \tag{2.40}$$

The Tabu lists and aspiration criteria will ensure that no previously analysed solution will be reanalysed. All variation models, i.e. S_1, …, S_{VTS} will thereafter be solved and steps 1-5

repeated. Following this the "best" solution S_{best} of the S_0, …, S_{VTS} solutions will be identified and used to create the model for the following iteration j+1. The determination of the "best" solution will follow a relatively simplistic principle; namely the ratio of Volume to Total Strain Energy (VTSE). The VTSE value for any VTS iteration will therefore be calculated as defined by equation (2.41) .

$$VTSE_k = \frac{\sum_{i=1}^{CNE} VE_i}{\sum_{i=1}^{CNE} ESE_i} \tag{2.41}$$

In (2.41) VE_i is the individual volume of the (non-void) elements and ESE_i is the total strain-energy of non-void elements (summed over Δt) and k represents the random Variation Run iteration number.

The above implementation method should help to maintain the robustness of the algorithm, because if any (or perhaps all) of the random variation solutions S_1, …, S_{VTS} are numerically unstable and error terminates these can simply be disregarded and the algorithm will continue using S_0 if no other or better solution is available.

The Tabu list implementation into the BEETS software will use the following principles. Firstly, the use of a Tabu list will be optional, and can be switched on or off using a simple Boolean operator as shall be further explained in sub-section 3.2.2. As discussed in sub-section 2.2.6, it is not generally possible to store entire solution sets in Tabu lists, mainly due to memory limitations. The Tabu lists of the BEETS software will store the element status values, as it is not feasible to store all element status values a fraction of the elements will be selected for the Tabu lists. This will utilise a Tabu List Fraction (TLF) value specified by the user. The Number of Elements Stored in the Tabu List (NESTL) can thereby be calculated as defined in equation (2.42).

$$NESTL = NE_0 \cdot TLF \tag{2.42}$$

NE_0 in (2.42) is the number of elements at iteration 0 (where all elements in the model are non-void). With the definition of $NESTL$ a list of element numbers whose statuses will be

stored in the Tabu list can now be created. This will be done by randomly selecting $NESTL$ number of elements.

In order to reduce the computational efforts associated with the Tabu list the first parameter used to detect potential "identical" iterations will be the volume of the FE model. If the volume of the current FE model (at iteration i) is identical to any of the volumes of preceding models (from iteration 0 to iteration i-1) the Tabu list will be activated to check if the element statuses of the elements are also all identical. If this is the case the software will recognise that two potential iterations are identical and prompt the user to decide whether or not the optimisation should continue or terminate.

3.1.5 Simulated Annealing Implementation

The focus is now turned to the implementation of Simulated Annealing optimisation principles into the BEETS algorithm and software. With the implementation of Tabu Search presented in the previous sub-section it was decided not to include all aspects of the SA optimisation principles; the only aspect which will be considered is the cooling function initially defined in equation (2.10). The fundamental reasoning for introducing the cooling function into BEETS is an attempt to maximise the efficiency of the algorithm, i.e. minimising the number of iterations required to obtain the optimised solution. If a relatively large Evolution Ratio, equation (2.32), is selected without the implementation of a cooling function the optimisation may rapidly converge towards a solution, but may struggle to refine the structure as a direct function of the large ER value. On the other hand if a small ER value is selected the number of iterations required to obtain the fundamental topology of the solution may be significantly increased reducing the efficiency of the software. Returning to the "original" cooling equation (2.10) it can be seen that this is a function of the "temperature" at iteration 0 (I0) and the number of iterations passed since I0. Implementing the cooling in such a way is likely to have a detrimental effect upon the "widened search space" enabled by the introduction of Tabu Search as previously presented. The reason for this is that a direct implementation of (2.10) would lead to a continuous decrease in the relative fractional volume change per iteration, significantly narrowing the search area. Instead it was therefore decided to introduce an amended Tabu Search cooling function which only becomes active towards the end of the optimisation, as the volume approaches the final target volume, i.e. V_{end}, equation (2.31). The activation of the SA cooling function will be determined by the use of expression (2.43).

$$TV_{j+1} \leq V_{end} \cdot (1 \pm COOLIT) \leq TV_{j+1} \tag{2.43}$$

Equation (2.43) simply states that if the updated Target Volume for the next iteration, TV_{j+1}, lies within a given interval of the final target volume V_{end}, determined by the Evolution Ratio (ER) the cooling function defined in will be activated.

$$ER_{cool} = ER_{cool} - ER_{cool} \cdot SF_{cool} \tag{2.44}$$

In (2.44) ER_{cool} is the updated Evolution Ratio (which may be reset if equation (2.43) becomes untrue at any given stage). At the first iteration where equation (2.43) activates equation (2.44) ER will be used to replace ER_{cool} on the right hand side of equation (2.44). SF_{cool} is a user defined scale factor dictating the rate of the cooling process. SF_{cool} must adhere to equation (2.45).

$$0.01 \leq SF_{cool} \leq 0.99 \tag{2.45}$$

Once an ER_{cool} value has been determined this will replace ER in equation (2.32) to calculate an updated TV_{j+1} value. Equations (2.43), (2.44) and (2.45) thus represent the methodology in which the cooling function principle of Simulated Annealing is introduced into the BEETS algorithm and software.

With the above changes and the “original” BESO algorithm illustrated in Figure 1, the flowchart for the overall BEETS algorithm can now be defined as illustrated in Figure 8. This flowchart is the backbone upon which contribution 6 of this PhD portfolio, the BEETS software, has been programmed. The next section will explain the overall details of the software programming and algorithm implementation process including initial verification of the algorithm and software.

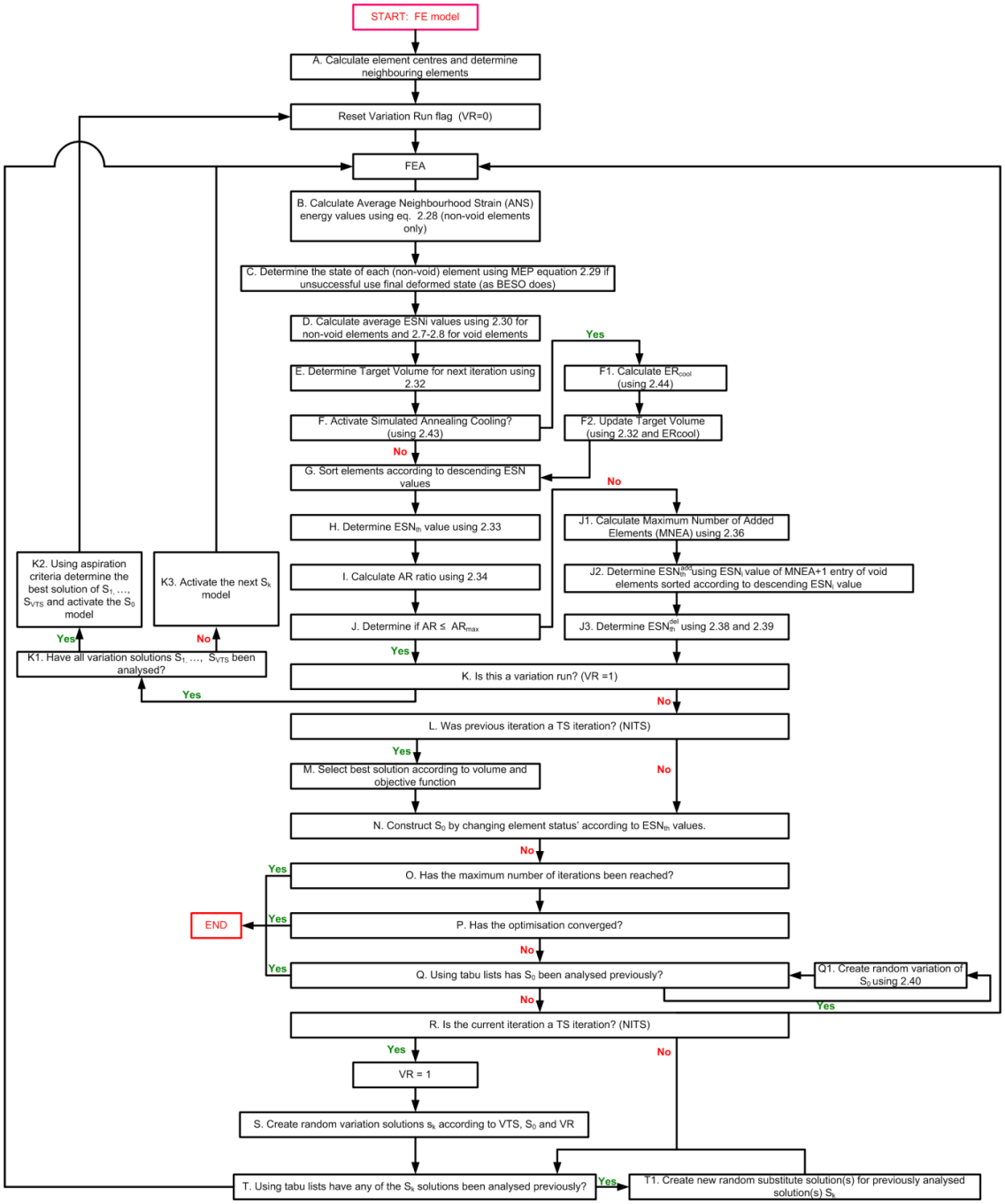


Figure 8, BEETS optimisation algorithm flowchart.

3.2 Algorithm implementation and Software Development

The purpose of this section is to explain the overall software, contribution 6 development process including the implementation of the BEETS algorithm and some of the intricate differences between the flowchart of Figure 8 and the actual software. The first task of any software development process is to select a suitable programming language / software platform.

3.2.1 Selection of programming language and software platform

Many different options are available for any new software development, the majority of which can be used in conjunction with each other. Programming languages such as Fortran, C and C++ are commonly used as base for commercial Finite Element software for example LS-DYNA and Altair HyperWorks. There are many reasons for this, one of the primary reasons is likely the facts that these programming languages:

a) Generally have extensive mathematical libraries and advanced functions.
b) Can be coupled with a wide array of other languages, such as Java, Html etc. to e.g. create advanced Graphic User Interfaces (GUIs).
c) Are very versatile and can run across multiple software platforms (e.g. Linux, Unix and Microsoft Windows) with only minor changes to the software.
d) Are public, which means that in general no royalties or software license fees are applicable to any software created using any of these languages.
e) Can be very specifically tailored to maximise CPU and memory capacities especially when handling large amounts of data.

The above facts imply that choosing a language such as C++ could indeed be advantageous for versatility, efficient hardware utilisation and High Performance Computing, and must therefore be serious contenders for a longer term strategic development of the BEETS algorithm if the potential of the algorithm can be justified. The advantages of such programming languages do however come at a cost, as they can be somewhat cumbersome to work in, and generally require specialist knowledge to fully utilise the potential in terms of e.g. efficient hardware utilisation. Although initial steps have been taken to program a dedicated FE solver including implementation of the BEETS algorithm these activities are at a relatively infantile stage and have therefore not been included in this PhD portfolio. Remembering that the purpose of the software at this stage is to investigate the feasibility and potential of the BEETS algorithm for large-deformation non-linear topology optimisation it was decided to utilise “validated”

commercial software wherever possible in an attempt to ensure that the “true” effects of the BEETS algorithm were clear. In other words it was decided to utilise commercial FE software for the FE analysis and reduce the programming to predominately focus on the actual BEETS algorithm. At this point is should be noted that this choice does not rule out the usage of the programming languages used above such as Fortran. With this change the software is however not required to actively solve any FE model, but merely send an FE model to the FE solver, extract / read the relevant data to inform the BEETS optimisation algorithm before updating the FE model and subsequently sending the updated model back to the FE solver. The core of the tasks ahead can therefore be solved using matrices and arrays, suggesting that commercial mathematics software such as MatLab or Maple may be a suitable choice.

There is however one significant issue with utilising such software; namely the fact that it is not possible for e.g. MatLab to adapt an FE file, which in general can be considered to be in text format. Hence it was deemed that a scripting language would be suitable for the task ahead. For simplicity it was chosen to use a Windows based platform and Microsoft PowerShell was ultimately chosen for the task as it enables to easily access and execute programs, conduct mathematical operations as well as reading, writing and adapting files of any given format. The BEETS software of Contribution 6 is therefore programmed in Microsoft PowerShell and requires an active LS-DYNA license for FEA. LS-DYNA was chosen for the solver as it is a commercial FE code aimed specifically at non-linear and dynamic analysis problems, is widely used within the automotive sector for e.g. crash analysis, and can be controlled via a command prompt (directly from within PowerShell). LS-DYNA is set up as an explicit FE solver, hence all subsequent analysis / optimisation will utilise explicit FEA unless otherwise stated.

3.2.2 Overview of BEETS PowerShell Software

As previously stated, the BEETS software of contribution 6 is based upon the overall flowchart in Figure 8. The software has been compiled using a modular approach, meaning that a series of scripts, subroutines and functions are called from the main script. There are two main advantages to this approach:

a) Obtaining an overview of the intricate details of the software is eased as individual steps (Figure 8) is simplified, this may also reduce the complexity of debugging.
b) The software becomes modular, which means that replacing, upgrading, adding or even removing aspects / parts of the software is easily achievable. This is an important attribute for future research.

The BEETS optimisation software consist of a total of 12 PowerShell files (*.ps1). The file names, a narrative description, a link to the flowchart (Figure 8) steps covered as well as the equations used within each *.ps1 file is listed in Table 8.

Table 8, overview of PowerShell files

File name	Description	Steps covered (Figure 8)	Equations used
MAIN.ps1	This is the main file which prompts the user for inputs; it contains the optimisation iteration loops and calls all functions (sub-routines) from separate *.ps1 files.	Start, end, K-T	
Element_Sensitivity_Numbers.ps1	This file calculates the Element Sensitivity Numbers (ESN)	D	(2.7) (2.8) (2.30)
Update_FE_file.ps1	This file updates the LS-DYNA keyword file for subsequent analysis of the updated design		
Determine_ESNth.ps1	This file calculates the target volume for the next iteration, determines threshold sensitivity values, including checking the admissions ratio and controlling the cooling function.	F-J	(2.32) (2.33) (2.34) (2.35) (2.36) (2.37) (2.38) (2.39) (2.43) (2.44) (2.45)

File name	Description	Steps covered (Figure 8)	Equations used
MEP.ps1	This file contains the Maximum Entropy Principle methodology to determine the likely state of each individual element	C	(2.22) (2.24) (2.29)
Element_Strain_Values.ps1	This file assembles the raw element strain data extracted by Read_elout_strain.ps1. The file also calculates the ANS_i values as well as calling the MEP.ps1 file	B	(2.28)
Read_elout_strain.ps1	This file extracts the raw element strain values from the elout file created from the LS-DYNA solver.		
Determine_neighbouring_elements.ps1	This file determines the neighbouring elements of any given element as a function of the centre coordinates of the elements, and a user defined distance.	A, B	
Read_FE_data.ps1	This file reads the vital FE data from the *.key / *.k LS-DYNA keyword file, such as number of elements, nodal coordinates etc..	A	

File name	Description	Steps covered (Figure 8)	Equations used
Remove_lines.ps1	This file is a “service file”; it contains a simple script to remove blank lines from the *.key / *.k files to ensure no errors occur when data is read using other *.ps1 files		
Read_file.ps1	This file is a “service file”; it contains a simple script which opens a file browser enabling the user to identify the input *.key / *.k file		

To run the software it is essential that PowerShell is installed on the PC, and that a working LS-DYNA installation is present. The software is initiated by right clicking the “MAIN.ps1” file and select “Run with PowerShell”. The software will thereafter prompt the user to select the *.key / *.k input file. There are a few requirements for the LS-DYNA input file:

a) The optimisation will only be conducted on *SHELL elements.
b) The shell elements must be defined in the global XY-plane only.
c) The file must contain a *SET_SHELL set containing all *SHELL elements within the model.
d) The *DATABASE_ASCII card must have the ELOUT option enabled.

The main reason for point D above is to ensure that the strain values are available for the BEETS optimisation algorithm. Throughout this PhD portfolio, including the literature review of this contribution it has persistently been suggested that the individual element strain energy be the measurable quantity for the optimisation algorithm. Extracting the element strain energy from an LS-DYNA FEA is however not a straightforward task. The only direct method of extracting the element strain energy in LS-DYNA is to model every single element as an individual part leading to a very extensive “organisational” task significantly (and unnecessarily) reducing the versatility of the software. It is possible to extract the element

strain values by post-processing via LS-PrePost, a dedicated LS-DYNA pre- and post-processor. The output format of this is however not a “conventional” text format making it impossible to read the data into PowerShell for the purpose of optimisation. Hence it was decided to change the input data into the BEETS software from strain energy to actual strain values, as these are readily available. This could potentially have an effect upon the accuracy of the results, however the strain should serve as a valid means for relative comparison between individual elements as intended. Due to the fact that large deformation, and thereby large strains are expected it was decided to use the total (sum) of strain values ε_{tot} to represent the individual element as defined in (2.46).

$$\varepsilon_{tot} = \varepsilon_{11} + \varepsilon_{22} + \varepsilon_{33} + \varepsilon_{12} + \varepsilon_{13} + \varepsilon_{23} \tag{2.46}$$

As implied above, changing the measurable quantity from strain energy to strain values may have an effect upon the accuracy of the algorithm, and warrants further in-depth analysis. For the remainder of this document, including the software of contribution 6 the total strain value will however be used as defined by equation (2.46).
The BEETS software will prompt the user for the inputs listed in Table 9.

.Table 9, BEETS user defined inputs.

Input prompt	**PowerShell variable name**	**Notes**
Please enter the maximum number of iterations using an integer	Max_iter	Max_iter defines the maximum number of optimisation iteration loops.
Please enter the frequency of Tabu Search Iterations using an integer	TSI	TSI defines how often a TS iteration, Figure 7, is completed. For example a value of 2 will dictate that iteration 2, 4, 6 etc. are Tabu iterations.
Please enter the number of Variation Runs per TS Iteration using an integer	VTS	VTS defines how many variation runs (Figure 7) are completed per Tabu search iteration.

Input prompt	PowerShell variable name	Notes
Please enter the filtering radius	RADIUS	RADIUS defines the maximum distance between two element centres to be used with the Von Neumann neighbourhood calculations, Figure 5.
Please enter the Tabu Search Fraction per Variation Run	TSF	TSF defines how many elements randomly are set to void for any given variation run. For example if a model contains 100 elements and TSF is set to 0.1 10 elements will be randomly removed. It should be noted that this does not take varying element sizes (volumes), boundary conditions, strain levels etc. into account.
Please enter an iteration number from which the Tabu Search Iterations should start	TABU_offset	TABU_offset provides an opportunity to offset the start of the TS iterations, this may for example be used to stabilise a given optimisation run by avoiding random removal of elements at the initial iterations.
Use Tabu list (Y/N)	TABU_LIST	If Yes is selected a Tabu list will be created to avoid repetition of running previous configurations / models.
Please enter the fraction of elements to be included in the Tabu list	TLF	This number dictates how many element statuses will be monitored per iteration. For example if a given model (at iteration 0) contains 100 elements and TLF is set to 0.1 the status of 10 elements (selected at random) will be stored for each optimisation iteration.

Input prompt	PowerShell variable name	Notes
Please enter the Evolution Ratio	ER	ER defines the fractional change of volume between two consecutive iterations as defined in equation (2.32).
Please enter the maximum Admissions Ratio as a fraction	ARMAX	ARMAX defines the maximum fraction of elements which can be added to a model for any given iteration. For example if a model contains 1000 (non-void) elements in a given iteration and ARMAX is set to 0.03 a maximum of 30 elements can be changed from void to-non-void for the next optimisation iteration. It is advised that ARMAX is not set less than 0.03 to encourage the growth of additional elements where required. (Huang and Xie, 2008).
Please enter the Target Volume Fraction	TVF	TVF defines the final target volume based upon the model volume at iteration 0. Using Read_FE_data.ps1, Table 8, the model volume is calculated by summation of individual shell volumes (areas). This volume is then multiplied by TVF to obtain the target volume.
Please enter the optimisation convergence tolerance	TERMTOL	TERMTOL determines if the optimisation terminates before MAX_iter is reached, provided that the average change over the previous 3 iterations is less than or equal to TERMTOL.
Use Cooling Function (Y/N)	COOLUSE	COOLUSE is a Boolean operator which dictates whether or not the Cooling Function is included in the optimisation,

Input prompt	PowerShell variable name	Notes
Please enter the Cooling interval	COOLIT	COOLIT is used to determine the interval in which the cooling function i.e. equation (2.44) becomes active. For example if COOLIT = 0.1 and TVF equals 0.5 the cooling function will be active if the target volume of any given iteration is between 0.4 and 0.6.
Please enter the cooling scale factor (between 0.01 and 0.99)	SFCOOL	SFCOOL is only activated if COOLUSE is set to 1 (on); SFCOOL dictates the rate of cooling using equation (2.44).
Use Entropy (Y/N)	MEPUSE	MEPUSE is a Boolean operator, which dictates whether or not the Maximum Entropy Principle is employed. If not the optimisation software will follow the BESO algorithm, potentially including the Tabu search iterations.

During the optimisation process the software creates a new folder per optimisation iteration within the folder which contains the base *.key / *.k file, Figure 9. The *key / *.k file is copied into the individual folders and altered according to the optimisation criteria.

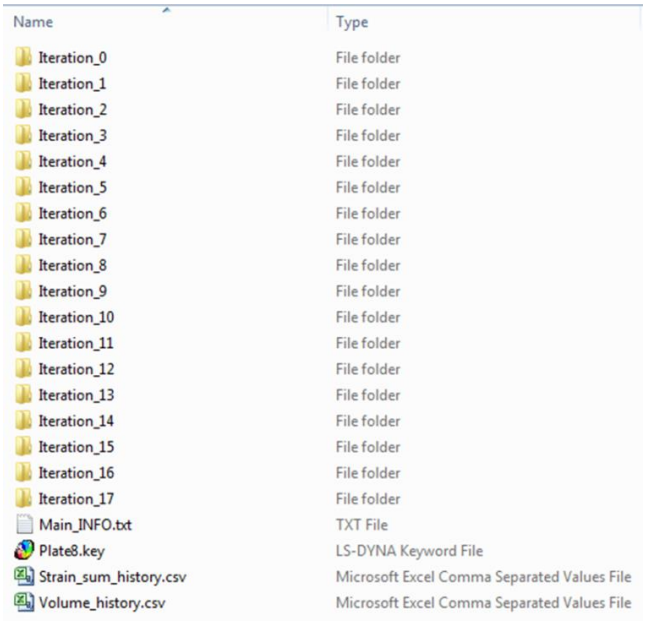


Figure 9, main folder contents.

