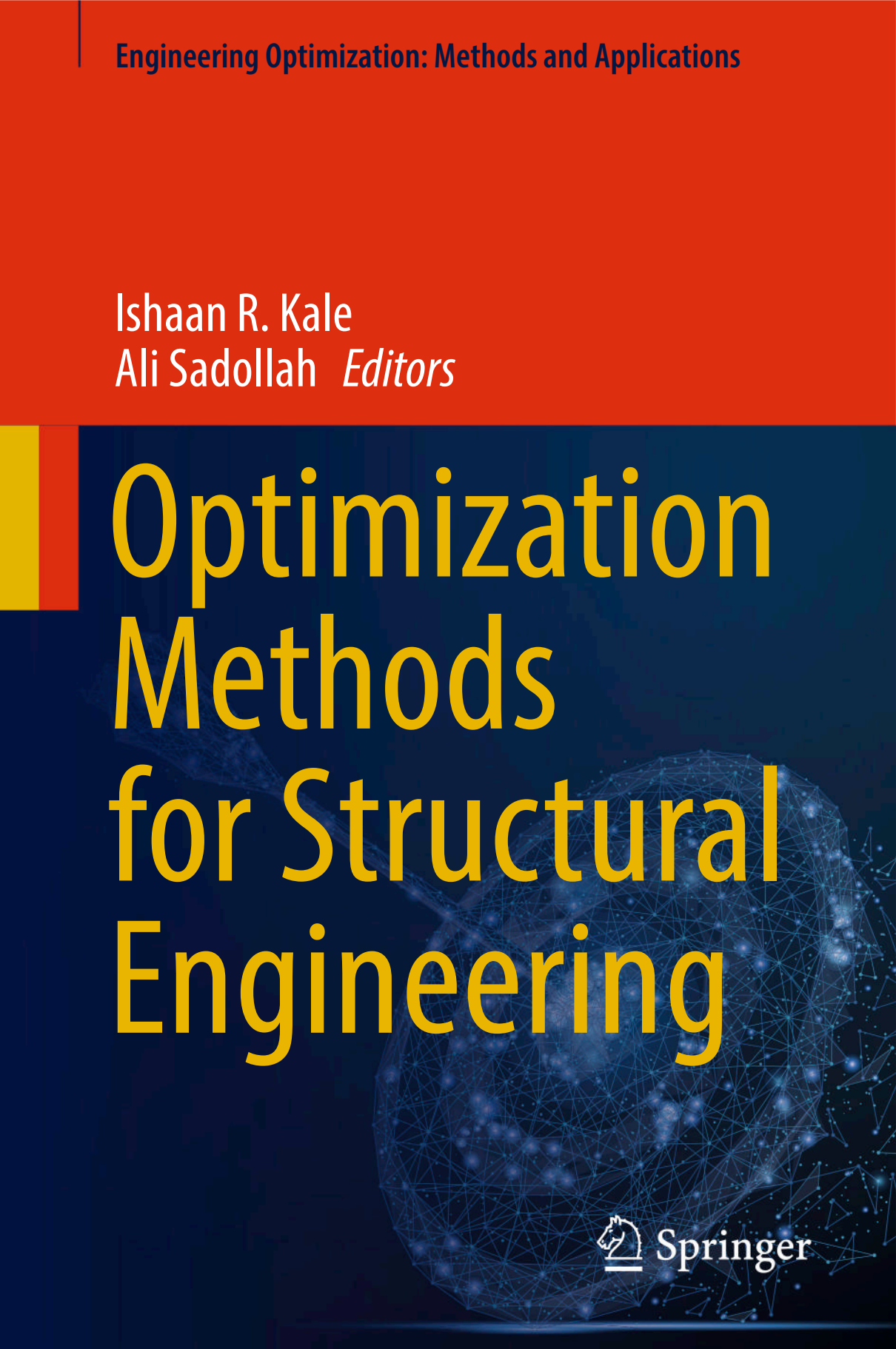
Engineering Optimization: Methods and Applications

Ishaan R. Kale
Ali Sadollah *Editors*

Optimization Methods for Structural Engineering



Springer

Engineering Optimization: Methods and Applications

Optimization carries great significance in both human affairs and the laws of nature. It refers to a positive and intrinsically human concept of minimization or maximization to achieve the best or most favorable outcome from a given situation. Besides, as the resources are becoming scarce there is a need to develop methods and techniques which will make the systems extract maximum from minimum use of these resources, i.e. maximum utilization of available resources with minimum investment or cost of any kind. The resources could be any, such as land, materials, machines, personnel, skills, time, etc. The disciplines such as mechanical, civil, electrical, chemical, computer engineering as well as the interdisciplinary streams such as automobile, structural, biomedical, industrial, environmental engineering, etc. involve in applying scientific approaches and techniques in designing and developing efficient systems to get the optimum and desired output. The multifaceted processes involved are designing, manufacturing, operations, inspection and testing, forecasting, scheduling, costing, networking, reliability enhancement, etc. There are several deterministic and approximation-based optimization methods that have been developed by the researchers, such as branch-and-bound techniques, simplex methods, approximation and Artificial Intelligence-based methods such as evolutionary methods, Swarm-based methods, physics-based methods, socio-inspired methods, etc. The associated examples are Genetic Algorithms, Differential Evolution, Ant Colony Optimization, Particle Swarm Optimization, Artificial Bee Colony, Grey Wolf Optimizer, Political Optimizer, Cohort Intelligence, League Championship Algorithm, etc. These techniques have certain advantages and limitations and their performance significantly varies when dealing with a certain class of problems including continuous, discrete, and combinatorial domains, hard and soft constrained problems, problems with static and dynamic in nature, optimal control, and different types of linear and nonlinear problems, etc. There are several problem-specific heuristic methods are also existing in the literature.

This series aims to provide a platform for a broad discussion on the development of novel optimization methods, modifications over the existing methods including hybridization of the existing methods as well as applying existing optimization methods for solving a variety of problems from engineering streams. This series publishes authored and edited books, monographs, and textbooks. The series will serve as an authoritative source for a broad audience of individuals involved in research and product development and will be of value to researchers and advanced undergraduate and graduate students in engineering optimization methods and associated applications.

Ishaan R. Kale · Ali Sadollah
Editors

Optimization Methods for Structural Engineering

Springer

Editors
Ishaan R. Kale
Institute of Artificial Intelligence
Dr. Vishwanath Karad MIT World Peace University
Pune, Maharashtra, India

Ali Sadollah
Department of Mechanical Engineering
University of Science and Culture
Tehran, Iran

ISSN 2731-4049 ISSN 2731-4057 (electronic)
Engineering Optimization: Methods and Applications
ISBN 978-981-99-2377-9 ISBN 978-981-99-2378-6 (eBook)
https://doi.org/10.1007/978-981-99-2378-6

Preface

This edited book aims to demonstrate and discuss the concepts of optimization in structural engineering. Several structural frameworks, such as cranes, towers, bridges, frames, vehicle chassis, are very much useful for the progression, living of human society, and smooth transportation. It is also equally important for industrial purposes such as safe handling of equipment, logistics, operations, and so on. Resources such as man, machines, materials, and financial aspects are needed to have efficient utilization so that structures can become viable or feasible in many ways. For the resource management, traditional techniques are being used which are based on trial and error, and some are deterministic and approximation methods. On the other hand, vehicle parts, such as car frames, chassis, are complex design and exceptionally challenging due to its nonlinear shape and dynamic loading conditions. This edited volume intends to provide a platform to state-of-the-art discussion on various structural applications, their mathematical modeling, complexity of problems, several metaheuristic optimization algorithms, and other software-based techniques using finite element analysis (FEM). This edited book also aims to discuss the literature survey on several nature-inspired optimization algorithms, its applicability for solving single-objective and multi-objective structural engineering problems, solution quality, challenges, opportunities, key features, and limitations.

All the chapters submitted in this volume are critically reviewed by at least two expert reviewers. The critical suggestion given by the reviewers helped the authors to enrich the quality of the chapter in terms of methodology, critical discussion performance and solution quality, representation, etc. The volume serves as a contribution of optimization for structural engineering domain. This book will be helpful for the researchers, professors, and industry persons working in the field of structural engineering.

The volume is divided into two parts. The detailed review on nature-inspired metaheuristic algorithms, their applicability, challenges, and several applications on structural engineering is discussed in Part I. The topology optimization of structures, car suspension, crashworthiness during the impact of car, design of car chassis under dynamic loading conditions using finite element methods are discussed in Part II. The contribution of every chapter is discussed below in detail.

Part I: Optimization of Weight, Shape, and Size of Structures Using Different Nature-Inspired Metaheuristic Algorithms

In Chap. 1 "Review of Turning Mass Dampers and Application of Improved Harmony Search," Bekdaş et al. discussed the overview of turned mass dampers which is generally installed at the top of the seismic structures to measure the amplitude during disasters. The critical review on passive and active turning mass dampers is discussed with its applications. Further, an improved harmony search metaheuristic algorithm is used to examine the performance of active turned mass damper and passive turned mass damper. The ten-story seismic structure is considered with multiple cases of time delay of the controller and stroke capacity. The performance active turned mass damper is observed to be better as it has reduced the displacement by 32.01% as compared to passive turned mass damper.

In Chap. 2 "Optimal Design of Trusses: The Force Density Perspective," Dzierżanowski and Wójcik-Grząba designed a combined model of truss geometry and topology optimization. The two-dimensional and three-dimensional structures of cable nets and cantilever elements are considered for the optimization. The combined geometry and topology optimization of the trusses in one numerical algorithm results in high nonlinearity. So, the force density method is used to optimize the combined geometry and topology of the truss structure to identify its reliability.

In Chap. 3 "CI-SAPF for Structural Optimization Considering Buckling and Natural Frequency Constraints," Kale and Khedkar investigated the constrained version of socio-inspired cohort intelligence (CI) algorithm solving two cases 18-bar truss structural problems for size optimization. Two types of constraints are associated with these problems such as buckling and natural frequency. These constraints are handled using self-adaptive penalty function (SAPF) approach. The CI-SAPF approach has obtained comparable results as compared to other nature-inspired techniques. In CI algorithm, the roulette wheel approach plays an important role which assists the CI candidate to improve their individual behavior (objective function). This further helps to improve the behavior of the entire cohort.

In Chap. 5 "Truss Structure Optimization Using Constrained Version of Variations of Cohort Intelligence" by Kale et al., the follow-best and follow-better variations of CI are used to optimize the weight of the truss structure. Instead of following random candidate using roulette wheel approach, the best candidate is followed by other candidates in follow-best approach, and better candidate is followed by other candidate in follow-better approach. The discrete variable 6-bar and 10-bar truss structures problems are considered. The constraints such as tensile/compression stress and deflections associated with these problems are handled using SAPF approach. Further CI algorithm is hybrid with fuzzy logic in Chap. 6 "Hybridization of Cohort Intelligence and Fuzzy Logic (CIFL) for Truss Structure Problems" by Patel et al. The two cases of 3-bar truss structure problems are solved using CIFL algorithm.

In Chap. 4 "Improved Drosophila Food-Search Algorithm for Structural and Mechanical Optimization Problems," the author Ali Mortazavi demonstrates the

improved version of metaheuristic Drosophila food-search optimization (DFO) algorithm to solve a spatial 72-bar truss structure problem and two design engineering problems such as welded beam design problem and tubular column system problem. In DFO, the three-phase search pattern is utilized to explore the desired problem's search domain, and to maintain the diversity of the population during the optimization process, the quadratic approximation search approach is used to produce new individuals.

In Chap. 7 "Optimum Design of BRB Frame Based on Drift Uniformity, Structure Weight, and Seismic Parameters Using Nonlinear Time History Analysis," Razavi and Shirjani presented the multi-objective problems of buckling-restrained braces (BRBs) which are a popular seismic resistant structural system. This work aims to minimize the cost of the braces by optimizing three objectives, i.e., bracing weight, total weight of the structure (without bracing weight), and deviation from the uniform drift over the height of the structure. The low-cycle fatigue constraint is considered to ensure that the braces do not rupture during seismic event. To solve this problem, several multi-objective optimization algorithms are used such as NSGA_II, MOPSO, MOEA_D, PESA_II, SPEA_II, and comparison of results is discussed in details.

Part II: Topology Optimization and Design of Structures Under Dynamic Conditions Using Finite Element Methods

Chapter 8 "Topology Optimization in Linear Elasticity, Plasticity and Fracture Mechanics" by Desai discusses the theoretical and numerical study of topology optimization of structures. For the topology optimization, the behaviors of elasticity and plasticity must be complimentary to avoid the damage or fatigue. It discusses the non-differentiable behavior of the governing equation of linear elasticity, plasticity, and damage which needs to be overcome to withstand the structure and its mechanical properties. For that, an approximation by penalization and regularization is constructed using a level-set method which allows the body-fitted remeshing and capture the boundary of shapes during the topology change. The topology optimization of several 2D and 3D Frames is carried out using the level-set method.

In Chap. 9 "Design of Quarter Car Model for Active Suspension System and Control Optimization," Patel et al. describe the behavioral relationship between suspension and the car body. The design and simulation of quarter car system are presented here to stable the working control systems. This is usually carried out to achieve the passenger's comfort design, the road handling design, and a balanced system. The H-infinity method is used to synthesize the control system and design a controller based on the defined states and control inputs of the system. In order to suppress the effect of the disturbances, the μ-Synthesis is used which helps to set

the control systems and balance the design. The performance of control system is measured on 7 cm and 10 cm bumps to investigate the proposed design of the quarter car system.

Dr. Ishaan R. Kale
Research Assistant Professor
Institute of Artificial Intelligence
Dr. Vishwanath Karad MIT World
Peace University
Pune, Maharashtra, India
ishaan.kale@mitwpu.edu.in

Dr. Ali Sadollah
Assistant Professor
Department of Mechanical Engineering
University of Science and Culture
Tehran, Iran
sadollah@usc.ac.ir

Contents

Editors and Contributors

About the Editors

Dr. Ishaan R. Kale received his Ph.D. at the faculty of mechanical engineering, Symbiosis International University, Pune, India, in 2021. He received his Master's in Mechanical Design Engineering from Maharashtra Institute of Technology (affiliated with Pune University, India). His research interest lies in design engineering, computational intelligence, and nature-inspired optimization techniques, viz. probability collectives, particle swarm optimization, genetic algorithm, hybrid metaheuristics, also in game theory, operation research and numerical methods. Dr. Kale has published several research articles and a book on constraint handling in cohort intelligence. Currently, he is working as a Research Assistant Professor at the Institute of Artificial Intelligence, MIT World Peace University, Pune, India. He is also involved as an active research member of the Optimization and Agent Technology Research Lab.

Dr. Ali Sadollah received his Ph.D. from the department of mechanical engineering, University of Malaya, Malaysia, in 2013. He served as a postdoctoral research fellow at universities such as Korea University and Nanyang Technological University. Currently, he is an assistant professor at the University of Science and Culture, Iran. He is among the top 2 % of the world's researchers, according to a recent Stanford University study, and the inventor of two metaheuristic optimization methods named as water cycle algorithm and neural network algorithm. His research interests include algorithm development, optimization and metaheuristics, applications of soft computing methods in engineering, computational solid mechanics, finite element method and cold roll forming process. Dr. Sadollah has published three books, contributed scientifically to over 100 scientific research and was invited to participate in numerous international and national events.

Contributors

S. Ali Razavi Department of Civil Engineering, University of Science and Culture, Tehran, Iran

Gebrail Bekdaş Department of Civil Engineering, Istanbul University-Cerrahpaşa, Istanbul, Turkey

J. Desai CNRS, Université de Paris and Sorbonne Université, Paris, France

Grzegorz Dzierżanowski Faculty of Civil Engineering, Warsaw University of Technology, Warsaw, Poland

Zong Woo Geem College of IT Convergence, Gachon University, Seongnam, South Korea

Vijaykumar S. Jatti Symbiosis Institute of Technology (SIT), Symbiosis International University (SIU), Lavale, Pune, Maharashtra, India

Vinaykumar S. Jatti Symbiosis Institute of Technology (SIT), Symbiosis International University (SIU), Lavale, Pune, Maharashtra, India

Ishaan R. Kale Institute of Artificial Intelligence, Dr Vishwanath Karad MIT World Peace University, Pune, MH, India

Aylin Ece Kayabekir Department of Civil Engineering, Istanbul Gelişim University, Istanbul, Turkey

Ayush Khedkar Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, MH, India

Anand J. Kulkarni Institute of Artificial Intelligence, Dr Vishwanath Karad MIT World Peace University, Pune, India

Ali Mortazavi Department of Civil Engineering, Izmir Democracy University, Izmir, Turkey

Sinan Melih Nigdeli Department of Civil Engineering, Istanbul University-Cerrahpaşa, Istanbul, Turkey

Saif Patel Institute of Artificial Intelligence, Dr Vishwanath Karad MIT World Peace University, Pune, India

Vyomkumar P. Patel D. Y. Patil College of Engineering, Akurdi, Pune, Maharashtra, India

Mandar S. Sapre Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, MH, India

Rouhollah Shirjani Department of Civil Engineering, University of Science and Culture, Tehran, Iran

Serdar Ulusoy Department of Civil Engineering, Turkish-German University, Istanbul, Turkey

Izabela Wójcik-Grząba Faculty of Civil Engineering, Warsaw University of Technology, Warsaw, Poland

Part I
Optimization of Weight, Shape and Size of Structures Using Different Nature-Inspired Metaheuristic Algorithms

Chapter 1
Review of Tuning Mass Dampers and Application of Improved Harmony Search

Gebrail Bekdaş, Sinan Melih Nigdeli, Aylin Ece Kayabekir, Serdar Ulusoy, and Zong Woo Geem

Abstract In this chapter, a review of the optimum design of passive and active tuned mass dampers for structures is presented. Metaheuristics have been often used in the optimum design of tuned mass dampers. As an example, an improved harmony search algorithm is presented for the optimum design of active tuned mass dampers (ATMDs) using proportional integral derivative type controllers. The ATMD was also compared via passive tuned mass damper (TMD) and the optimum design results are presented for a 10-story structure with multiple cases of the time delay of the controller and stroke capacity of ATMD. ATMD is better than TMD in the reduction of displacements up to 32.01%.

Keywords Structural control · Harmony search algorithm · Tuned mass dampers · Active tuned mass damper

G. Bekdaş (✉) · S. M. Nigdeli
Department of Civil Engineering, Istanbul University-Cerrahpaşa, 34320 Istanbul, Turkey
e-mail: bekdas@iuc.edu.tr

S. M. Nigdeli
e-mail: melihnig@iuc.edu.tr

A. E. Kayabekir
Department of Civil Engineering, Istanbul Gelişim University, 34310 Istanbul, Turkey
e-mail: aekayabekir@gelisim.edu.tr

S. Ulusoy
Department of Civil Engineering, Turkish-German University, 34820 Istanbul, Turkey
e-mail: serdar.ulusoy@tau.edu.tr

Z. W. Geem
College of IT Convergence, Gachon University, Seongnam 13120, South Korea
e-mail: zwgeem@yahoo.com

I. R. Kale and A. Sadollah (eds.), *Optimization Methods for Structural Engineering*, Engineering Optimization: Methods and Applications,
https://doi.org/10.1007/978-981-99-2378-6_1

1.1 Introduction

To mitigate the structural responses of tall buildings and wide-span bridges against earthquake or wind-induced vibrations has been the subject of many studies in recent years. However, the energy absorption capacity of these types of structures alone is not sufficient without using additional mechanisms developed with technology. Therefore, many passive or active control devices have been installed in existing or newly designed structures. The main difference between active and passive control is whether the external force supply is present in the structure or not. The important representatives of passive control systems are base isolation, friction or visco-elastic dampers and tuned mass damper (TMD) while active control systems are active tendon control and active TMD (ATMD). Although base isolation, friction or visco-elastic dampers are widely used in practical applications, TMD consisting of mass, spring and viscous damper is one of the preferred control systems due to its simplicity, reliability and energy absorbability compared to the other passive control systems. In addition, ATMD as an active control system attracts the attention of researchers to achieve less oscillation in structures subjected to near-fault ground motions which have impulsive characteristics such as high peak ground velocity. As practical examples for both control systems, the Taipei 101 building in Taiwan; ORC 2000 Symbol tower in Osaka; the Citicorp Center in New York; Berlin Television Tower in Berlin; Shinjuku Park Tower (227 m) in Tokyo; Shanghai World Financial Center in Shanghai can be given. TMD and its active form ATMD should be optimally tuned to do their job well. Therefore, some applications and simplifications are needed to determine the parameters of TMD and ATMD. Since the harmony search algorithm (HS) has successfully obtained the objective function in many different studies, the application examples in structural control systems have gradually increased.

1.2 Review of TMD and ATMD

1.2.1 Tuned Mass Dampers

The first thought of using TMD with a mass bonded to spring is considered to avoid the resonance vibrations in vehicles such as ships and aircraft (Frahm 1911). TMD is recreated with the addition of damping to increase the effectiveness of the system against different frequencies other than its frequency (Ormondroyd and Hartog 1928). As a result of extensive research of Den Hartog, the design parameters of TMD such as basic frequency and damping ratio have emerged for undamped single degree of freedom (SDOF) main systems (Hartog 1947). The equations are used in many studies considering the effect of inherent damping (Bishop and Welbourn 1952; Snowdon 1959; Ioi and Ikeda 1978a). The formulations of frequency and damping ratio of undamped SDOF main systems are expressed by Warburton using the white noise excitation instead of harmonic excitation (Warburton 1982). A curve fitting

method is proposed by Sadek et al. with the help of numerical algorithms to determine the expressions of main SDOF systems including inherent damping (Sadek et al. 1997). Also, the graphical method for these values is presented by Falcon et al. depending on the different types of excitations (Falcon et al. 1967). Different equations are applicable in determining the optimum parameters of multiple degrees of freedom (MDOF) systems with TMD taking into account the critical mode although these equations are derived for SDOF (Warburton and Ayorinde 1980). There are many studies on passive control with TMD in the literature to assess the effects on the structure. For example, a small mass for the design parameters of TMD leads to a decrease in the first modal response (Wirsching and Campbell 1973). The seismic performance of TMD is quite successful in terms of reducing the maximum displacement (Adam and Furtmüller 2010) or both from the displacement and dissipated power perspectives (Greco and Marano 2013). The reduction rate of the structural responses is observed using a different combination of TMD parameters (McNamara 1977). The optimization method of the spring constant of the damper and damping factor is explained to alleviate the vibration considering the three different possible scenarios (Ioi and Ikeda 1978). Also, the damping constant and the duration of earthquake records play an important role in changing the responses of structures with TMD (Kaynia et al. 1981). The influence of damping, which is one of the TMD parameters, on structural reactions is illustrated using numerical examples (Villaverde 1985). The optimally designed absorber under the band-limited white noise excitation is detected by reducing the response variance of the building (Wang and Wang 1988). The reduction of the vibrations of a long span bridge (Lin et al. 2000), wind turbine (Hemmati et al. 2019; Gaur et al. 2020) and pedestrian bridge (Alhassan et al. 2020) are investigated using TMD. The effect of the soil–structure interaction (SSI) is examined to realize a real design of TMD (Pietrosanti et al. 2017). The robustness of optimally tuned mass damper with an inverter (mechanical devices) is evaluated and compared with classical TMD (Kamgar and Khatibinia 2019) or base isolation with supplemental damping (Domenico and Ricciardi 2018). On the other hand, optimally designed multiple tuned mass dampers (MTMD), multiple tuned mass dampers-inverters (MTMDI) and TMD with magnetoreological (MR) have better outcomes than TMD to suppress the structural reactions (Joshi and Jangid 1997; Cao et al. 2020; Aldemir 2003). Moreover, metaheuristic algorithms inspired by nature instead of conventional mathematical methods are used to obtain optimum values of TMD or MTMD parameters under different conditions. Examples of these are particle swarm optimization (PSO) (Leung and Zhang 2009; Khatibinia et al. 2016), differential evolution method (DEM) (Caicedo et al. 2021), genetic algorithm (GA) (Pourzeynali et al. 2013; Mohebbi et al. 2013), bat algorithm (BA) (Bekdaş et al. 2018), flower pollination algorithm (FPA) (Yucel et al. 2019), artificial bee colony algorithm (ABC) (Farshidianfar and Soheili 2013), gravitational algorithm (Khatibinia et al. 2018), simulated annealing (SA) (Yang and Li 2017), Jaya algorithm (JA) (Bekdaş et al. 2019), teaching–learning-based optimization (TLBO) (Nigdeli and Bekdas 2015), whale optimization algorithm (WOA) (Lara-Valencia et al. 2021) and chaotic optimization algorithm (COA) (Kaveh et al. 2020).