The FEA is subsequently solved from the individual iteration folders, meaning that output files are contained within these folders, Figure 10 A). For Tabu Search iterations Variation Runs are created within subfolders inside the iteration folder, Figure 10 B).

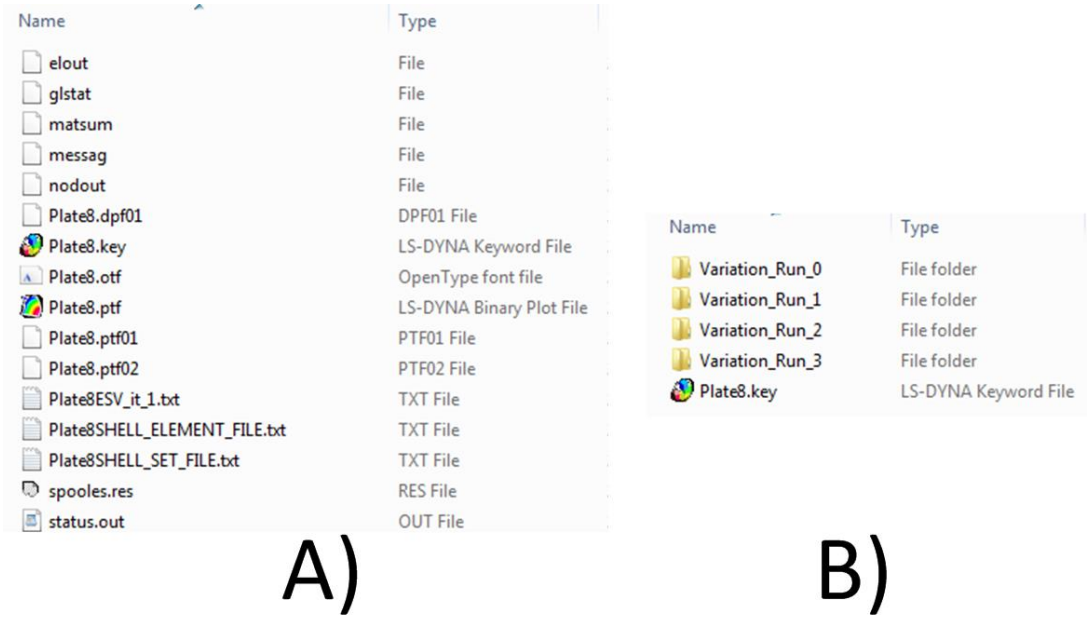
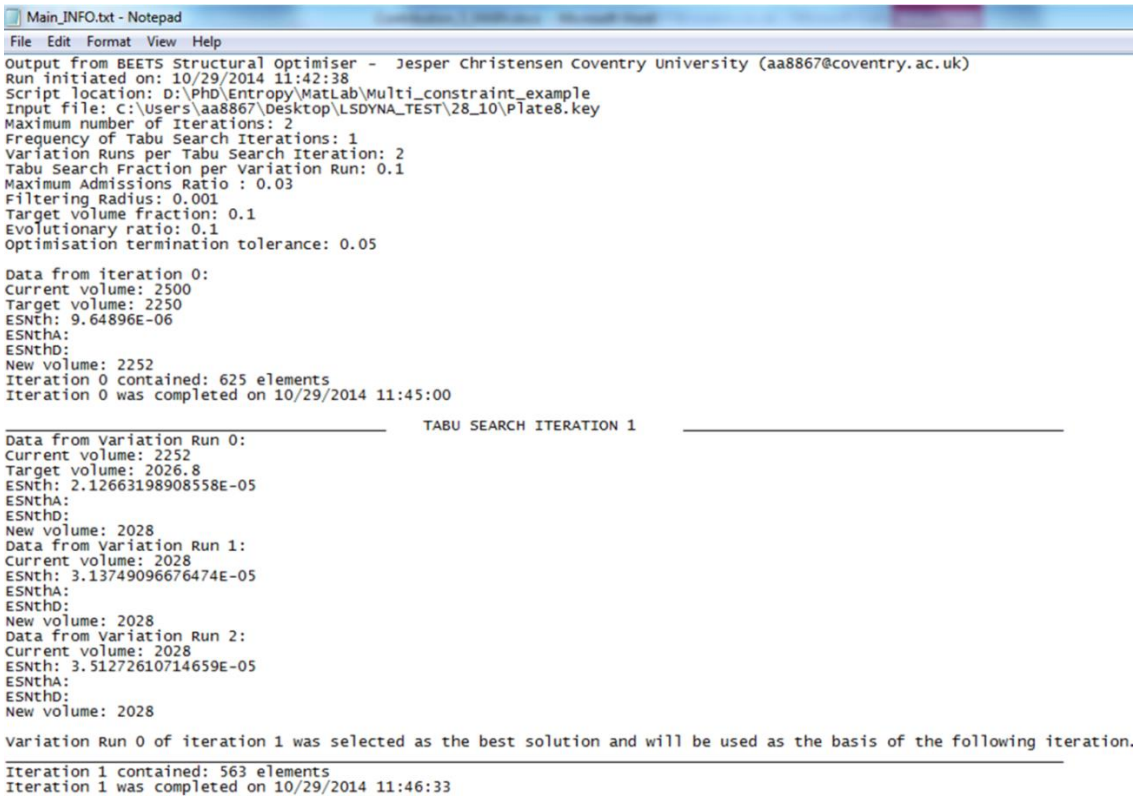


Figure 10, example folder contents A) standard iteration, B) Tabu Search iteration.

In addition to the actual updated *.key / *.k files, the software also outputs general data about the overall optimisation procedure, the input values and specific data about the individual iterations. This data is stored in the Main_INFO.txt file placed in the main folder, i.e. the folder

which contains the original *.key / *.k file selected by the user. An example of the output is illustrated in Figure 11.



Main_INFO.txt - Notepad

File Edit Format View Help

```
Output from BEETS Structural Optimiser -  Jesper Christensen Coventry University (aa8867@coventry.ac.uk)
Run initiated on: 10/29/2014 11:42:38
Script location: D:\PhD\Entropy\MatLab\Multi_constraint_example
Input file: C:\Users\aa8867\Desktop\LSDYNA_TEST\28_10\Plate8.key
Maximum number of Iterations: 2
Frequency of Tabu Search Iterations: 1
Variation Runs per Tabu Search Iteration: 2
Tabu Search Fraction per Variation Run: 0.1
Maximum Admissions Ratio : 0.03
Filtering Radius: 0.001
Target volume fraction: 0.1
Evolutionary ratio: 0.1
Optimisation termination tolerance: 0.05

Data from iteration 0:
Current volume: 2500
Target volume: 2250
ESNth: 9.64896E-06
ESNthA:
ESNthD:
New volume: 2252
Iteration 0 contained: 625 elements
Iteration 0 was completed on 10/29/2014 11:45:00

_____ TABU SEARCH ITERATION 1 _____
Data from Variation Run 0:
Current volume: 2252
Target volume: 2026.8
ESNth: 2.12663198908558E-05
ESNthA:
ESNthD:
New volume: 2028
Data from Variation Run 1:
Current volume: 2028
ESNth: 3.13749096676474E-05
ESNthA:
ESNthD:
New volume: 2028
Data from Variation Run 2:
Current volume: 2028
ESNth: 3.51272610714659E-05
ESNthA:
ESNthD:
New volume: 2028

Variation Run 0 of iteration 1 was selected as the best solution and will be used as the basis of the following iteration.
_____
Iteration 1 contained: 563 elements
Iteration 1 was completed on 10/29/2014 11:46:33
```

Figure 11, example of Main_INFO.txt file content.

In addition to the general output data from the MAIN_INFO.txt file the total volume and total strain values (per iteration) are exported to two separate *.csv files which can subsequently be imported to e.g. Excel.

With the above, the main features of the BEETS software of Contribution 6 have now been presented, including the algorithm development steps, overall flowchart and significant variable names. The following chapter will examine the feasibility of the software for non-linear structures exposed to large deformations.

4. Case Studies

This chapter will contain a series of case studies using the BEETS software and will aim to compare BEETS to other previously presented algorithms and software. The first case study will be used to verify the fundamental algorithm embedded within the BEETS software.

4.1 Linear topology optimisation case studies

The purpose of this sub-section is to verify that the base algorithm of BEETS which in essence is a BESO algorithm follows the principles of BESO and obtains results similar to that of BESO optimisation. For the first example a simple bracket structure will be analysed. The general FE model is illustrated in Figure 12.

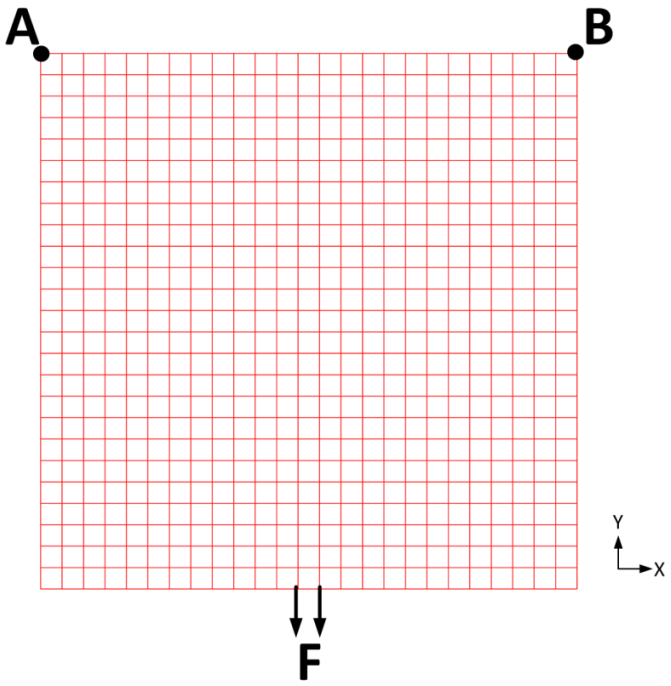


Figure 12, general FE model for case study 1 - bracket.

The model illustrated in Figure 12 will be fully constrained at points A and B and exposed to two single point loads in the negative Y-direction as indicated by F. The model in Figure 12 contains 625 elements and utilises the MAT_001:Elastic LS-DYNA keyword 971 material model. This is an isotropic linear elastic material model. The specific material parameters used are listed in Table 10.

Table 10, material parameters for case study 1 - bracket.

Parameter	Value	Unit
Young's Modulus	210,000	MPa
Poisson's Ratio	0.3	

The loading F will be applied in a quasi-static manner, however the FE analysis will utilise explicit computation, hence a time-history definition of the load is required. This has been defined as illustrated in Figure 13.It should be noted that the termination time for the analysis will be 1.1s, and that the load value is merely an indication of no load (0) and full load (1). The actual load magnitude will be defined by a scale factor, which initially will be set to 200N.

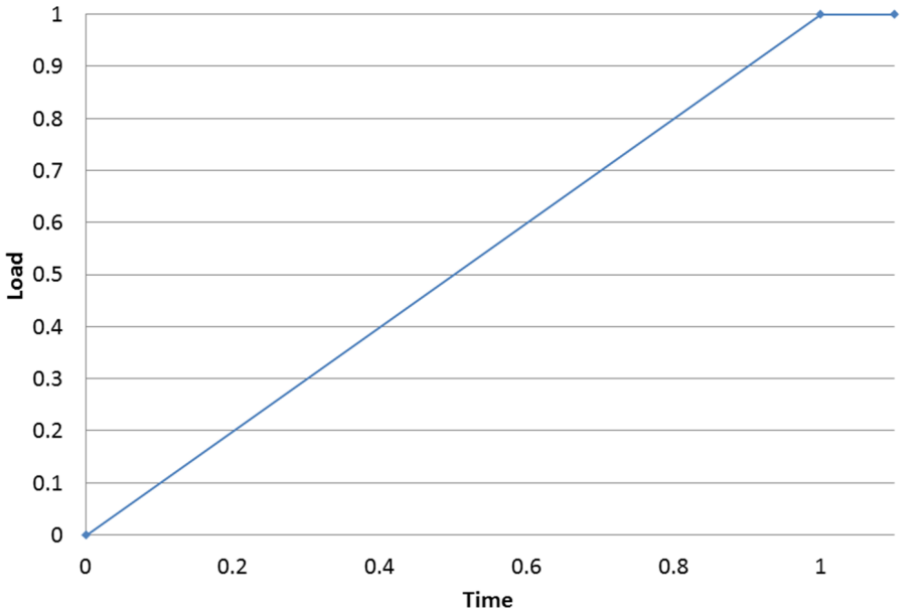


Figure 13, load vs time.

4.1.1 BEETS Case Study 1

Using the above described model as an input file to the BEETS optimisation software the model was run using the optimisation parameters defined in Table 11.

.

Table 11, BEETS input parameters for case study 1.

PowerShell variable	Value
Max_iter	25
TSI	26 (no Tabu Search included)
VTS	N/A
TSF	N/A
TABU_offset	N/A
TABU_LIST	N/A
TLF	N/A
ARMAX	0.03
RADIUS	0.0 (no neighbourhood influence)
TVF	0.1
ER	0.1
TERMTOL	0.05
MEPUSE	No
COOLUSE	No
COOLIT	N/A
SFCOOL	N/A

With the above parameters set the BEETS software uses the “standard” BESO algorithm as previously presented. The optimisation completed after iteration 23 as the average change in volume over the final 3 iteration was 2.99%. Figure 14 illustrates the total strain as a function of the iteration number.

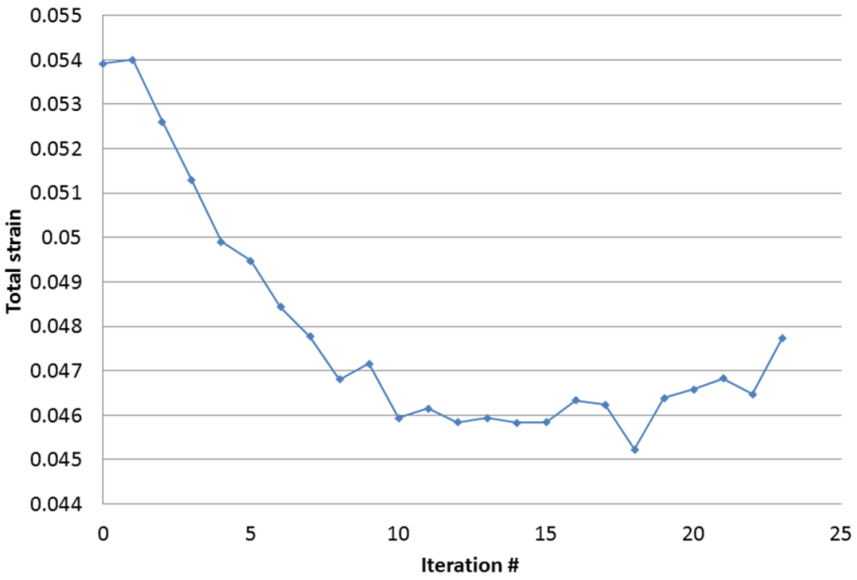


Figure 14, sum of strain per iteration case study 1.

As illustrated by Figure 14 the total strain has decreased significantly between iteration 0 and iteration 23. This is as expected as elements which did not contribute significantly to counteracting the applied load has been removed from the structure, i.e. optimised. The lowest strain value is however not found in iteration 23, but in iteration 18. At this stage it is convenient to review the volume as a function of iteration number, this graph can be found in Figure 15.

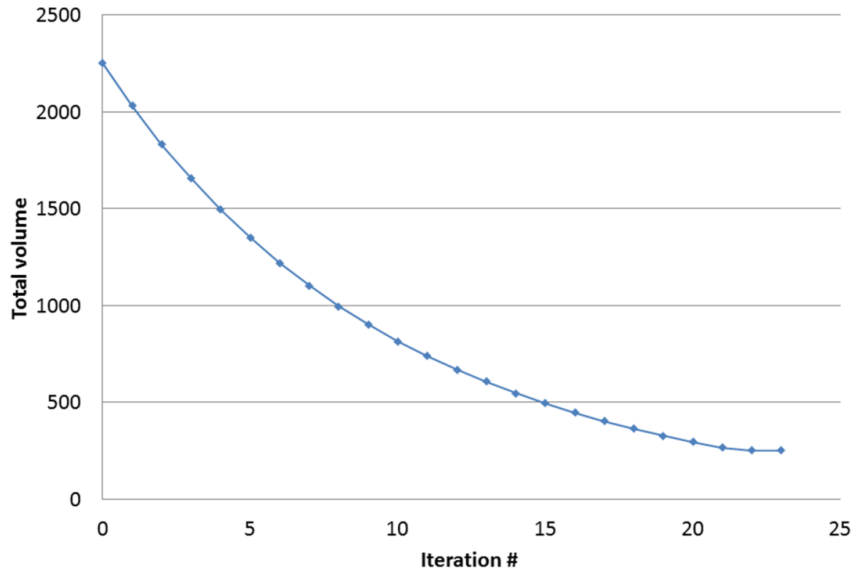


Figure 15, total volume per iteration case study 1.

Figure 15 clearly indicates a steady convergence tendency with a continuous reduction of volume, i.e. elements; this is substantiated when reviewing the MAIN_INFO.txt file which confirms that no elements were added at any point during the optimisation. The tendency in Figure 14 is therefore not repeated in Figure 15. The tendency of Figure 14 could perhaps be partially attributed to numerical error, i.e. rounding off errors. It is however more likely that the tendency observed in Figure 14 can be explained by reviewing the strain plots of selected iterations. Figure 16 illustrates the Von Mises strain values (which are representative of the total strain values in this case) for iteration 0, at the termination time, i.e. t =1.1. Figure 16 clearly shows that there are a significant number of elements with very low strain values, indicating that these can be removed during the subsequent optimisation process.

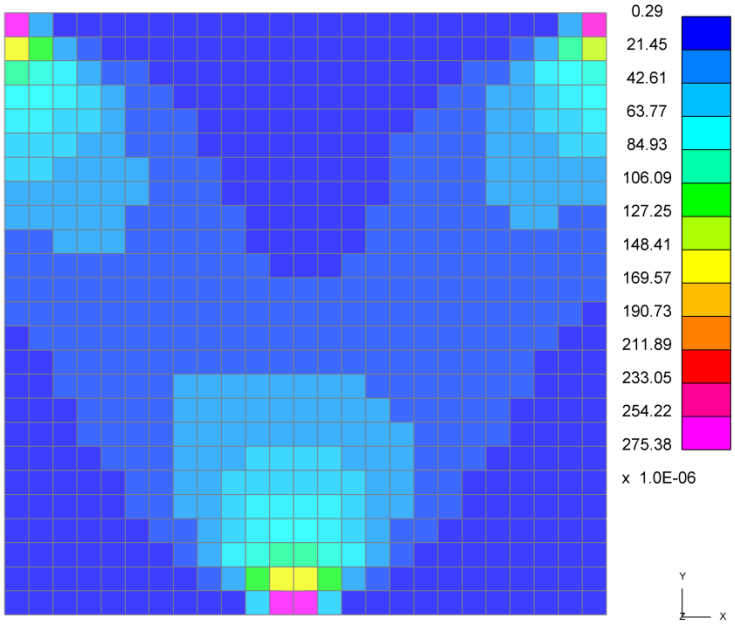


Figure 16, total strain (Von Mises strain) case study 1, iteration 0, t = 1.1.

Figure 17 is the “corresponding” stage of iteration 18 to that of iteration 0, Figure 16. As illustrated in Figure 17 a significant number of elements have been removed from the element; it can also be seen that the maximum strain level has decreased from iteration 0 to iteration 18, both of which would contribute to the drop seen in Figure 14.

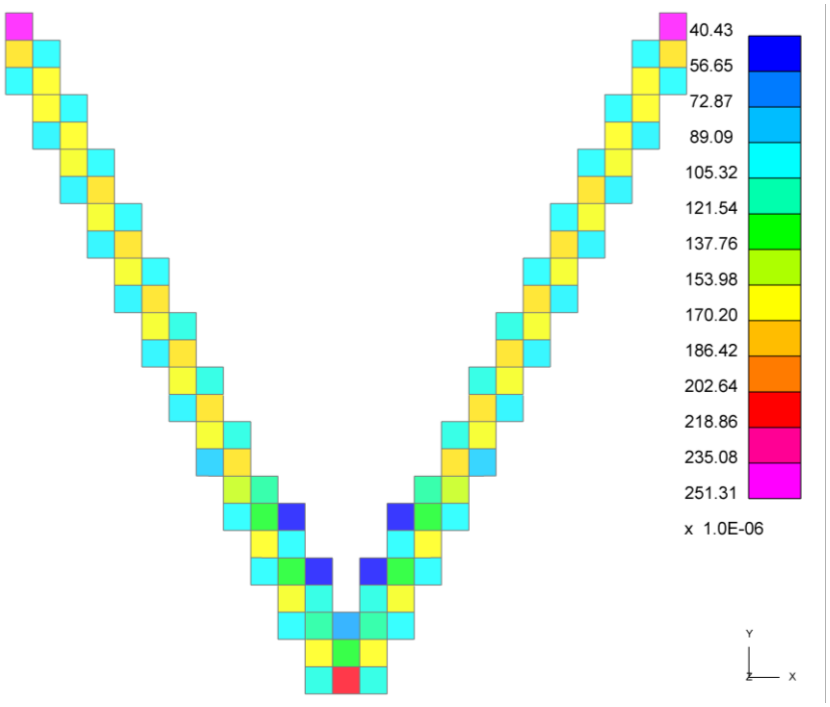


Figure 17, total strain (Von Mises strain) case study 1, iteration 18, t = 1.1.

Figure 18 is the corresponding stage of iteration 23 to those of iteration 0 (Figure 16) and 18 (Figure 17). Comparing the 3 figures the reason for the trend seen in Figure 14 can be identified. Although the number of elements have decreased between iterations 15 and 23 it is clear that the maximum strain value has increased; Figure 18 also indicates that the average strain value per element has increased as can be expected.