1.2.2 Active Tuned Mass Dampers

The active control of structures is firstly by Zuk proposed (Zuk 1968). Comparative studies between ATMD and other active control systems have been conducted in terms of their performance against earthquakes (Abdel-Rohman and Leipholz 1983; Ulusoy et al. 2021). Various feedback strategies such as displacement, velocity and acceleration feedback are used to actively control the structures with ATMD subjected to wind loading (Ankireddi et al. 1996). In the study of Mackriell et al., the best reduction of the first mode responses of structures with ATMD is obtained via acceleration algorithms (Mackriell et al. 1997). The accurate analytical expressions for the design of structures with ATMD under more complex wind load are examined (Yan et al. 1999). A reduced-order modeling technique is presented to avoid the impractical installation of sensors on all floors of tall buildings with ATMD (Qu et al. 2001). The performance comparison between the linear quadratic regulator (LQR) and the fuzzy logic controller (FLC) is carried out in the benchmark model with ATMD under seismic excitations (Samali and Al-Dawod 2003). In another study of the same authors, the performance of the controllers such as FLC and linear quadratic Gaussian (LQG) controller for the 76-story building exposed to cross wind load are examined (Samali et al. 2004). The structures with several ATMD called active multiple tuned mass dampers (AMTMD) are tested under the historical ground motions to achieve further reduction in structural reactions (Han and Li 2006; Li and Xiong 2008). With the combination of genetic algorithm and FLC, it is aimed to design optimum parameters of ATMD and to achieve the best possible reduction in structural reactions (Pourzeynali et al. 2007). Also, a comparative study of Guclu and Yazıcı about the structural control with ATMD has shown that the FLC performs better than proportional-derivative (PD) controller under different loads and earthquakes (Guclu and Yazici 2008). Self-tuning fuzzy logic controllers (STFLC) and fuzzy proportional-integral-derivative (PID) controllers are suggested by the same authors for a nonlinear structural system with ATMD (Guclu and Yazici 2009a, 2009). An optimum design methodology is proposed for the asymmetric structures with ATMD to minimize the translational and the torsional responses in case of an earthquake (Li et al. 2010). Then, an asymmetric structure having soil–structure interaction (SSI) is investigated for the same purpose (Li 2012). A new method including linear quadratic regulator (LQR), discrete wavelet transform (DWT) and particle swarm optimization (PSO) is introduced to calculate the required control force for structures generated by ATMD (Amini et al. 2013). Linear quadratic Gaussian (LQG) and sliding mode controller (SMC) are applied to optimally control the structures with ATMD under fluctuating along-wind load (You et al. 2014) or earthquake loading (Khatibinia et al. 2020), respectively. The efficiency of FLC optimized PSO on the structural responses is checked using an ATMD under near or far fault ground motions (Shariatmadar and Meshkat Razavi 2014). The interval type-2 FLC (IT2FLC) is offered for an ATMD system and has better outcomes than type-1 FLC in suppressing the structural responses (Shariatmadar et al. 2014). A methodological simulation approach with the help of a multi-objective adaptive genetic-fuzzy controller is introduced for

benchmark 76-story building with ATMD to mitigate the vibrations because of the strong winds or high levels of seismic excitations (Soleymani and Khodadadi 2014). A hybrid controller consisting of a combination of LQR-PID controllers is used to actively control the structures with ATMD (Heidari et al. 2018). The optimization of both the parameters of PID controller and physical parameters of ATMDs is determined using a hybrid metaheuristic method (Kayabekir et al. 2021).

1.3 Modified Harmony Search Algorithm

In this chapter, an adaptive version of harmony search (HS) was presented for optimum design of TMD and ATMD on structures. HS was developed by Geem et al. (2001) by the investigation of musical performances. The features such as playing popular notes, playing new notes and playing notes of similar known notes are the options of a musician who tries to adjust tune the aim of admiration of audience.

In this study, the modified HS is used, and it contains adaptive parameter configuration with respect to iteration number. A new design variable which will be evaluated for the next iteration ($X_{,}^{j,t+1}$) is generated as Eq. (1.1) for global optimization, and this type of optimization is used if harmony memory considering rate (HMCR) bigger than a random number between 0 and 1 (r_1).

Global optimization is the formulation of playing new notes. To consider the known notes, local optimization is used. If HMCR is smaller or equal to r_1, local optimization is formulated as Eq. (1.2).

$$X_i^{j,t+1} = X_{i,\min} + \text{rand}(1)(X_{i,\max} - X_{i,\min}) \text{ if HMCR} > r_1 \tag{1.1}$$

$$X_i^{j,t+1} = X_i^{j+ni,t} + \text{rand}(1)\text{FW}(X_{i,\max} - X_{i,\min}) \; if \text{ HMCR} \leq r_1 \tag{1.2}$$

rand(1) defines a random number between 0 and 1. FW is the specific parameter called fret width. The values of i, j and t define number for a problem with n design variables ($i = 1$ to n), harmony or population ($j = 1$ to harmony memory size; HMS) and iteration number ($t = 1$ to maximum number of iteration), respectively. The ranges of design variables are defined by minimum ($X_{i,\min}$) and maximum ($X_{i,\max}$) limits. ni is the neighborhood index, and it is adjusted to select a random harmony from the existing solutions. Also, it is chosen as the best solution with the minimum objective function by checking a probability called best solution considering rate (BSCR). The parameters, FW and HMCR, are modified according to Eqs. (1.3) and (1.4) by using initial values of FW and HMCR defined as FW_{in} and HMCR_{in}, respectively.

$$\text{FW} = \text{FW}_{\text{in}}\left(1 - \frac{t}{\text{mt}}\right) \tag{1.3}$$

$$\mathrm{HMCR} = \mathrm{HMCR}_{\mathrm{in}}\left(1 - \frac{t}{\mathrm{mt}}\right) \tag{1.4}$$

The flowchart of the optimization process is shown in Fig. 1.1. Firstly, the user-defined values (design constants, ranges of design variables, ground motion acceleration data and algorithm parameters) are defined. Then, an initial solution matrix is generated including sets of design variables that are randomly assigned within the desired range. The number of sets is equal to the harmony memory size (HMS). As the iterative optimization process, new sets of design variables are generated via global and local optimization according to HMCR value. The existing solutions are updated if the newly generated one is better in the solution. The iterative optimization continues for the maximum number of iterations. In the numerical examples, the algorithm parameters are taken as listed in Table 1.1.

1.4 The Optimization Problem

For seismic structures, tuned mass dampers are generally installed on the top of the structure since the maximum amplitude of the first mode shape is seen on the top of the structure. In this section, the equation of motion of structure with a TMD on the top of a shear building is also presented. The control algorithm, which is optimized via metaheuristic methods, is also summarized for ATMD which is also given.

The equations of the motion of structure with ATMD are given in matrix form. Compared to the coupled equation of uncontrolled structures, the additional terms for ATMD are the control force and it is generated via the employed PID control algorithm. Also, the interaction of stiffness and damping force resulting via ATMD are added to the system matrices. As known, with the installment of ATMD or TMD on the top, the n degrees of freedom structural system will be transformed into an $n + 1$ degrees of freedom system. The shear building models with passive and active control are given in Fig. 1.2.

Mass (m_i), stiffness (k_i) and damping coefficient (c_i) of ith story are respectively shown in Fig. 1.2. The parameters of TMD and ATMD for mass, stiffness and damping are shown as m_d, k_d and c_d, respectively. These parameters can be also used in the calculation of the period (T_d) and damping ratio (ξ_d) of TMD or ATMD as given in Eqs. (1.5) and (1.6).

Generally, the mass of TMD or ATMD was taken as a design constant, and it is defined according to axial capacity and economic issues. The other TMD or ATMD parameters such as T_d and ξ_d are considered as design variables.

$$\mathrm{T}_d = 2\pi\sqrt{\frac{m_d}{k_d}} \tag{1.5}$$

$$\xi_d = \frac{c_d}{2m_d\sqrt{\frac{k_d}{m_d}}} \tag{1.6}$$

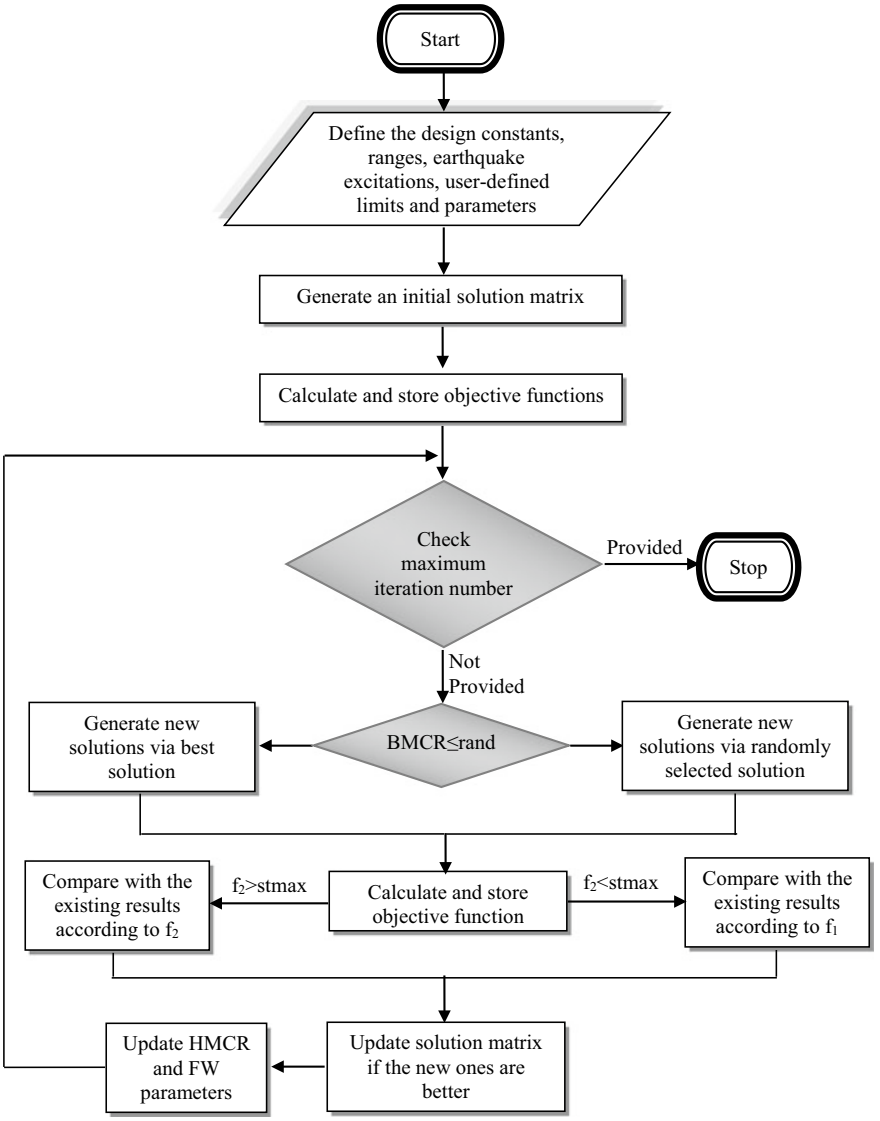


Fig. 1.1 Flowchart of the optimization methodology

The equations of motion in matrix form are shown as Eq. (1.7) having system matrices (mass (M), damping (C) and stiffness (K) matrices), a unit vector ($\{1\}$), the ground acceleration ($\ddot{x}_g$), the control force vector ($F(t)$) including the generated force via ATMD (F_u) and the displacement vector ($x(t)$) including the corresponding derivative of it. All matrices and vectors of Eq. (1.7) are shown as Eqs. (1.8–1.12). The story displacements and ATMD are shown as x_i (for $i = 1$ to n) and x_d, respectively,

Table 1.1 Algorithm parameters

Symbol	Definition	Value
HMS	Harmony memory size	10
mt	Maximum iteration number	5000
$HMCR_{in}$	Initial harmony memory considering rate	0.5
FW_{in}	Initial fret width	0.05
BSCR	Best solution considering rate	0.3
sp	Switch probability	0.5

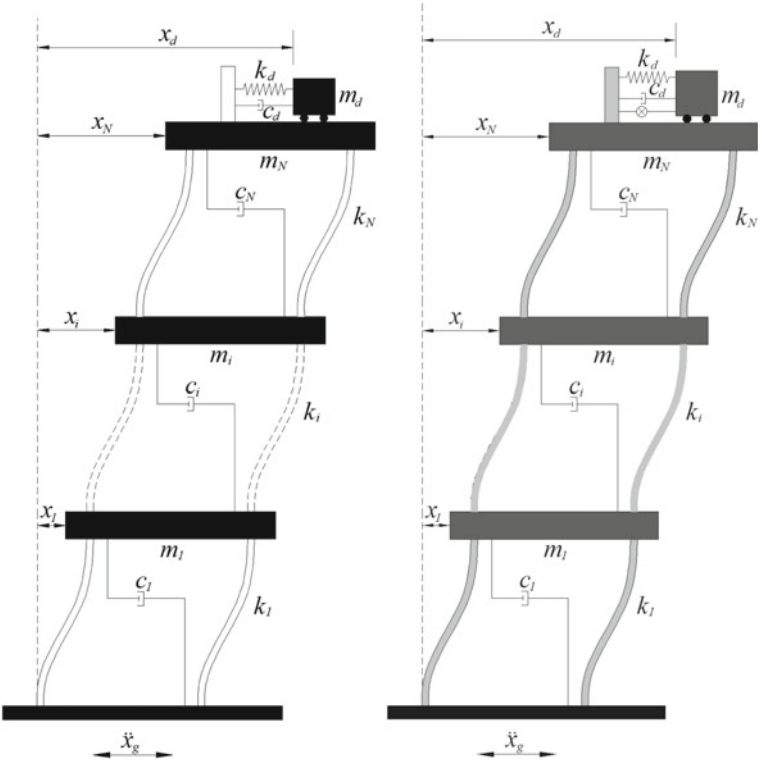


Fig. 1.2 Shear building with TMD (left) and ATMD (right)

and these displacements are respect to the ground.

$$M\ddot{x}(t) + C\dot{x}(t) + Kx(t) = -M\{1\}\ddot{x}_g(t) + F(t) \tag{1.7}$$

$$M = \text{diag}[m_1 m_2 \ldots m_N m_d] \tag{1.8}$$

$$C = \begin{bmatrix} (c_1 + c_2) & -c_2 & & & & & \\ -c_2 & (c_2 + c_3) & -c_3 & & & & \\ & \cdot & \cdot & & & & \\ & \cdot & \cdot & \cdot & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & -c_N & (c_N + c_d) & -c_d & \\ & & & & -c_d & c_d \end{bmatrix} \tag{1.9}$$

$$K = \begin{bmatrix} (k_1 + k_2) & -k_2 & & & & & \\ -k_2 & (k_2 + k_3) & -k_3 & & & & \\ & \cdot & \cdot & & & & \\ & \cdot & \cdot & \cdot & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & -k_N & (k_N + k_d) & -k_d & \\ & & & & -k_d & k_d \end{bmatrix} \tag{1.10}$$

$$x(t) = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \\ x_d \end{Bmatrix} \tag{1.11}$$

$$F(t) = \begin{Bmatrix} 0 \\ 0 \\ \vdots \\ F_u \\ -F_u \end{Bmatrix} \tag{1.12}$$

For ATMD, the three parameters of PID controller such as proportional gain (K_p), derivative time (T_d) and integral time (T_i) are also taken as design variables. As Eq. 1.13, the control force is found via the trust constant (K_f) and current of the armature coil (i_{ATMD}). By using Eq. (1.14), i_{ATMD} is calculated via R taken as the resistance value, K_e taken as the induced voltage constant of armature coil and u taken as the generated control signal. A structural response such as the velocity of the top story ($\dot{x}_N$) and ATMD ($\dot{x}_d$) is also used in Eq. (1.14).

$$F_u = K_f i_{\text{ATMD}} \tag{1.13}$$

$$R i_{\text{ATMD}} + K_e(\dot{x}_d - \dot{x}_N) = u \tag{1.14}$$

The control signal was generated by using Eq. (1.15), and the controller transforms an error signal that is a feedback of the response of the system into a control signal. The top story velocity of the structure was taken as the error signal in this study.

$$u = K_p\left[e(t) + T_d\frac{\mathrm{d}e(t)}{\mathrm{d}t} + \frac{1}{T_i}\int e(t)\mathrm{d}t\right] \tag{1.15}$$

During the iterative stages of optimization, the equations of motion formulations of structure with and without control are analyzed via the code generated by using MATLAB with Simulink (MATLAB 2018), and the maximum of dynamic analysis as top story displacement (x_N) is used as the objective function that is tried to be minimized. The objective (f_1) is formulated as Eq. (1.16).

$$f_1 = \max|x_N| + \text{pen} \tag{1.16}$$

A penalty function (pen) is used in the objective, and it represents the consideration of the maximum control force. As Eq. (1.17), the pen function is calculated and added to the objective if it exceeds the limit.

$$\text{pen} = \max|F_u| \tag{1.17}$$

A secondary objective is used in the methodology to consider the stroke capacity of the control system. This objective function (f_2) is given as Eq. (1.18). This function must be smaller than a user-defined parameter called stmax. Always, the values of f_2 lower than stmax are considered as a better value. If both compared values are lower than stmax for f_2, the comparison is done via the minimization of f_1. In cases of both exceeding values of stmax, f_2 minimization is considered as better in solution.

$$f_2 = \frac{\max(|x_d - x_N|)_{\text{with ATMD}}}{\max(|x_N|)_{\text{without ATMD}}} \tag{1.18}$$

In the optimization, the optimum results are defined according to multiply analyses under different ground motion records. A set of far-faults records given in FEMA P:695: Quantification of Building Seismic Performance Factors (FEMA 2009) was taken.

1.5 Numerical Examples

In this section, the optimum results for the control system that is attached to the last story of a 10-story shear building are presented for different cases of stroke limits (stmax) and time delay (td). Tables 1.2 and 1.3 show the constant problem values in the example for structure and control system, respectively.

Table 1.2 Structure design constants and design variable limits

Parameter	Name	Value	Unit
m_i	Mass of the ith story	360	Ton
k_i	Stiffness of the ith story	650	MN/m
c_i	Damping coefficient of the ith story	6.2	MNs/m

Table 1.3 Controller system design constants and design variable limits

Parameter	Name	Value	Unit
m_d	Mass of TMD or ATMD	180	Ton
T_d	Period of TMD or ATMD	0.5–1.5 times of period of structure	s
ξ_d	Damping ratio of TMD or ATMD	1–50	%
K_p	Proportional gain	(−10,000) to (10,000)	Vs/m
T_d	Derivative time	(−10,000) to (10,000)	s
T_i	Integral time	(−10,000) to (10,000)	s
stmax	Stroke limit	2 and 3	–
td	Time delay	10–50	ms
R	Resistance	4.2	Ω
K_f	Trust constant	2	N/A
K_e	Induced voltage constant	2	V

The problem was optimized under a total of eight different cases. These are the combinations of two stroke limits and four time delays. The optimization results for ATMD are presented in Table 1.4. N_{opt} represents the number of iterations where the optimum result is obtained. $F_{\max}$ defines the maximum control force, and the limit is taken as the amount of 10% of the total weight of the structure. The optimum results for TMD systems are also presented in Table 1.5.

The critical earthquake record is found as the BOL090 component of the Duzce earthquake record within 44 record components. Top story displacement are plotted for the critical earthquake excitation in Fig. 1.3, and it is found that both systems are very effective in reducing the top amplitude and providing a quick steady-state response.

For all earthquake records used in the optimization, the maximum results of the top story displacement are plotted as Fig. 1.4. The optimum ATMD system is also effective under other earthquake records that are not the critical one, and the efficiency of ATMD is maximum under the excitations resulting big displacements at the structure.

The results of these analyzes are presented and discussed in the following section.

Table 1.4 Optimum results for ATMD

			stmax = 2			stmax = 3		
td	10	20	30	50	10	20	30	50
T_d (s)	0.944	0.9731	0.953	0.974	1.021	1.009	0.975	0.957
ξ_d (%)	26.283	27.613	28.711	29.091	11.121	11.461	11.920	11.363
K_p (Ns/m)	803.94	−2214.8	582.190	−1313.5	−491.93	618.385	514.182	−947.5
T_d (s)	−665.74	232.17	−953.67	379.956	1059.3	−821.484	−1055.8	515.50
T_i (s)	618.19	7967.5	8305.46	−3845.9	6124.1	7420.85	9328.27	−4194
f_2	1.999	1.999	1.999	1.999	2.999	2.999	2.999	2.999
N_{opt}	4999	4999	4999	4992	4998	4995	4998	4998
F_{max} (kN)	3530.2	3530	3530.12	3530.15	3530.2	3530.15	3530.11	3530.2
f_1 (m)	0.2498	0.2528	0.2544	0.2612	0.2123	0.2173	0.2200	0.2285

Table 1.5 Optimum results for TMD

stmax	TMD	
	2	3
T_d (s)	0.9418	0.9434
ξ_d (%)	5.64	4.69
f_2	1.9999	2.0755
f_1 (m)	0.2820	0.2803

1.6 Conclusions

According to the optimum period values, the increasing trend is observed with the increasing stmax values in both TMD and ATMD cases except for td = 50. Oppositely, a decreasing behavior of optimum damping ratios due to stmax values is seen. The change observed for TMD system is very low compared to the ATMD system. Also, the damping ratio and period values are lower than ATMD for TMD system.

Two conclusions have been found about the stroke limits of TMD and ATMD systems. The stroke values for optimum results are different for TMD and ATMD systems. Secondly, for all stmax values, the values of second objective are equal to the upper limit in ATMD systems, but it is only at the upper limit for stmax 2 for TMD. This shows the need for big stroke capacity for the ATMD.

According to the displacement values of the ATMD-controlled structure, the values show a big decrease when stmax value increases, but a significant change is not seen for TMD. ATMD has a more significant reduction performance of displacements than TMD, and it is possible to see that the reduction differences between systems are between 7.95% and 12.89% for stmax 2, and 22.69% and 32.01% for stmax 3.