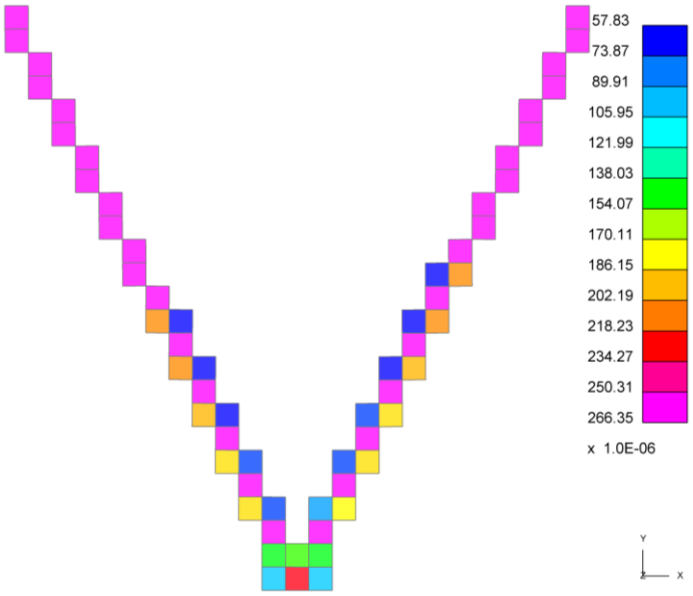


Figure 18, total strain (Von Mises strain) case study 1, iteration 23, t = 1.1.

Despite the above discussions the optimised structure illustrated by Figure 18 is indeed an optimised structure given the input model and associated data. It should be noted that the BEETS algorithm contains no symmetry constraints; the above bracket is symmetric purely because the model including Boundary Conditions are symmetric. The stress, strain and displacement values confirm that the model has remained in the linear elastic region of the material curve. Figure 15 also confirms that the TVF value of 0.1 (or 10%) has been obtained.

4.1.2 BESO Case Study 1

A free (2D) downloadable version of the BESO algorithm is available from (RMIT, 2014) the authors of which are Huang and Xie, this software is therefore a reasonable representation of the BESO algorithm capability and functionality, and can serve as a good means of comparison to the fundamental BESO algorithm of BEETS. Using the BESO software it is possible to repeat the bracket optimisation study presented above, using identical parameters and general model setup. The only aspect it is not possible to repeat is the filtering radius as the BESO software does not allow for a filtering radius less than the element size. The results obtained from the BESO software can be seen in Figure 19.

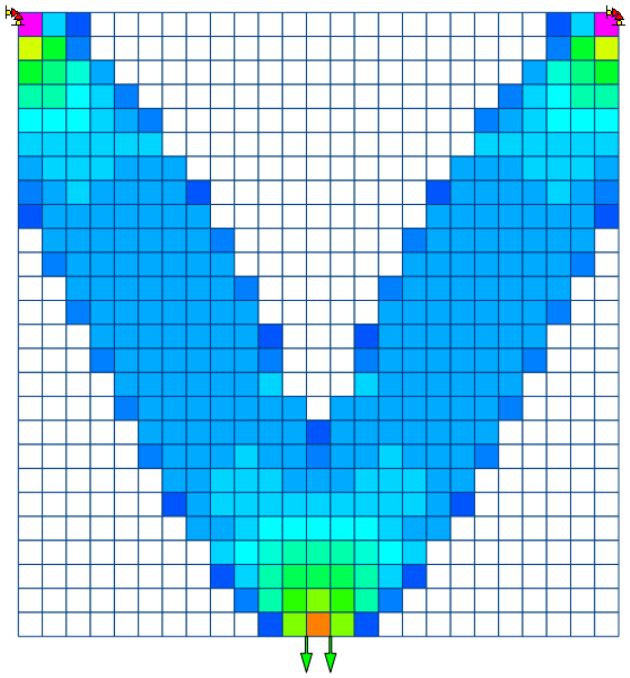


Figure 19, optimised bracket from BESO.

Figure 19 reveals that the general shape of the bracket is identical between the BEETS and BESO algorithms, the major difference is most likely caused by the filtering radius as explained above. Reviewing the output data from the BESO software, including Figure 19, it is clear that the algorithm has not met its target volume fraction objective of 10%. Nevertheless there is seemingly a good correlation between BEETS and BESO in terms of the proposed bracket topology for this simple example.

4.1.3 HyperWorks Case Study 1

In this sub-section the bracket of case study 1 will be optimised using HyperWorks / Optistruct. Remembering that Optistruct is a linear static solver using VDM and SIMP the model has been set up using static loading only. All other attributes of the model are identical to those of the LS-DYNA model previously presented. The objective of the optimisation was set to minimise volume with no constraints applied in accordance with the BEETS and BESO models above. The outcome from Optistruct is illustrated in Figure 20.

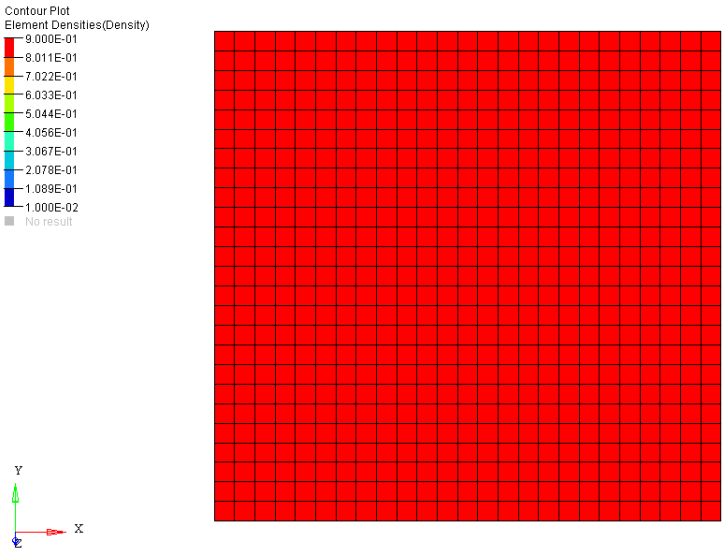


Figure 20, optimised bracket from Optistruct using unconstrained optimisation.

As anticipated the outcome of Optistruct is not a feasible engineering design; the reason for this infeasible result is as previously discussed the fact that Optistruct in general will not "function correctly" using unconstrained optimisation as is the case with BEETS and BESO as presented above. The Optistruct model was therefore repeated this time including a maximum compliance constraint (of 10 times the compliance of the initial model) imposed. The outcome of this optimisation is illustrated in Figure 21.

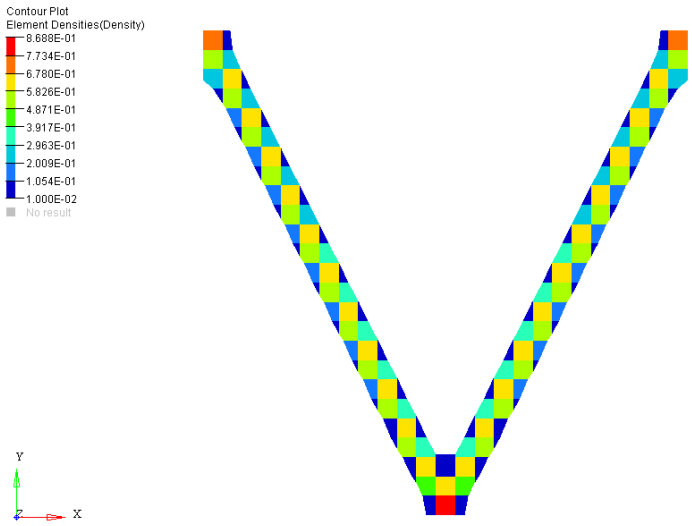


Figure 21, Optimised bracket from Optistruct using compliance constrained optimisation.

As illustrated by Figure 21 it can be seen that the resulting bracket is indeed identical to the BEETS result and very similar to that of BESO.

With the above in mind it is clear that the bracket design problem above is a fairly simplistic optimisation problem; a more advanced (linear static) case study will now be considered.

4.1.4 BEETS Case Study 2

For case study 2 a Warren bridge structure will be analysed, as illustrated in Figure 22.

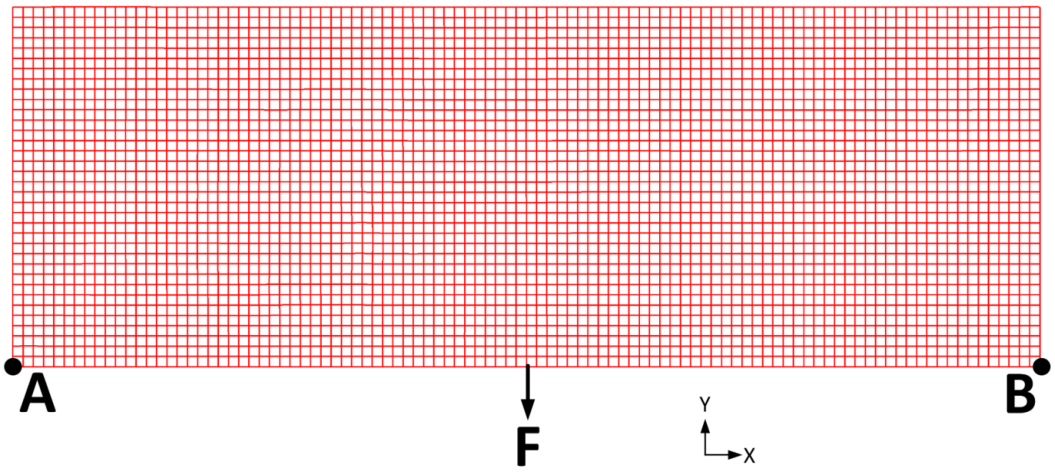


Figure 22, general FE model for case study 2 - Warren bridge.

The Warren Bridge is fully constrained at point A and simply supported, i.e. constrained in the Y-direction only at point B Figure 22. The model contains 3500 shell elements of unit thickness

and is subjected to a single point load indicated by F in Figure 22. The model utilises the material parameters listed in Table 10 as well as the general loading curve defined in Figure 13; the total load magnitude applied was 100N in the negative Y-direction, Figure 22.

Table 12, BEETS input parameters for case study 2 – Warren bridge.

PowerShell variable	Value
Max_iter	75
TSI	76 (no Tabu Search included)
VTS	N/A
TSF	N/A
TABU_offset	N/A
TABU_LIST	N/A
TLF	N/A
ARMAX	0.03
RADIUS	0.0 (no neighbourhood influence)
TVF	0.6
ER	0.04
TERMTOL	0.01
MEPUSE	No
COOLUSE	No
COOLIT	N/A
SFCOOL	N/A

With the above parameters set the BEETS software uses the “standard” BESO algorithm as was the case for case study 1. The optimisation of case study 2 completed after iteration 75 i.e. when Max_iter, Table 12 was reached. Figure 23 illustrates the total strain as a function of the iteration number for case study 2.

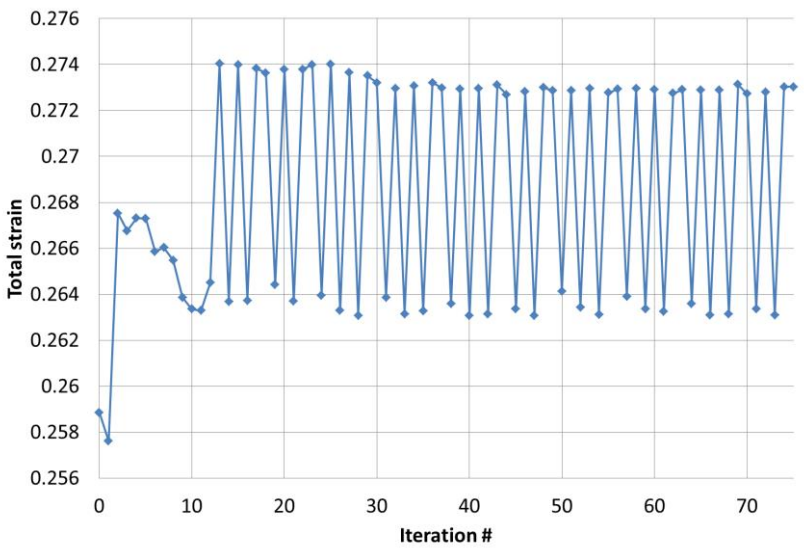


Figure 23, sum of strain per iteration case study 2.

Figure 23 indicates that a tendency to oscillate exists; as the total strain value after iteration 12 varies between the two "extremes" of approximately 0.274 and 0.263 for all successive iterations. This tendency is initially confirmed by viewing the total volume per iteration from the case study, Figure 24.

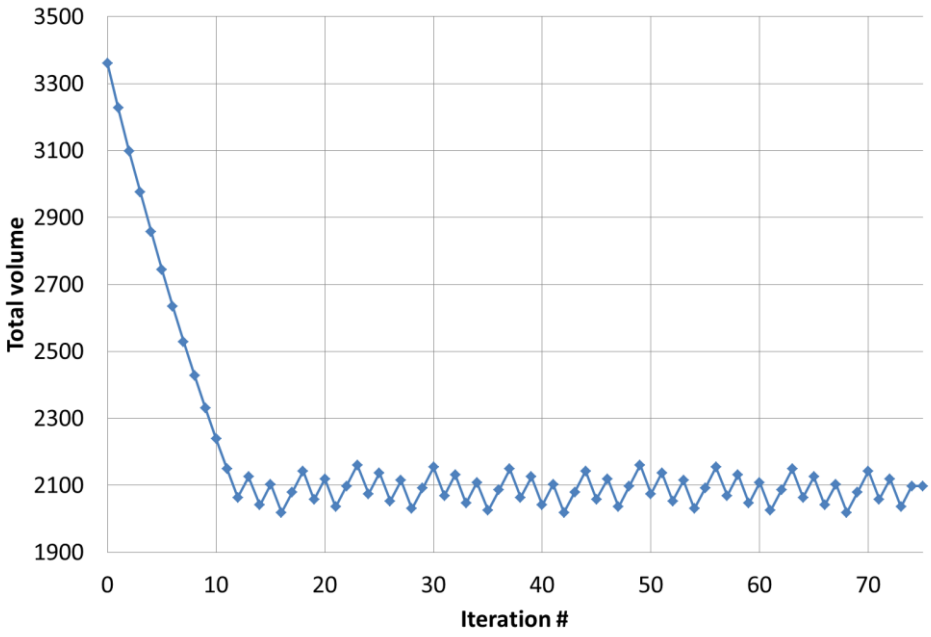


Figure 24, sum of volume per iteration case study 2.

Reviewing the MAIN_INFO.txt file from case study it was confirmed that the target volume was indeed 2100; explaining the oscillation tendency. This thereby indicates that the optimisation could have been terminated after approximately 12 iterations; however this was not the case as TERMTOL was set too low and thus not activated. Figure 25 illustrates the optimised structure after iteration 12 and the "final" structure after iteration 75.

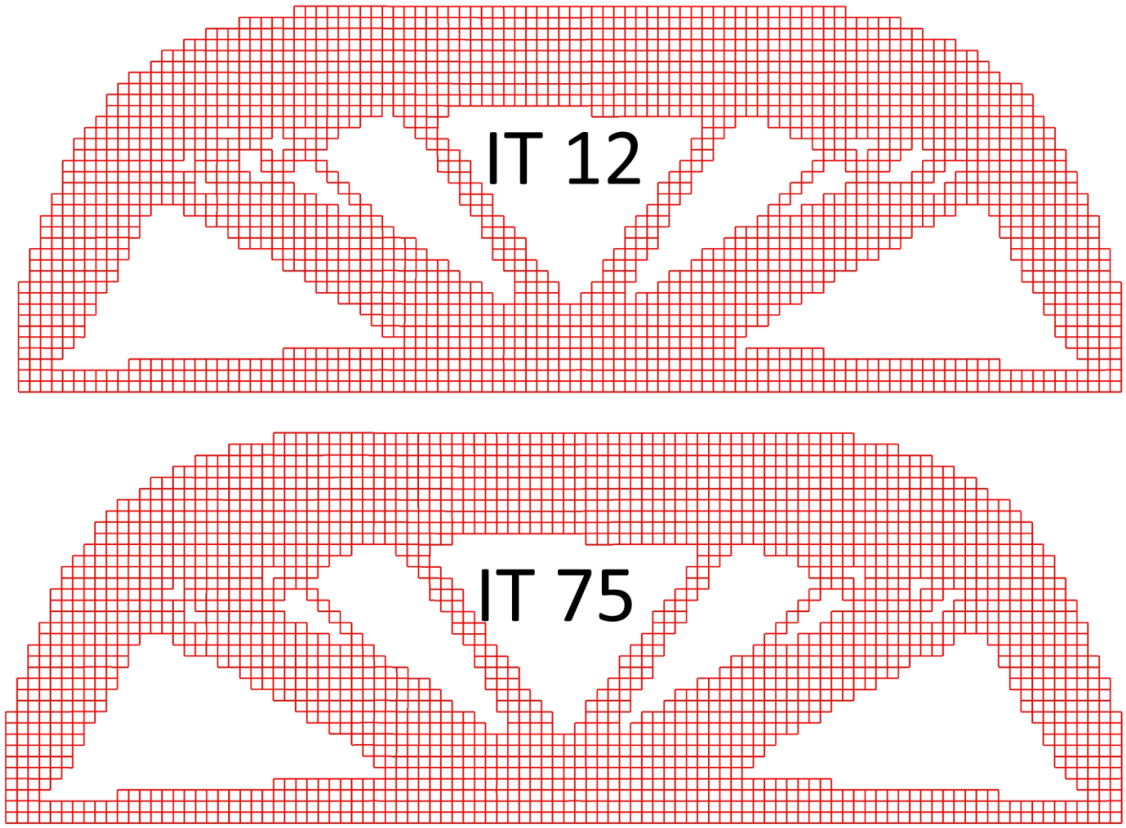


Figure 25, case study 2 iteration 12 and iteration 75.

Figure 25 clearly illustrate that the difference between the two iterations are minimal. Due to the BEETS settings, Table 12, the algorithm used is in essence the BESO algorithm as previously discussed. Figure 23, Figure 24 and Figure 25 are thus good indicators of the previously discussed "oscillation tendency. One of the aims of the BEETS algorithm is, as previously defined, to reduce this tendency by using a cooling function and Tabu lists; however neither were activated in this particular case. Despite the oscillation tendency the results of the analysis seem logic given the boundary conditions, material parameters and general optimisation settings applied.

4.1.5 BESO case study 2

Using the BESO software (RMIT, 2014) the Warren bridge model was recreated and optimised using the relevant parameters from Table 12. Figure 26 illustrates the outcome from the BESO software.

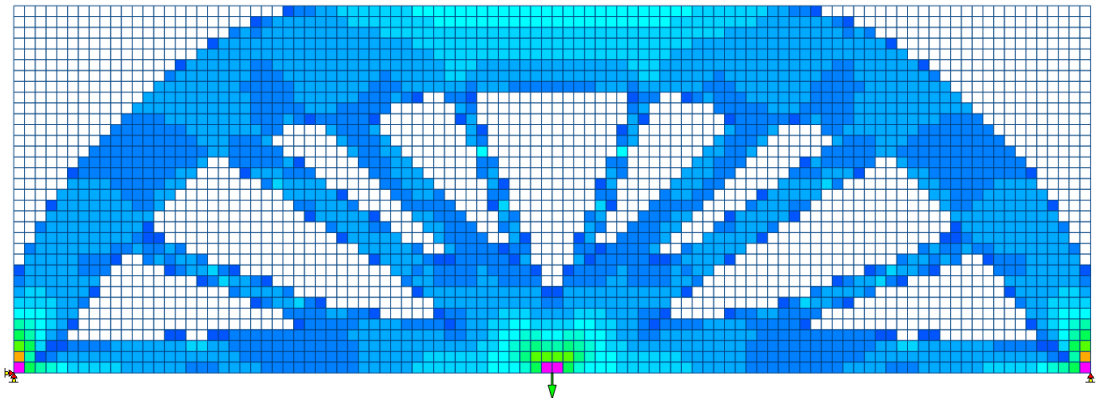


Figure 26, optimised Warren bridge from BESO.

Comparing Figure 26 to Figure 25 some similarities do appear; there is however also some subtle differences; particularly around the “internal holes” in the structure. In an attempt to get a better overview of these differences iteration 75 from Figure 25 has been overlaid onto Figure 26; Figure 27.

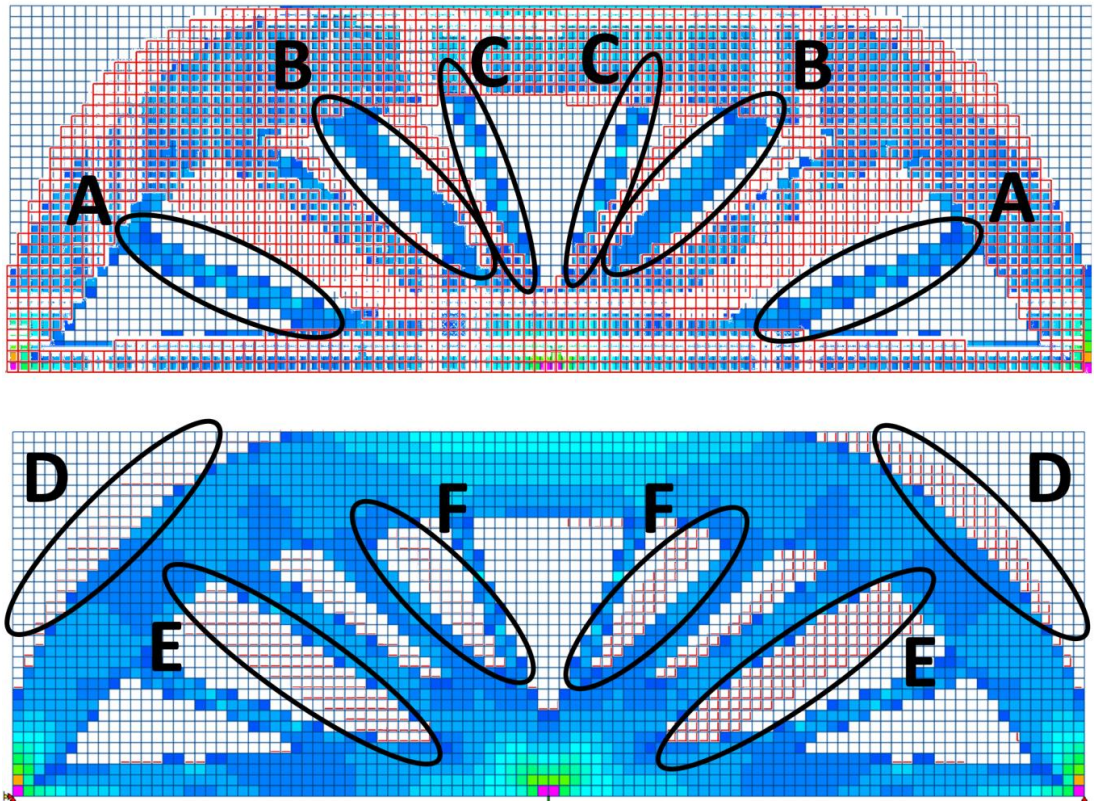


Figure 27, overlay of BEETS and BESO results.

The top image in Figure 27 illustrates Figure 25 on top of Figure 26, whereas the bottom image in Figure 27 represents Figure 26 on top of Figure 25. Areas A, B and C in Figure 27 thus represent areas where the BESO software has retained elements while the BEETS software has not. Analogue areas D, E and F in Figure 27 indicate areas where the BEETS software has retained elements whereas the BESO software has not. The results thus indicate that the BEETS software has grown “wider” than the BESO software, areas D Figure 27. Generally it can also be noted that the “supports” of the BEETS results i.e. E and F, Figure 27 , lie in between those of the BESO results. In other words Figure 27 reveals that E lies in between A and B in addition F lies in between B and C. It is not known exactly what has caused these differences; it may be due to the neighbourhood influence, which as previously stated cannot be turned off in the BESO software. Other factors could relate to slight differences in the algorithm, the exact algorithm within the BESO software is not known, in addition the BESO software does utilise linear static FEA, hence there may be a fundamental difference in the results guiding the optimisation.

4.1.6 HyperWorks case study 2

The Warren bridge model was also solved using HyperWorks / Optistruct . The model utilised the same optimisation settings as used in case study 1, including a compliance constraint. The resulting topology from this optimisation is illustrated in Figure 28.

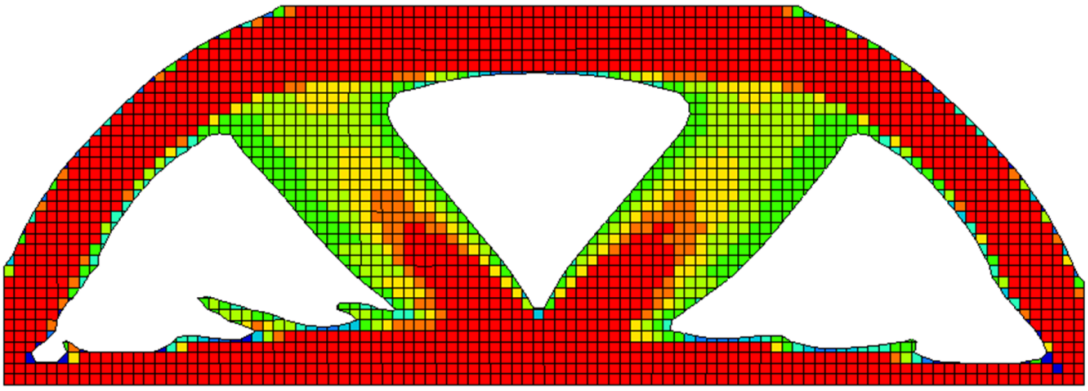


Figure 28, optimised Warren bridge from Optistruct using compliance constrained optimisation

The “external profile” of the optimised structure in Figure 28 is similar to those obtained using the BEETS and BESO software; the “internal supports” are however significantly different. The Optistruct results utilise two main supports, whereas the BEETS results, Figure 25, indicate 4 supports and finally the BESO software, Figure 26, indicate 8 supports. The volumes of the 3 results are almost identical: 2100 for BEETS, 2096 for BESO and 2070 for Optistruct despite the fact that the volume was not specified for Optistruct; as opposed to BEETS and BESO which aimed for the specific target volume of 2100. The above results thus highlight the potential differences between the three software / algorithms which may be profound for relatively simplistic problems such as the two illustrated above. In an attempt to determine whether or not the above differences may be affected by the different types of FEA an additional case study was defined.

4.1.7 BEETS case study 3

The setup of case study 3 is in essence identical to case study 1 including material characteristics element size and the external load curve; the only difference is the number of boundary conditions and the specific locations where they are applied. These differences are illustrated in Figure 29.

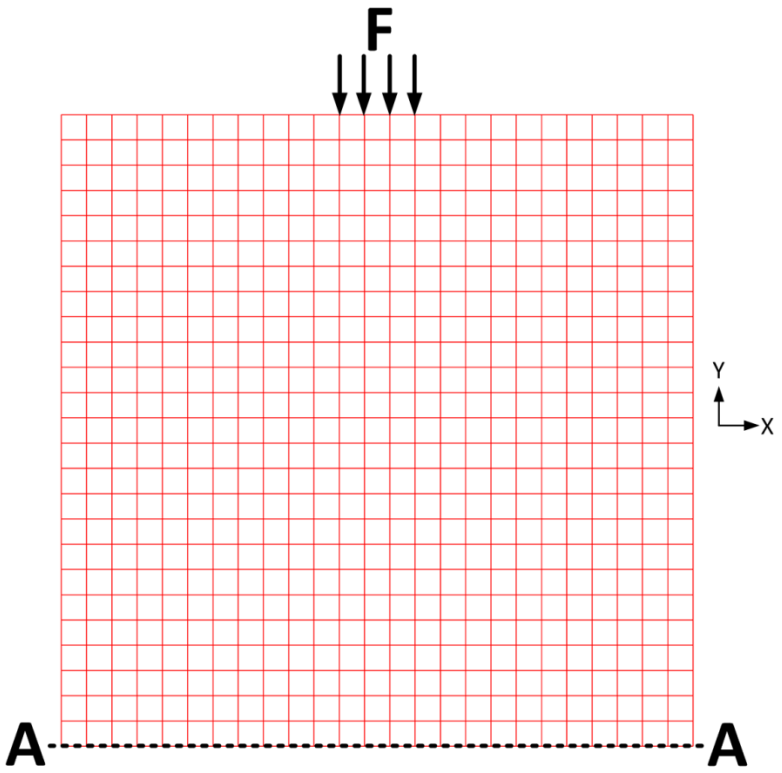


Figure 29, FE model for case study 3.

As indicated in Figure 29 the model of case study 3 is exposed to compressive loading exerted by 4 single point loads. The loads all follow the load vs time curve defined in Figure 13 and each have a magnitude of 100N. All DOF are constrained on all nodes along the A-A line in Figure 29. This optimisation problem should have two distinctly different solutions when approached from a static and a dynamic FE viewpoint. The static solution should theoretically be a column, whereas the dynamic solution should resemble and upside down V. The BEETS specific optimisation input parameters for case study 3 are defined in Table 13.

.

Table 13, BEETS input parameters for case study 3.

PowerShell variable	Value
Max_iter	25
TSI	26 (no Tabu Search included)
VTS	N/A
TSF	N/A
TABU_offset	N/A
TABU_LIST	N/A
TLF	N/A
ARMAX	0.03
RADIUS	0.0 (no neighbourhood influence)
TVF	0.3
ER	0.1
TERMTOL	0.05
MEPUSE	No
COOLUSE	No
COOLIT	N/A
SFCOOL	N/A

The BEETS optimisation terminated at iteration 13 as the average change over the final 3 iterations was 2.46%. The resulting topology is illustrated in Figure 30.

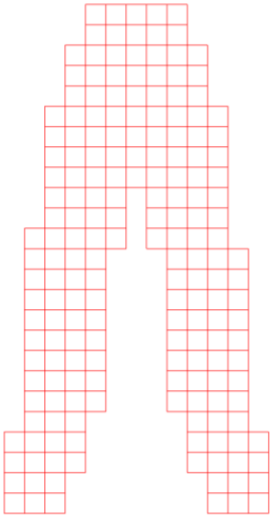


Figure 30, BEETS optimisation result from case study 3.

As anticipated the resulting structure resembles and upside down V despite the fact that the loading is applied in a quasi-static like manner. This is supported by the fact that the two "legs" of the V are relatively close to the centre line of the structure indicating a relatively low / moderate rate of deformation i.e. load application.

4.1.8 BESO case study 3

Using an identical model setup including relevant optimisation parameters from Table 13 the above case study was analysed using the linear static BESO software. The optimisation converged and terminated at iteration 20; the resulting topology is illustrated in Figure 31.

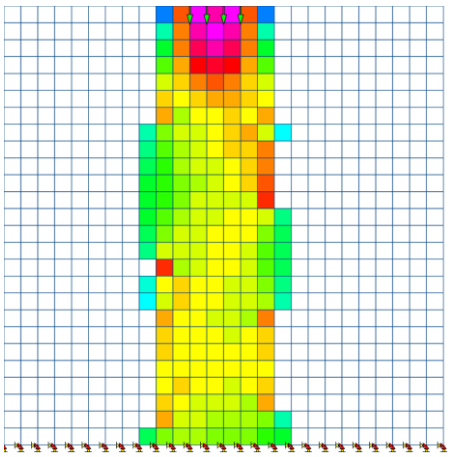


Figure 31, BESO optimisation result from case study 3.

Figure 31 clearly depicts a more “column like” structure when compared to that of Figure 30. These results thereby substantiate the initial speculation that the difference between BESO and BEETS optimisation results in case study 2 may be influenced by the difference in FEA approach.

4.1.9 HyperWorks case study 3

Adapting the above model setup into HyperWorks / Optistruct and including a compliance constraint the optimisation study was repeated. The Optistruct optimisation study converged at iteration 19; the resulting topology is illustrated in Figure 32.

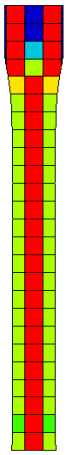


Figure 32, Optistruct optimisation result from case study 3.

Figure 32 thereby supports the statement that the “static” solution to the optimisation problem of case study 3 is a column, whereas Figure 30 supports the statement that the “dynamic” solution is an upside down V. Case study 3 thereby supports the notion that the difference in results of case study 2 is influenced by the difference in FE solution methodology. The difference between the two “columns” of Figure 31 and Figure 32 may be caused by the neighbourhood influence included in the BESO software. For case study 3 it is also interesting to note that the BEETS algorithm terminated at iteration 13, whereas the BESO software converged at iteration 20 and Optistruct at iteration 19.

Based on the 3 case studies above it can reasonably be argued that the “basic” results from the BEETS software are sensible thus initially verifying that the software provides ”correct” and “relevant” results. From this point onwards the focus of attention will now be turned to the overall area of interest: non-linear topology optimisation.

4.2 Non-linear topology optimisation case studies

This section will cover a series of case studies aimed at determining the suitability of the BEETS algorithm for non-linear topology optimisation. The initial case studies will in essence consist of utilising the "basic" BESO algorithm of the BEETS software. The starting point will contain material as well as geometrical non-linearity. Due to various factors including limitations of the initial BEETS software in addition to considerations in relation to stability of the FEA the fundamental setup of the FE model has been adapted. Instead of applying the external load directly onto individual nodes the external load will now be applied by defining an infinitely rigid plate moving at a constant velocity. The general setup of the "revised model" is illustrated in Figure 33.

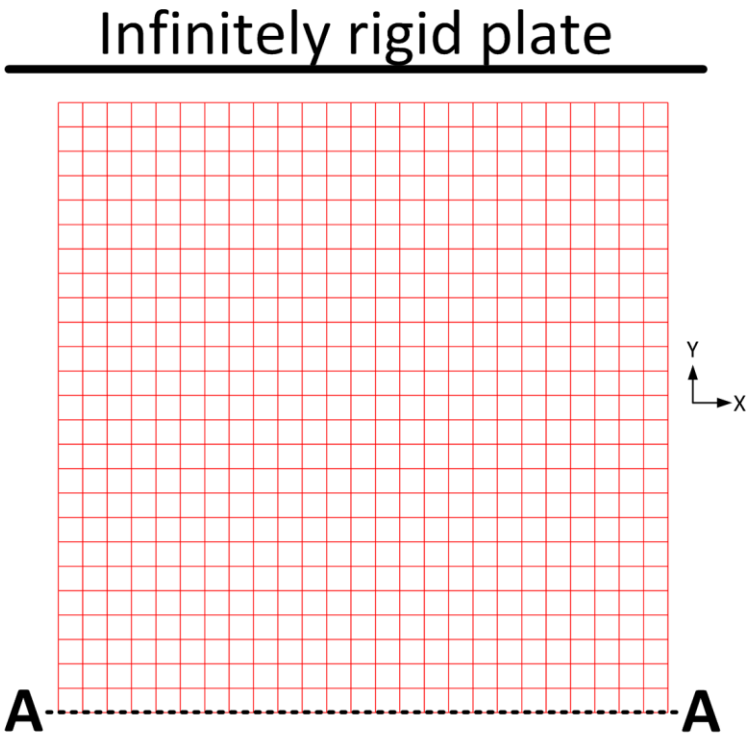


Figure 33, fundamental FE model for non-linear case studies.

The Infinitely Rigid Plate (IRP) in Figure 33 will move in the negative y-direction at a predefined constant velocity. The model will be fully constrained at all nodes along the A-A line. The IRP will have a predefined mass hence the external load is a function of the mass and the velocity of the IRP.

The material data used for the non-linear case studies will be based on data from actual physical test of Boron steel, (Bastien, 2014). The LS-Dyna material model used will be *MAT_024:Piecewise_linear_plasticity. This is an isotropic linear-elastic piecewise plastic material model with strain rate sensitivity. The general material parameters apart from the actual test curves are listed in Table 14.

Table 14, material properties for non-linear case studies.

Parameter	Value	Unit
Young's Modulus	210,000	MPa
Poisson's Ratio	0.3	
Yield Stress	374.9	MPa

4.2.1 BEETS case study 4

For this case study the mass of the IRP will be 100 kg, and the initial velocity in the negative y-direction, Figure 33 will be 5 m/s i.e. 18 km/h, which in comparison to a vehicle crash scenario is a relatively modest impact velocity. The BEETS optimisation parameters for this particular case study will remain unchanged to those from case study 3; i.e. as specified in table 13.

.

A better understanding of the fundamental differences between case study 4 and the preceding linear optimisation case studies can be obtained by reviewing Figure 34 which represents the final deformed shape of the structure at iteration 0 before any optimisation has occurred.

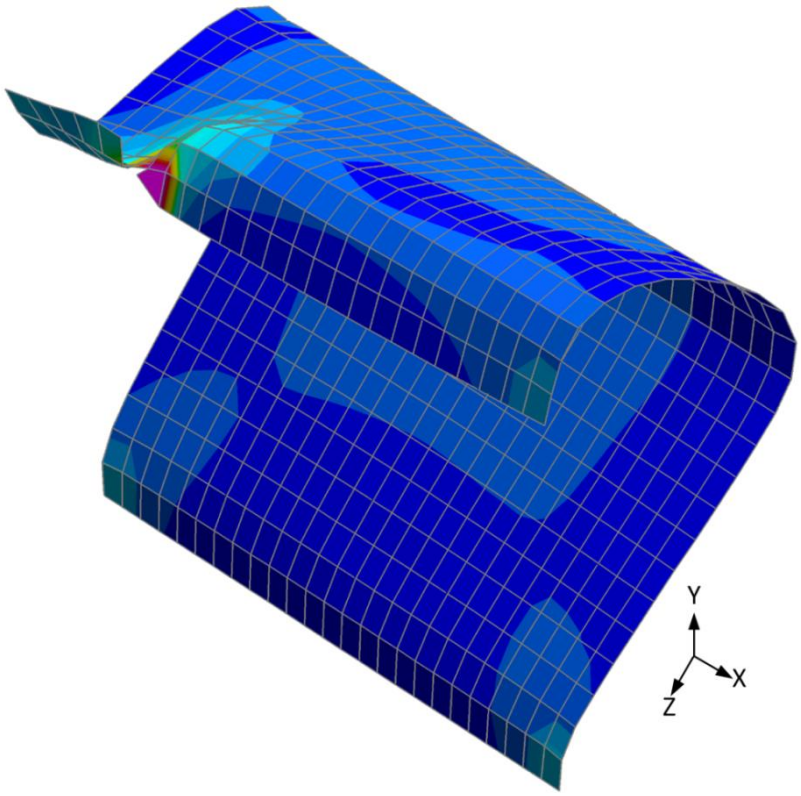


Figure 34, final deformed shape at iteration 0.

Figure 34 clearly illustrates a significant level of deformation and can therefore be argued to represent a structure exposed to large deformations. The maximum displacement resultant in the final deformation shape of Figure 34 is 29.50 mm. Figure 34 also clearly depict that a failure has occurred on the “top left hand side” of the structure.

The BEETS optimisation of the updated FE model presented above terminated at iteration 13 as the average change over the final 3 iterations was 3.3%. The resulting topology is illustrated in Figure 35.

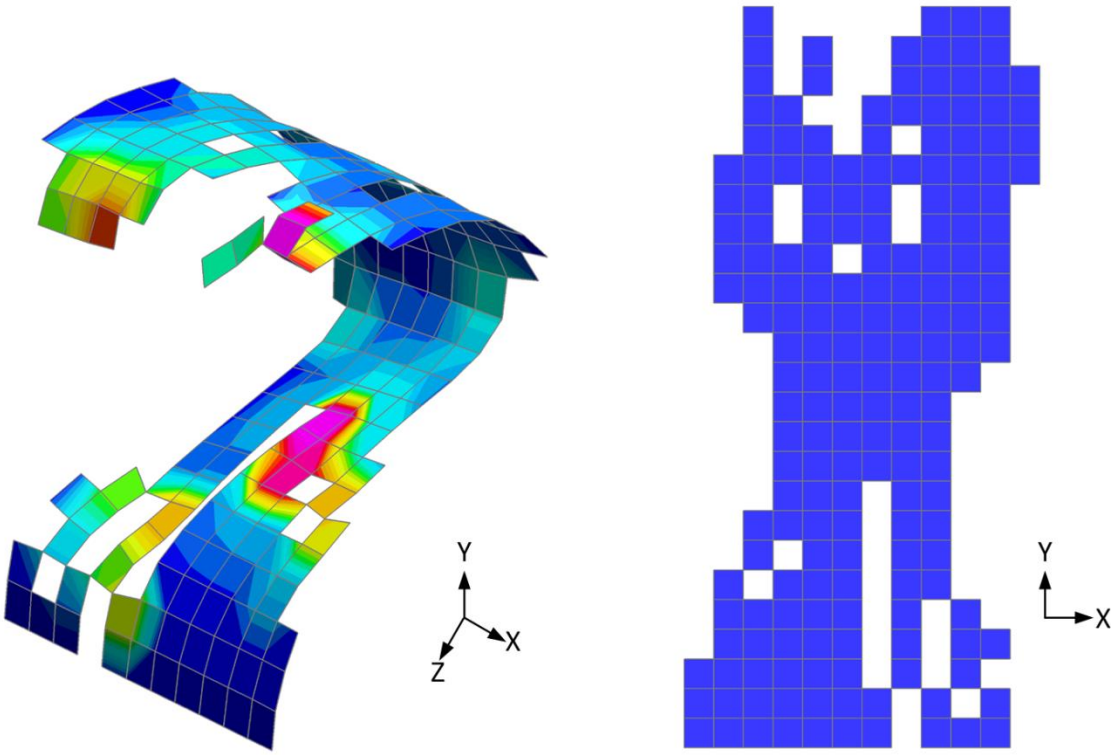


Figure 35, BEETS optimisation results case study 4.

The left hand side of Figure 35 illustrates the optimised structured at its final deformed shape. When comparing the final deformed shape on the left hand side of Figure 35 to that of the original structure in Figure 34 it can be seen that the overall deformation shape has not changed significantly. Furthermore, the maximum resultant displacement at the final deformed state has only changed significantly as well. The value from the original structure, i.e. in Figure 34 was 29.5 mm; the corresponding value from Figure 35 was 29.88mm. At this point it should be noted that the optimised structure only contains 30% of the volume of the original structure; it is also worth remembering that no stipulations with respect to maximum displacement values are applied to the optimisation. The right hand side of Figure 35 represent the undeformed optimised structure, which unsurprisingly is noticeably different from the optimised BEETS structure of case study 3; but does interestingly bare some resemblance with the results from the BESO software of case study 3. At this stage it does however not make any sense to directly compare the BEETS results from case study 4 those obtained via the BESO software or indeed Optistruct; the main reason being the significant levels of non-linearity and large deformations. It was also decided not to conduct any comparison studies to the ESLM method as these have been thoroughly examined in contribution 4 to this PhD portfolio. The only relevant remaining option is therefore to compare the above results to the HCA algorithm

of LS-TaSc initially presented in section 2.2.1; the purpose of these case studies are however to evaluate the change of results from the fundamental BESO algorithm, which resides inside the BEETS software. Once these are understood the results from the BEETS algorithm should be compared and evaluated against those of LS-TaSc.

Figure 36 and the volume history, Figure 37 represents the strain energy history from case study 4.

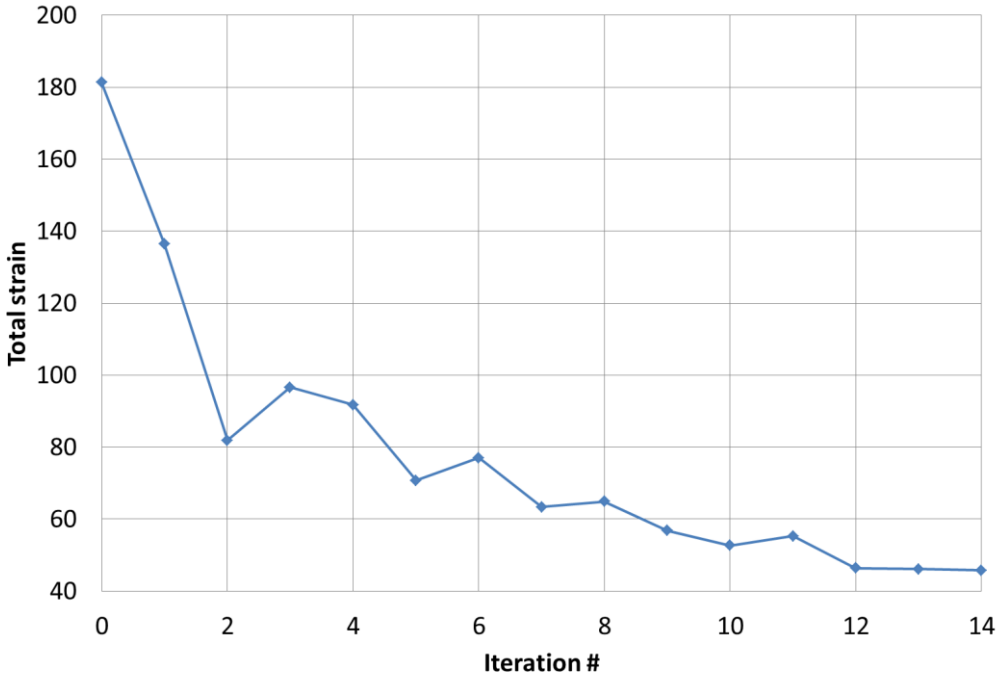


Figure 36, sum of strain per iteration case study 4.