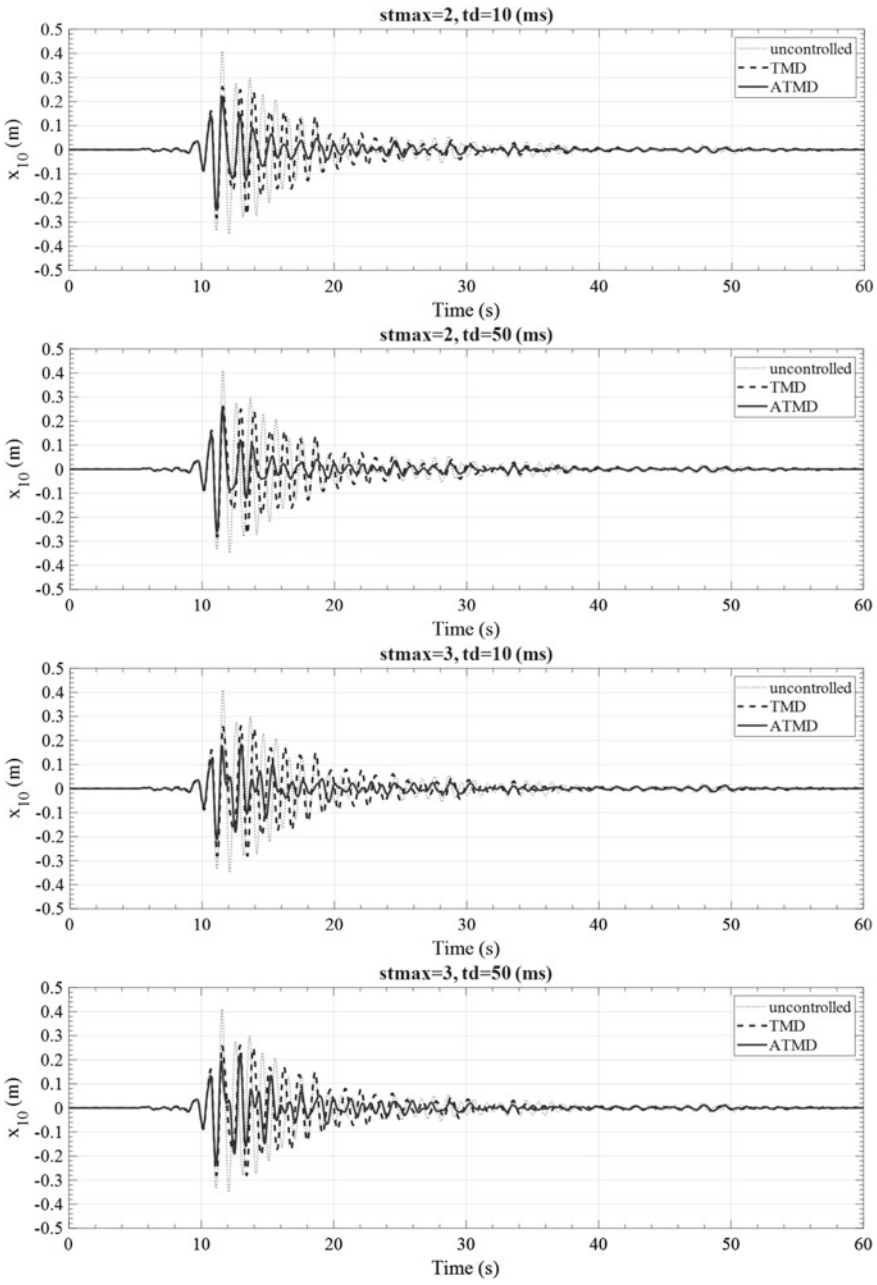


Fig. 1.3 Top story displacements under critical earthquake record

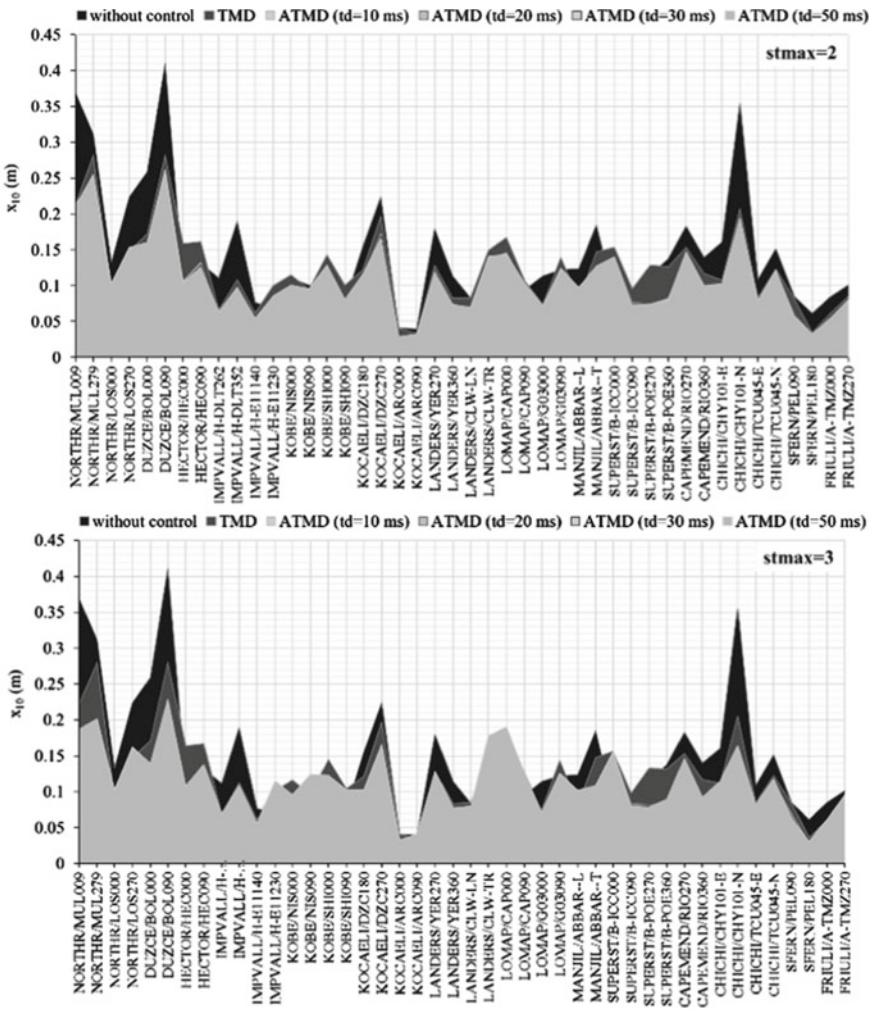


Fig. 1.4 Maximum values of top story displacements for FEMA P-695 far-field ground motions

References

Abdel-Rohman M, Leipholz HH (1983) Active control of tall buildings. J Struct Eng (ASCE) 109:628–645

Adam C, Furtmüller T (2010) Seismic performance of tuned mass dampers. In: Mechanics and model-based control of smart materials and structures. Springer, Vienna, pp 11–18

Aldemir U (2003) Optimal control of structures with semiactive-tuned mass dampers. J Sound Vib 266(4):847–874

Alhassan MA, Al-Rousan RZ, Al-Khasawneh SI (2020) Control of vibrations of common pedestrian bridges in Jordan using tuned mass dampers. Procedia Manuf 44:36–43

Amini F, Hazaveh NK, Rad AA (2013) Wavelet PSO-based LQR algorithm for optimal structural control using active tuned mass dampers. Comput Aided Civ Infra-Struct Eng 28(7):542–557
Ankireddi S, Yang HTY (1996) Simple ATMD control methodology for tall buildings subject to wind loads. J Struct Eng 122(1):83–91
Bekdaş G, Nigdeli SM, Yang XS (2018) A novel bat algorithm based optimum tuning of mass dampers for improving the seismic safety of structures. Eng Struct 159:89–98
Bekdaş G, Kayabekir AE, Nigdeli SM, Toklu YC (2019) Tranfer function amplitude minimization for structures with tuned mass dampers considering soil-structure interaction. Soil Dyn Earthq Eng 116:552–562
Bishop RED, Welbourn DB (1952) The problem of the dynamic vibration ab-sorber. Engineering, Lond, 174:769
Caicedo D, Lara-Valencia L, Blandon J, Graciano C (2021) Seismic response of high-rise buildings through metaheuristic-based optimization using tuned mass dampers and tuned mass dampers inerter. J Build Eng 34:101927
Cao L, Li C, Chen X (2020) Performance of multiple tuned mass dampers-inerters for structures under harmonic ground acceleration. Smart Struct Syst 26(1):49–61
Den Hartog JP (1947) Mechanical vibrations. McGraw-Hill, New York
De Domenico D, Ricciardi G (2018) Earthquake protection of existing structures with limited seismic joint: base isolation with supplemental damping versus rotational inertia. Adv Civ Eng
Falcon KC, Stone BJ, Simcock WD, Andrew C (1967) Optimization of vibration absorbers: a graphical method for use on idealized systems with restricted damping. J Mech Eng Sci 9:374381
Farshidianfar A, Soheili S (2013) ABC optimization of TMD parameters for tall buildings with soil structure interaction, Interact. Multiscale Mech 6:339–356
FEMA (2009) P-695: Quantification of building seismic performance factors. Washington
Frahm H (1911) Device for damping of bodies. U.S. Patent No: 989,958
Gaur S, Elias S, Höbbel T, Matsagar VA, Thiele K (2020) Tuned mass dampers in wind response control of wind turbine with soil-structure interaction. Soil Dyn Earthq Eng 132:106071
Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. simulation 76(2):60–68
Greco R, Marano GC (2013) Optimum design of tuned mass dampers by displacement and energy perspectives. Soil Dyn Earthq Eng 49:243–253
Guclu R, Yazici H (2008) Vibration control of a structure with ATMD against earthquake using fuzzy logic controllers. J Sound Vib 318(1–2):36–49
Guclu R, Yazici H (2009a) Seismic-vibration mitigation of a nonlinear structural system with an ATMD through a fuzzy PID controller. Nonlinear Dyn 58(3):553
Guclu R, Yazici H (2009b) Self-tuning fuzzy logic control of a non-linear structural system with ATMD against earthquake. Nonlinear Dyn 56(3):199
Han B, Li C (2006) Seismic response of controlled structures with active multiple tuned mass dampers. Earthq Eng Eng Vib 5(2):205–213
Heidari AH, Etedali S, Javaheri-Tafti MR (2018) A hybrid LQR-PID control design for seismic control of buildings equipped with ATMD. Front Struct Civ Eng 12(1):44–57
Hemmati A, Oterkus E, Khorasanchi M (2019) Vibration suppression of offshore wind turbine foundations using tuned liquid column dampers and tuned mass dampers. Ocean Eng 172:286–295
Ioi T, Ikeda K (1978a) On the dynamic vibration damped absorber of the vibration system. Bulletin of the JSME 21:6471
Ioi T, Ikeda K (1978b) On the dynamic vibration damped absorber of the vibration system. Bulletin of JSME 21(151):64–71
Joshi AS, Jangid RS (1997) Optimum parameters of multiple tuned mass dampers for base-excited damped systems. J Sound Vib 202(5):657–667
Kamgar R, Khatibinia M (2019) Optimization criteria for design of tuned mass dampers including soil–structure interaction effect. Iran Univ Sci Technol 9(2):213–232

Kaveh A, Javadi SM, Moghanni RM (2020) Optimal structural control of tall buildings using tuned mass dampers via chaotic optimization algorithm. Structures 28:2704–2713. Elsevier
Kayabekir AE, Nigdeli SM, Bekdaş G (2021) A hybrid metaheuristic method for optimization of active tuned mass dampers. Comput Aided Civ Infrastructure Engineering.
Kaynia AM, Veneziano D, Biggs JM (1981) Seismic effectiveness of tuned mass dampers. J Struct Div 107(8):1465–1484
Khatibinia M, Gholami H, Kamgar R (2018) Optimal design of tuned mass dampers subjected to continuous stationary critical excitation. Int J Dyn Control 6(3):1094–1104
Khatibinia M, Mahmoudi M, Eliasi H (2020) Optimal sliding mode control for seismic control of buildings equipped with ATMD. Iran Univ Sci Technol 10(1):1–15
Khatibinia M, Gholami H, Labbafi SF (2016) Multi–objective optimization of tuned mass dampers considering soil–structure interaction
Lara-Valencia LA, Caicedo D, Valencia-Gonzalez Y (2021) A novel whale optimization algorithm for the design of tuned mass dampers under earthquake excitations. Appl Sci 11(13):6172
Leung AYT, Zhang H (2009) Particle swarm optimization of tuned mass dampers. Eng Struct 31(3):715–728
Li C (2012) Effectiveness of active multiple-tuned mass dampers for asymmetric structures considering soil–structure interaction effects. Struct Design Tall Spec Build 21(8):543–565
Li C, Xiong X (2008) Estimation of active multiple tuned mass dampers for asymmetric structures. Struct Eng Mech 29(5):505–530
Li C, Li J, Qu Y (2010) An optimum design methodology of active tuned mass damper for asymmetric structures. Mech Syst Signal Process 24(3):746–765
Lin YY, Cheng CM, Lee CH (2000) A tuned mass damper for suppressing the coupled flexural and torsional buffeting response of long-span4bridges. Eng Struct 22(9):1195–1204
Mackriell LE, Kwok KCS, Samali B (1997) Critical mode control of a wind-loaded tall building using an active tuned mass damper. Eng Struct 19(10):834–842
MATLAB (2018) 9.7. 0.1190202 (R2019b), Natick, Massachusetts: The MathWorks Inc
McNamara RJ (1977) Tuned mass dampers for buildings. J Struct Div 103(9):1785–1798
Mohebbi M, Shakeri K, Ghanbarpour Y, Majzoub H (2013) Designing optimal multiple tuned mass dampers using genetic algorithms (GAs) for mitigating the seismic response of structures. J Vib Control 19(4):605–625
Nigdeli SM, Bekdas G (2015) Teaching-learning-based optimization for estimating tuned mass damper parameters. In: 3rd International conference on optimization techniques in engineering (OTENG'15), 7–9 Nov, Rome, Italy
Ormondroyd J, Den Hartog JP (1928) The theory of dynamic vibration absorber. T ASME 50:922
Pietrosanti D, De Angelis M, Basili M (2017) Optimal design and performance evaluation of systems with Tuned Mass Damper Inerter (TMDI). Earthquake Eng Struct Dynam 46(8):1367–1388
Pourzeynali S, Lavasani HH, Modarayi AH (2007) Active control of high rise building structures using fuzzy logic and genetic algorithms. Eng Struct 29(3):346–357
Pourzeynali S, Salimi S, Kalesar HE (2013) Robust multi-objective optimization design of TMD control device to reduce tall building responses against earthquake excitations using genetic algorithms. Scientia Iranica 20(2):207–221
Qu ZQ, Shi Y, Hua H (2001) A reduced-order modeling technique for tall buildings with active tuned mass damper. Earthquake Eng Struct Dynam 30(3):349–362
Sadek F, Mohraz B, Taylor AW, Chung RM (1997) A method of estimating the parameters of tuned mass dampers for seismic applications. Earthquake Eng Struct Dynam 26:617635
Samali B, Al-Dawod M (2003) Performance of a five-storey benchmark model using an active tuned mass damper and a fuzzy controller. Eng Struct 25(13):1597–1610
Samali B, Al-Dawod M, Kwok KC, Naghdy F (2004) Active control of cross wind response of 76-story tall building using a fuzzy controller. J Eng Mech 130(4):492–498
Shariatmadar H, Meshkat Razavi H (2014) Seismic control response of structures using an ATMD with fuzzy logic controller and PSO method. Struct Eng Mech 51

Shariatmadar H, Golnargesi S, Akbarzadeh Totonchi MR (2014) Vibration control of buildings using ATMD against earthquake excitations through interval type-2 fuzzy logic controller. Asian J Civ Eng Build Housing:15

Snowdon JC (1959) Steady-state behavior of the dynamic absorber. J Acoust Soc Am 31:10961103

Soleymani M, Khodadadi M (2014) Adaptive fuzzy controller for active tuned mass damper of a benchmark tall building subjected to seismic and wind loads. Struct Design Tall Spec Build 23(10):781–800

Ulusoy S, Kayabekir AE, Nigdeli SM, Bekdaş G (2021) Metaheuristic-based structural control methods and comparison of applications. In: Nature-inspired metaheuristic algorithms for engineering optimization applications. Springer, Singapore, pp 251–276

Villaverde R (1985) Reduction seismic response with heavily-damped vibration absorbers. Earthquake Eng Struct Dynam 13(1):33–42

Wang YZ, Wang KS (1988) The optimal design of a dynamic absorber for an arbitrary planar structure. Appl Acoust 23(2):85–98

Warburton GB (1982) Optimum absorber parameters for various combination of response and excitation parameters. Earthquake Eng Struct Dynam 10:381401

Warburton GB, Ayorinde EO (1980) Optimum absorber parameters for simple systems. Earthquake Eng Struct Dynam 8:197217

Wirsching PH, Campbell GW (1973) Minimal structural response under random excitation using the vibration absorber. Earthquake Eng Struct Dynam 2(4):303–312

Yan N, Wang CM, Balendra T (1999) Optimal damper characteristics of ATMD for buildings under wind loads. J Struct Eng 125(12):1376–1383

Yang Y, Li C (2017) Performance of tuned tandem mass dampers for structures under the ground acceleration. Struct Control Health Monit 24(10):e1974

You KP, You JY, Kim YM (2014) LQG control of along-wind response of a tall build-ing with an ATMD. Mathe Probl Eng

Yucel M, Bekdaş G, Nigdeli SM, Sevgen S (2019) Estimation of optimum tuned mass damper parameters via machine learning. J Build Eng 26:100847

Zuk W (1968) Kinetic structures. Civil Eng 39(12):62–64

Chapter 2
Optimal Design of Trusses: The Force Density Perspective

Grzegorz Dzierżanowski and Izabela Wójcik-Grząba

Abstract This paper concerns simultaneous geometry and topology optimization of truss structures. To be specific, coordinates of truss nodes and axial stress resultants are admitted as design variables in the optimization process. Within this scope, two- and three-dimensional systems are considered; the latter also include cable nets in tension. In the analysis of such structures, curvilinear elements are represented as polygonal chains of straight members. Such approach is legitimate in light of computational line of discussion in the paper, and it also allows for a consistent description of statics in terms of the theory of trusses. Combining the optimal design of truss geometry and topology in one numerical algorithm results in a highly nonlinear problem involving algebraic functions. However, employing the idea of *force densities*—i.e., ratios of axial stress resultants to member lengths—makes it possible to recast the problem into the computationally less demanding form involving polynomials. The computational part of the study was performed in Scilab and MATLAB. The results obtained show that the proposed approach to optimal design of trusses and truss-like structures is scientifically reliable and may be used by civil engineers and architects.

Keywords Force density method · Minimum volume problem · Trusses

2.1 Introduction

2.1.1 Motivation of the Study

This study regards a non-classical approach to the optimal design of trusses and truss-like structures. The proposed optimization algorithm can be categorized as a

G. Dzierżanowski (✉) · I. Wójcik-Grząba
Faculty of Civil Engineering, Warsaw University of Technology, Warsaw, Poland
e-mail: grzegorz.dzierzanowski@pw.edu.pl

I. Wójcik-Grząba
e-mail: izabela.grzaba@pw.edu.pl

I. R. Kale and A. Sadollah (eds.), *Optimization Methods for Structural Engineering*,
Engineering Optimization: Methods and Applications,
https://doi.org/10.1007/978-981-99-2378-6_2

simultaneous geometry and topology optimization method, because it incorporates the coordinates of truss nodes and axial stress resultants as design variables. In this regard, it differs from typically used *topology optimization methods*, which keep the geometry of nodes fixed and deal with truss member forces only. Mathematically, the novel approach leads to nonlinear and non-convex optimization problems, which are generally considered to be more CPU time demanding than the traditional ones. This shortcoming, however, is compensated for (to some extent) by the relatively small number of design variables required for obtaining the results, which are both scientifically reliable and useful for civil engineers and architects in pursuing structural stiffness and efficiency of the use of materials.

Limiting the scope of computations to structures with a moderate number of nodes and members has several advantages over large-scale optimization algorithms. First, it leads to solutions with a clear hint toward rational, manufacturable projects. Second, the proposed approach is very robust in tackling the problem of optimal positioning of load, i.e., a design challenge of finding such an architectural form of the structure, which optimally adjusts to the profile of the applied load. The optimization procedure allows for the migration of external forces in space and the shape of the structure closely follows their location. This process is known in the literature as *form-finding*, see (Adriaenssens et al. 2014) and references therein. However, reducing the number of truss members makes the proposed approach not suitable for the numerical study of *Michell structures*, and therefore, they lie outside of interest in this paper. For a comprehensive treatment of the topic, the reader is directed to Lewiński et al. (2019) and papers cited.

A special emphasis of the research is placed on the form-finding of cable nets and archgrids. Mechanical understanding of the term *cable net* pertains to a spatial system entirely composed of elements in tension (ties). The term *archgrid* is consistent with the one in Rozvany and Prager (1979)—it is a structure whose elements (struts) carry compressive stresses only. In this regard, cable nets and archgrids are complementary to one another. They are bending- and shear-free, with stresses distributed evenly in cross sections of all members, and the only difference is in the sign of stresses. Consequently, both bar assemblies can be analyzed by the same computational algorithm.

Another point that needs to be explained is that the minimum volume problem can be fully based on the static analysis of trusses, without accounting for their kinematic response. Consequently, it is not necessary to provide the full data on mechanical properties and stress–strain relation for a material. In spite of this, the ability to control the accuracy of computations is not affected. It is routinely provided by the duality principle in mathematical optimization. This topic, however, lies outside of the scope of the paper.

2.1.2 Archgrids, Cable Nets, and Polygonal Chains

In the computational study of optimal archgrids and cable nets, it is convenient to implement the theory of trusses, that is pin-jointed, bending-, and shear-free bar systems, with external loads equilibrated by axial stress resultants only. Curved axes of single arches and cables are approximated by polygonal chains—line segments of any such chain are equivalent to truss members, while the vertices represent truss nodes (member joints). Mathematically speaking, the equilibrium equation for trusses plays a twofold role: (i) it serves as a condition for spatial assembly of polygonal chains to carry a load of a given intensity, and (ii) it allows to determine the distribution of chain vertices.

To visualize the concept, consider a truss with m members, j pin joints (nodes), and r reactions at supports, with $r = r_x + r_y + r_z$, since the number of reactions may be different in x-, y-, and z-directions in the chosen $Oxyz$ reference system. Supporting a node in some direction is equivalent to fixing its spatial coordinate in this direction. The terms *fixed node coordinate* and *free node coordinate* are used in the sequel to describe the features of nodes in the supported and unsupported directions, respectively. In what follows, vectors $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ of node coordinates are such that $\dim \boldsymbol{x} = \dim \boldsymbol{y} = \dim \boldsymbol{z} = j$. Vector $\boldsymbol{x}$ has r_x components fixed, while the remaining $k_x = j - r_x$ are free, thus subject to optimization. Similar arguments apply to the components of vectors $\boldsymbol{y}$ and $\boldsymbol{z}$.

Equilibrium equation for a truss reads

$$\mathbf{B}\boldsymbol{n} = \boldsymbol{f}, \mathbf{B} = \begin{bmatrix} \mathbf{B}_x \\ \mathbf{B}_y \\ \mathbf{B}_z \end{bmatrix}, \boldsymbol{f} = \begin{bmatrix} \boldsymbol{f}_x \\ \boldsymbol{f}_y \\ \boldsymbol{f}_z \end{bmatrix}, \tag{2.1}$$

where

- $\boldsymbol{f}_x, \boldsymbol{f}_y, \boldsymbol{f}_z$, with $\dim(\boldsymbol{f}_x) = \dim(\boldsymbol{f}_y) = \dim(\boldsymbol{f}_z) = j$, are the vectors of forces applied at nodes. In light of the above, r_x components of $\boldsymbol{f}_x$ are the effective forces (loads minus reactions) x-aligned with fixed node coordinates, and k_x components correspond to external loads applied along free node coordinates. Components of load vectors $\boldsymbol{f}_y$ and $\boldsymbol{f}_z$ are identified similarly. It is important to notice that the equilibrium in all supported directions is guaranteed upfront. In the flow of calculations, it can be simply assumed that reactions take the values necessary to equilibrate axial stress resultants in truss members. However, at the post-processing stage of static calculations, equilibrium equations in supported directions determine the values of reaction components.
- $\boldsymbol{n}$, with $\dim(\boldsymbol{n}) = m$, is the vector of axial stress resultants.
- $\mathbf{B}_x, \mathbf{B}_y, \mathbf{B}_z$, with $\dim(\mathbf{B}_x) = \dim(\mathbf{B}_y) = \dim(\mathbf{B}_z) = j \times m$, denote the matrices of direction cosines of member axes in the $Oxyz$ system. Note that $\mathbf{B} = \mathbf{B}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$.

2.1.3 Basic and Optimal Form-Finding Problems. Polygonal Chains of Minimum Weight

Basic form-finding problems are straightforward to solve. They simply use (2.1) to determine $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ for fixed $\boldsymbol{f}$ and $\boldsymbol{n}$. In the case of optimal form-finding, only the load vector $\boldsymbol{f}$ is fixed and vector $\boldsymbol{n}$ is subject to optimization. Usually, to compute $\boldsymbol{n}$, some functional $\mathcal{F}(\boldsymbol{n})$ is introduced and then minimized by solving

$$\mathcal{F}_{\min} \equiv \mathcal{F}(\boldsymbol{n}_{\min}) = \min\{\mathcal{F}(\boldsymbol{n}) \mid \mathbf{B}\,\boldsymbol{n} = \boldsymbol{f}\}. \tag{2.2}$$

Note that definitions introduced in Sect. 2.1.2 additionally assume $\boldsymbol{n} \leq \mathbf{0}$ ($N_M \leq 0$, $M = 1, \ldots, m$) in optimal form-finding problems for archgrids and $\boldsymbol{n} \geq \mathbf{0}$ ($N_M \geq 0$) in case of cable nets.

Any form-finding problem, basic or optimal, needs to be well-posed in terms of linear algebra. In particular, vectors $\boldsymbol{f}, \boldsymbol{n}$ and matrix $\mathbf{B} = \mathbf{B}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ have to obey, see (Strang 2009),

$$\begin{cases} \boldsymbol{f} \in \operatorname{colsp}(\mathbf{B}) \text{ and } \boldsymbol{n} \notin \operatorname{nullsp}(\mathbf{B}) & \text{if } \boldsymbol{f} \neq \mathbf{0}, \\ \boldsymbol{n} \in \operatorname{nullsp}(\mathbf{B}) & \text{if } \boldsymbol{f} = \mathbf{0}. \end{cases} \tag{2.3}$$

Here, colsp$(\cdot)$ and nullsp$(\cdot)$ denote the column space and null space of a matrix, respectively. Another aspect of well-posedness of the form-finding problem is the existence of solutions under the constraints of $\boldsymbol{n} \leq \mathbf{0}$ (archgrids) and $\boldsymbol{n} \geq \mathbf{0}$ (cable nets). This question is thoroughly addressed in Bouchitté et al. (2019).