Figure 36 clearly indicates that the sum of strain has substantially decreased from iteration 0 to iteration 14, i.e. a significant decrease in the total strain contained within the structure; despite the fact that the structure is still withstanding an identical amount of external loading. The information in Figure 36 coupled with the marginal increase in maximum displacement is a positive indicator of an optimised structure given the fact that the volume, i.e. mass, of the structure has also decreased by 70%. The volume history of case study 4, Figure 37, also indicates that the optimisation has converged.

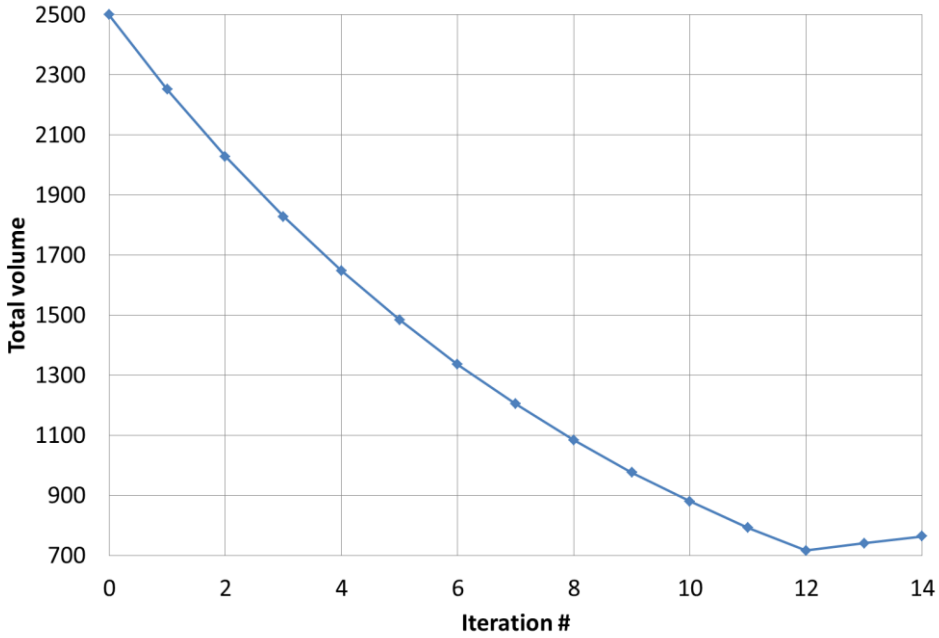


Figure 37, sum of volume per iteration case study 4.

At this point it is however worth remembering that the Evolutionary ratio (ER) dictates a consistent rate of change in elements per iteration. This coupled with the maximum allowed admissions ratio (ARMAX), and the termination tolerance (TERMTOL) dictates when the optimisation will stop, provided the cooling function is not activated. This means that it can generally be accurately predicted at which iteration the optimisation will terminate. It could be argued that a good rule of thumb is that TERMTOL should be less than both the ER and the ARMAX values; it naturally follows that ER must be larger than TERMTOL. This can be mathematically formulated as expression (2.47).

$$TERMTOL < ARMAX < ER \tag{2.47}$$

Expression (2.47) will be used as a rule of thumb for all subsequent BEETS analysis.
The focus will now shift entirely onto the BEETS algorithm; the remaining case studies will primarily focus on 5 aspects of the BEETS algorithm / software:

1. Effects or ER and ARMAX magnitudes.
2. Cooling function influence.
3. Neighbourhood influence.
4. Tabu search influence.
5. Effects of Entropy.

All 5 points above will be the individual focus of attention in the subsequent sections; for all cases the fundamental FE model of Figure 33 and the material parameters in Table 14 will be utilised unless otherwise specified. This also means that BEETS case study 4 will be used as the "benchmark" for the subsequent studies.

4.2.2 BEETS case study 5: effects of ER and ARMAX magnitudes

The aim of this case study is to initially investigate the effects of ER and ARMAX magnitudes. Expression (2.47) of section 4.2.1 suggested a rule of thumb for the relationship between the ER, ARMAX and TERMTOL values. Using this rule of thumb; is it possible to increase the ER and ARMAX magnitudes to reduce the overall computation time of the optimisation problem? In order to get an overview of this the optimisation model from BEETS case study 4 was repeated using the input parameters listed in Table 15.

Table 15, BEETS input parameters for case study 5.

PowerShell variable	Value
Max_iter	25
TSI	26 (no Tabu Search included)
VTS	N/A
TSF	N/A
TABU_offset	N/A
TABU_LIST	N/A
TLF	N/A
ARMAX	0.1
RADIUS	0.0 (no neighbourhood influence)
TVF	0.3
ER	0.35
TERMTOL	0.05
MEPUSE	No
PowerShell variable	**Value**
COOLUSE	No
COOLIT	N/A
SFCOOL	N/A

As defined by Table 15, the only changes between BEETS case study 4 and case study 5 is therefore a significant increase in ER and ARMAX values. This should mean that the optimisation will be able to reach its target volume fraction (TVF) of 0.3 in approximately 3 iterations; it is subsequently possible to significantly increase the number of elements as ARMAX has been increased to 0.1.

The optimisation study concluded due to the maximum number of 25 iterations been reached. At this point it is therefore very clear that the desired effect of reduced overall CPU time was not obtained. Reviewing the change of the structure during the initial 3 iterations provide a very direct and unsurprising clue as to why this is, Figure 38.

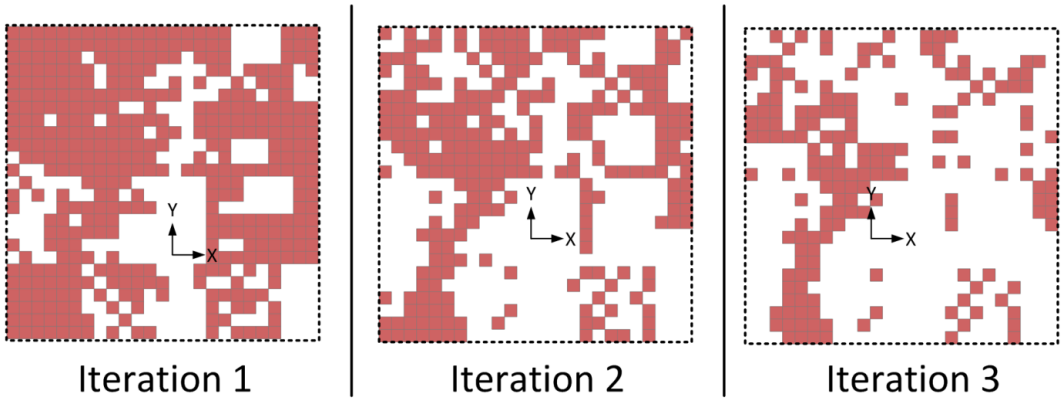


Figure 38, first 3 iterations BEETS case study 5.

As indicated by Figure 38, the optimisation algorithm has simply become “too aggressive” as too much material, i.e. elements, are removed in the initial iterations causing the (optimisation) model to become unstable. The explicit FE model does however remain stable allowing the optimisation to continue. Figure 39 illustrates the final structure obtained from case study 5. This bares some parallels to the ESLM case study of contribution 4; where not only the optimisation but also the FE analysis became unstable causing the optimisation to error terminate.

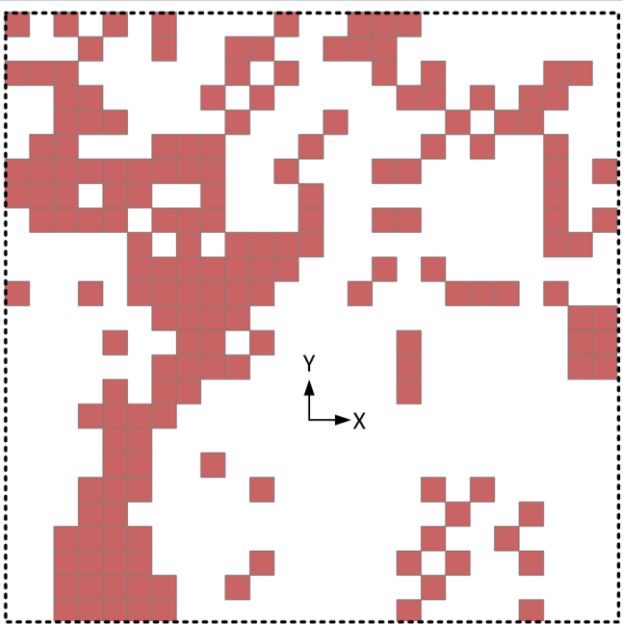


Figure 39, final structure from BEETS case study 5.

Figure 40 illustrates the volume history for BEETS case study 5. Observing the trends in the figure it can be seen that the volume goes below the target value of 750 (corresponding to a TVF value of 0.3) after approximately 3 iterations.

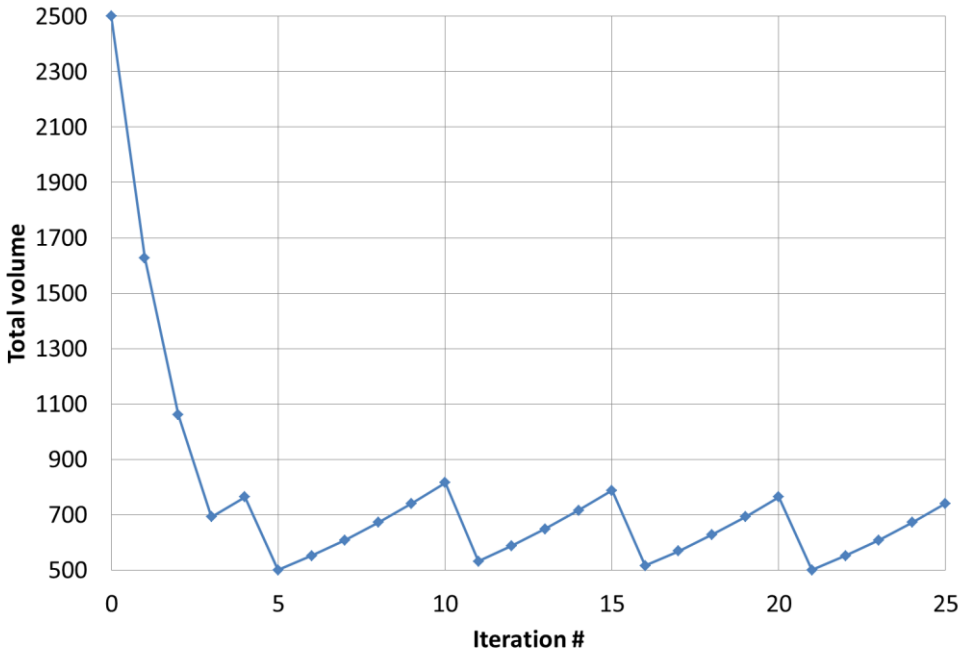


Figure 40, sum of volume per iteration case study 5.

From iteration 3 to iteration 4 the volume increases by 9.5%, i.e. very close to the maximum allowable value; meaning that the volume exceeds the target value of 750. The optimisation therefore reduces the value by 35% according to the ER value; meaning that the volume is significantly less than the target value at iteration 5. From iteration 5 to iteration 10 the volume steadily increases by 10% for each iteration as dictated by the ARMAX value. At iteration 10 the volume has once again exceeded the target volume and is decreased by 35% according to the ER value. From this point onwards this cyclic increase and decrease is repeated until the maximum number of iterations is reached as shown in Figure 40. Part of this trend identified in Figure 40 is an inherent flaw within the BESO algorithm; simply because the final target volume is not necessarily specified as a target volume for the individual iteration. This will naturally lead to the cyclic tendencies seen in Figure 40. Although case study 5 utilises "extreme" ER and ARMAX values it nevertheless highlights a flaw which may be solved by the introduction of a cooling function which will be the focus of attention in the following section.

4.2.3 BEETS case study 6: cooling function influence

The introduction of a cooling function may help to incrementally decrease or increase the "aggressiveness" of the optimisation algorithm in order to better control and guide the optimisation towards the target volume thus reducing or completely eliminating the cyclic behaviour illustrated in Figure 40. BEETS case study 6 is therefore a "replica" of case study 5 with the cooling function activated. The BEETS input parameters for case study 6 are listed in Table 16.

Table 16, BEETS input parameters for case study 6.

PowerShell variable	Value
Max_iter	25
TSI	26 (no Tabu Search included)
VTS	N/A
TSF	N/A
TABU_offset	N/A
TABU_LIST	N/A
TLF	N/A
ARMAX	0.1
RADIUS	0.0 (no neighbourhood influence)
TVF	0.3

PowerShell variable	Value
ER	0.35
TERMTOL	0.05
MEPUSE	No
COOLUSE	YES
COOLIT	0.1
SFCOOL	0.5

As defined in Table 16 the cooling function has been activated and will become active within an interval of ±10% of the final target volume. Within this range the ER value will be updated according to the SFCOOL value of 0.5. Using the cooling function the optimisation terminated at iteration 6 as the average change over the last 3 iterations was 3.04%. The resulting topology remains unfeasible i.e. similar to the topologies of Figure 38 and Figure 39. Figure 41 represent the volume as a function of iteration for case study 6.

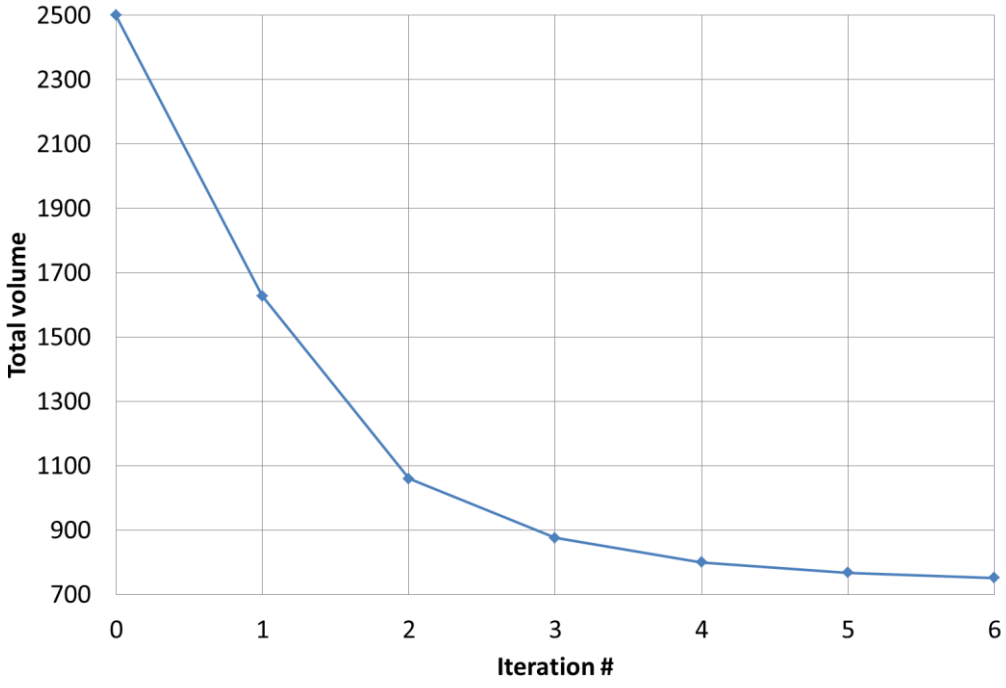


Figure 41, sum of volume per iteration case study 6.

As indicated in Figure 41 the introduction of the cooling function has completely removed the cyclic behaviour of the BESO algorithm indicated in Figure 40. The particular cooling parameters; i.e. COOLIT and SFCOOL does of course warrant further analysis and must be carefully chosen for each individual analysis; at this point the investigation into the cooling

function will however be temporarily abandoned in order to focus upon the next adaptation of BESO in the BEETS software; namely the neighbourhood influence.

4.2.4 BEETS case study 7: neighbourhood influence

The purpose of this sub-section is to initially investigate the influence of the neighbouring elements; i.e. the neighbourhood influence. For this particular purpose the BEETS input parameters for case study 7 are listed in Table 17.

Table 17, BEETS input parameters for case study 7.

PowerShell variable	Value
Max_iter	25
TSI	26 (no Tabu Search included)
VTS	N/A
TSF	N/A
TABU_offset	N/A
TABU_LIST	N/A
TLF	N/A
ARMAX	0.05
RADIUS	0 (M1); 2.1 (M2); 4.1(M3); 6.1(M4)
TVF	0.3
ER	0.1
TERMTOL	0.03
MEPUSE	No
COOLUSE	No
COOLIT	N/A
SFCOOL	N/A

The only change between BEETS case study 4 and 7 is thereby the RADIUS parameter. As indicated in Table 17 case study 6 will actually contain 4 models; M1, M2, M3 and M4. M1 will not contain any neighbourhood influence, but has been included as the ARMAX and TERMTOL parameters are not identical to those of BEETS case study 4. The average element size of the model is 2 mm; which means that the distance between two adjacent element centres is 2 mm. M2 should therefore contain the immediately adjacent elements only, whereas M3 and

M4 will contain an increasing amount of neighbourhood elements / influence as illustrated in Figure 5.

All four models of case study 7 terminated due to convergence prior to the maximum number of iterations. M1, M2 and M3 all terminated at iteration 24 with an average percentage difference over the final 3 iterations of 2.51%. M4 terminated at iteration 17 with an average volume change over the final 3 iterations of 2.93%. Figure 42 contains the total volume per iteration for all four models.

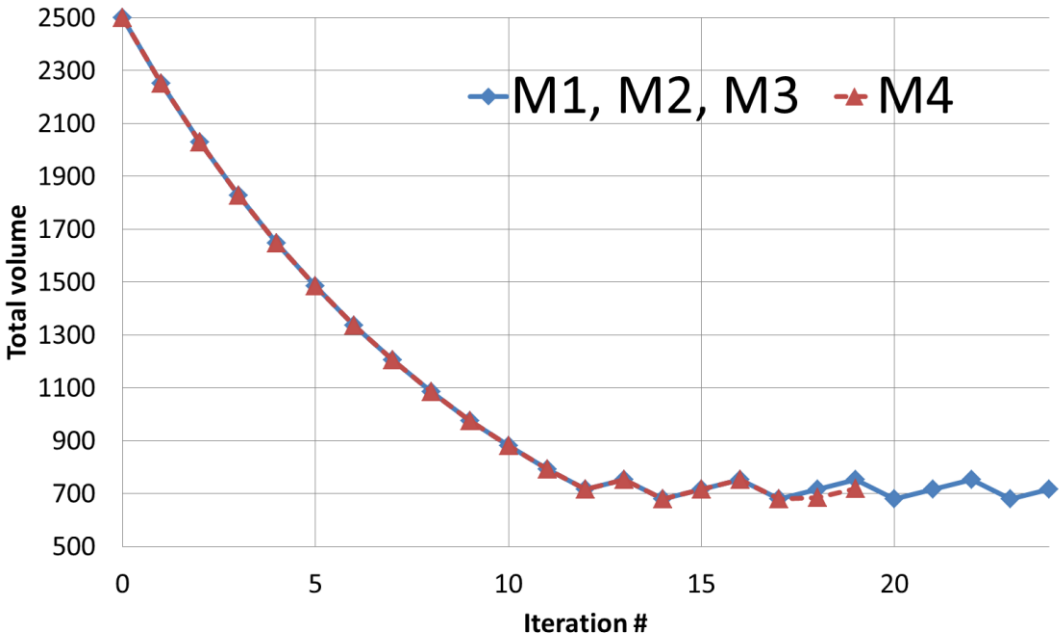


Figure 42, sum of volume per iteration case study 7.

As expected Figure 42 reveals that all 4 models follow the same pattern for volume reduction because this is dictated by the ER and ARMAX parameters; remembering that no cooling function is activated in case study 7. Figure 42 also reveals a slight change between M1-M3 and M4 from iteration 17 onwards causing the optimisation of M4 to terminate premature to the other 3 models. This is an interesting and rather unexpected fact; which indicates that the neighbourhood function may also have an influence on the termination, and indeed help reduce the cyclic tendencies which may occur if the cooling function is not activated as discussed in section 4.2.3. Figure 42 thereby also indicates “locking” of element sensitivity numbers indicating that difficulties with “singling out” elements has been introduced preventing the optimisation from reaching the iteration specific target volume. The existence

of this tendency may be further substantiated by observing the final optimised structures as depicted in Figure 43.