In 1872, James Clerk Maxwell, (Maxwell 1872), considered a special class of trusses that are either statically determinate or carry a self-equilibrate load. In particular, Maxwell proved the theorem stating that regardless of the spatial layout of members in such systems the difference between the *tension load path* (the product of tensile force, $T > 0$, and tie length, $L_T > 0$, summed over all ties) and the *compression load path* (the product of compressive force, $C > 0$, and strut length, $L_C > 0$, summed over all struts) is constant. It can be written

$$\sum_{\text{ties}} T L_T - \sum_{\text{struts}} C L_C = \mu, \tag{2.4}$$

where μ is insensitive to the layout of truss members and depends only on the intensity of applied forces (including reactions) and their location in space.

In fact, both sums in (2.4) apply to all truss members. Indeed, it is legitimate to write $T > 0$ and $C = 0$ for ties, similarly $T = 0$ and $C > 0$ for struts, while "zero" members admit $T = C = 0$. For the convenience of further exposition, fix $\boldsymbol{l} = [L_M]$, $L_M \geq 0$, $M = 1, \ldots, m$, and

$$\boldsymbol{n} = \boldsymbol{t} - \boldsymbol{c}, \quad \boldsymbol{t} = [T_M], \quad \boldsymbol{c} = [C_M], \quad T_M \geq 0, \quad C_M \geq 0. \tag{2.5}$$

Hence, Maxwell theorem (2.4) now reads

$$\boldsymbol{l}^T(\boldsymbol{t}-\boldsymbol{c})=\mu, \text{ where } \boldsymbol{l}=\boldsymbol{l}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}), \tag{2.6}$$

and the requirements in (2.3) apply accordingly. The notation $\boldsymbol{a}^T$ stands for the transposition of vector $\boldsymbol{a}$.

Almost 30 years later, in the seminal paper (Michell 1904), posed the question of finding the truss of minimum volume of material. His line of reasoning was the following. Assuming $\sigma_T > 0$ and $\sigma_C > 0$ for the limit tensile and compressive stresses in truss members, respectively, it is possible to express the minimum cross-sectional areas of ties and struts as $A_T = T/\sigma_T$ and $A_C = C/\sigma_C$. By this, the least volume of material, for which stresses in truss members stay within the $(-\sigma_C, \sigma_T)$ range, is

$$\begin{aligned}\mathcal{V}_{\text{truss}}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{t},\boldsymbol{c}) &= \sum_{\text{ties}} A_T L_T + \sum_{\text{struts}} A_C L_C \\ &= \boldsymbol{l}^T(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})\left(\frac{1}{\sigma_T}\boldsymbol{t}+\frac{1}{\sigma_C}\boldsymbol{c}\right),\end{aligned} \tag{2.7}$$

and the minimum volume problem for a truss reads,

$$(\mathcal{P}_{\text{truss}}) \;:\; \mathcal{V}^{\min}_{\text{truss}} = \min\left\{ \mathcal{V}_{\text{truss}}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{t},\boldsymbol{c}) \text{as in} (2.7) \;\middle|\; \begin{array}{l} \mathbf{B}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})(\boldsymbol{t}-\boldsymbol{c})=\boldsymbol{f}, \\ \boldsymbol{t}\circ\boldsymbol{c}=\mathbf{0}, \\ (2.3)_1 \;\text{ applies.} \end{array} \right\}$$

Here, $\boldsymbol{a}\circ\boldsymbol{b}$ denotes the Hadamard (component-wise) product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.

If additionally, the truss is statically determinate or subject to a self-equilibrated set of forces, then any candidate structure allowed in $(\mathcal{P}_{\text{truss}})$ admits the same value of Maxwell's constant μ, but different layout of members (hence, different tension and compression load paths). Introducing the affine transformation,

$$\mathcal{F}_{\text{truss}} = \frac{2\sigma_T\sigma_C}{\sigma_T+\sigma_C}\mathcal{V}_{\text{truss}} + \frac{\sigma_T-\sigma_C}{\sigma_T+\sigma_C}\mu = \boldsymbol{l}^T(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})(\boldsymbol{t}+\boldsymbol{c}), \tag{2.8}$$

immediately proves that the second component of the sum in the middle of (2.8) is constant. This means that $\mathcal{F}_{\text{truss}}$ has the same minimizing argument $(\boldsymbol{x}_{\min},\boldsymbol{y}_{\min},\boldsymbol{z}_{\min},\boldsymbol{t}_{\min},\boldsymbol{c}_{\min})$ as $\mathcal{V}_{\text{truss}}$. In other words, among all trusses with the same μ, truss of minimum volume is the one for which the sum of tension and compression load paths is minimal, see (Baker et al. 2015) for illustrative examples regarding this topic.

In light of the discussion in Sect. 2.1.2, the results above apply also to polygonal chains, with archgrids and cable nets fitting naturally in the framework by considering compression- or tension-only constraints, respectively. For definiteness, consider a cable net, thus setting $\boldsymbol{n} \geq \mathbf{0}$. With this, $\boldsymbol{c} = \mathbf{0}$ in the entire cable net, $\boldsymbol{n} = \boldsymbol{t}$, hence

(2.6) and (2.7) are replaced by

$$\mu = \boldsymbol{l}^T(\boldsymbol{x}, \boldsymbol{y}, z)\boldsymbol{t}, \quad \mathcal{V}_{\text{cable}} = \frac{\mu}{\sigma_T}, \tag{2.9}$$

and ($\mathcal{P}_{\text{truss}}$) is modified to

$$(\mathcal{P}_{\text{cable}}) \ : \ \mathcal{V}_{\text{cable}}^{\min} = \min \left\{ \mathcal{V}_{\text{cable}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{t}) \text{ as in} (2.9) \middle| \begin{array}{l} \mathbf{B}(\boldsymbol{x}, \boldsymbol{y}, z)\boldsymbol{t} = \boldsymbol{f}, \\ \boldsymbol{t} \geq \mathbf{0}, \\ (2.3)_1 \text{ applies.} \end{array} \right\}$$

It is important to notice here that minimizing the volume of a cable net is equivalent to minimizing the value of Maxwell's constant μ or optimizing the location of nodes at which the external load is applied. This observation follows immediately from (2.9) and ($\mathcal{P}_{\text{cable}}$). The above is valid, *mutatis mutandis*, also in the case of archgrids, that is structures with $\boldsymbol{n} \leq \mathbf{0}$ ($\boldsymbol{t} = \mathbf{0}$ in the entire system).

Cable nets are often pre-tensioned to achieve structural stiffness, where pre-tensioning amounts to stretching a slack net by pulling the cables at the supports. Tensile forces in net members are self-equilibrated, and no other external load is applied. Hence, $\boldsymbol{f} = 0$ in ($\mathcal{P}_{\text{cable}}$), which now takes the following form,

$$(\mathcal{P}_{\text{pre}}) \ : \ \mathcal{V}_{\text{pre}}^{\min} = \min \left\{ \mathcal{V}_{\text{cable}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{t}) \text{as in (2.9)} \middle| \begin{array}{l} \mathbf{B}(\boldsymbol{x}, \boldsymbol{y}, z)\boldsymbol{t} = \mathbf{0}, \\ \boldsymbol{t} \geq \mathbf{0}, \\ (2.3)_2 \text{ applies.} \end{array} \right\}$$

2.2 Force Density Method and Its Applications

2.2.1 The Concept of Force Density

In the original work, Maxwell expressed (2.4) in terms of the ratios of forces in ties and struts to member lengths. Setting $\gamma = T/L_T$ and $\pi = C/L_C$, we get

$$\sum_{\text{ties}} \gamma L_T^2 - \sum_{\text{struts}} \pi L_C^2 = \mu. \tag{2.10}$$

The same quantities were re-invented in 1974 by Schek, (Schek 1974), who coined the term *force density*. In this section, the force density notation is used in reformulating the optimal form-finding problems.

For this, first define m-vectors of member length projections at the axes of fixed Cartesian system $Oxyz$,

$$\begin{aligned} \boldsymbol{l}_x &= \mathbf{H}_x \boldsymbol{x}, \\ \boldsymbol{l}_y &= \mathbf{H}_y \boldsymbol{y}, \\ \boldsymbol{l}_z &= \mathbf{H}_z \boldsymbol{z}, \end{aligned} \tag{2.11}$$

where $\mathbf{H}_x, \mathbf{H}_y, \mathbf{H}_z$ are *incidence matrices* with $\dim(\mathbf{H}_x) = \dim\big(\mathbf{H}_y\big) = \dim(\mathbf{H}_z) = m \times j$. Now, the truss is understood to be a *directed graph* with m edges (truss members) and j vertices (truss nodes), and the term *directed* pertains to axes of truss members. They span two endpoints (nodes of a truss), and, for definiteness, it is assumed that each axis originates at the endpoint with the lower number. If A and B, such that $1 \leq \ldots \leq A < \ldots < B \leq \ldots \leq j$, are the numbers of endpoints of M-th truss member, then, by the axis direction convention, $l_x^M = x_B - x_A$, $l_y^M = y_B - y_A$, and $l_z^M = z_B - z_A$. Thus, the components of incidence matrices are either -1, 0, or 1, and the nonzero values correspond to the coordinates of endpoints, see (Dzierżanowski and Wójcik-Grząba 2020) for more detailed explanations of this issue. Truss member lengths are given by the m-vector

$$\boldsymbol{l} = \left((\boldsymbol{l}_x)^{\circ 2} + (\boldsymbol{l}_y)^{\circ 2} + (\boldsymbol{l}_z)^{\circ 2}\right)^{\circ \frac{1}{2}}, \tag{2.12}$$

where $\boldsymbol{a}^{\circ p}$ is the p-th Hadamard (component-wise) power of vector $\boldsymbol{a}$.

Throughout the paper, lower-case bold characters (Latin or Greek) are used to denote vectors and upper-case bold characters symbolize matrices. It is also convenient to adopt the "vector-diagonal matrix" correspondence: if $\boldsymbol{w} = [w_I]$, $\dim(\boldsymbol{w}) = i$, denotes a vector, then $\mathbf{W} = \mathrm{diag}[w_I]$, $\dim(\mathbf{W}) = i \times i$, is the corresponding matrix. This notation, combined with (2.11), allows to split $(2.1)_1$ into separate equilibrium equations in the x-, y-, and z-directions according to the equations,

$$\begin{aligned} \left(\mathbf{H}_x^T \mathbf{L}_x \mathbf{L}^{-1}\right)(\boldsymbol{t} - \boldsymbol{c}) &= \boldsymbol{f}_x, \\ \left(\mathbf{H}_y^T \mathbf{L}_y \mathbf{L}^{-1}\right)(\boldsymbol{t} - \boldsymbol{c}) &= \boldsymbol{f}_y, \\ \left(\mathbf{H}_z^T \mathbf{L}_z \mathbf{L}^{-1}\right)(\boldsymbol{t} - \boldsymbol{c}) &= \boldsymbol{f}_z. \end{aligned} \tag{2.13}$$

Defining the *force density vectors*,

$$\begin{aligned} \boldsymbol{\gamma} &= \mathbf{L}^{-1} \boldsymbol{t} \geq \mathbf{0}, \\ \boldsymbol{\pi} &= \mathbf{L}^{-1} \boldsymbol{c} \geq \mathbf{0}, \end{aligned} \tag{2.14}$$

gives

$$\begin{aligned} \mathbf{H}_x^T \mathbf{L}_x (\boldsymbol{\gamma} - \boldsymbol{\pi}) &= \boldsymbol{f}_x, \\ \mathbf{H}_y^T \mathbf{L}_y (\boldsymbol{\gamma} - \boldsymbol{\pi}) &= \boldsymbol{f}_y, \\ \mathbf{H}_z^T \mathbf{L}_z (\boldsymbol{\gamma} - \boldsymbol{\pi}) &= \boldsymbol{f}_z. \end{aligned} \tag{2.15}$$

Next, introducing (2.11) in (2.15) leads to

$$\begin{aligned} &\mathbf{D}_x \boldsymbol{x} = \boldsymbol{f}_x, && \mathbf{D}_x = \mathbf{H}_x^T(\boldsymbol{\Gamma} - \boldsymbol{\Pi})\mathbf{H}_x, \\ &\mathbf{D}_z \boldsymbol{z} = \boldsymbol{f}_z, \ \text{where} && \mathbf{D}_y = \mathbf{H}_y^T(\boldsymbol{\Gamma} - \boldsymbol{\Pi})\mathbf{H}_y, \\ &\mathbf{D}_y \boldsymbol{y} = \boldsymbol{f}_y, && \mathbf{D}_z = \mathbf{H}_z^T(\boldsymbol{\Gamma} - \boldsymbol{\Pi})\mathbf{H}_z. \end{aligned} \tag{2.16}$$

It follows that (2.16) determine free components of vectors $\boldsymbol{x}$, $\boldsymbol{y}$, z for given $\boldsymbol{\gamma}$, $\boldsymbol{\pi}$, $\boldsymbol{f}_x$, $\boldsymbol{f}_y$ and $\boldsymbol{f}_z$. Hence, they provide the solution to the basic form-finding problem for a truss. Additionally, with $\boldsymbol{f}_x = \boldsymbol{f}_y = \boldsymbol{f}_z = \mathbf{0}$, (2.16) are suitable for the form-finding of pre-stressed truss-like structures composed of ties and struts. In the literature, any such structure is referred to as *tensegrity*.

2.2.2 Optimal Form-Finding in Terms of Force Densities

The force density notation is useful in recasting $\mathcal{V}_{\text{truss}}$, ($\mathcal{P}_{\text{truss}}$), as well as the subsequent optimization problems, in the forms that are easier to deal with computationally. To this end, by abuse of notation, re-write (2.7),

$$\begin{aligned} \mathcal{V}_{\text{truss}}(\boldsymbol{x}, \boldsymbol{y}, z, \boldsymbol{\gamma}, \boldsymbol{\pi}) &= \sum_{\text{ties}} \gamma L_T^2 + \sum_{\text{struts}} \pi L_C^2 \\ &= \boldsymbol{l}^T(\boldsymbol{x}, \boldsymbol{y}, z)\left(\frac{1}{\sigma_T}\boldsymbol{\Gamma} + \frac{1}{\sigma_C}\boldsymbol{\Pi}\right)\boldsymbol{l}(\boldsymbol{x}, \boldsymbol{y}, z), \end{aligned} \tag{2.17}$$

and set

$$(\mathcal{P}_{\text{truss}}) \ : \ \mathcal{V}_{\text{truss}}^{\min} = \min\left\{ \mathcal{V}_{\text{truss}}(\boldsymbol{x}, \boldsymbol{y}, z, \boldsymbol{\gamma}, \boldsymbol{\pi}) \text{as in (2.17)} \,\middle|\, \begin{array}{l} \mathbf{D}_x(\boldsymbol{\gamma}, \boldsymbol{\pi})\boldsymbol{x} = \boldsymbol{f}_x, \\ \mathbf{D}_y(\boldsymbol{\gamma}, \boldsymbol{\pi})\boldsymbol{y} = \boldsymbol{f}_y, \\ \mathbf{D}_z(\boldsymbol{\gamma}, \boldsymbol{\pi})z = \boldsymbol{f}_z, \\ \boldsymbol{\gamma} \circ \boldsymbol{\pi} = \mathbf{0}, \\ (2.3)_1 \text{ applies.} \end{array} \right\}$$

In [1974, Ch. 4], Schek put forward minimization problems of form-finding, alternative to ($\mathcal{P}_{\text{truss}}$), noticing their simple description in the force density framework. As shown below, Schek's problems of: (i) *minimum sum of squared lengths*, and (ii) *minimum sum of lengths* follow as a straightforward corollary from the works of Maxwell and Michell.

To show this, introduce the quadratic form,

$$\mathcal{Q}(\boldsymbol{l}, \boldsymbol{q}) = \boldsymbol{l}^T \mathbf{Q} \boldsymbol{l}, \tag{2.18}$$

where $\boldsymbol{l} = \boldsymbol{l}(\boldsymbol{x}, \boldsymbol{y}, z)$ and $\boldsymbol{v}$ is such that $\mathcal{Q}(\boldsymbol{l}, \boldsymbol{q}) > 0$, i.e., the form is strictly positive. Next, minimize $\mathcal{Q}$ with respect to $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$. More specifically, note that

$$\frac{\partial \boldsymbol{l}_x}{\partial \boldsymbol{x}} = \mathbf{H}_x, \frac{\partial \boldsymbol{l}_y}{\partial \boldsymbol{y}} = \mathbf{H}_y, \frac{\partial \boldsymbol{l}_z}{\partial \boldsymbol{z}} = \mathbf{H}_z, \tag{2.19}$$

results from (2.11), and

$$\begin{aligned} \frac{\partial \mathcal{Q}}{\partial \boldsymbol{x}} &= 2\boldsymbol{l}_x^T \mathbf{Q} \frac{\partial \boldsymbol{l}_x}{\partial \boldsymbol{x}}, \\ \frac{\partial \mathcal{Q}}{\partial \boldsymbol{y}} &= 2\boldsymbol{l}_y^T \mathbf{Q} \frac{\partial \boldsymbol{l}_y}{\partial \boldsymbol{y}}, \\ \frac{\partial \mathcal{Q}}{\partial \boldsymbol{z}} &= 2\boldsymbol{l}_z^T \mathbf{Q} \frac{\partial \boldsymbol{l}_z}{\partial \boldsymbol{x}}, \end{aligned} \tag{2.20}$$

is a direct consequence of symmetry of $\mathcal{Q}$. Combining (2.19) with (2.20) leads to necessary conditions of stationarity of the quadratic form in (2.18). They read,

$$\begin{aligned} \mathbf{H}_x^T \mathbf{Q} \mathbf{H}_x \boldsymbol{x} &= \mathbf{0}, \\ \mathbf{H}_y^T \mathbf{Q} \mathbf{H}_y \boldsymbol{y} &= \mathbf{0}, \\ \mathbf{H}_z^T \mathbf{Q} \mathbf{H}_z \boldsymbol{z} &= \mathbf{0}. \end{aligned} \tag{2.21}$$

Taking particular representations of vector $\boldsymbol{q}$, paves way to geometrical minimum properties reported in Schek's paper. Indeed, setting $\boldsymbol{q} = \mathbf{1}\,(q_M = 1, M = 1, \ldots, m)$, gives $\mathcal{Q}(\boldsymbol{l}, \boldsymbol{q}) = \boldsymbol{l}^T\boldsymbol{l} = l_1^2 + l_2^2 + \cdots + l_M^2$. Hence, (2.21) with $\mathbf{Q} = \mathrm{diag}[q_M]$ provide the solution to the *minimum sum of squared lengths* problem. Alternatively, fixing $\boldsymbol{q} = \boldsymbol{l}^{\circ -1}$ (Hadamard inverse of $\boldsymbol{l}$, $q_M = 1/l_M, M = 1, \ldots, m$) leads to $\mathcal{Q}(\boldsymbol{l}, \boldsymbol{q}) = \boldsymbol{l}^T\mathbf{1} = l_1 + l_2 + \ldots + l_M$, thus (2.21) with $\mathbf{Q} = \mathrm{diag}[1/l_M]$ solve the problem of *minimum sum of lengths*.

Note that (2.21) represent equilibrium equations of a tensegrity with tensile and compressive force densities given by vector $\boldsymbol{q}$. Furthermore, setting $\boldsymbol{q} = \boldsymbol{\gamma}$ and applying $(2.14)_1$ in (2.9) gives $\mathcal{V}_{\mathrm{cable}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\gamma}) = (1/\sigma_T) \cdot \mathcal{Q}(\boldsymbol{l}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}), \boldsymbol{\gamma})$. As a result, the minimum volume problem for the pre-stressed cable net, $(\mathcal{P}_{\mathrm{pre}})$, becomes

$$(\mathcal{P}_{\mathrm{pre}}) \;:\; \mathcal{V}_{\mathrm{pre}}^{\min} = \frac{1}{\sigma_T} \min \left\{ \mathcal{Q}(\boldsymbol{l}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}), \boldsymbol{\gamma}) \;\middle|\; \begin{array}{l} \mathbf{D}_x(\boldsymbol{\gamma}, \mathbf{0})\boldsymbol{x} = \mathbf{0}, \\ \mathbf{D}_y(\boldsymbol{\gamma}, \mathbf{0})\boldsymbol{y} = \mathbf{0}, \\ \mathbf{D}_z(\boldsymbol{\gamma}, \mathbf{0})\boldsymbol{z} = \mathbf{0}, \\ (2.3)_2 \text{ applies.} \end{array} \right\}.$$

The minimum sum of lengths problem provides a certain variant of $(\mathcal{P}_{\mathrm{pre}})$. Namely, it amounts to the minimum volume problem for a cable net with constant cross section of cables. Note that if $A_M = A_0$ for all $M = 1, \ldots, m$, then tensile axial forces are also constant in the entire structure, $\boldsymbol{t} = A_0\sigma_T\mathbf{1}$. This feature refers to optimality of "constructions" resulting from natural processes, e.g., spider webs or trunks of trees. Defining the force density vector $\boldsymbol{\gamma} = A_0\sigma_T\boldsymbol{l}^{\circ -1}$ and setting $\boldsymbol{q} = \boldsymbol{\gamma}$ in (2.18)

reformulates (2.21) to

$$\begin{aligned} \mathbf{H}_x^T \mathbf{H}_x \boldsymbol{x} &= \mathbf{0}, \\ \mathbf{H}_y^T \mathbf{H}_y \boldsymbol{y} &= \mathbf{0}, \\ \mathbf{H}_z^T \mathbf{H}_z \boldsymbol{z} &= \mathbf{0}. \end{aligned} \tag{2.22}$$

Thus, the optimization problem for a spider-web-like cable net reads

$$(\mathcal{P}_{\text{spider}}) \ : \ \mathcal{V}_{\text{spider}}^{\min} = A_0 \min\left\{ \boldsymbol{l}^T \mathbf{1} \mid \boldsymbol{l} = \boldsymbol{l}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}), \text{ with } \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \text{ as in (2.22)} \right\}.$$

2.3 Numerical Examples

Simultaneous optimization of spatial distribution of truss nodes and values of axial stress resultants is a highly nonlinear problem in general. Indeed, taking $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$, $\boldsymbol{t}$, $\boldsymbol{c}$ as design variables leads to $(\mathcal{P}_{\text{truss}})$ as in Sect. 2.1.3, and such a problem involves algebraic functions of the form $f(u) = g(u)/h(u)$, with $u = u(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ and $h(u) = \sqrt{u}$. Introducing force densities, thus replacing $(\boldsymbol{t}, \boldsymbol{c})$ by $(\boldsymbol{\gamma}, \boldsymbol{\pi})$, recasts $(\mathcal{P}_{\text{truss}})$ into a form, where all functions are polynomials in five indeterminates: $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$, $\boldsymbol{\gamma}$, and $\boldsymbol{\pi}$. Treatment of polynomials is numerically less demanding than that for algebraic functions, even for nonlinear and non-convex optimization.

2.3.1 Single Cable of Minimum Volume

Consider a single cable shown in Fig. 2.1, pinned at the endpoints A and B and subjected to the lateral load $p = p(x)$ of arbitrary intensity. Here, H denotes horizontal thrust, i.e., the constant horizontal component of the tensile force in the cable, such that

$$N(x) = \frac{H}{\cos \varphi(x)}, \tag{2.23}$$

where $N = N(x)$ represents the tensile force and $\varphi = \varphi(x)$ stands for the angle of inclination of the tangent to the sag function $z = z(x)$ at given x.

By assumption, the cable is bending- and shear-free and thus remains in the static equilibrium under the action of lateral load p and axial forces N. It is a classical matter to conclude that in such a case, H, N, φ and p are linked through

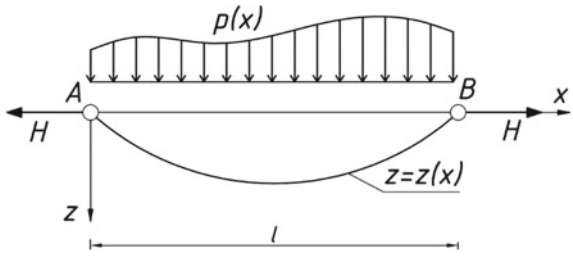


Fig. 2.1 Single cable under lateral load

$$\frac{dQ}{dx} + p = 0,\ Q(x) = H \cdot \tan\varphi(x),\ N(x) = \frac{H}{\cos\varphi(x)}. \tag{2.24}$$

Assuming constant stress σ_T in the entire cable, its minimum cross section is defined by

$$A(x) = \frac{N(x)}{\sigma_T} = \frac{1}{\sigma_T}\frac{H}{\cos\varphi(x)} = \frac{H}{\sigma_T}\left(1 + \tan^2\varphi(x)\right)^{\frac{1}{2}}. \tag{2.25}$$

Hence, with $s = s(x)$ denoting the natural coordinate along cable axis, the formula

$$\begin{aligned} V(H, Q) &= \int_0^{s(l)} A(x)ds(x) = \int_0^l A(x)\frac{dx}{\cos\varphi(x)} = \frac{1}{\sigma_T}\int_0^l \left(H + \frac{Q^2(x)}{H}\right)dx \\ &= \frac{l}{\sigma_T}\left(H + \frac{\|Q\|_2^2}{Hl}\right), \end{aligned} \tag{2.26}$$

expresses the volume of a single cable. Here, $\|f\|_2$ is the norm in the space of square-integrable functions, $f \in L^2([0, l]; \mathbb{R})$. Minimizing V with respect to H gives

$$H_{\text{opt}} = \frac{\|Q\|_2}{\sqrt{l}}, \tag{2.27}$$

and defines $H = H_{\text{opt}}$ for fixed Q satisfying $(2.24)_1$. From $(2.24)_2$ and $\tan\varphi = dz/dx$, it follows that

$$\frac{1}{\sqrt{l}}\left\|\frac{dz_{\text{opt}}}{dx}\right\|_2 = 1, \tag{2.28}$$

which is the so-called *Rozvany-Prager mean squared slope condition*, see (Lewiński et al. 2019). Originally, the formula (2.28) was derived in the context of an arch in compression, but it is also valid for checking the optimality of sag functions of cables. Combining (2.27) and (2.28) in (2.26) leads to

Table 2.1 Cable volume for different values of horizontal thrust H

H[N]	$\sigma_T V$ [cm^3]	$V_{\text{FDM}}/V_{\text{exact}}$			Rozvany-Prager's condition (2.28)		
		Number of segments					
		8	16	32	8	16	32
10	933.333	0.986	0.996	0.999	8.203	8.301	8.325
20	616.667	0.989	0.997	0.999	2.051	2.075	2.081
28.868	577.350	0.992	0.998	0.999	0.984	0.996	0.999
30	577.778	0.992	0.998	0.999	0.911	0.922	0.925
40	608.333	0.995	0.999	≈ 1.0	0.513	0.519	0.520
50	666.667	0.996	0.999	≈ 1.0	0.328	0.332	0.333

$$V_{\text{opt}} = 2\frac{\sqrt{l}}{\sigma_T}\|Q_{\text{opt}}\|_2 = 2\frac{H_{\text{opt}}\sqrt{l}}{\sigma_T}\left\|\frac{dz_{\text{opt}}}{dx}\right\|_2 = 2\frac{H_{\text{opt}}l}{\sigma_T}. \tag{2.29}$$

For detailed derivation of the equations above, refer to Dzierżanowski and Hetmański (2021); see also Dzierżanowski and Czubacki (2021).

In Table 2.1, the exact, analytically computed, volume of the single cable, V_{exact}, related to the given horizontal thrust H, are compared with the approximations, V_{FDM}, computed through the numerical simulations by the force density method for 8, 16, and 32 straight segments. In the calculations, it is assumed that $l = 10$m and $p = 10$N/m. Additionally: the differential Eq. $(2.24)_1$ with $Q(0) = -Q(l) = pl/2$ is used to calculate Q, formula (2.27) gives H_{opt}, Eq. $(2.24)_2$ with $z(0) = z(l) = 0$ leads to z_{opt}, and Eq. (2.29) provides V_{opt}.

Thus,

$$\begin{aligned} Q_{\text{opt}}(x) &= pl\left(\frac{x}{l} - \frac{1}{2}\right), \\ z_{\text{opt}}(x) &= \sqrt{3}x\left(1 - \frac{x}{l}\right), \end{aligned} \tag{2.30}$$

and

$$\begin{aligned} H_{\text{opt}} &= \frac{50}{\sqrt{3}}\text{N} \approx 28.868\ \text{N}\,, \\ V_{\text{opt}} &= \frac{1000}{\sqrt{3}\sigma_T}\text{cm}^3 \approx \frac{577.35}{\sigma_T}\text{cm}^3. \end{aligned} \tag{2.31}$$

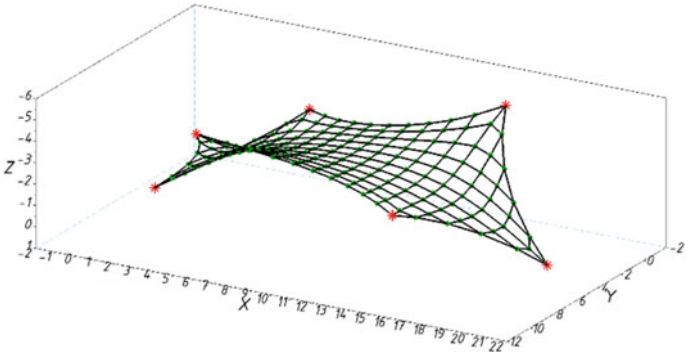


Fig. 2.2 Ground structure with member layout (*black lines*) and support location (*red stars*) of the cable net for study case 2.3.2.1—the axonometric view

2.3.2 Cable Nets of Minimum Volume

The example in Sect. 2.3.1 and also the problem ($\mathcal{P}_{\text{pre}}$) deal with cables of variable cross section, $A(x)$, and such cables are seldom used in real structures (the formula for $A_{\text{opt}}(x)$ is not reported here). Also, it is a matter of straightforward calculations to check, that $z_{\text{opt}}(l/2) = \left(\sqrt{3}/4\right)l \approx 4.33\,\text{m} = 0.433\,l$ is the sag at midspan of the optimal cable. Therefore, ($\mathcal{P}_{\text{pre}}$) and ($\mathcal{P}_{\text{cable}}$) are not practical in cable net optimization due to poor manufacturability and limited functionality.

Several realistic designs of cable nets with constant cross-sectional area, $A_M = A_0$, $M = 1, \ldots, m$, obtained through ($\mathcal{P}_{\text{spider}}$) are presented below. It is necessary to remark here that the problem is generically three-dimensional, as the computational algorithm for $\left(\mathcal{P}_{\text{spider}}\right)$ determines $\boldsymbol{x}_{\text{opt}}$, $\boldsymbol{y}_{\text{opt}}$, and $\boldsymbol{z}_{\text{opt}}$ for fixed forces $\boldsymbol{t} = A_0\sigma_T\mathbf{1}$. Mathematically, vector $\boldsymbol{t}$ is not subject to optimization, it merely parameterizes the minimization problem.

For each study case, the common goal was to optimize the location of cable joints assuming that forces in all—or almost all—cable segments are equal to 50 kN. Stresses in the entire structure are tensile and equal to the prefixed value σ_T. Initial member layout—usually referred to as the *ground structure*—is chosen arbitrarily and next gradually transformed to the optimal one by the computational algorithm. We emphasize that the topology of the ground structure is of secondary concern in the research, and therefore, this topic is intentionally not discussed in the paper.

Study cases considered in this section are rooted in civil engineering, though the analyzed structures are simplified for the purpose of clarity. The algorithm for numerical simulations in this section has been developed from scratch in Scilab.

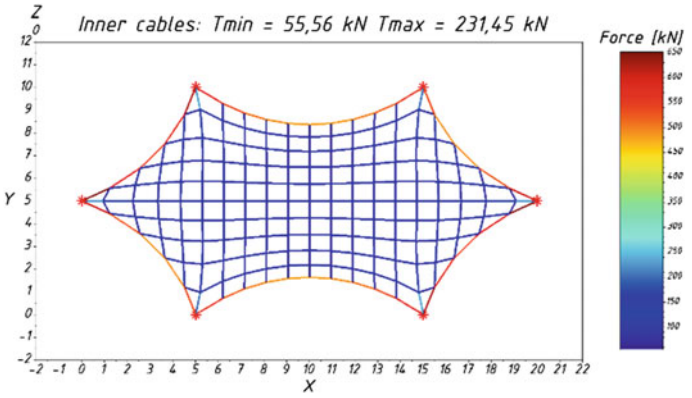


Fig. 2.3 Initial layout of members and nodes (in the top view) and tensile forces in cable segments for study case 2.3.2.1

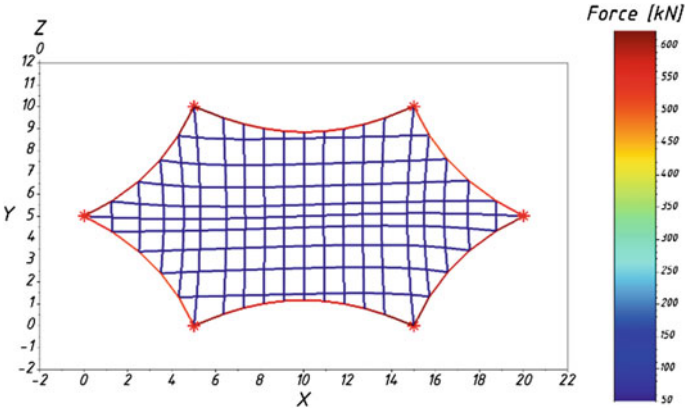


Fig. 2.4 Optimum layout of members and nodes (in the top view) and values of tensile forces in cable segments for study case 2.3.2.1. Forces in inner cable segments are fixed to 50kN; forces in outer cable segments result from the solution of ($\mathcal{P}_{\text{spider}}$)

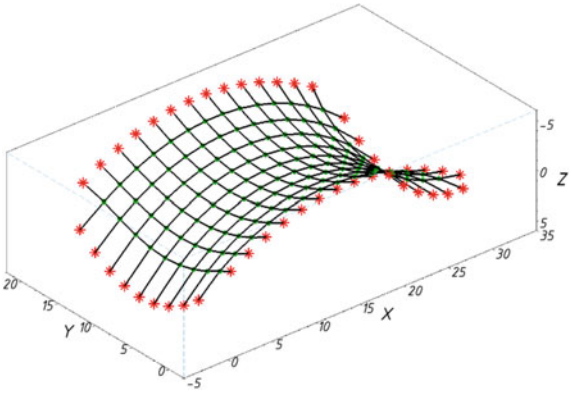


Fig. 2.5 Ground structure with member layout (*black lines*) and support location (*red stars*) of the cable net for study case 2.3.2.2—the axonometric view

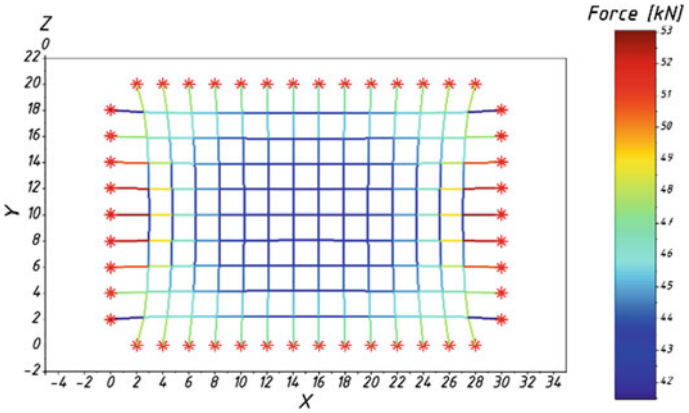


Fig. 2.6 Initial layout of members and nodes (in the top view) and tensile forces in cable segments for study case 2.3.2.2

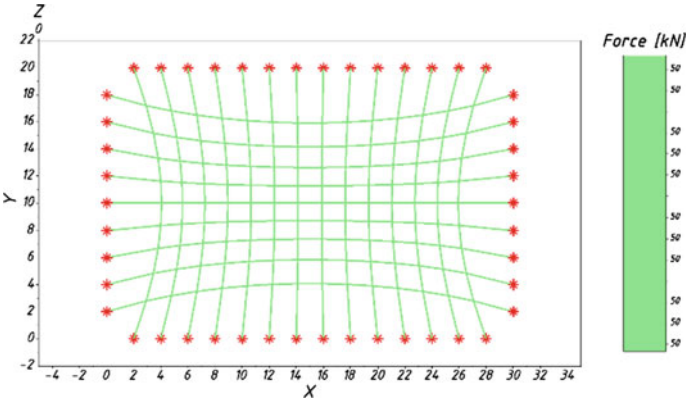


Fig. 2.7 Optimum layout of members and nodes (in the top view) and values of tensile forces in cable segments for study case 2.3.2.2. Forces in all cable segments are fixed to 50kN

2.3.2.1 Cable Net Pinned at the Vertices of a Hexagon

2.3.2.2 Cable Net Pinned at Four Parabolic Arches

2.3.2.3 Cable Net Pinned at the Parabolic Contour Over the Circular Ring

2.3.3 Trusses of Minimum Volume

In this section, the force density method is used for numerical simulations of ($\mathcal{P}_{\text{truss}}$), see Sect. 2.2.2. Now, structural members of a truss are allowed to carry either tensile

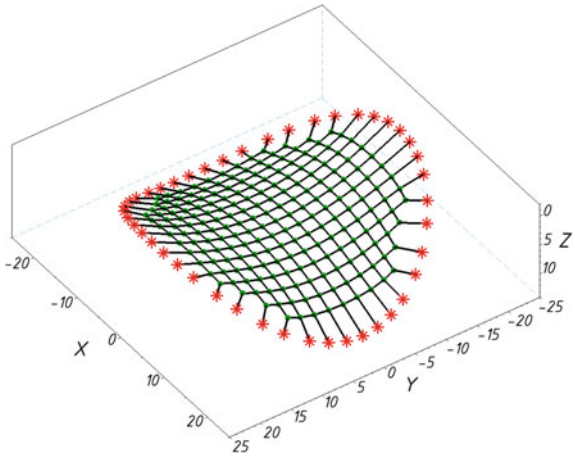


Fig. 2.8 Ground structure with member layout (*black lines*) and support location (*red stars*) of the cable net for study case 2.3.2.3—the axonometric view

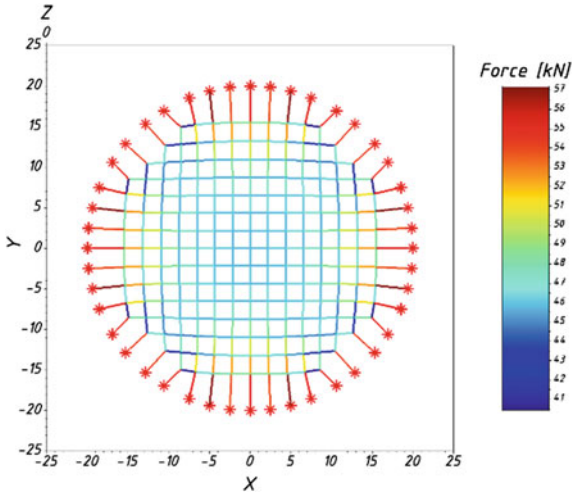


Fig. 2.9 Initial layout of members and nodes (in the top view) and tensile forces in cable segments for study case 2.3.2.3

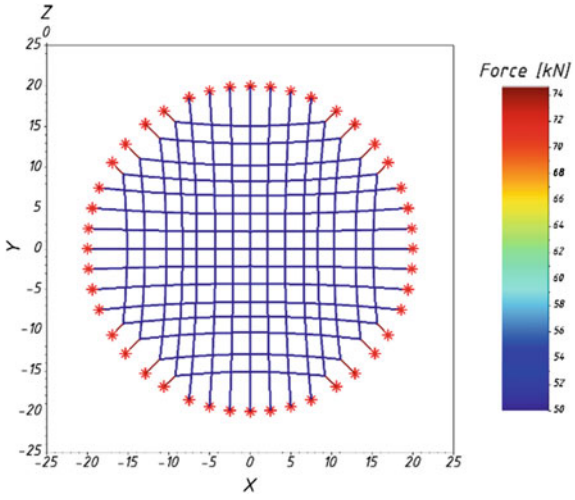


Fig. 2.10 Optimum layout (in the top view) and values of tensile forces in cable segments for study case 2.3.2.3. Forces in cable segments are fixed to 50kN, except 16 segments (*dark red*) whose axes run at 45° to the X- and Y-coordinate lines

or compressive forces, and structures are optimized for equal limit stress in tension and compression, that is for $\sigma_T/\sigma_C = 1$. This simplification is made for clarity only; the algorithm admits any value of σ_T/σ_C in general.

Similarly to Sec. 2.3.2, the initial configurations (ground structures) of trusses are chosen arbitrarily. The algorithm for numerical simulations has been developed from scratch in MATLAB.

2.3.3.1 Cantilever Truss

The initial configuration of a truss, as well as support and force locations, is shown in Fig. 2.11. Results of optimization are displayed in Fig. 2.12 with two subcases considered: (a) the design area is limited to the rectangle of dimensions $a \times 3a$, and (b) the design area is a strip of width $3a$ and infinite height. Note that enlarging the larger design area results in lower volume.

2.3.3.2 Beam Girder

Beam girder in this study case is optimized for $\sigma_T/\sigma_C = 1$. The initial configuration, as well as support and force locations, is shown in Fig. 2.13. Similarly, to study case 2.3.3.1, two optimization cases are considered: (a) the design area is limited to the rectangle of dimensions $a \times 3a$, and (b) the design area is a strip of width $3a$ and infinite height, see Fig. 2.14 for the results.

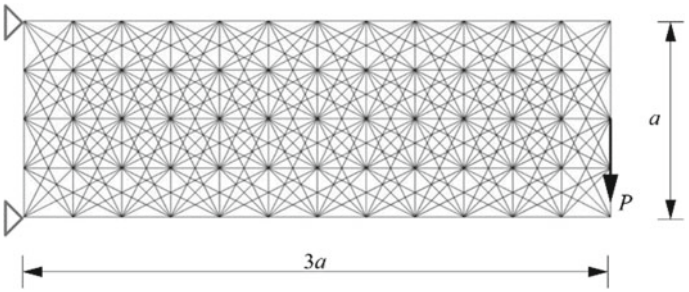


Fig. 2.11 Initial configuration and loading of a cantilever truss pinned at two nodes and loaded by force P along the vertical line distanced by $3a$ from the line of supports—study case 2.3.3.1. The ground structure truss consists of 65 nodes and 368 members

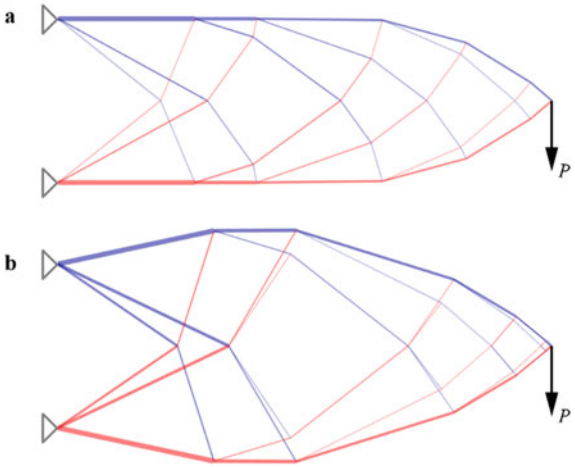


Fig. 2.12 Optimum layout of truss nodes and members for the study case 2.3.3.1 Members in tension are shown in *navy color* and those in compression are shown in *red color*, **a** the design area is limited to a rectangle of dimensions $a \times 3a$, minimum volume $V_{\text{opt,a}} = 13.76432 Pa/\sigma_T$, is only 1.2% larger than the analytical solution obtained in Graczykowski and Lewiński (2010); **b** the design area is a strip of width $3a$ and infinite height, minimum volume, $V_{\text{opt,b}} = 13.324597\ Pa/\sigma_T$, is only 2.5% larger than the one obtained in Sokół (2011) by topology optimization

2.4 Conclusions

The idea of force densities significantly helps in reformulating the problem of simultaneous geometry and topology optimization of truss and truss-like structures. To be specific, it allows to re-write the numerically demanding, original formulation involving node coordinate vectors $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ and axial force vectors $\boldsymbol{t}$, $\boldsymbol{c}$ as design variables into more tractable version with $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ and axial force density vectors $\boldsymbol{\gamma}$, $\boldsymbol{\pi}$. As a result, it is possible to define optimization problems with a smaller total number of

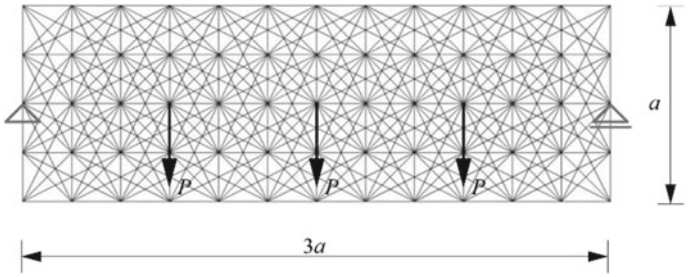


Fig. 2.13 Initial configuration and loading of a plane, simply supported beam girder loaded by three forces P, evenly distributed along the line spanning the supports. The ground structure truss consists of 65 nodes and 368 members

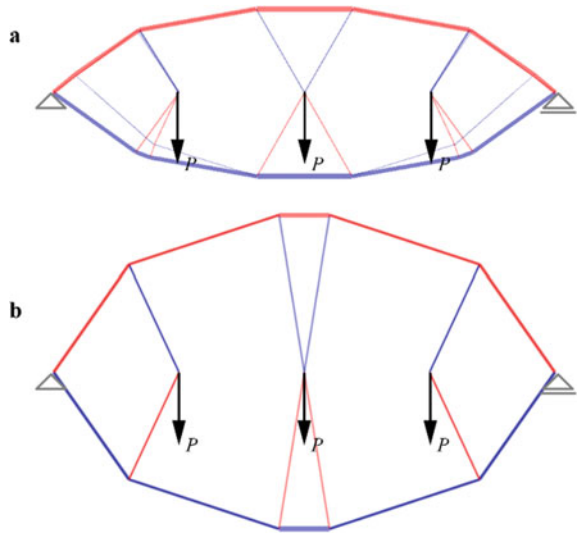


Fig. 2.14 Optimum layout of truss nodes and members for the study case 2.3.3.2 Members in tension are shown in *navy color* and those in compression are shown in *red color*, **a** the design area is limited to a rectangle of dimensions $a \times 3a$, minimum volume $V_{\text{opt,a}} = 10.581999\ Pa/\sigma_T$; **b** the design area is a strip of width $3a$ and infinite height, minimum volume, $V_{\text{opt,b}} = 8.908424\ Pa/\sigma_T$

unknowns. The proposed approach compensates for the larger CPU time than usually required for solving nonlinear problems and (more importantly) does not reduce the accuracy of numerical analysis.

By-passing the large computational cost of the method is possible by incorporating the *adaptive ground structure* strategy, see (Gilbert and Tyas 2003). Briefly speaking, the method consists in making the algorithm of ($\mathcal{P}_{\text{truss}}$) capable of modifying the topology of the ground structure, that is adding (or removing) the truss members. For this, the kinematic response of a structure is taken into account along the static one. Mathematically, it amounts to considering both primal and dual solutions to ($\mathcal{P}_{\text{truss}}$). This topic lies outside of the scope of the present paper, but it is a subject of extensive research by the authors.

Incorporating spatial distribution of truss nodes as design variables makes it possible to tackle the problem of optimal positioning of load, or, in other words,

finding an architectural form of the structure, which optimally adjusts to the profile of the applied load. Allowing for the migration of external forces in space and shaping the structure in accordance to their actual location is known in the literature as *form-finding*. The proposed algorithm, self-coded in Scilab and MATLAB, solves the form-finding problem for truss systems in both two- and three-dimensional settings.

Acknowledgements The paper was prepared within the Research Grant no 2019/33/B/ST8/00325 funded by the National Science Centre (Poland), entitled:

Merging the optimum design problems of structural topology and of the optimal choice of material characteristics. The theoretical foundations and numerical methods.