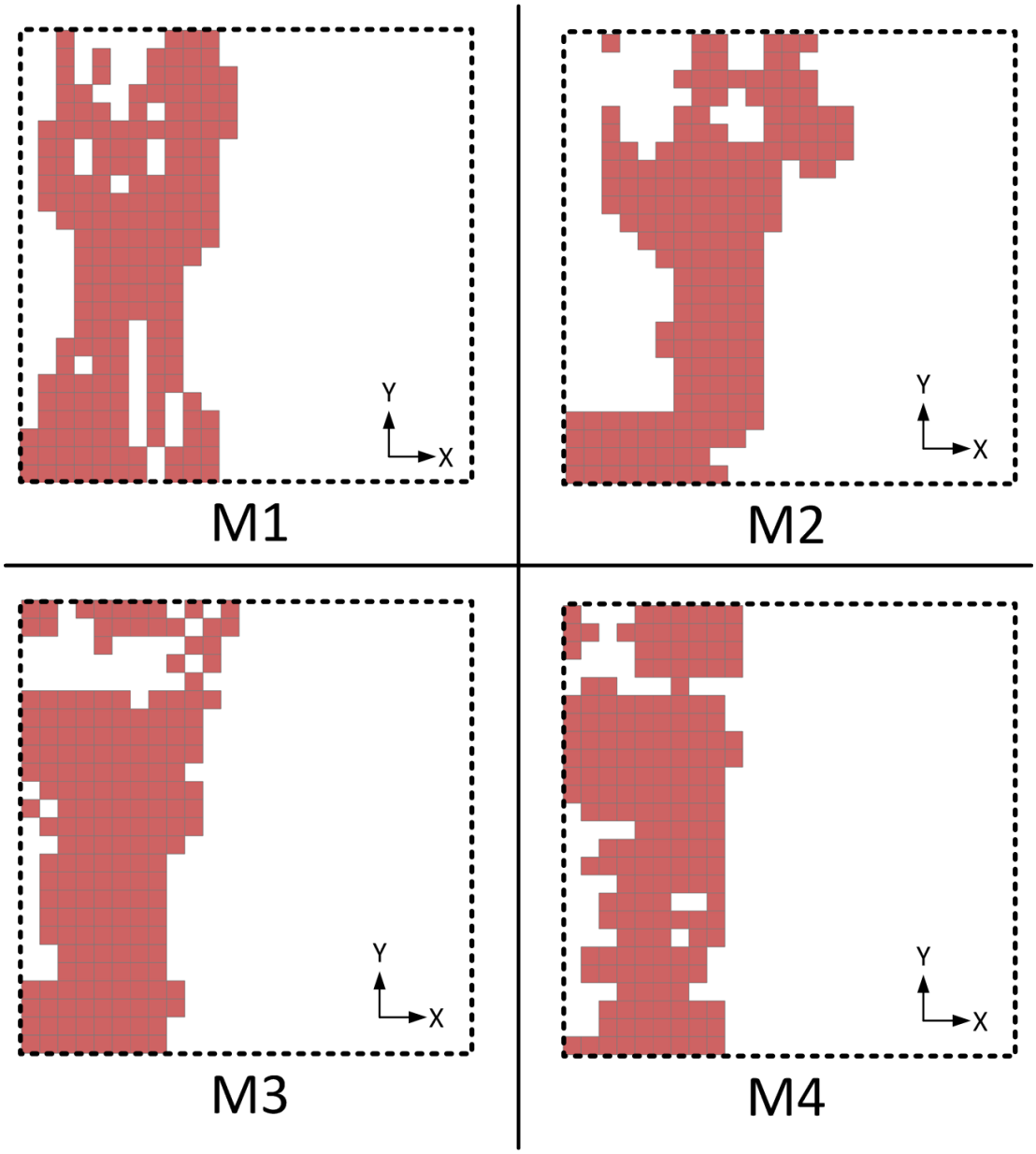


Figure 43, optimised shapes case study 7.

The dotted boxes in Figure 43 denote the design volume, i.e. the outline of the "original" FE model. Figure 43 thereby reveals that the load paths of the optimised structure are situated towards the left hand side of the design volume. This may initially seem odd as the model setup including constraints and applied loading is completely symmetrical. The reason for this

fact is however very simply that a single element fails (due to a very high strain level) on the left hand side of the model during the initial FE analysis, i.e. at iteration 0; which can also be seen in Figure 34.

Based on the results of all 4 models as illustrated in Figure 43 some interesting initial conclusions may be drawn:

1. **The inclusion of the neighbouring elements undoubtedly has a significant effect upon the resulting topology.** This is immediately obvious by comparing M1 to M2. There is also a significant difference between M2-M3 and M2-M4. Although there is detailed difference between M3-M4 there is also a general similarity in the location of the "primary load path". This is not the case when comparing M2-M3 and M2-M4.
2. **Including only the immediately adjacent neighbouring elements seemingly create a structure prone to "bending".** This statement is purely based on intuition; it is however clear that there is not a "straight" load path between the points of applied loading and the constraints, Figure 33. In fact by observing the results from M2 in Figure 43 it can be argued that a significant moment is introduced based on this "misalignment". This conclusion is however not straight forward as this may be obvious in the "undeformed" structures, Figure 43; however significant deformation and "self-contact" of the structure during deformation is not easily taken into account.
3. **Increased neighbourhood influence, i.e. RADIUS, seemingly lead to "instable structures" and "isolated elements".** As previously mentioned the introduction of the neighbourhood influence may prevent the optimisation from "singling out elements" for deletion. This is of course one of the main reasons for introducing the neighbourhood influence; in order to avoid the chequerboard effect; making the neighbourhood influence a positive attribute. However observing the "top left hand side" of the results of M2, M3 and M4 in Figure 43 it can be seen that "isolated element(s)" which are not connected to the "main structure" may also start to appear. In other words there are element(s) which are directly connected to the applied loading but not to the constraints. This is more prominent in the "high neighbourhood influence" models i.e. M3 and M4. This is of course not a positive attribute and can be partially contributed to the neighbourhood influence. These elements may lead to instability of the FEA; however due to the "inherent stability" of explicit FE the FEA and thereby the optimisation is able to continue. It does however seem that the

optimisation algorithm is not able to remove these elements as their element sensitivity number will remain relatively high. Furthermore it can be seen that the high neighbourhood elements M3 and M4 have an inherent instability as they have only a single element connecting the "top structure" where the external loading is applied to the "lower structure" which is directly connected to the constraints, in other words a chequerboard effect.

The above observations are of course based upon a very limited amount of information as additional models and scenarios must be carefully explored before any general conclusions and trends can be substantiated; the above discussions are therefore not meant as "final conclusions" but merely as indicators for additional analysis and areas of exploration of future work.

4.2.5 BEETS case study 8: Tabu Search influence

The purpose of this section is to initially explore the effects of applying Tabu Search (TS) onto the fundamental BESO algorithm. The primary reason behind the application of TS is an attempt to avoid premature convergence towards a local optimum point; in other words to expand the search within the solution space. For BEETS case study 8 the optimisation parameters are defined in Table 18.

Table 18, BEETS input parameters for case study 8.

PowerShell variable	Value
Max_iter	25
TSI	3
VTS	5
TSF	0.2
TABU_offset	5
TABU_LIST	Y
TLF	0.5
ARMAX	0.05
RADIUS	0.0 (no neighbourhood influence)
TVF	0.3
ER	0.1
TERMTOL	0.03

PowerShell variable	Value
MEPUSE	No
COOLUSE	No
COOLIT	N/A
SFCOOL	N/A

The optimisation of case study 8 terminated after iteration 25 due to the maximum allowable number of iterations being reached. Figure 44 represents the volume as a function of iteration number for case study 8.

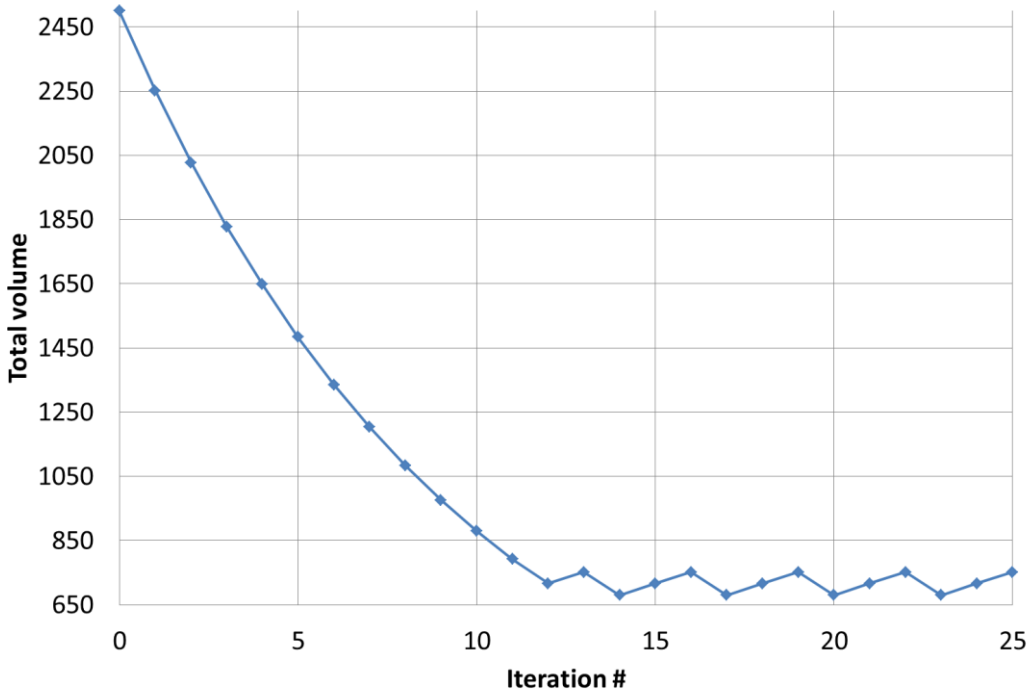


Figure 44, sum of volume per iteration case study 8.

As indicated in Figure 44 case study 8 also exhibited the cyclic behavior of the fundamental BESO algorithm without a cooling function. Figure 44 also indicate that the introduction of TS has not had any effect upon the change in volume per iteration. This is however not unexpected as the target volume for TS runs within a given iteration number is identical to that of the "normal" iteration. In other words the introduction of TS will not lead to any drastic changes of volume; instead it may lead to a different solution / base structure for subsequent iterations. If the base structure has changed as a function of TS the information will be stored in the Main_INFO.txt file output from the BEETS software. Reviewing this file for case study 8 it

seems that in for this case study no change in base structure has been chosen as a function of introducing TS. There may be several reasons for this, including but not limited to:

1. The optimum structure for the individual iterations is in fact the structure obtained using the "fundamental BESO" algorithm. This could be further explored by exploring other FE models, preferably where "known" optimum solutions are available.

2. The design volume is "too coarse" for the effects of TS to become noteworthy. This could be further explored by increasing the element count.

3. The evaluation expression for "measuring" the effectiveness of TS is incorrect and/or too simplistic. At present this is simply measured by dividing the sum of strain (for the entire model) by the volume of that model.

4. The constraint of utilising an identical volume for the TS may be too restrictive. Allowing the individual TS iterations to reduce the volume simultaneously with "randomly" removing / adding elements may enhance the influence of the TS iterations.

As was the instance with case study 7 the above statements are based on a very limited amount of information and cannot be justified to be generally applicable. The statements should merely be considered as pointers for additional research, analysis and scrutiny.

4.2.6 BEETS case study 9: effects of MEP

The purpose of this section is to initially explore the effects of introducing the Maximum Entropy Principle (MEP) into the fundamental BESO algorithm. This case study will contain a total of 4 models. M1 and M2 which will be the first areas of attention will simply include the FE model originally introduced at the start of section 4.2, i.e. illustrated in Figure 33 and utilising the material parameters in Table 14; with the Entropy parameter deactivated (M1) and activated (M2). The input parameters for the BEETS optimisations of case study 9 are defined in Table 19.

.

Table 19, BEETS input parameters for case study 9.

PowerShell variable	Value
Max_iter	25
TSI	26 (no Tabu Search included)
VTS	N/A
TSF	N/A
TABU_offset	N/A
TABU_LIST	N/A
PowerShell variable	**Value**
TLF	N/A
ARMAX	0.05
RADIUS	0.0 (no neighbourhood influence)
TVF	0.3
ER	0.1
TERMTOL	0.03
MEPUSE	No (M1, M3) / Yes(M2, M4)
COOLUSE	No
COOLIT	N/A
SFCOOL	N/A

M1 and M2 of BEETS case study 9 terminated at iteration 25 due to the convergence criteria being met as the average change of volume over the final 3 iterations was 2.51%. Figure 45 represents the volume as a function of iteration number for case study 9 M1 and M2.

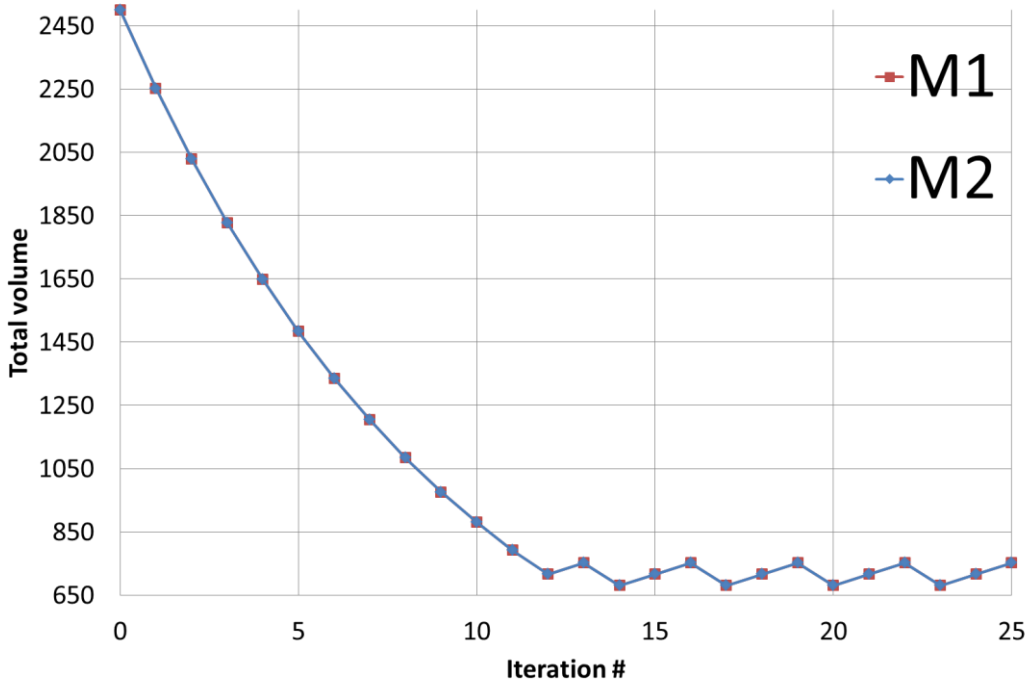


Figure 45, sum of volume per iteration case study 9 M1 and M2.

Comparing the two graphs in Figure 45 to each other it is clear that there is no difference at all between the graphs indicating that the introduction of MEP has had no effect upon the volume as a function of iteration; this is however as expected.

Figure 46 represents the total strain per iteration which indicates a significant difference between the two graphs from M1 and M2 suggesting a significant influence of introducing MEP.

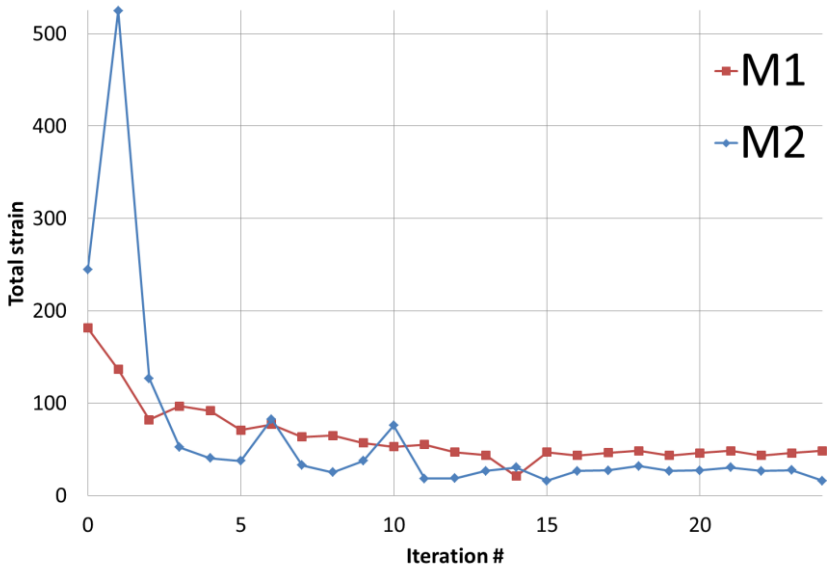


Figure 46, sum of strain per iteration case study 9 M1 and M2.

Figure 46 reveals that the model without MEP (M1) generally detects a lower total strain value as opposed to M2 (with MEP) for the first 3 iterations. Thereafter the total strain of M2 is generally lower than M1; and consistently lower from iteration 15 onwards. Due to the methods in which MEP has been implemented in the BEETS software, as described in section 3.1.1, it was anticipated that the MEP model would indicate higher total strain values (due to the method of summation); the fact that the MEP model consistently displays lower total strain values from iteration 15 onwards is a strong indicator that this model (M2) is a more "optimised" model than M1 as the volume of the two remains identical as documented by Figure 46. The identical volume is confirmed when reviewing the resulting FE models from M1 and M2 which both contain 188 elements after the optimisation. The resulting FE models / topology from the two models are illustrated in Figure 47.

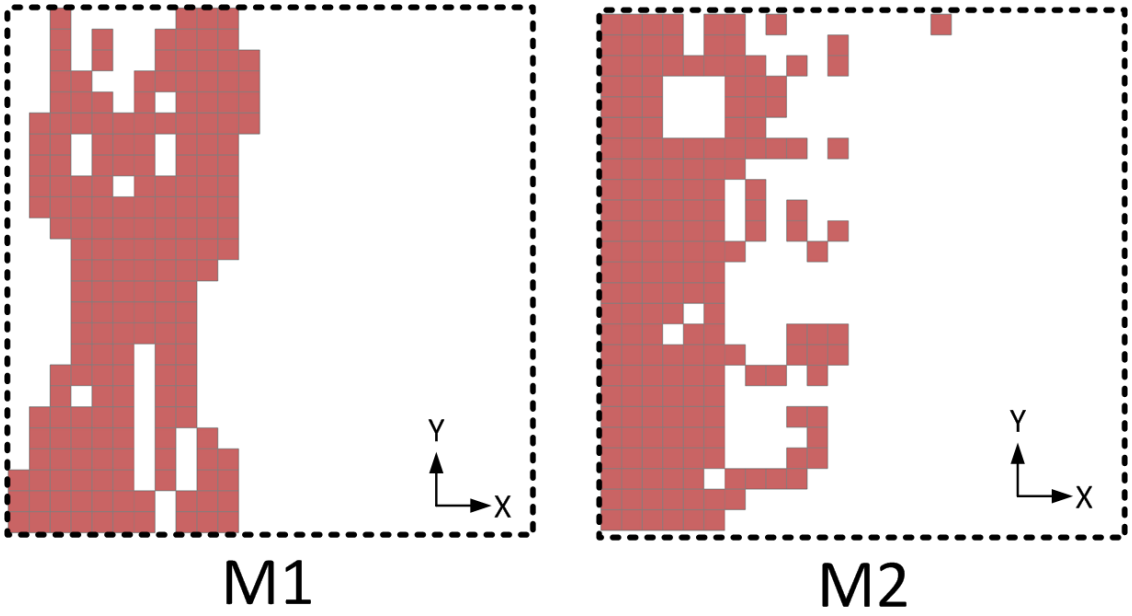


Figure 47, resulting topologies case study 9 M1 and M2.

Based on Figure 47 there can be no doubt that the introduction of MEP has a significant effect upon the results of the optimisation. On one hand the results from M2 demonstrates a very distinct load path on the far left of the design volume; but on the other hand it also seems to "suffer" from chequerboard effect. The effects of introduction MEP are therefore very interesting and most certainly warrant further analysis. In order to further analyse this, an additional two models are introduced; M3 and M4. The fundamental setup of these two models is identical to those of M1 and M2; the method in which the loading is applied upon the infinitely rigid plate is however different. Observing a "typical" automotive crash scenarios it is clear that the loading does not necessarily remain consistent throughout the impact; in fact there is generally a point where the two impacting vehicles move away from each other, this is generally known as "rebound". In the context of an FE based crash analysis this means that the maximum deflection does not necessarily occur at the final timestep; which is very likely to have an effect upon optimisation using BESO; but what are the effects using BEETS? In order to initially investigate this, the rigid plates of M3 and M4 will be subjected to a prescribed motion; not an initial velocity following the values listed in Figure 48.

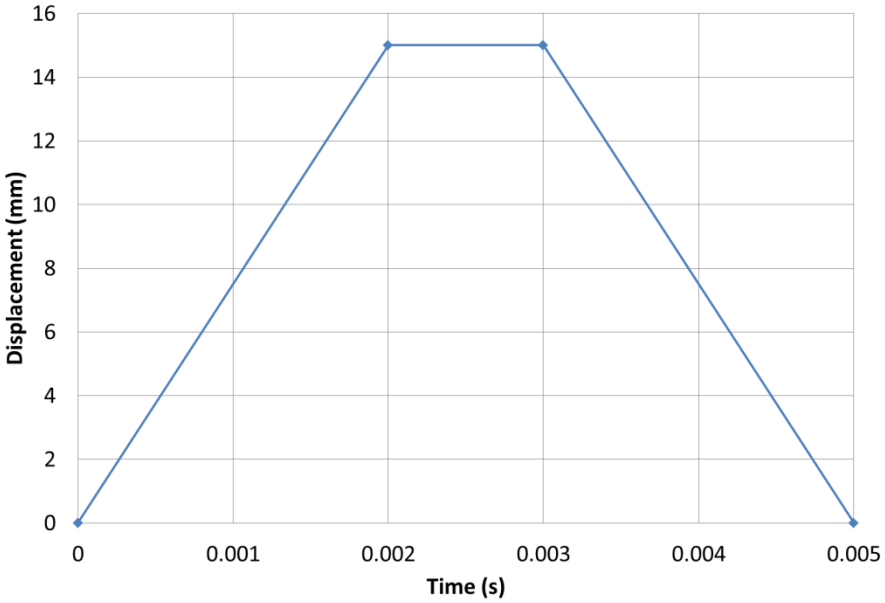


Figure 48, load vs. time M3 and M4 case study 9.

In Figure 48 the maximum displacement value of 15 mm has been found by observing the maximum displacement value of the rigid plate from case study 4. In case study 4 this value was however obtained at 0.005s, as defined in Figure 48 the maximum displacement will initially be obtained at 0.002s indicating a significant increase in strain rate. The two models M3 and M4 will thereby not only represent a difference in when the maximum deflection occurs but also a significant increase in deformation velocity.

Both M3 and M4 terminated at iteration 25 due to the convergence criteria being met; the average change of volume over the final 3 iterations was 2.51%. The volume versus iteration graphs for M3 and M4 are identical to those of M1 and M2 illustrated in Figure 45 and will therefore not be illustrated. The total model strain as a function of iteration number for both M3 and M4 is illustrated in Figure 49.