References

Adriaenssens, S., Block, Ph., Veenendaal, D., Williams, Ch. (eds.): Shell Structures Form Finding and Optimization, Routledge – Taylor and Francis Group, London, New York (2014)

Baker WF, Beghini LL, Mazurek A, Carrion J, Beghini A (2015) Structural innovation: combining classical theories with new technologies. Eng J 52:203–217

Bouchitté G, Mattei O, Milton GW, Seppecher P (2019) On the forces that cable webs under tension can support and how to design the cable webs to channel stresses. Proc Royal Soc A 475:20180781

Dzierżanowski G, Czubacki R (2021) Optimal archgrids spanning rectangular domains. Comp Struct 242:106371

Dzierżanowski G, Hetmański K (2021) Optimal design of archgrids: the second-order cone programming perspective. Arch Civil Eng 67:469–486

Dzierżanowski G, Wójcik-Grząba I (2020) Optimal form-finding of cable systems. Arch Civil Eng 66:305–321

Gilbert M, Tyas A (2003) Layout optimization of large-scale pin-jointed frames. Eng Comput 20:1044–1064

Graczykowski C, Lewiński T (2010) Michell cantilevers constructed within a half strip. Tabulation of selected benchmark results. Struct Multidiscip Optimiz 42:869–877

Lewiński T, Sokół T, Graczykowski C (2019) Michell structures. Springer International Publishing AG, Cham, Switzerland

Maxwell JC (1872) On reciprocal figures, frames, and diagrams of forces. Trans Royal Soc Edinburgh 26:1–40

Michell AGM (1904) The limits of economy of material in frame-structures. Phil Mag 8:589–597

Rozvany GIN, Prager W (1979) A new class of structural optimization problems: optimal archgrids. Comp Meth Appl Mech Eng 19:127–150

Schek H-J (1974) The force density method for form finding and computation of general networks. Comp Meth Appl Mech Eng 3:115–134

Sokół T (2011) A 99 line code for discretized Michell truss optimization written in Mathematica. Struct Multidiscip Optimiz 43:181–190

Strang G (2009) Introduction to linear algebra (4th edn). Wellesley—Cambridge Press, Wellesley, USA

Chapter 3
CI-SAPF for Structural Optimization Considering Buckling and Natural Frequency Constraints

Ishaan R. Kale and Ayush Khedkar

Abstract Structural analysis is important as it provides the basis for structural design and assesses whether a particular structure design will be able to withstand external and internal pressures and forces. While solving the structural problems, we find that such problems are quite complex in solving and the number of iterations taken to find the optimal solution is quite high. Various techniques were used to solve such types of problems such as the artificial bee colony algorithm (ABC), fuzzy-controlled genetic algorithm (FCGA), approximation method algorithm, harmony search algorithm (HS), constrained mean–variance mapping algorithm, buckling restrained sizing, and shaping algorithm. The following paper uses the CI-SAPF approach to find the size optimization of truss structure using buckling and natural frequency constraints along with discrete variables. In this paper, two separate problems of 18-bar truss structure are considered. The solutions obtained from CI-SAPF algorithm are compared with other contemporary techniques.

Keywords Truss structure · Cohort intelligence · Buckling constraint · Natural frequency constraint

3.1 Introduction

Over the past few years, many natures inspired optimization algorithms were applied to optimize the size, shape, and weight of the truss structures. These problems are associated with various constraints such as stress, deflection, Euler buckling,

I. R. Kale (✉)
Institute of Artificial Intelligence, Dr Vishwanath Karad MIT World Peace University, Pune, MH 411038, India
e-mail: ishaan.kale@mitwpu.edu.in

A. Khedkar
Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, MH 412115, India

I. R. Kale and A. Sadollah (eds.), *Optimization Methods for Structural Engineering*, Engineering Optimization: Methods and Applications,
https://doi.org/10.1007/978-981-99-2378-6_3

frequency, and other topological constraints. These algorithms include the artificial bee colony (ABC), particle swarm optimization (PSO), ant colony optimization (ACO), fuzzy-controlled genetic algorithm (FCGA), approximation methods, harmony search (HS), constrained mean–variance mapping algorithm, buckling restrained sizing, and shaping algorithm so on and so forth (Kashani et al. 2022). Such algorithms have been applied on various truss structures such as 10-bar structures, 17-bar structures, 18-bar structures, 72-bar structures, and so on.

Jawad et al. (2021) solved the truss structure problems using ABC combining size and shape optimization by having the discrete and continuous variables. A well-known penalty function constraint handling approach was used to handle the displacement, size, and buckling constraints. This algorithm demonstrated a robust performance with a success rate of 100%. The FCGA was flexible to deal with rigidly jointed structures and it did not impose any kind of limitations as compared to other optimization methods. This method provided an easy way of dealing with complex problems because fuzzy models containing vagueness and impreciseness in knowledge representation could be built easily. This algorithm resulted in lesser weight with a smaller number of iterations and reduced the computational time and enhanced the search efficiency (Soh' and Yang 1996). Approximation method for configuration optimization of trusses proposed by Hansen and Vanderplaats (1990) used the first order Taylor series expansion for the structural analysis. The degree of coupling between sizing and geometry design variables was reduced by force approximation which resulted an increase in the accuracy of the final result approximation. The HS algorithm was based on the concept of musical process of searching for a perfect state of harmony. This algorithm did not require any initial values and it used a random search instead of gradient search which made the derivative information unnecessary. The results indicated that this technique was a powerful search and optimization method for solving the structural problems as compared to conventional mathematical methods. From the convergence curve of this algorithm, it can be observed that this technique outperformed the GA-based algorithms, but fuzzy-controlled method was better than this approach (Lee and Geem 2004). Another algorithm consisted of a single-solution and population-based variants of mean variance mapping optimization (MVMO) with an adaptive exterior penalty scheme to handle geometric and mechanical constraints. The results of this approach showed a rapid rate of convergence as compared to other algorithms. MVMO was a robust and reliable tool for computationally expensive problems with hundreds of constraints as this method was the combination of mapping functions, the archive of best points, the adaptive strategies employed in MVMO, and the dynamic exterior penalty function. A globally optimum solution with a minimum number of function evaluations was achieved for a population size of twice the number of variables using the MVMO method. Geometrically nonlinear forward model-based optimization algorithm was used for the analysis of derivative-free directionality-based optimization scheme to have minimum structural weight, as a large number of buckling constraints on cost function reduced computational efficiency and also violated constraints (Aslani et al. 2018). From the results, it was interpreted that the optimized cross-sectional

area may become smaller than in the case of linear analysis solution by incorporating geometric nonlinearity in the forward model. Although the proposed algorithm appeared efficient, many aspects were needed to be investigated (Venkatesh Varma et al. 2020).

The CI algorithm was implemented and proposed by Kulkarni et al. (2013). It is inspired from the socially learning behaviour of the with every other candidate in the cohort. This approach is also applied for solving real-world combinatorial problems from healthcare and logistics domains as well as for large-sized complex problems from the Cross-Border Supply Chain (Kulkarni et al. 2016) Domain Travelling Salesman Problem (TSP) and several benchmark problems (Kulkarni et al. 2017). It was also implemented for constrained problems from truss structure domain and mechanical engineering domain (Kale and Kulkarni 2018; Kale et al. (2022); Kale and Kulkarni (2021a, b)). Further, a hybrid version CI-SAPF-CBO was proposed (Kale and Kulkarni 2021a, b) to overcome the limitation of CI (i.e. sampling space reduction factor). In this, self-adaptive penalty function (SAPF) is the modified version of penalty function approach which does not require tuning of penalty parameter.

In this chapter, CI is investigated for solving the truss structure problems having bucking and nature frequency constraints. The truss structure problems are basically aimed to minimize the weight. For the application of the CI-SAPF algorithm, an 18-bar as well as a 10-bar truss structure problem is taken into consideration. The members of the 18-bar truss structure are divided into four groups having the same cross-sectional area while handling the buckling constraint, whereas the 10-bar structure problem is solved while handling the frequency constraint. This paper aims at reducing the weight of the 18-bar and 10-bar truss structures.

3.2 Cohort Intelligence (CI)

Cohort intelligence is a self-organizing system which is based on artificial intelligence (AI) concepts. The candidates in CI model interact with each other and try to compete with other candidates in order to achieve a certain objective. While trying to achieve this goal, each candidate tries to improve its own behaviour by observing and following the behaviour of other candidates present in that cohort. A candidate follows certain type of behaviour that results into the improvement of its own behaviour. A candidate having the best behaviour is followed by the other candidates by taking help of the roulette wheel approach which may result in the improvement of their own behaviour. Hence, each candidate learns from one another, and the overall cohort behaviour is evolved. If after considerable number of learning attempts the candidate does not show significant change in its behaviour, then the cohort behaviour is considered to be saturated. All the truss structure problems are successfully solved by using CI-SAPF algorithm. The SAPF function is used to handle the inequality constraints (Fig. 3.1).

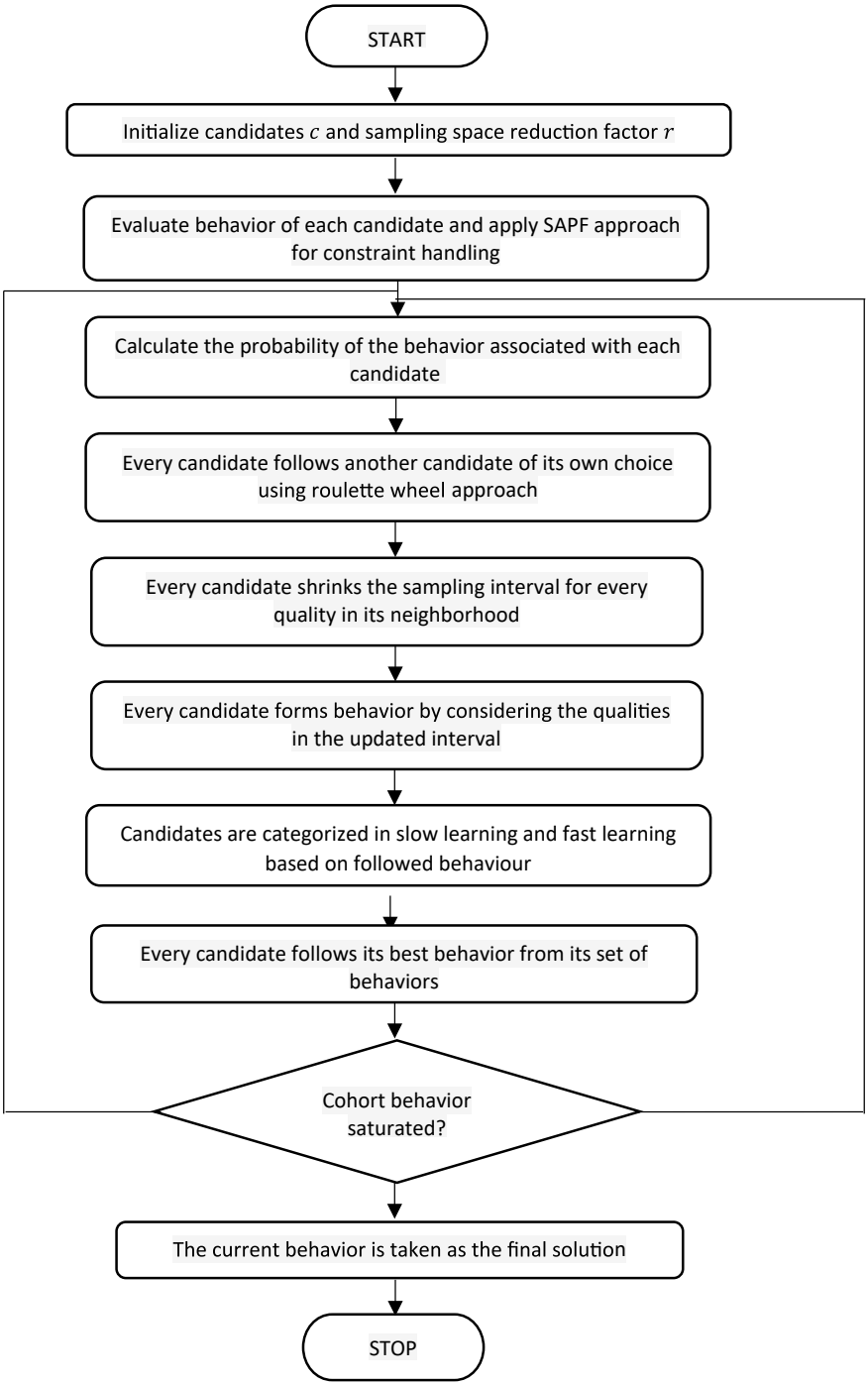


Fig. 3.1 Cohort Intelligence flowchart

3.3 Test Problems

The main objective of the truss structural optimization problems is to reduce the weight of the structure by achieving the optimum nodal coordinates and optimum element cross-sectional areas while satisfying constraints. These problems are previously solved using several metaheuristic algorithms such as HS (Lee and Geem 2004), multiplier method (Imai and Schmit 1981), mean–variance mapping optimization (MVMO) (Aslani et al. 2018), geometric nonlinear (GNL) (Venkatesh Varma et al. 2020), democratic particle swarm optimization (DPSO) (Kaveh and Zolghadr 2012), school-based optimization (SBO) (Farshchin et al. 2016), vibrating particles system (VPS) (Kaveh and Ghazaan 2017), (Roulette Wheel Selection-Elitist-Differential Evolution) (ReDe) (Ho-Huu et al. 2018), improved symbiotic search algorithm (ISOS) (Tejani et al. 2018), adaptive hybrid evolutionary firefly algorithm (AHEFA) (Lieu et al. 2018), modified simulated annealing algorithm (MSAA) (Millan-Paramo 2020). In general, the truss structural problem is defined as follows:

$$\begin{aligned}
&\text{Minimize} \quad W(X) = \sum_{i=1}^{n} \rho A_i L_i \\
&\text{Subject to} \quad f_q - f_{q\ \min} \geq 0 \\
&\qquad\qquad f_r - f_{r\ \max} \leq 0 \\
&\qquad\qquad A_{i\ \min} \leq A_i \leq A_{i\ \max} \\
&\qquad\qquad NC_{j\ \min} \leq NC_j \leq NC_{j\ \max} \\
&A = \{A_1, A_2, \ldots., A_n\} \text{ and } NC = \{NC_1, NC_2, \ldots., NC_m\}
\end{aligned}$$

where

W	is the weight of the structure;
n	is the total number of members of the structure;
m	is the total number of nodes;
ρ_i	is the material density;
A_i	is the cross-sectional area;
L_i	is the length of the ith member;
NC_j	is a nodal coordinate (x_j, y_j, z_j) of node jth of the truss;
f_q and f_r	are the qth and rth natural frequencies of the structure respectively and the superscripts, max and min denote maximum and minimum allowable limits, respectively.

Test Problem 1: 18-bar truss structure problem

The 18-bar truss structure problem (refer to Fig. 3.2) is taken into consideration for the investigation of CI-SAPF algorithm. The 18-bar structure problem was previously solved using different algorithms such as Multiplier method (Imai and Schmit 1981), HS (Lee and Geem 2004), MVMO (Aslani et al. 2018), and GNL (Venkatesh Varma

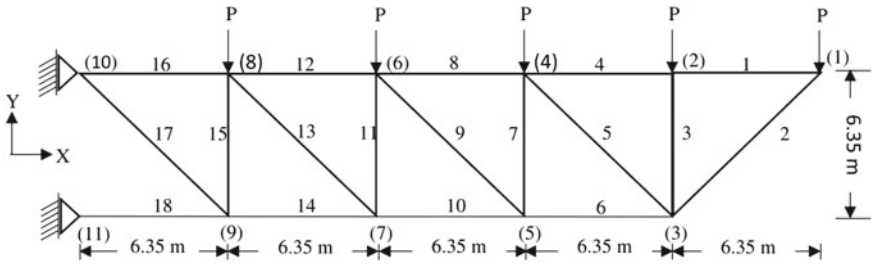


Fig. 3.2 Spatial 18-bar truss structure

et al. 2020). A force of 20 kips is applied in the downward direction on nodes 1, 2, 4, 6, and 8, and the material density is considered to be 0.1 lb/m^3. The members of the truss structure are divided into four groups having the same cross-sectional area. These are represented as A1 for members 1, 4, 8, 12, and 16; A2 for members 2, 6, 10, 14, and 18; A3 for members 3, 7, 11, and 15, and A4 for members 5, 9, 13, and 17. Euler buckling constraints are also taken into consideration and represented by:

$$\sigma_b = -(K \times E \times A_i)/L^2$$

where K is the buckling constant having a value of 4; E is the modulus of elasticity ($E = 10{,}000$ ksi); and L is the length of element.

For solving the 18-bar truss structure problem, the results obtained using CI-SAPF approach are compared with other contemporary algorithms (refer Table 3.1). The best, mean, and worst function values obtained from 30 trials using CI-SAPF are 2780.3374 lb, 2780.3430 lb, and 2780.3488 lb, respectively, with standard deviation 0.0034, average function evaluations 3486, and average CPU time 23.87 s. It is observed that CI-SAPF algorithm received the significantly improved solution with slightly higher function evaluations as compared to multiplier method, HS algorithm, MVMO, and GNL algorithm.

Test Problem 2: 10-bar truss structure problem

A 10-bar truss structure problem was taken into the consideration for the investigation of CI-SAPF algorithm. The 10-bar truss structure was previously analysed using various algorithms DPSO (Kaveh and Zolghadr 2012), SBO (Farshchin et al. 2016), VPS (Kaveh and Ilchi Ghazaan 2017), ReDe (Ho-Huu et al. 2018), ISOS (Tejani et al. 2018), AHEFA (Lieu et al. 2018), and MSAA (Millan-Paramo 2020). A lumped mass of 454 kg was applied on the nodes 1, 2, 3, and 4. The material density considered is 2770 kg/m^3, and Young's Modulus acting on the truss structure is 6.98×10^{10} N/m^2. The size variables are within the limits of $0.645 \leq A \leq 50$, whereas the frequency constraints are within the limits $f_1 \geq 7$, $f_2 \geq 15$ and $f_3 \geq 20$ (Fig. 3.3).

For solving the 10-bar truss structure problem, the results obtained using CI-SAPF approach are compared with other contemporary algorithms (refer Table 3.2). The best and average function values obtained from 30 trials using CI-SAPF

Table 3.1 Comparison of results for 18-bar truss structure problem

Weight *lb*	Multiplier method (Imai and Schmit 1981)	HS (Lee and Geem 2004)	MVMO (Aslani et al. 2018)	GNL (Venkatesh Varma et al. 2020)	CI-SAPF
W-mean	–	–	–	44,564 (2021.4 kg)	2780.343
W-best	6430.0	6421.88	6418.725	44,024 (1966.9 kg)	2780.337
W-worst	–	–	–	46,590 (2113.3 kg)	2780.349
FE	–	2000	2200	2170	3486
SD	–	–	–	–	0.0034

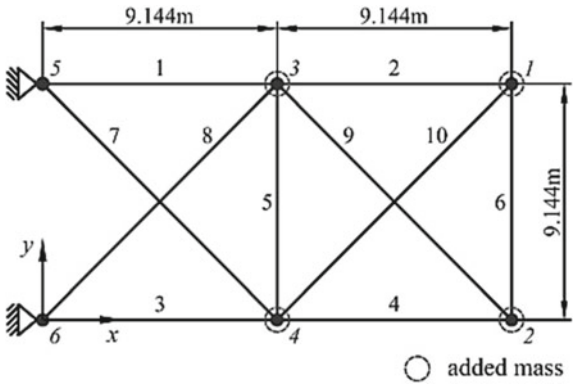


Fig. 3.3 10-Bar truss structure

are 524.5624 kg and 525.2325 kg, respectively, with standard deviation of 0.7366, average function evaluations 1185, and average CPU time 25.17 s. Along with the weight, the frequency of each variable was also calculated (refer Table 3.3) and was compared with other algorithms.

From the comparison table, it is observed that CI-SAPF obtained best weight of the truss structure as compared to DPSO, SBO, VPS, and MSAA algorithms. While the best weight obtained by CI-SAPF is precisely similar as compared to ReDe, ISOS, and AHEFA algorithms, the CI-SAPF algorithm also obtained the robust solution as compared to other cotemporary algorithm within less number of function evaluations.

3.4 Result Analysis and Discussion

The CI-SAPF algorithm is successfully validated by solving the 18-bar and 10-bar truss structure problems involving buckling and frequency constraints. The CI-SAPF

Table 3.2 Comparison of results for 10-bar truss structure problem

Variables	DPSO (Kaveh and Zolghadr 2012)	SBO (Farshchin et al. 2016)	VPS (Kaveh and Ilchi Ghazaan 2017)	ReDe (Ho-Huu et al. 2018)	ISOS (Tejani et al. 2018)	AHEFA (Lieu et al. 2018)	MSAA (Millan-Paramo 2020)	CI-SAPF
A1	35.9440	35.5994	35.1471	35.1565	35.2654	35.1714	32.09710	35.5600
A2	15.5300	14.9956	14.6668	14.7605	14.6803	14.7203	15.5925	14.3700
A3	38.2850	35.4806	35.6889	35.1187	34.4273	35.1074	32.8514	34.8800
A4	15.3850	14.7646	15.0929	14.7275	14.9605	14.6986	15.5942	14.5600
A5	0.6480	0.6450	0.6450	0.6450	0.6450	0.6451	0.6454	0.6450
A6	4.5830	4.6305	4.6221	4.5558	4.5927	4.5593	4.6552	4.580
A7	23.6100	24.3272	23.5552	23.7199	23.3417	23.7330	26.1179	23.8100
A8	23.5990	23.8528	24.4680	23.6304	23.8236	23.6795	26.1350	24.0800
A9	13.1350	12.6797	12.7198	12.3827	12.8497	12.3987	11.9983	12.0800
A10	12.3570	12.6375	12.6845	12.4580	12.5321	12.4231	11.9339	12.5100
Best weight (kg)	532.3900	532.0500	530.7700	524.4500	524.7300	524.4500	532.0400	524.5624
Average weight (kg)	537.80	533.4500	535.6400	524.7600	530.0300	525.1600	532.0600	525.2325
SD	4.0200	2.2000	2.5500	1.1100	3.4800	1.9200	0.0100	0.7366
FE	6000	10,000	30,000	8300	4000	5860	7130	1185

Table 3.3 Comparison of frequency for 10-bar truss structure problem

Frequency number (Hz)	DPSO (Kaveh and Zolghadr 2012)	SBO (Farshchin et al. 2016)	VPS (Kaveh and Ilchi Ghazaan 2017)	ReDe (Ho-Huu et al. 2018)	ISOS (Tejani et al. 2018)	AHEFA (Lieu et al. 2018)	MSAA (Millan-Paramo 2020)	CI-SAPF
1	7.0000	7.0000	7.0000	7.0000	7.0001	7.0000	7.0000	7.0000
2	16.1870	16.1660	16.1599	16.1924	16.1703	16.1920	15.8458	16.19411
3	20.0000	20.0000	20.0000	20.0000	20.0024	20.0000	20.0000	20.0000
4	20.0210	20.0004	20.0001	20.0002	–	20.0000	20.0001	20.0001
5	28.4700	28.5834	28.6008	28.5517	–	28.5551	27.9633	28.49678
6	29.2430	29.0034	29.0628	–	–	28.9588	28.0334	28.87237
7	48.7690	48.5051	48.4904	–	–	48.5777	46.8421	48.500031
8	51.3890	51.0259	51.0476	–	–	51.0712	48.8795	50.93509

algorithm is run for 30 times for both the problems. The results are presented in Table 3.1, Table 3.2, and Table 3.3. Tables 3.1 and 3.2 present the best and mean weight including the standard deviation, average CPU time taken, and the function evaluations obtained from CI-SAPF along with result comparisons with other algorithms. The buckling and frequency constraints associated with these problems are handled using SAPF approach and CI worked as a global optimizer. The results obtained from CI-SAPF algorithm for solving 18-bar and 10-bar truss structure problems are compared with several other metaheuristics optimization algorithms such as HS (Lee and Geem 2004), multiplier method (Imai and Schmit 1981), MVMO (Aslani et al. 2018), GNL (Venkatesh Varma et al. 2020), DPSO (Kaveh and Zolghadr 2014), SBO (Farshchin et al. 2016), VPS (Kaveh and Ilchi Ghazaan 2017), ReDe (Ho-Huu et al. 2018), ISOS (Tejani et al. 2018), AHEFA (Lieu et al. 2018), and MSAA (Millan Paramo 2020).

The HS algorithm (Lee and Geem 2004) uses a stochastic random search approach incorporated with harmony memory considering rate (HMCR) and pitch adjusting rate (PAR). These two parameters play a crucial role to obtain the improved solution vectors. However, it is necessary to set an appropriate value of HMCR and PAR to obtain the better solution matrix. On the other hand, CI-SAPF does not require parameter tuning. Although it has computational parameters such as number of cohort candidates and sampling space reduction factor, these are selected based on analysis conducted in Kale and Kulkarni (2017). This helps in lower down the initial computational efforts. In Aslani et al. (2018), the gradient-based mean–variance mapping optimization (MVMO) is proposed to solve the constrained truss problems. An adaptive penalty function approach is proposed which is a modified version of exterior penalty function approach to handle the constraints. The gradient-based algorithms are associated with Hessian matrix which became a limitation to the proposed constrained handling approach in MVMO. The DPSO algorithm does not tune the parameter thus reducing the tedious task of tuning and making this algorithm comparatively faster than other algorithms (Kaveh and Zolghadr 2012). The CI-SAPF achieved the same results with much lesser function evaluation due to the use of SAPF. In CI-SAPF algorithm, every candidate is self-supervised which follows a certain behaviour using a roulette wheel approach. This assists to get better position in the search space and make the algorithm to obtained better solutions. In SAPF approach, the function value is used as a penalty parameter which is updated in every iteration of the algorithm. This function value keeps on improving as the iteration progresses which accelerates the convergence rate and obtains better results with lesser computational cost. The main advantage of the proposed SAPF approach is that it can be directly applicable to a variety of constrained optimization problems without any previous trials. The CI-SAPF algorithm is reported to be more efficient which reduces the efforts of trial-and-error process of setting the suitable penalty parameters. The CI using SAPF approach obtained better results than other algorithm, and it significantly reduced the weight of the truss structure. It was observed that CI-SAPF gave better results than other algorithms.

3.5 Conclusion

The following paper made use of the CI-SAPF approach to solve the 18-bar truss problem having buckling constraint as well as 10-bar truss structure problem having the frequency constraint. The CI-SAPF approach is used for the first time to solve problems having dynamic constraints. It was observed that CI-SAPF approach produced better solutions both the times. The solution obtained from this algorithm showed significant reduction in the weight of the truss structure for 18-bar as well as 10-bar truss structure problems. The frequency obtained for each variable in the 10-bar truss structure problem was also better than other algorithms. CI-SAPF reduces the time taken to solve the problem and also gives better solution using a smaller number of iterations. The CI-SAPF algorithms can be applied for solving real-world mechanical design engineering, transportation problems in the near future. Based on the advantages and disadvantages, other nature inspired algorithms can be hybridized with CI-SAPF for solving challenging and complex real-world problems from various domains.

References

Aladeemy M, Adwan L, Booth A, Khasawneh MT, Poranki S (2020) New feature selection methods based on opposition-based learning and self-adaptive cohort intelligence for predicting patient no-shows. Appl Soft Comput 86:105866

Aslani M, Ghasemi P, Gandomi AH (2018) Constrained mean-variance mapping optimization for truss optimization problems. Struct Design Tall Spec Build 27(6):p1449

Farshchin M, Camp CV, Maniat M (2016) Optimal design of truss structures for size and shape with frequency constraints using a collaborative optimization strategy. Expert Syst Appl 66:203–218

Hansen SR, Vanderplaats GN (1990) Approximation method for configuration optimization of trusses. AIAA journal, 28(1):161–168

Ho-Huu V, Nguyen-Thoi T, Truong-Khac T, Le-Anh L, Vo-Duy T (2018) An improved differential evolution based on roulette wheel selection for shape and size optimization of truss structures with frequency constraints. Neural Comput Appl 29(1):167–185

Imai K, Schmit LA Jr (1981) Configuration optimization of trusses. J Struct Div 107(5):745–756

Jawad FK, Ozturk C, Dansheng W, Mahmood M, Al-Azzawi O, Al-Jemely A (2021) Sizing and layout optimization of truss structures with artificial bee colony algorithm. In Structures 30:546–559. Elsevier

Kale IR, Kulkarni AJ (2018) Cohort intelligence algorithm for discrete and mixed variable engineering problems. Int J Parallel, Emergent Distrib Syst 33(6):627–662

Kale IR, Kulkarni AJ (2021a) Cohort intelligence with self-adaptive penalty function approach hybridized with colliding bodies optimization algorithm for discrete and mixed variable constrained problems. Complex Intell Syst 7(3):1565–1596

Kale IR, Kulkarni AJ (2021b) Constraint handling in cohort intelligence algorithm. CRC Press

Kale IR, Pachpande MA, Naikwadi SP, Narkhede MN (2022) Optimization of advanced manufacturing processes using socio inspired cohort intelligence algorithm. Int J Simul Multi Design Optim 13:6

Kashani AR, Camp CV, Rostamian M, Azizi K, Gandomi AH (2022) Population-based optimization in structural engineering: a review. Artif Intell Rev 55:345–452

Kaveh A, Ilchi Ghazaan M (2017) Vibrating particles system algorithm for truss optimization with multiple natural frequency constraints. Acta Mech 228(1):307–322

Kaveh A, Zolghadr A (2012) Truss optimization with natural frequency constraints using a hybridized CSS-BBBC algorithm with trap recognition capability. Comput Struct 102:14–27

Kulkarni AJ, Baki MF, Chaouch BA (2016) Application of the cohort-intelligence optimization method to three selected combinatorial optimization problems. Eur J Oper Res 250(2):427–447

Kulkarni AJ, Krishnasamy G, Abraham A (2017) Cohort intelligence: a socio-inspired optimization method. Springer International Publishing, Switzerland

Kulkarni AJ, Durugkar IP, Kumar M (2013) Cohort intelligence: a self supervised learning behavior. In: 2013 IEEE international conference on systems, man, and cybernetics, pp 1396–1400

Lee KS, Geem ZW (2004) A new structural optimization method based on the harmony search algorithm. Comput Struct 82(9–10):781–798

Lieu QX, Do DT, Lee J (2018) An adaptive hybrid evolutionary firefly algorithm for shape and size optimization of truss structures with frequency constraints. Comput Struct 195:99–112

Millan-Paramo C, Abdalla Filho JE (2020) Size and shape optimization of truss structures with natural frequency constraints using modified simulated annealing algorithm. Arab J Sci Eng 45(5):3511–3525

Soh CK, Yang J (1996) Fuzzy controlled genetic algorithm search for shape optimization. J Comput Civ Eng 10(2):143–150

Tejani GG, Savsani VJ, Patel VK (2016) Adaptive symbiotic organisms search (SOS) algorithm for structural design optimization. J Comput Des Eng 3(3):226–249

Tejani GG, Savsani VJ, Patel VK, Mirjalili S (2018) Truss optimization with natural frequency bounds using improved symbiotic organisms search. Knowl-Based Syst 143:162–178

Venkatesh Varma T, Sarkar S, Mondal G (2020) Buckling restrained sizing and shape optimization of truss structures. J Struct Eng 146(5):04020048

Chapter 4
Improved Drosophila Food-Search Algorithm for Structural and Mechanical Optimization Problems

Ali Mortazavi

Abstract The metaheuristic optimization methods are non-gradient-based search techniques that wildly utilized for solving various optimization cases in the different disciplines. Drosophila food-search algorithm (DFO) is a metaheuristic search approach that models the food-search mechanism of the Drosophila Melanogaster incest in the nature. It applies a three-phase search pattern to explore the desired problem's search domain. To provide a proper level of population diversity during the optimization process, it applied the quadratic approximation search approach. In this chapter, an improved version of the DFO is utilized for solving structural optimization problems. The attained outcomes are reported and compared with five other well-stablished methods as differential evolution (DE), firefly algorithm (FA), teaching and learning-based optimization (TLBO), Drosophila food-search optimization (DFO), and integrated particle swarm optimization (iPSO). The acquired results demonstrate that the proposed IDFO, specially compared with the standard DFO, provides an appropriate dynamic balance between exploration and exploitation search behaviors during the optimization process. Also, the proposed method in comparison with all other selected methods has the most stable behavior with the lowest standard deviation value.

Keywords Drosophila food-search algorithm · Metaheuristic · Structural problems · Mechanical problems

4.1 Introduction

In the many cases, the optimal configuration (e.g., balancing the consumed material, serviceability, and economical properties) is the most important goal of the engineering design process (Kazikova et al. 2021). Applying a proper optimization technique plays a key role to establish such an optimal condition (Mortazavi 2020a). The optimization methods can be categorized into two main sub-groups: stochastic and

A. Mortazavi (✉)
Department of Civil Engineering, Izmir Democracy University, Izmir, Turkey
e-mail: ali.mortazavi.phd@gmail.com; ali.mortazavi@idu.edu.tr

I. R. Kale and A. Sadollah (eds.), *Optimization Methods for Structural Engineering*, Engineering Optimization: Methods and Applications,
https://doi.org/10.1007/978-981-99-2378-6_4

deterministic approaches (Mortazavi 2019a). The deterministic approaches indicate the conventional gradient-based approaches. Although these methods are efficient, they suffer from two major shortcomings; first, they required differentiable objective functions and its different grade gradients to define the steps sizes and directions during the procedure. Generally, in many complex optimization problems, it is very difficult or even impossible to define such an objective function (Mortazavi 2020b). Next, they are extremely vulnerable to the initial conditions of running procedure; it means if the population is selected through an inappropriate location of search space, the algorithm would simply converge to nearest local optima. Based on these facts, the non-deterministic approaches provide an alternative search mechanism to solve optimization problems. Metaheuristic methods form one of the main groups of the non-deterministic methods. These methods usually provide a mathematical model which mimics natural phenomena, physical rules, or social behaviors in the society (Degertekin et al. 2018). Distinct methods that their search patterns are inspired by different natural and physical models are generated and used in different fields of science and engineering (Kale and Kulkarni 2018, 2021). Also, the different reinforced methods are also generated to solve global optimization problems (Mortazavi 2022, 2021; Mortazavi and Moloodpoor 2021a, b).

Drosophila food-search mechanism (DFO) (Das and Singh 2014) is a metaheuristic search approach that simulates the food-search mechanism of Drosophila Melanogaster bug in the nature. This method uses a quadratic approximation approach to maintain the required diversity in the population during the optimization process. In this phase, the DFO applies the data stored in three arbitrary agents, resulting a high level of exploration implementation during the optimization process. However, a proper search process requires an appropriate balance between local and global search behaviors (Mortazavi 2019b). To cover this aim, using the concept of the weighted agent, the quadratic search pattern of standard DFO is enhanced. The new method is called improved Drosophila food-search optimization (IDFO). Mostly in an engineering problem, to reach a feasible solution, the search domain of the problems should be limited by different restrictions. This issue makes the engineering problem more complex in comparison with standard unconstrained mathematical problems. Thus, these problems provide high challenging area for optimization techniques (Mei and Wang 2021). In the current chapter, the search performance of the developed IDFO is assessed on solving constrained structural and mechanical optimization problems. The results are evaluated and compared with five other techniques. In the next two subsections, the standard DFO and its improved version are described in detail, respectively. Then the numerical problems are solved using different methods, and results are compared.

4.2 Drosophila Food-Search Algorithm (DFO)

Drosophila food-search algorithm (DFO) is a metaheuristic optimization algorithm that imitates the mating and/or food-search behavior of *Drosophila Melanogaster*

bug. The DFO is the population-based algorithm (Das and Singh 2014). This method consists of two main search paradigms as proximity source searching and quadratic approximation (QA). The first paradigm is mathematically formulated as below:

$$\begin{aligned}
&U_{i,k} = V_{i,k}\left|V_{r3,k} - V_{r4,k}\right| \\
&W_{i,k} = V_{i,k} + \left|V_{r3,k} - V_{r4,k}\right| \text{ for } k = r1 \text{ and } r2; \\
&\text{for } j \neq r1 \text{ and } j \neq r2,\ U_{i,k} = V_{i,j} \text{ and } W_{i,j} = V_{i,j} \\
&V_{i,j}^{'} = \text{Min}\{f\left(V_{i,j}\right), f\left(U_{i,j}\right), f\left(W_{i,j}\right)\} \text{ for } i = 1, 2, \ldots, P \text{ and } j = 1, 2, \ldots, D
\end{aligned} \tag{4.1}$$

in which, $i \in \{1, 2, \ldots, p\}$, and $j \in \{1, 2, \ldots, D\}$; U and W are two alternative vectors which are used to update the location of the current agent; D and p respectively indicate dimension of the problem and population size. $r1, r2$ are random scalars selected from interval $[1, D]$. Also, the current and updated location for the current agent is shown by $V_{i,k}$ and $V_{i,j}^{'}$, respectively.

The employed quadratic approximation search pattern in the DFO is mathematically formulated as follows:

$$\text{Child} = 0.5\frac{\left(R_2^2 - R_3^2\right)f(R_1) + \left(R_3^2 - R_1^2\right)f(R_2) + \left(R_1^2 - R_2^2\right)f(R_3)}{\left(R_2 - R_3\right)f(R_1) + \left(R_3 - R_1\right)f(R_2) + \left(R_1 - R_2\right)f(R_3)} \tag{4.2}$$

where $f(.)$ specifies the objective function value for any randomly selected individuals R_1, R_2 and R_3 in the colony, while $R_1 \neq R_2 \neq R_3$.

To provide more information, a pseudo-code for the DFO is given as follows (Table 4.1).

In the next section, the proposed improved Drosophila-food search algorithm (IDFO) method is described in detail.

Table 4.1 Pseudo-code for the DFO method

```
Set the algorithm adjusting factors
Calculate the objective value of each agent
while (termination condition are not met)
Apply tournament selection
  for (each agent)
    Do proximity search and renew the current agent's location using Eq. (4.1)
    Calculate the objective function of the agent f(Xi)
     The best position of the current agent is saved
     If the difference between updated objective function value and its previous value is among 1% radius, then
      employ quadratic search using Eq. (4.2)
    Hold the updated location of the current if it is better than its old value, else reject it
  end
end
```

4.3 Improved Drosophila Food-Search Algorithm (IDFO)

In the standard Drosophila food-search method (DFO), the algorithm makes a quadratic search around the three randomly selected agents of the colony. Such a random search may be useful in the very first iterations by providing a high level of exploration, but more exploitation is needed as the process progresses. To mitigate this problem, the quadratic search approach in the standard DFO is modified by replacing a random agent with a weighted agent. The weighted agent is mathematically defined as follow:

$$\begin{aligned}
&\boldsymbol{X}^{W} = \sum_{i=1}^{M} \overline{c}_i^{w} \boldsymbol{X}_i^{P} \\
&\overline{c}_i^{w} = \left(\hat{c}_i^{w} / \sum_{i=1}^{M} \hat{c}_i^{w} \right) \\
&\hat{c}_i^{w} = \frac{\max_{1\le k\le M}\left(f\left(\boldsymbol{X}_k^{P}\right)\right) - f\left(\boldsymbol{X}_i^{P}\right)}{\max_{1\le k\le M}\left(f\left(\boldsymbol{X}_k^{P}\right)\right) - \min_{1\le k\le M}\left(f\left(\boldsymbol{X}_k^{P}\right)\right) + \varepsilon}, \quad i = 1, 2, \ldots, M
\end{aligned} \tag{4.3}$$

where M is the population size; $\boldsymbol{X}^{W}$ is the proposed weighted agent; $\hat{c}_i^{w}$ is the impact factor coefficient of each agent; $f(.)$ is the objective function value; $\max_{1\le kw\le M}\left(f\left(\boldsymbol{X}_{kw}^{P}\right)\right)$ and $\min_{1\le kw\le M}\left(f\left(\boldsymbol{X}_{kw}^{P}\right)\right)$ are, respectively, the worst and best objective values of the population. The ε is a small positive number to avoid any probable division by zero. Thus, by substituting the R_1 with $\boldsymbol{X}^{w}$, quadratic search part of the proposed DFO is modified as follows:

$$\begin{aligned}
&R^{\text{new}} = 0.5 \\
&\frac{\left(R_2^2 - R_3^2\right) f\left(X^{\text{W}}\right) + \left(R_3^2 - \left(X^{\text{W}}\right)^2\right) f(R_2) + \left(\left(X^{\text{W}}\right)^2 - R_2^2\right) f(R_3)}{\left(R_2 - R_3\right) f\left(X^{\text{W}}\right) + \left(R_3 - X^{\text{W}}\right) f(R_2) + \left(X^{\text{W}} - R_2\right) f(R_3)}
\end{aligned} \tag{4.4}$$

4.4 Numerical Tests

In this section, the developed IDFO method is tested on handling of structural and mechanical optimization problems. The details about the problem such as constraints and objective function are given in the problem. For more clarity, the internal parameters settings for applied selected method (i.e., comparative methods) are given in flowing table (Table 4.2).

Table 4.2 Parameter adjusting for the utilized methods

Method	Pop. size	Parameter
DE (Storn and Price 1997)	20	$F \in [0, 2]$
FA (Yang 2009)	20	$\alpha = 0.5,\ \beta_{\min} = 0.2,\ \gamma = 1$
TLBO (Sadollah et al. 2012)	20	$TF \in [1, 2]$
iPSO (Mortazavi and Toğan 2016)	20	$\alpha = 0.4,\ C_3 = C_4 = 1,\ C_2 = 2$
DFO (Das and Singh 2014)	20	–
IDFO	20	–

4.5 Design of Welded Beam Problem

In this section, as demonstrated in Fig. 4.1, an optimal welding parameter for a moment-resistance connection is considered as a mechanical optimization problem with some constraints. The total cost of the welding mechanism should be minimized. The minimization process is constrained with seven different linear and non-linear limitations. This objective function and the corresponding constraints are given in Eq. 4.5. In this equation, the terms are defined as follows: normal stress (σ), shear stress (τ), and displacement (δ). Design variables of this problem are considered as the welding part's (shown in yellow), length (l), and height (h) and cross-sectional specifications as width (b) and thickness (t). The archived results for tested methods are tabulated in Table 4.3. According to the obtained results, the IDFO outperforms the other applied techniques in the term of stability; however, the iPSO requires the lowest number of objective function evaluation (OFEs).

To minimize

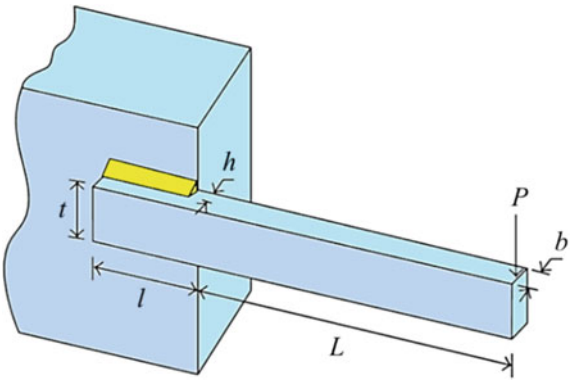


Fig. 4.1 Welded beam problem (Yang 2009)

Table 4.3 Optimal outcomes for the welded beam problem

Variables	DE (Storn and Price 1997)	FA (Yang 2009)	TLBO (Sadollah et al. 2012)	iPSO (Mortazavi and Toğan 2016)	DFO (Das and Singh 2014)	IDFO
x_1(h)	0.248729	0.202369	0.206428	0.205729	0.202369	0.205730
x_2(l)	2.95378	3.544214	3.461341	3.470493	3.544214	3.470489
x_3(t)	8.362973	9.048210	9.021457	9.036626	9.048210	9.036624
x_4(b)	0.249008	0.205723	0.206422	0.205729	0.205723	0.205730
$f(X)$						
Best	1.900415	1.728024	1.727353	1.724857	1.728024	1.724856
Mean	2.559287	1.748831	1.729874	1.724853	1.748831	1.824997
Std	1.983998	0.013	0.005898	6.9 E−019	0.024100	0.005569
OFEs	8865	47,850	28,650	1240	24,450	5680

$$\begin{aligned} &\text{cost}\,(\boldsymbol{X}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \\ &\boldsymbol{X} = \{x_1, x_2, x_3, x_4\} \end{aligned} \tag{4.5}$$

Subjected to

$$\begin{aligned} &g_1(x) = \tau(x) - \tau_{\max} \le 0 \\ &g_2(x) = \sigma(x) - \sigma_{\max} \le 0 \\ &g_3(x) = x_1 - x_4 \le 0 \\ &g_4(x) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \le 0 \\ &g_5(x) = 0.125 - x_1 \le 0 \\ &g_6(x) = \delta(x) - \delta_{\max} \le 0 \\ &g_7(x) = p - p_c(x) \le 0 \\ &0.1 \le x_1 \le 2 \quad 0.1 \le x_2 \le 10 \quad 0.1 \le x_3 \le 10 \quad 0.1 \le x_4 \le 2 \end{aligned} \tag{4.6}$$

where,

$$\begin{aligned} &\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2} \\ &\tau' = \frac{P}{\sqrt{2}x_1x_2} \quad \tau'' = \frac{MR}{J} \quad M = P\left(L + \frac{x_2}{2}\right) \\ &R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \end{aligned}$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$$P_c(x) = \frac{4.013\sqrt{E\left(x_2^3x_4^6/36\right)}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \tag{4.7}$$

4.6 Design of Tubular Column System

In this section, the tubular column presented in Fig. 4.2 is desired to be designed in minimum cost. The density, yielding stress, and elasticity modulus of the utilized material are $\rho = 0.0025\,\text{kgf/cm}^3$, $\sigma_y = 500\,\text{kgf/cm}^2$, and $E = 0.85 \times 10^6\,\text{kgf/cm}^2$, respectively. The column's length (l) is 250 cm, and it is carrying a compressive load as $P = 2500\,\text{kgf}$. There are two decision variables: the column diameter (d as $\times$ 1) and its wall thickness (t as $\times$ 2).

To minimize

$$f(\boldsymbol{X}) = 9.82x_1x_2 + 2x_1$$
$$\boldsymbol{X} = \{x_1, x_2\} \tag{4.8}$$

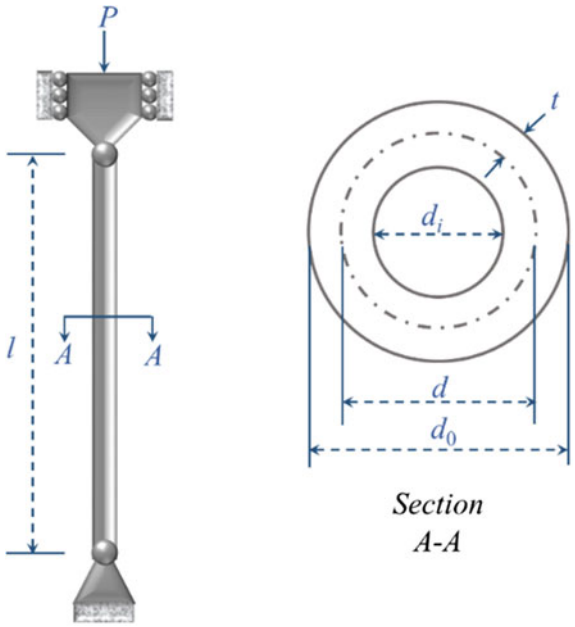


Fig. 4.2 Tubular column system (Yang 2009)

Table 4.4 Results for the tubular column design problem

Variables	DE (Storn and Price 1997)	FA (Yang 2009)	TLBO (Sadollah et al. 2012)	iPSO (Mortazavi and Toğan 2016)	DFO (Das and Singh 2014)	IDFO
x_1(d)	5.324786	5.451206	5.451156	5.451235	5.451241	5.451151
x_2(t)	0.313227	0.292091	0.291965	0.291961	0.294960	0.291963
$f(X)$						
Best	27.028023	26.538349	26.531328	26.531571	26.692041	26.531201
Mean	27.028210	26.658410	26.400236	26.531653	26.535960	26.531556
Std	2.31E−05	7.04E−06	3.05E−04	8.50E-06	4.26E−05	3.61E−06
OFEs	4400	11,950	7600	2600	10,050	3750

Subjected to

$$
\begin{aligned}
g_1(\boldsymbol{X}) &= \frac{P}{\pi x_1 x_2 \sigma_y} - 1 \le 0 \\
g_2(\boldsymbol{X}) &= \frac{8Pl^2}{\pi^3 E x_1 x_2 \left(x_1^2 + x_2^2\right)} - 1 \le 0 \\
g_3(\boldsymbol{X}) &= \frac{2.0}{x_1} - 1 \le 0 \\
g_4(\boldsymbol{X}) &= \frac{x_1}{14.0} - 1 \le 0 \\
g_5(\boldsymbol{X}) &= \frac{0.2}{x_2} - 1 \le 0 \\
g_6(\boldsymbol{X}) &= \frac{x_2}{0.8} - 1 \le 0 \\
2 \le x_1 \le 14,\ & 0.2 \le x_2 \le 0.8
\end{aligned}
\tag{4.9}
$$

Optimal results obtained using the selected methods are given in Table 4.4. According to the achieved outcomes, the IDFO algorithm puts forward more accurate values. Based on the reported statistical data (i.e., Std. values), the stable behavior of the IDFO algorithm is apparent. In addition, the number of objective function evaluations (OFEs) presents the speed of the approaches in finding the optimal result. The iPSO is more rapid than the other approaches.