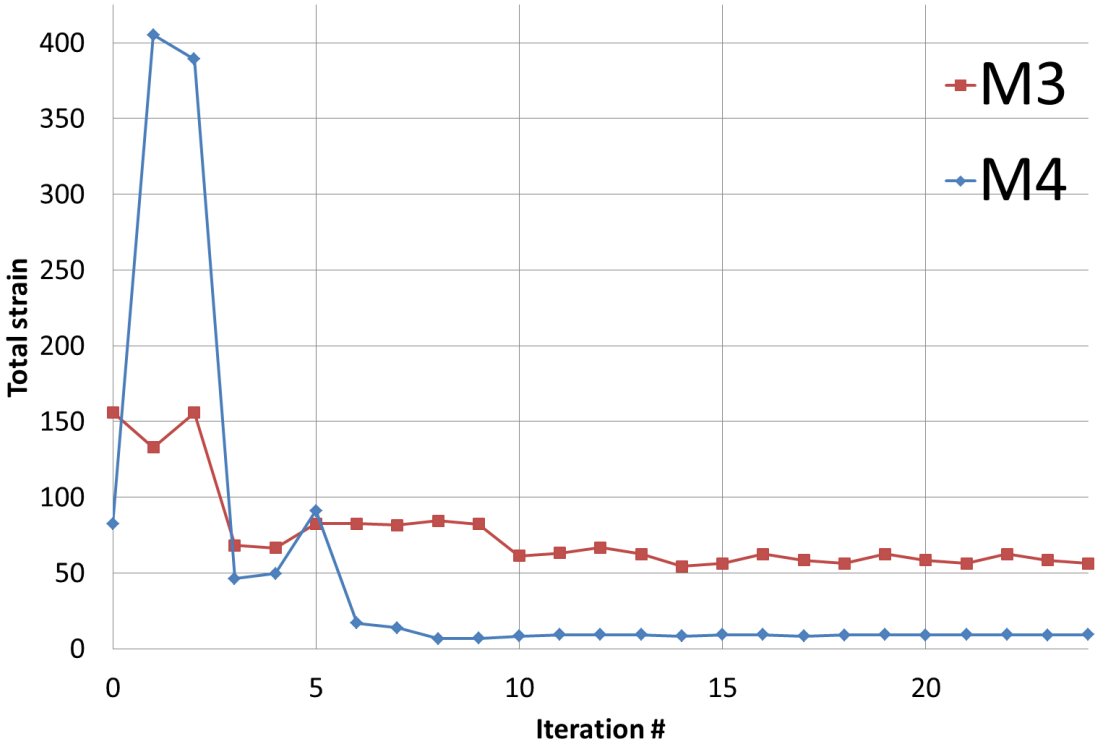


Figure 49, sum of strain per iteration case study 9 M3 and M4.

Analogue to the trends identified in Figure 46 Figure 49 also indicate that the MEP enabled model (M4) initially displays a higher total strain value than the model without MEP (M3). From iteration 6 onwards the total strain is consistently less for the MEP enabled model; furthermore the total strain value is seemingly very consistent, almost constant from iteration 10 onwards.

The resulting topologies from M3 and M4 are illustrated in Figure 50.

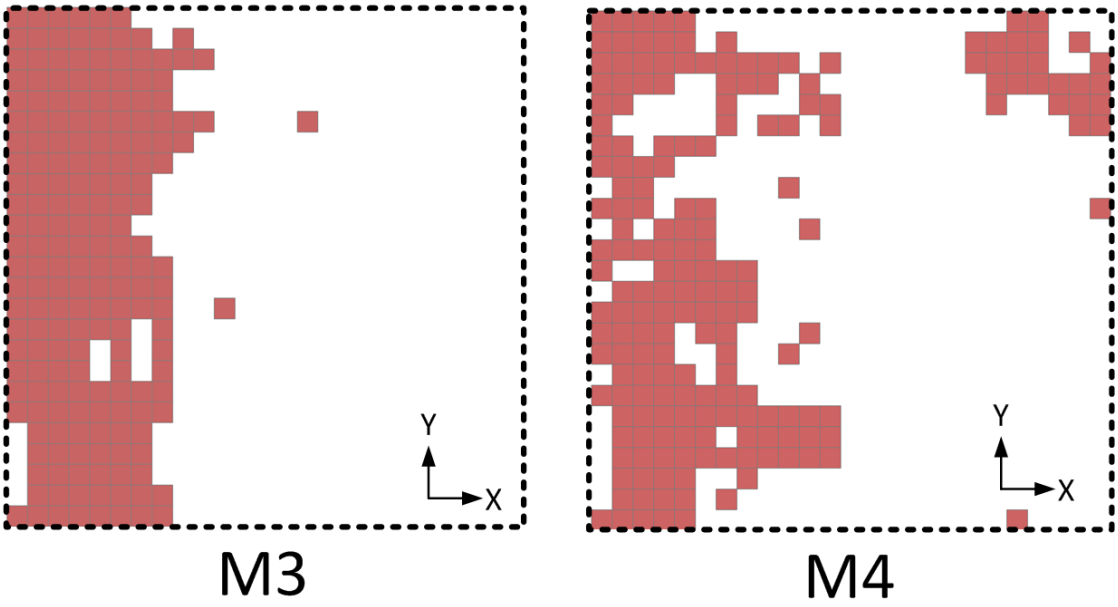


Figure 50, resulting topologies case study 9 M3 and M4

Figure 50 reveals a relatively simple and well defined structure for the MEP disabled model (M3); whereas the MEP enabled structure, i.e. M4, is less well defined; and seemingly subjected to chequerboard effect. The overall trends from M3 and M4 are thus very similar to those of M1 and M2; in general:

1. **The MEP enabled models seemingly produce more optimised structures.** This statement is purely based upon the total strain to volume ratio.

2. **The MEP enabled models are generally exposed to the chequerboard effect.** This is very visible in both Figure 47 and Figure 50.

Although case study 9 contains 4 models the above statements are still based on a very limited amount of information and cannot be justified to be generally applicable. The statements should therefore merely be considered as pointers for additional research, analysis and scrutiny.

At this point the "Individual" variations of BESO in the BEETS algorithm have been separately investigated; the combined effect / influence has however not. This will be the focus of attention in the next section.

4.2.7 BEETS case study 10: combined effect of BEETS

The purpose of this section is to briefly investigate the combined effects of the principles and adaptations implemented into the BEETS algorithm / software versus the "original" BESO

algorithm. For this purpose the "original" FE model presented in section 4.2 will be utilised. The input parameters for case study 10 M1 of the BEETS optimisation study are defined in Table 20.

Table 20, BEETS input parameters for case study 10 M1.

PowerShell variable	**Value**
Max_iter	25
TSI	3
VTS	5
TSF	0.3
TABU_offset	1
TABU_LIST	Yes
TLF	0.5
PowerShell variable	**Value**
ARMAX	0.05
RADIUS	2.1
TVF	0.3
ER	0.1
TERMTOL	0.03
MEPUSE	Yes
COOLUSE	Yes
COOLIT	0.1
SFCOOL	0.5

The "benchmark model" for case study 10 in general is M1 from case study 7, where ER, ARMAX and TERMTOL are identical to those defined in Table 20; hence M1 from case study 7 will be used for comparison. Figure 51 illustrates the sum of volume per iteration of case study 10 M1 and also includes case study 7 M1 for comparison.

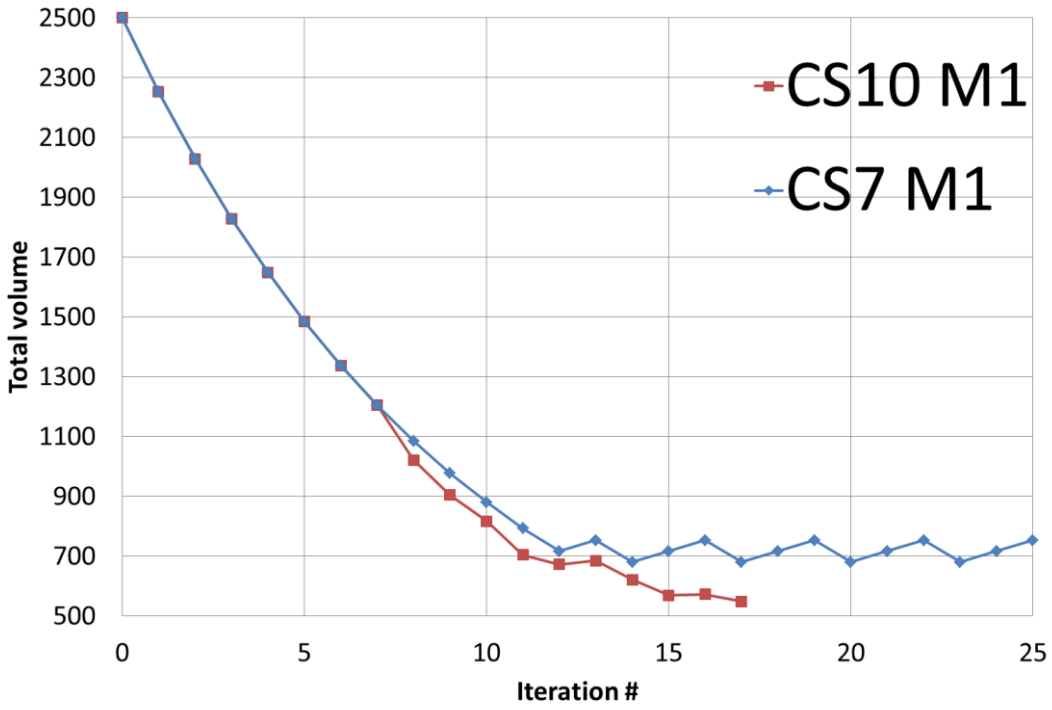


Figure 51, sum of volume per iteration case study 10 M1 and case study 7 M1.

As indicated in Figure 51 the BEETS algorithm terminates prior to the BESO algorithm; avoiding the cyclic behaviour of the BESO algorithm. Figure 51 does however also reveal that the BEETS algorithm undershoots the target volume of 750. The primary reason for this can be found by reviewing the Main_INFO.txt file. After iteration 9 the volume is in excess of the final target volume of 750; however as the volume has entered the cooling interval the ER value has been decreased. During iteration 10 the software determines that a number of elements need to be added to the model; however due to the reduced ER ratio and the limiting ARMAX value the software is unable to add as many elements as desired; leading to an undershoot in actual volume after iteration 10. This is simply because the ARMAX value is not cross-linked with the reduction in ER leading to an inconsistency in iteration target volume (which was 775 for iteration 10) and the actual volume of the model after the iteration (which was 704 for iteration 10). This trend continues until the optimisation criterion has been met; leading to a significant undershoot in target volume. This problem needs to be resolved by reviewing the interaction between ER, ARMAX and the cooling function.

The Main_INFO.txt file also reveals that the TS has an effect upon the overall optimisation as “alternative” more efficient solutions was chosen based upon the TS at iteration 3, 6, 9 and 15. Figure 52 illustrates the strain per iteration for case study 10 M1 and case study 7 M1.

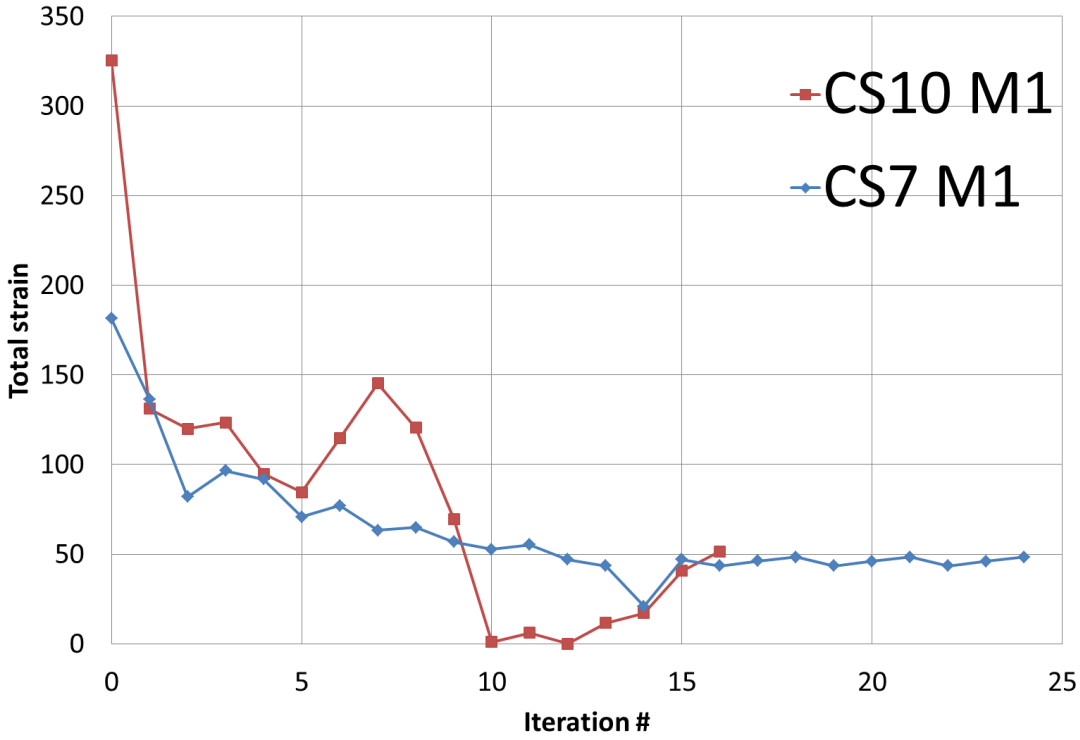


Figure 52, sum of strain per iteration case study 10 M1 and case study 7 M1.

As indicated in Figure 52 the MEP enabled model (M1 of case study 10) initially displays a higher total strain value; which then drops below that of the MEP disabled model (M1 of case study 7), despite a significantly lower volume. The resulting topologies from both models are illustrated in Figure 53.

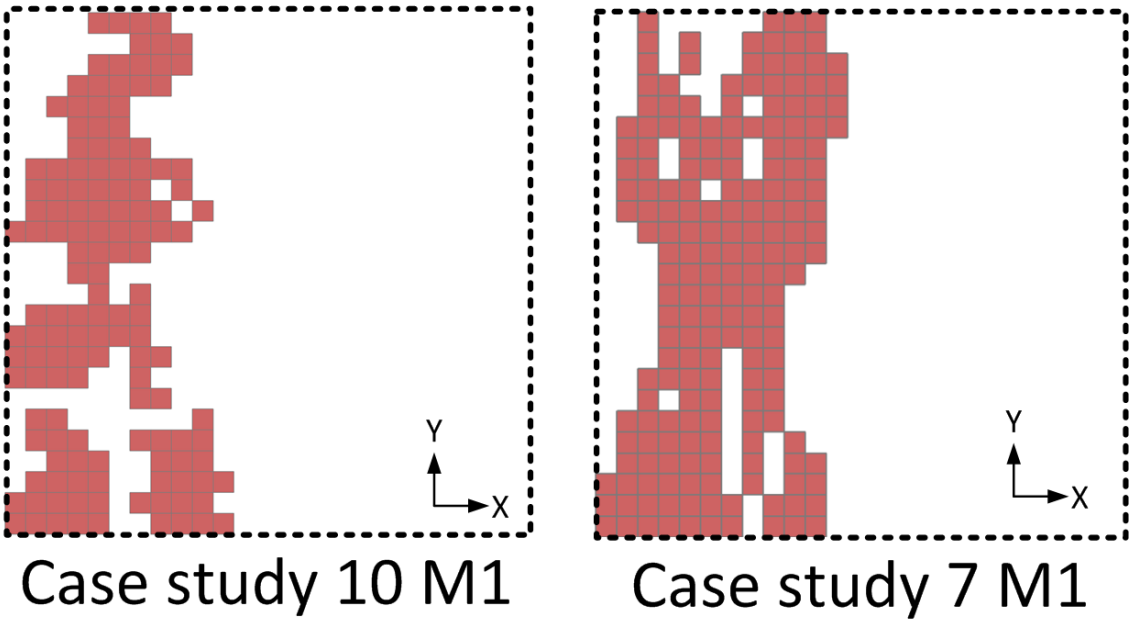


Figure 53, resulting topologies case study 10 M1 and case study 7 M1.

As can be seen in Figure 53 the MEP enabled model is subjected to the chequerboard effect; however not necessarily to a large extent. The main issue with case study 10 M1 is that the model is “disconnected”; i.e. there is no direct connection between the external loading applied at the top and the constraint applied at the bottom. This could however be a function of the undershoot phenomenon discussed above and to a greater extent a function of the limited neighbourhood influence. In an attempt to determine whether the latter is true or not an additional model is introduced. The FE model of this additional model, case study 10 M2, is identical to that of case study 10 M1; only the RADIUS and TERMTOL BEETS input parameters have been changed as specified in Table 21.

.

Table 21, BEETS input parameters case study 10 M2.

PowerShell variable	Value
Max_iter	25
TSI	3
VTS	5
TSF	0.3
PowerShell variable	**Value**
TABU_offset	1
TABU_LIST	Yes
TLF	0.5
ARMAX	0.05
RADIUS	6.1
TVF	0.3
ER	0.1
TERMTOL	0.01
MEPUSE	Yes
COOLUSE	Yes
COOLIT	0.1
SFCOOL	0.5

The optimisation of case study 10 M2 termiated at iteration 15 due to the convergence criteria being met. The average change over the final 3 iterations was 0.87%. The volume as a function of iteration for case study 10 M2 is illustrated in Figure 54.

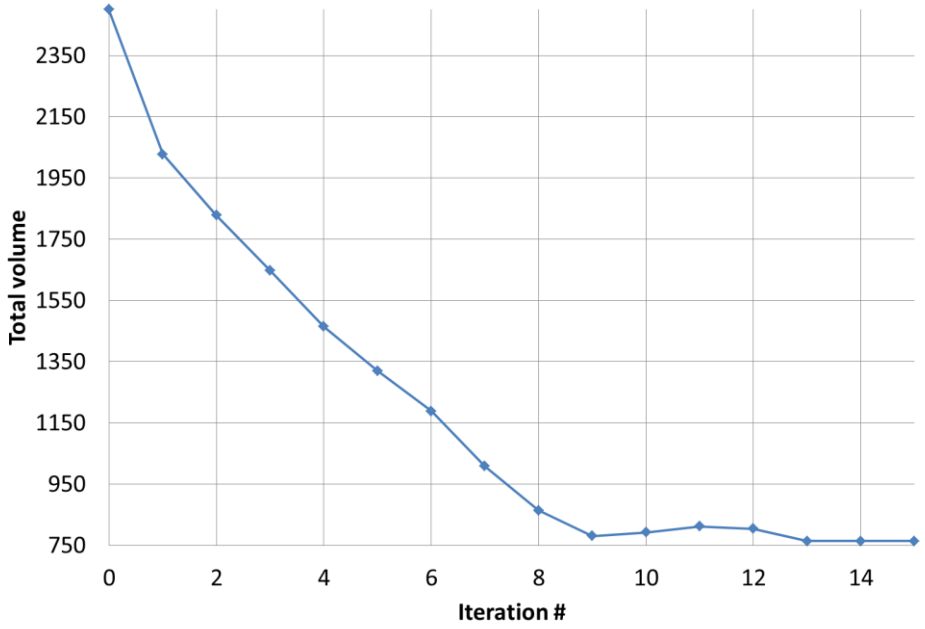


Figure 54, sum of volume per iteration case study 10 M2.

By comparing Figure 54 to Figure 51 it can be seen that the BEETS optimisation more closely converges towards the final volume fraction of 750; and that the neighbourhood influence helps control this as no undershoot has occurred in case study 10 M2.

Figure 55 represents the total strain as a function of iteration number for case study 10 M2.

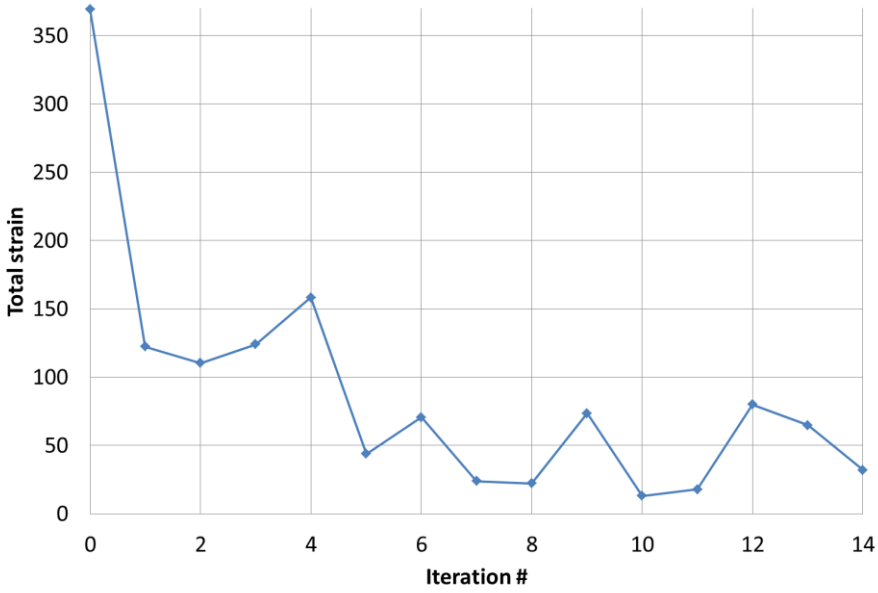


Figure 55, sum of strain per iteration case study 10 M2.

Figure 55 also indicates that case study 10 M2 is more stable than M1 simply because the total strain is less fluctuant than that displayed in Figure 52. Figure 56 represents the resulting topology from case study 10 M2.

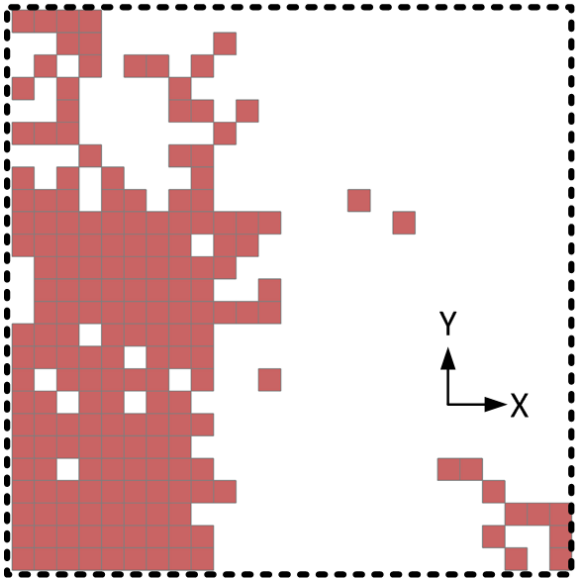


Figure 56, resulting topology case study 10 M2.

Comparing Figure 56 to Figure 53 it can be seen that the former retains the direct link between the external loading and the constraints. There is however no denying that Figure 56 is also prone to the chequerboard effect, perhaps even more so than case study 10 M1 of Figure 53. The increase in neighbourhood influence therefore seems to have eased some of the problems from case study 10 M1 but introduced or enhanced others.

4.3. Conclusion of the potential of BEETS for non-linear topology optimisation

Based on the case studies documented in section 4.1 and 4.2 of this report it is possible to draw some general preliminary conclusions of the suitability of the BEETS algorithm for non-linear topology optimisation. Firstly it should be emphasised that the conclusions are not final; and that they are based upon relatively limited amount of information in essence based on a single case study with variable parameters; hence little can be concluded about the overall versatility and robustness of the algorithm. Ignoring these limitations for the time being it is clear that the BEETS algorithm has introduced some changes with respect to the BESO algorithm; and attempted to reduce and in some cases even eliminate the limitations of the

BESO algorithm. The disadvantages of particular interest were initially listed in Table 7. Table 22 lists these "original" disadvantages and summarises the findings from the BEETS algorithm based upon the case studies of this chapter.

Table 22, findings in response to "original" BESO disadvantages.

"Original" BESO advantage from Table 2	**Importance of resolving disadvantage**	**Preliminary conclusions of BEETS based upon case studies**
Quasi-static loading assumption.	Medium	The quasi-static load assumption plays a lesser role with the introduction of MEP. A number of case studies have shown that a structure may be optimised even when exposed to very high strain rates. The limitations of the quasi-static loading assumption are still those of explicit FEA; and thereby dependent upon the timestep size.
Non-convergent and potential oscillation between "equal" design solutions.	High	This problem has been addressed by the introduction of the cooling function which clearly reduced or even eliminated the oscillation phenomenon in case study 6.
No direct consideration of "intermediate" deformation modes / levels.	High	This has been addressed by the introduction of the MEP; case study 9 revealed the suggested benefits of this supposedly leading to a more optimised solution; the feasibility of these solutions could however be questioned.
Relatively high computational cost.	Low	This aspect has in essence not been analysed. The primary reason for this is simply that this is highly dependent upon the actual programming; handling and storage of large data sets. This falls outside the scope of the current work meaning that no relevant and factual comparison can be completed at this stage.

"Original" BESO advantage from Table 2	**Importance of resolving disadvantage**	**Preliminary conclusions of BEETS based upon case studies**
Significant issues with high levels of non-linear material behaviour.	Medium	The BEETS model does not seem to have experienced any noteworthy problems with the higher levels of non-linear behaviour or even higher strain rates. However the limited scope of these initial case studies must be taken into consideration at this stage.
Low dispersion of search within potential solution space, high probability of "converging" towards local optimum points.	High	It is evident that the potential solution space has been significantly increased (on a variable scale) due to the adaptation of the TS; which in certain cases actively changed the structure in given iterations.

Based on the previous case studies and the outcomes summarised in Table 22 above it can be concluded that further refinements and adaptations of the BEETS software would be advantageous before an extended case study, including robustness, sensitivity and versatility is conducted. From the preceding case studies 4 main areas of interest for future work is suggested:

1. **Objective function definition and monitoring.** Although it has been suggested that the strain energy is the best measure for the objective function this has not been analysed. Furthermore, the current implementation of monitoring and assessing the total strain is rather crude and simplistic as previously discussed.
2. **MEP effects and implementation.** Although the introduction of MEP seemingly has had a positive effect upon the volume to total strain ratio it also seems to have introduced higher levels of the chequerboard effect. Furthermore, the actual influence of the intermediate strain values and the "built in" neighbourhood influence need to be further addressed. Finally, the MEP influence upon the total strain value per iteration must also be advanced.

3. **Direct link and influence between ARMAX, cooling function and ER.** As discussed throughout section 4.2 there seems to be an inherent flaw in the BEETS algorithm which causes the optimisation to undershoot or even overshoot the final target volume simply because the ARMAX ratio is not affected by the cooling function and vice versa. The link between ARMAX, the cooling function and the ER must therefore be reviewed.
4. **Limitations of current BEETS version.** At present the BEETS software only caters for structures in a 2D geometrical space. This needs to be expanded to 3D in order to review more relevant case studies, such as for example an automotive crush can.

5. Conclusion

This monologue consists of 3 main parts; an extended literature review; an algorithm definition and implementation and finally a series of case studies.

The literature review enabled a greater overview of potentially applicable optimisation algorithms that could be adapted to cater for large deformation non-linear topology optimisation. The general advantages, disadvantages and overall potential of each individual algorithm were evaluated in addition to being compared to the "competing" algorithms. Although this literature review was extensive and contains the general principles, it does not contain all potential optimisation algorithms; in fact it will be almost impossible to map all existing algorithms quite simply because there are countless variations of the general algorithms adapted to individual problems. Nevertheless the extended literature review provided good and relevant background information which helped guide the continued research into the definition and implementation of a non-linear topology optimisation algorithm. Based upon the literature review in addition to the information gathered via other research activities; the majority of which are documented in the other contributions to this PhD portfolio, it was decided to "review and upgrade" the BESO algorithm as it was deemed to hold the highest potential for resolving the need for a non-linear topology optimisation algorithm, despite the fact that it does not contain any evidence or convergence justification.

The second main part of this monologue consisted of reviewing the advantages and more importantly the disadvantages of BESO optimisation including rating the importance of resolving the specific disadvantages as well as ideas of how these could be resolved. The result was the BEETS optimisation algorithm which was subsequently implemented into a BEETS software package based upon PowerShell scripting and utilising LS-Dyna as an explicit FE solver. This software constitutes contribution 6 to this PhD portfolio. Initial verification relating to e.g. the correctness of the software importing the FE data was also completed.

Finally a series of case studies were set up and used to firstly evaluate the performance of the BEETS software in a linear static environment, including "benchmarking" to established FE based optimisation tools such as the VDM-SIMP method of HyperWorks / Optistruct as well as comparison to the "original" BESO algorithm using an independent BESO program and the "base" BESO algorithm in the BEETS software itself. The outcome of the linear static study concluded that there were differences between the software; these could however be

contributed to circumstantial parameters such as the inability of the external BESO software to disable the neighbourhood influence as well as the inherent differences of implicit versus explicit FEA.

The second part of the case studies focused on topology optimisation of non-linear structures exposed to large deformations and relative high strain rates. These case studies exclusively focused on the BEETS software; as no feasible means of comparison exist. It should also be remembered that the purpose of this second part of the case studies was to initially evaluate the feasibility of the BEETS algorithm for large deformation and non-linear topology optimisation before further development and research is conducted. The individual variations applied to the fundamental BESO algorithm was examined in isolation and finally the combined effects were also analysed. The initial results were not fully conclusive in establishing the effectiveness or indeed feasibility of resulting topology from the BEETS optimisation; they were however indicative that further analysis would be beneficial to fully explore the potential of the algorithm / software, as some of the disadvantages listed in Table 7 were successfully addressed as summarised in Table 22. The case studies also revealed some areas of further refinement which should be addressed before extensive case studies are undertaken; these mainly related to the interaction between ER, ARMAX and the cooling function in addition to the information gathering and handling with respect to strain values output from LS-Dyna.

Once the above issues have been addressed and a revised version of the BEETS software created more advanced case studies should be solved and compared to results from e.g. LS-TaSc as well as conducting physical experiments to verify the performance, validity and indeed manufacturing feasibility of the results obtained.

References

Alavala, C. R. (2008), “Fuzzy Logic and Neural Networks” (online) available from: <https://www.academia.edu/1435724/Fuzzy_Logic_and_Neural_Networks_by_Chennakesava_R._Alavala> (01 September 2014)

Altair (2009), “blog” (online) available from: <http://blog.altair.co.kr/wp-content/uploads/2011/03/optistruct_optimization_10-0.pdf> (15 October 2014)

Barati, R. (2014) “A novel approach in optimization problem for research reactors fuel plate using a synergy between cellular automata and quasi-simulated annealing methods” (online) available from: <http://www.sciencedirect.com/science/article/pii/S0306454914001091> (14 September 2014)

Bastien, C. (2014), confidential LS-Dyna data from research project (28 November 2014)

Bianci et al. (2008) ”A survey on metaheuristics for stochastic combinatorial optimization” (online) available from: < http://link.springer.com/article/10.1007%2Fs11047-008-9098-4> (27 August 2014)

Bianci et al. (2008a) ”A survey on metaheuristics for stochastic combinatorial optimization” (online) available from: <http://link.springer.com/article/10.1007%2Fs11047-008-9098-4> page 246 (27 August 2014)

Bureerat, S., Limtragool, J. (2008) “Structural topology optimisation using simulated annealing with multiresolution design variables” (online) available from: <http://www.sciencedirect.com/science/article/pii/S0168874X08000681> (22 September 2014)

Bureerat, S., Limtragool, J. (2008a) “Structural topology optimisation using simulated annealing with multiresolution design variables” (online) available from: <http://www.sciencedirect.com/science/article/pii/S0168874X08000681> page 747 (22 September 2014)

Chen, W. (2012) “Robust Shape &Topology Optimization under Uncertainty” (online) available from: <http://www.google.co.uk/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&sqi=2&ved=0CDsQFjAC&url=http%3A%2F%2Fwww.altairhyperworks.com%2F(S(abosnynyy2lsvkdbzowbzg05))%2FResLibDownload.aspx%3Ffile_id%3D1434%26from%3DResourceLibrary.aspx%253Fkeywords%253DRobust%252BShape%252Band%252BTopology%252BOptimization&ei=yP_ZU4akPMf17AbOwIHQBA&usg=AFQjCNE440sQsyqlLmJH77yf3--XRjz5-w&sig2=jsVrHwXIRksfY1YIDVlQsQ&bvm=bv.72185853,d.ZGU> (03 November 2014)

Chuang, C.H., Yang, R.J. (2012) “Benchmark of Topology Optimization Methods for Crashworthiness Design” (online) available from: < http://www.dynalook.com/international-conf-2012/optimization-metal-forming18-a.pdf> (22 September 2014)

Chuang, C.H., Yang, R.J. (2012a) “Benchmark of Topology Optimization Methods for Crashworthiness Design” (online) available from: <http://www.dynalook.com/international-conf-2012/optimization-metal-forming18-a.pdf> page 3 (22 September 2014)

Garcia-Lopez, M. et. al. (2011) “A hybrid topology optimization methodology combining simulated annealing and SIMP” (online) available from: <http://www.sciencedirect.com/science/article/pii/S0045794911001210> (10 November 2014)

Garcia-Lopez, M. et. al. (2011a) “A hybrid topology optimization methodology combining simulated annealing and SIMP” (online) available from: <http://www.sciencedirect.com/science/article/pii/S0045794911001210> page 1514 (10 November 2014)

Garcia-Lopez, M. et. al. (2013) “An improved robust topology optimization approach using multiobjective evolutionary algorithms” (online) available from: < http://www.sciencedirect.com/science/article/pii/S0045794913001569> page 1 (01 November 2014)

Garcia-Lopez, M. et. al. (2013a) “An improved robust topology optimization approach using multiobjective evolutionary algorithms” (online) available from: < http://www.sciencedirect.com/science/article/pii/S0045794913001569> (01 November 2014)

Hosny, M. I. (2010) page 16 “Investigating Heuristic and Meta-Heuristic Algorithms for Solving Pickup and Delivery Problems” (online) available from: <http://users.cs.cf.ac.uk/C.L.Mumford/Research%20Topics/PDPTW/papers/Manar.pdf> (03 September 2014)

Huang, X. and Xie Y.M. (2008) “Topology optimization of nonlinear structures under displacement loading” (online) available from: <http://www.sciencedirect.com/science/article/pii/S0141029608000205> (02 May 2014)

Huang, X. and Xie Y.M. (2008a) “Topology optimization of nonlinear structures under displacement loading” (online) available from: <http://www.sciencedirect.com/science/article/pii/S0141029608000205> page 2059 (02 May 2014)

Huang, X. and Xie Y.M. (2008b) “Topology optimization of nonlinear structures under displacement loading” (online) available from: <http://www.sciencedirect.com/science/article/pii/S0141029608000205> page 2060 (02 May 2014)

Huang, X. and Xie Y.M. (2008c) “Topology optimization of nonlinear structures under displacement loading” (online) available from: <http://www.sciencedirect.com/science/article/pii/S0141029608000205> page 2061 (02 May 2014)

Huang, X. and Xie Y.M. (2008d) “Topology optimization of nonlinear structures under displacement loading” (online) available from: <http://www.sciencedirect.com/science/article/pii/S0141029608000205> page 2065 (02 May 2014)

Huang, X. and Xie, Y. M. (2010) "A further review of ESO type methods for topology optimization", *Structural and Multidisciplinary Optimization*, *volume 41, issue 5*, *pp. 671-683*

Jaynes, E.T. (1957) “Information Theory and Statistical Mechanics” (online) available from: <http://bayes.wustl.edu/etj/articles/theory.1.pdf> (18 July 2014)

Kennedy, J. and Eberhart, K. (1995) “Particle Swarm Optimization” (online) available from: <http://www.cs.tufts.edu/comp/150GA/homeworks/hw3/_reading6%201995%20particle%20swarming.pdf> (30 July 2014)

Kennedy, J. and Eberhart, K. (1995a) “Particle Swarm Optimization” (online) available from: <http://www.cs.tufts.edu/comp/150GA/homeworks/hw3/_reading6%201995%20particle%20swarming.pdf> page 1943(30 July 2014)

Kirkpatrick, S., Gelatt Jr., C. D., Vecchi, M. P. (1983) “Optimization by Simulated Annealing” (online) available from: <http://minds.jacobs-university.de/sites/default/files/uploads/teaching/share/KirkpatrickSimulatedAnnealing.pdf>
(30 July 2014)

Lin, C.Y., Lin, S.H. (2005) “Artificial neural network based hole image interpretation techniques For integrated topology and shape optimization” (online) available from: < http://www.sciencedirect.com/science/article/pii/S0045782504004657#> (23 October2014)

LSTC (2011) “Topology and Shape Computations for LS-Dyna” (online) available from: < http://www.lsoptsupport.com/documents/manuals/ls-tasc/ls-tasc-2.0-manual.pdf/view > (June 2014)

LSTC, (2011a) “Topology and Shape Computations for LS-Dyna” (online) available from: < http://www.lsoptsupport.com/documents/manuals/ls-tasc/ls-tasc-2.0-manual.pdf/view > page 11 (June 2014)

Luh, G.C., Lin, C.I., Lin, Y. S. (2011) “A binary particle swarm optimization for continuum structural topology optimization” (online) available from: <http://www.sciencedirect.com/science/article/pii/S1568494610002905 (08 September 2014)

Luh, G.C., Lin, C.I., Lin, Y. S. (2011a) “A binary particle swarm optimization for continuum Structural topology optimization” (online) available from: <http://www.sciencedirect.com/science/article/pii/S1568494610002905 page 2844 (08 September 2014)

Marsalli, M. (2006) ”McCulloch Pitts Neurons” (online) available from: <http://www.mind.ilstu.edu/curriculum/mcp_neurons/mcp_neuron_1.php>
(6 September 2014)

MatLab (2014) “Monte Carlo Simulation” (online) available from: <http://www.mathworks.co.uk/discovery/monte-carlo-simulation.html> (08 August 2014)

Park, G.J. (2011) “Technical overview of the equivalent static loads method for non-linear static response structural optimization” (online) available from: <http://link.springer.com/article/10.1007%2Fs00158-010-0530-x> (03 September 2014)

Penfield, P. Jr (2003) “Principle of Maximum Entropy” (online) available from: <http://www-mtl.mit.edu/Courses/6.050/notes/chapter9.pdf> (08 July 2014)

Penfield, P. Jr (2003a) “Principle of Maximum Entropy” (online) available from: <http://www-mtl.mit.edu/Courses/6.050/notes/chapter9.pdf> page 80 (08 July 2014)

Penfield, P. Jr (2003b) “Principle of Maximum Entropy” (online) available from: <http://www-mtl.mit.edu/Courses/6.050/notes/chapter9.pdf> page 82 (08 July 2014)

Penfield, P. Jr (2003c) “Physical Systems” (online) available from: <http://www-mtl.mit.edu/Courses/6.050/notes/chapter10.pdf> page 89 (12 July 2014)

Penfield, P. Jr (2003d) “Physical Systems” (online) available from: <http://www-mtl.mit.edu/Courses/6.050/notes/chapter10.pdf> page 90 (12 July 2014)

Penfield, P. Jr (2003e) “Physical Systems” (online) available from: <http://www-mtl.mit.edu/Courses/6.050/notes/chapter10.pdf> page 91 (12 July 2014)

Researchgate (2014) “Yi Min Xie – publications” (online) available from: <http://www.researchgate.net/profile/Yi_Xie4/publications> (28 July 2014)

RMIT (2014) “Software – RMIT University (online) available from: <http://www.rmit.edu.au/research/cism/software> (08 August 2014)

Roux, W. (2011) “ Topology Design using LS-TaSC Version 2 and LS-Dyna” available from: < http://www.dynamore.de/de/download/papers/konferenz11/papers/session4-paper4.pdf> (12 June 2014)

Roux, W. (2011a) “ Topology Design using LS-TaSC Version 2 and LS-Dyna” available from: < http://www.dynamore.de/de/download/papers/konferenz11/papers/session4-paper4.pdf> page 2 (12 June 2014)

Schrijver, A. (2003) “Combinatorial Optimization” Springer: London.

Singh, S. (2014) *Re-design of car seat rails and its height elevation system*. Unpublished MEng Thesis. Coventry: Coventry University.

Sorensen, R. and Lund, E. (2012) “Ply-based Optimization of Laminated Composite Shell Structures under Manufacturing Constraints” [online] available from: <http://www.escm.eu.org/eccm15/data/assets/1870.pdf> [23 August 2014]

Tovar, A. (2004) “Bone Remodelling as a Hybrid Cellular Automaton optimization Process ” (online) available from: <http://etd.nd.edu/ETD-db/theses/available/etd-12082004-004120/unrestricted/TovarA122004.pdf> (2 August 2014)

Vazirani, V. (2001) “Approximation Algorithms” (online) available from: <http://www.cc.gatech.edu/fac/Vijay.Vazirani/book.pdf> (23 August 2014)

Appendix A

MatLab MEP code:

```
clear;
clc;
%Maximum entropy example - 1 constraint only.

%Reset error warnings
   ERROR = 0;
%Initial problem definitions:
        %The sum of probabilities must equal 1:
                Psumresidual = 1E-5; %Residual for final check
         %Design parameters: e.g. "individual" element density
                D(1) = 1;
                D(2) = 1;
                D(3) = 1;
                D(4) = 1;

        %Constraint 1: e.g. overall strain energy
         Con1 = 2.5;
        Con1residual = 2;%E-5; %Residual to determine ERROR termination or not.
        %Known quantities (e.g. from FEA) Q is a known quantity; e.g. the combined srain
        energy must equal 0.3 --> Q = 0.3; qA(i) is the quantity associated with the individual
        parameter  / option; i.e.:
                 Q = Con1;
                 qD(1) = 1;
                 qD(2) = 2;
                qD(3) = 3;
                qD(4) = 8;
%Solution methodology:
   %Step 1: Setup and solve f_Beta = SUM((qAi-Q)*2^(-Beta(qAi-Q)))i = 0 --> maximum entropy
   S (by finding the Lagrangian multipliers Alpha and Beta):
        %Define initial Beta guess
                Beta=1;%0.25;
         %Newton-Raphson
                %Define maximum number of allowed iterations
                        maxiter = 10000;

                %Define termination tolerance
                         termtol = 0.000005;
       %NR outer loop:
         for j = 1:maxiter
                %Define/reset f_Beta and df_dBeta functions
```

```
                f_Beta = 0;
                df_dBeta = 0;
            for i = 1:4
                f_Beta = f_Beta + (qD(i)-Q)*2^(-Beta*(qD(i)-Q));
            end
            %Check for convergence
                if ((f_Beta)^2)^(1/2) <= termtol
                    str = ('Solution found; Beta = ',num2str(Beta), ' at iteration
                    number: ',num2str(j));
                    break
                end
            %Check for maximum iterations
                if j == maxiter
                    str = ('Solution could not be obtained after
                    ',num2str(maxiter),' iterations, please alter max # of iterations,
                    convergence tolerance and / or initial Beta value')
                     break
                 end
            %Calculate the derivative of f_Beta
                for i = 1:4
                    df_dBeta = df_dBeta + (qD(i)-Q)*log(2)*(-1*(qD(i)-
Q))*2^(Beta);
                 end
            %Update Beta
                Beta = Beta - (f_Beta/df_dBeta);
        end
  %Step 2: Determine Alpha: log(SUM(2^(-Beta*qA(i)))i:
        %Define Alphasum (temp value):
          Alphasum = 0;
           for i = 1:4
                Alphasum = Alphasum + 2^(-1*Beta*qD(i));
           end

        %Obtain Alpha:
          Alpha = log2(Alphasum);

   %Step 3: Determine the individual probabilities:
      for i = 1:4
          pD(i) = 2^(-1*Alpha)*2^(-1*Beta*qD(i));
      end

  %Step 4: Calculate the Entropy:
      S = Alpha + Beta*Q;
```

```
%Step 5: Verify results:
    % Calculate values
        Psum = 0;
       Con1sum = 0;
        for i =1:4
                Psum = Psum + pD(i); %Probability sum
                Con1sum = Con1sum + pD(i)*D(i); % Constraint sum
        end
    %Check values
       Psum=Psum-1;
        if (Psum >= Psumresidual)
                str = ('ERROR: The sum of probabilities does not equal 1; the actual value is:
                ',num2str(Psum))
                ERROR=1;
        end
       CCon1 = ((Con1sum-Con1)^2)^(1/2);
       if (CCon1 >= Con1residual)
                str = ('ERROR: The residual tolerance set for constraint 1 of
                ',num2str(Con1residual),' has been violated; the current value is:
                ',num2str(CCon1))
                ERROR=1;
    end
    if (ERROR ==0)
       str=('Program successfully completed, the resulting values are:')
       str=('Alpha = ',num2str(Alpha))
       str=('Beta = ',num2str(Beta))
       str=('Entropy S = ',num2str(S))
    end

    if (ERROR==1)
            str=('ERROR TERMINATION, please see ERROR message(s) above')
     end

    Pd % Write outcome to screen
```

This page is intentionally left blank.

This page is intentionally left blank.

CONTRIBUTION 6

BEETS Optimisation software

Completed December 2014

The whole of contribution 6 has been removed from this thesis.

This page is intentionally left blank.