4.6.1 Design of 72-Bar Truss Structure

In this section, a 72-bar truss system (given in Fig. 4.3) is applied as a structural optimization problem to evaluate the search capability of the proposed EQB technique.

The density and module of elasticity of the material of the structure are 0.1 lb/in^3 and 10,000 ksi, respectively. The structure's elements are gathered into 16 independent categories. For all nodes, allowable displacements in all directions are limited to ± 0.25 in. For all principal directions also, the stress of all compressive and tensile members is restricted up to 25 ksi. Node 17 of this structure is subjected to $P_x = 5$, $P_y = 5$ and $P_z = -5$ kips. Upper and lower bounds of sizing variables are restricted to 0.01 in^2 and 0.1 in^2, respectively. Achieved results for current example are given in Table 4.5. Based on these achievements, the standard DFO shows the costliest search performance (i.e., require 19,260 number of objective function evaluation). This is due to multiple structural analyses required in each iteration of the DFO method. However, considering the standard deviation (Std.) values reported for optimization processes, the IDFO shows the most stable search behavior for all runs (Table 4.5).

4.7 Non-Parametric Statistical Tests

In this to check the significance of the obtained results, a Friedman rank test is performed on the attained outcomes. The achieved results are tabulated in Table 4.6. Based on these results, the IDFO in comparison with other methods, specially in comparison with standard DFO, represents an advanced performance from both accuracy and stability aspects.

4.8 Conclusions

This chapter deals with developing a new search methos so-called improved Drosophila food-search (IDFO) and assessing its search capability on handling the structural and mechanical minimization problems (which are categorized as the constrained engineering optimization problem). The results are compared with five other well-stablished metaheuristic algorithms as differential evolution (DE) method, firefly algorithm (FA), teaching and learning-based optimization (TLBO), Drosophila food-search optimization (DFO), and integrated particle swarm optimization (iPSO). In IDFO algorithm, the totally random behavior of the standard DFO is modified process by contributing a new agent so-called weighted agent into the search process of the algorithm.

It is notable that the costliest part of a complex engineering (e.g., structural, and mechanical) optimization problem is spent to evaluate the objective function of the problem, so the reaching to the optimal state with the lower number of objective function evaluation (OFEs) is one of the key factors that should be considered in selecting proper optimization technique for handling this class of optimization problems. On the other hand, the stability is also one of the important specifications that is required for a robust search technique.

In this regard, reported standard deviation values reveal that the IDFO shows the most stable search behavior among all other selected methods. Attained outcomes

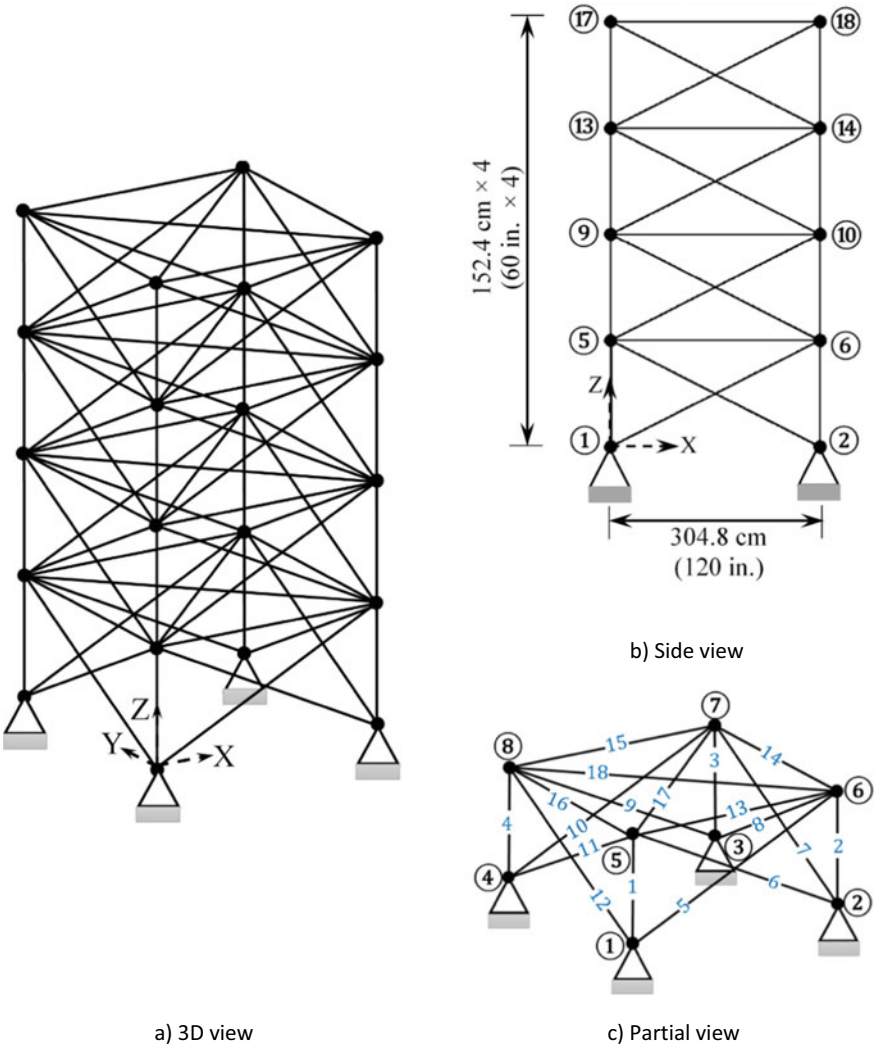


Fig. 4.3 72-bar truss structure (Yang 2009)

shows that required OFEs for the proposed IDFO are less than the other selected methods, especially in comparison with the standard DFO method. This indicates that additional agents can work properly in increasing the performance of the optimization method. The accuracy of the obtained solutions demonstrates that the search performance of the proposed method is also improved. Furthermore, the acquired statistical results prove the significancy of the obtained solutions by proposed IDFO method. Consequently, based on the acquired results, the IDFO method can be applied as a robust method for solving the constrained engineering optimization problems.

Table 4.5 The optimal outcomes for the 72-bar truss structure

Design variables	Optimal cross–sectional area (in^2)					
	DE (Storn and Price 1997)	FA (Yang and 2009)	TLBO (Sadollah et al. 2012)	iPSO (Mortazavi and Toğan 2016)	DFO (Das and Singh 2014)	IDFO
A_1–A_4	1.9	1.9	1.9	1.9	1.9	1.9
A_5–A_{12}	0.5	0.5	0.5	0.5	0.5	0.5
A_{13}–A_{16}	0.1	0.1	0.1	0.1	0.1	0.1
A_{17}–A_{18}	0.1	0.1	0.1	0.1	0.1	0.1
A_{19}–A_{22}	1.2	1.2	1.3	1.3	1.3	1.3
A_{23}–A_{30}	0.5	0.5	0.5	0.5	0.5	0.5
A_{31}–A_{34}	0.1	0.1	0.1	0.1	0.1	0.1
A_{35}–A_{36}	0.1	0.1	0.1	0.1	0.1	0.1
A_{37}–A_{40}	0.5	0.5	0.5	0.5	0.5	0.5
A_{41}–A_{48}	0.5	0.5	0.5	0.5	0.5	0.5
A_{49}–A_{52}	0.1	0.1	0.1	0.1	0.1	0.1
A_{53}–A_{54}	0.1	0.1	0.1	0.1	0.1	0.1
A_{55}–A_{58}	0.2	0.2	0.2	0.2	0.2	0.2
A_{59}–A_{66}	0.6	0.6	0.5	0.5	0.5	0.5
A_{67}–A_{70}	0.4	0.4	0.4	0.4	0.4	0.4
A_{71}–A_{72}	0.6	0.6	0.6	0.6	0.6	0.6
Best weight (lb)	379.9	379.9	379.7	379.7	379.7	379.7
Mean weight (lb)	382.2	383.6	381.7	381.7	381.8	381.9
Std. (lb)	1.9	2.8	1.4	1.3	2.2	1.3
OFEs	13,200	22,200	17,220	10,200	17,220	10,520

Table 4.6 Friedman rank test for mean and Std. values for engineering problems

	Test for optimal mean value			Test for optimal Std. value		
Method	Friedman value	Normalized value	Rank	Friedman value	Normalized value	Rank
DE (Storn and Price 1997)	17.5	0.257143	6	14	0.321429	5
FA (Yang 2009)	14	0.321429	5	12	0.375000	3.5
TLBO (Sadollah et al. 2012)	7.5	0.600000	2.5	12	0.375000	3.5
iPSO (Mortazavi and Toğan 2016)	7.5	0.600000	2.5	5.5	0.818182	2
DFO (Das and Singh 2014)	12	0.375000	4	15	0.300000	6
IDFO	4.5	1.000000	1	4.5	1.000000	1

References

Das KN, Singh TK (2014) Drosophila food-search optimization. Appl Math Comput 231:566–580

Degertekin SO, Lamberti L, Ugur IB (2018) Sizing, layout and topology design optimization of truss structures using the Jaya algorithm. Appl Soft Comput 70:903–928

Kale IR, Kulkarni AJ (2018) Cohort intelligence algorithm for discrete and mixed variable engineering problems. Int J Parallel Emergent Distrib Syst 33(6):627–662

Kale IR, Kulkarni AJ (2021) Cohort intelligence with self-adaptive penalty function approach hybridized with colliding bodies optimization algorithm for discrete and mixed variable constrained problems. Complex and Intell Syst 7(3):1565–1596

Kazikova A, Pluhacek M, Senkerik R (2021) How does the number of objective function evaluations impact our understanding of metaheuristics behavior? IEEE Access 1–1

Mei L, Wang Q (2021) Structural optimization in civil engineering: a literature review. Buildings 11(2):66

Mortazavi A (2019a) Comparative assessment of five metaheuristic methods on distinct problems. Dicle Univer J Eng 10(3):879

Mortazavi A (2019b) The performance comparison of three metaheuristic algorithms on the size, layout and topology optimization of truss structures. Mugla J Sci Technol 5(2):28–41

Mortazavi A, Moloodpoor M (2021a) Enhanced butterfly optimization algorithm with a new fuzzy regulator strategy and virtual butterfly concept. Knowl-Based Syst 228:107291

Mortazavi A, Moloodpoor M (2021b) Differential evolution method integrated with a fuzzy decision-making mechanism and virtual mutant agent: theory and application. Appl Soft Comput 112:107808

Mortazavi A, Toğan V (2016) Simultaneous size, shape, and topology optimization of truss structures using integrated particle swarm optimizer. Struct Multidiscip Optim 54(4):715–736

Mortazavi A (2020a) Size and layout optimization of truss structures with dynamic constraints using the interactive fuzzy search algorithm. Eng Optim 1–23

Mortazavi, A., Large-scale structural optimization using a fuzzy reinforced swarm intelligence algorithm. Adv Eng Softw 142:102790

Mortazavi A (2021) Bayesian interactive search algorithm: a new probabilistic swarm intelligence tested on mathematical and structural optimization problems. Adv Eng Softw 155:102994

Mortazavi A (2022) Interactive fuzzy Bayesian search algorithm: a new reinforced swarm intelligence tested on engineering and mathematical optimization problems. Expert Syst Appl 115954

Sadollah A et al (2012) Mine blast algorithm for optimization of truss structures with discrete variables. Comput Struct 102:49–63

Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11(4):341–359

Yang X-S (2009) Firefly algorithms for multimodal optimization. In: Stochastic algorithms: foundations and applications. Berlin, Heidelberg, Springer

Chapter 5
Truss Structure Optimization Using Constrained Version of Variations of Cohort Intelligence

Ishaan R. Kale, Ayush Khedkar, and Mandar S. Sapre

Abstract An especially significant class of structurally constrained optimization problems is truss design. This study presents a constrained version of two variations of the Cohort Intelligence (CI) algorithm. In this work, discrete variable truss structures with six bars and two cases with ten bars are studied using follow-best and follow-better approaches, as well as the self-adaptive penalty function (SAPF). These problems are associated with two linear constraints: tensile/compressive stress and deflection. Algorithm efficiency is evaluated by counting the function evaluations, computing CPU time, and determining the total weight of the truss structure. Compared to follow-better and other contemporary optimizers from literature, follow-best performs significantly better.

Keywords Self-adaptive penalty function · Discrete variables · Variations of CI · Design of trusses

5.1 Introduction

Truss structure problems are structural constrained optimization problems consisting of continuous, discrete, or mixed variables. The constraints are usually nonlinear in nature. There have been several techniques inspired by nature to solve truss structures problems. Genetic Algorithm (GA), Firefly Algorithm (FA) (Gandomi et al. 2011), Particle Swarm Optimization (PSO) (Li et al. 2009), and Artificial Bee Colony (ABC) (Sonmez 2011) are few optimizers from literature applied in this domain.

I. R. Kale (✉)
Institute of Artificial Intelligence, Dr Vishwanath Karad MIT World Peace University, Pune, MH 411038, India
e-mail: ishaan.kale@mitwpu.edu.in

A. Khedkar · M. S. Sapre
Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, MH 412 115, India
e-mail: mandar.sapre@sitpune.edu.in

I. R. Kale and A. Sadollah (eds.), *Optimization Methods for Structural Engineering*, Engineering Optimization: Methods and Applications,
https://doi.org/10.1007/978-981-99-2378-6_5

In an earlier study, Kale and Kulkarni (2018) used Cohort Intelligence (CI) with a static penalty function (SPF). Some limitations were observed when CI-SPF was used for solving the constrained problems. SPF approach is associated with a penalty parameter which needed to be set for every problem. To set the appropriate penalty parameter, certain preliminary trials are required. This may increase the initial computational cost. Self-adaptive penalty functions (SAPFs) have been proposed as a solution to overcome these limitations (Kale and Kulkarni 2021). SAPF-based constraint handling with CI algorithm facilitates the solution of constrained problems involving variables of discrete, continuous, and mixed nature. Furthermore, the hybrid CI-SAPF-CBO, refined the results.

Patankar and Kulkarni (2018) developed seven variations of CI. These were applied to mesh smoothing of complex objects (Sapre et al. 2019) and for optimizing the abrasive water jet machining process (Gulia and Nargundkar 2019). Two variations of CI are applied in this paper to solve three test problems from the truss structural domain, namely a six-bar test problem and two ten-bar test problems. These are the follow-best and follow-better approaches. For constrained problems, other rules such as roulette, alienation and random selection, follow-worst, and follow-itself are not effective. Round-off integer sampling is used to handle discrete variables, and SAPF is used to handle constrained variables. The results obtained from follow-best and follow-better approaches are compared with those from GA, CI-SAPF, CI-SAPF-CBO, ABC, Adaptive Dimensional Search Algorithm (ADS), and Probability Collectives (PC).

The work is organized as follows: The mechanism of follow-best and follow-better approach using CI-SAPF is explained in Sect. 5.2. The solution to the truss structure problems follows next. In Sect. 5.4, the results are analyzed and discussed in details. The last section represents conclusion and future directions.

5.2 Mechanism of Follow-Best and Follow-Better Approach with SAPF

In follow-best approach, the candidate follows other candidates in the cohort situated at the best behavior. This assists the individuals to learn faster and achieve the cohort goal within less computational efforts. In follow-better approach, the candidate follows subsequent candidate exhibiting a better behavior than itself. The pseudo-code of variations of CI using follow-best and follow-better mechanism incorporated with SAPF approach (Kale and Kulkarni 2021) is presented in Fig. 5.1.

Let

K Candidate Count in the cohort $(k = 1, \ldots, K)$

S Sampling space

r Sampling space reduction factor

P Penalty function

g A violation of a constraint measure

Initialize K, S, r

While

1. Generate the random solutions $\mathbf{X}$ using uniform distribution method.
2. Determine the value of the function $f(\mathbf{X}^k)$ for each candidate.
3. Use SAPF approach to evaluate a pseudo-objective function / behavior:
   $\Phi(\mathbf{X}^k) = f(\mathbf{X}^k) + P(\mathbf{X}^k)$
   where $P(\mathbf{X}^k) = f(\mathbf{X}^k) \times \sum_{k=1}^{K} g(\mathbf{X}^k)$
4. Determine the probability p^k of each candidate k in the cohort as: $p^k = \frac{1/\Phi^*(\mathbf{X}^k)}{\sum_{k=1}^{K}[1/\Phi^*(\mathbf{X}^k)]}$
5. (i) Using follow-best approach every candidate k follows the behavior with highest probability.
   (ii) Using the follow-better strategy, all candidates with a lower probability of selection than the candidate itself are discarded, and the remaining candidates are selected at random. As a result, every candidate follows themselves or a candidate who is better than themselves.
6. The sampling interval $\boldsymbol{S^c}$ in the neighborhood of each candidate, shrinks/ expands using sampling space reduction parameter r:
   $\left[\boldsymbol{S^{k,lower}}, \boldsymbol{S^{k,upper}}\right] = \left[\mathbf{X}^k - \left\|\frac{S^{upper} - S^{lower}}{2}\right\| \times r,\ \mathbf{X}^k + \left\|\frac{S^{upper} - S^{lower}}{2}\right\| \times r\right]$
7. **If**: Function value $\Phi^*(\mathbf{X}^k)$does not significantly improve, there is a state of saturation in the solution.
   Every candidate k expands/ shrink iteratively the sampling interval $\boldsymbol{S^k}$ to its original interval $\boldsymbol{S}$
   Accept $\Phi(\mathbf{X})$ as current behavior of cohort with the associated attributes $\mathbf{X}$.
   Else
   Go to Step 3
   End If

End While

Fig. 5.1 Pseudocode of variations of CI using follow-best and follow-better rule

5.3 Truss Structure Test Problems

This work is investigation of application of constrained version of variations of CI with SAPF approach in truss design. The six-bar and ten-bar examples were solved in the literature using GA (Nanakorn and Meesomklin 2001), CI-SAPD, CI-SAPF-CBO (Kale and Kulkarni 2021), ABC (Sonmez 2011), ADS (Hasançebi and Azad 2015), PC (Kulkarni et al. 2016). The mathematical formulation is shown in Eq. (5.1) as follows:

$$\begin{aligned} \text{Minimize} \quad & W = \sum_{i=1}^{N} \rho A_i l_i \\ \text{subject to} \quad & |\sigma_i| \leq \sigma_{\max} i = 1, 2 \ldots N \\ & |u_i| \leq u_{\max} j = 1, 2 \ldots M \end{aligned} \tag{5.1}$$

where

W Objective function (Weight)

A_i Design variables—Cross section area of ith truss member where, $i = 1, 2, \ldots, N$
ρ Material density
l_i Length of each truss member $i, i = 1, 2, \ldots, N$
$\sigma_{\max}$ Maximum allowable stress.
$u_{\max}$ Maximum allowable displacement.

Weight reduction of the truss structure is the goal with the maximum allowable tensile and compressive stresses at every node, as well as maximum allowable displacements as the limitations. There are as many variables as members in a truss problem. So, a six-bar truss has six variables. Each link of these trusses is a separate entity. In both the cases, distinct discrete set is utilized for the selection of variables.

Test Problem-1: Six-Bar Truss Structure

The six-bar truss structure (refer to Fig. 5.2) problem was formerly discussed by Nanakorn and Meesomklin (2001) Kale and Kulkarni (2021). There are six design variables (cross-sectional area) equal to number of truss members. Here, $A_i \in \{1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.50, 13.50, 13.90, 14.20, 15.50, 16.00, 16.90, 18.80, 19.90, 22.00, 22.90, 26.50, 30.00, 33.50\}$ in^2. The allowable stress is given as 25,000 psi, and allowable deflection is given as 2 in. The weight density of the material is $0.1\,\text{lb/in}^3$, and the modulus of elasticity is 10^7 psi.

Test Problem-2: Ten-Bar Truss Structure

The next example is shown in Fig. 5.3 and was previously discussed in (Nanakorn and Meesomklin 2001, Li et al. 2009; Sonmez 2011; Hasançebi et al. 2015). A ten-bar truss structure made of aluminum 2024-T3 is used in the analysis. The material density ρ is $0.1\,\text{lb/in}^3$, and the modulus of elasticity E is $10,000$ ksi. As shown in Fig. 5.3, a represents the longest length of the truss member. The maximum allowable tensile and compressive stresses $\sigma_{\max}$ on every member i are $\pm\,25$ ksi. The maximum allowable horizontal and vertical displacement $u_{\max}$ at every node are

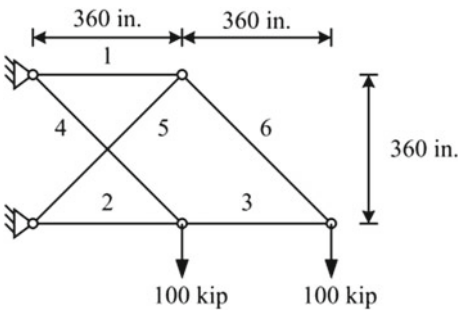


Fig. 5.2 Planar six-bar truss structure

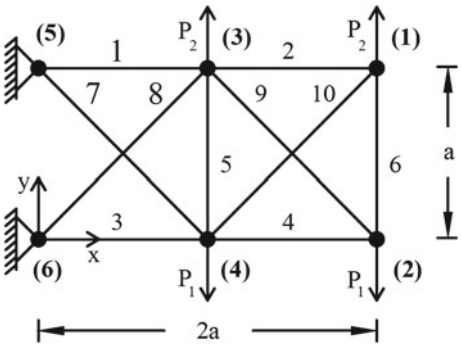


Fig. 5.3 Planar ten-bar truss structure, $a = 360$ in

± 2 in. The applied forces are $P_1 = 100$ kips and $P_2 = 0$. This problem involves ten design variables and two sub-cases.

Case 1: $A_i \in \{1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.50, 13.50, 13.90, 14.20, 15.50, 16.00, 16.90, 18.80, 19.90, 22.00, 22.90, 26.50, 30.00, 33.50\}\text{in}^2$.

Case 2: $A_i \in \{0.1, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.0, 10.5, 11.0, 11.5, 12.0, 12.5, 13.0, 13.5, 14.0, 14.5, 15.0, 15.5, 16.0, 16.5, 17.0, 17.5, 18.0, 18.5, 19.0, 19.5, 20.0, 20.5, 21.0, 21.5, 22.0, 22.5, 23.0, 23.5, 24.0, 24.5, 25.0, 25.5, 26.0, 26.5, 27.0, 27.5, 28.0, 28.5, 29.0, 29.5, 30.0, 30.5, 31.0, 31.5\}\ \text{in}^2$.

5.4 Results and Discussion

The use of follow-best and follow-better approaches for discrete variable problems is pioneered for the first time ever for truss structural problems. CI's follow-best version is much more efficient than other algorithms. In CI-SAPF and CI-SAPF-CBO, the candidate's follow other candidate in a cohort probabilistically due to which there was a possibility of following even the worst behavior of the candidate. This may require a greater number of learning attempts (iterations) for the convergence. In CI-SAPF, the performance of this approach is dependent on roulette wheel approach as well as the value of r. However, in these proposed approaches, solution value is driven by setting a suitable value of r (Kale and Kulakrni 2018). This model incorporates SAPF to handle linear constraints associated with test problems.

The comparison is shown in Table 5.1. The standard deviation using the follow-best approach is 9.7666, average function evaluation count is 615, while the average CPU time is 0.64 sec. In terms of function evaluations and computational time, follow-best approach has shown much better performance in comparison with follow-better, CI-SAPF, CI-SAPF-CBO, and GA. The convergence trend can be observed from Figs. 5.4 and 5.5, respectively.

Table 5.1 Comparative analysis of optimizers for six-bar truss structures

Design variables (in^2)	GA (Nanakorn and Meesomklin 2001)	CI-SAPF (Kale and Kulkarni 2021)	CI-SAPF-CBO (Kale and Kulkarni 2021)	Follow-best	Follow-better
A_1	30	30	30	30	30
A_2	19.9	19.9	19.9	19.9	19.9
A_3	15.5	15.5	15.5	15.5	15.5
A_4	7.22	7.22	7.22	7.22	7.22
A_5	22	22	22	22	22
A_6	22	22	22	22	22
Truss weight W(lb)	4962.0966	4962.0966	4962.0966	4962.0966	4962.0966
Function evaluations	–	2250	1740	615	1865
Time	–	–	–	0.64	0.26

NA Not Available

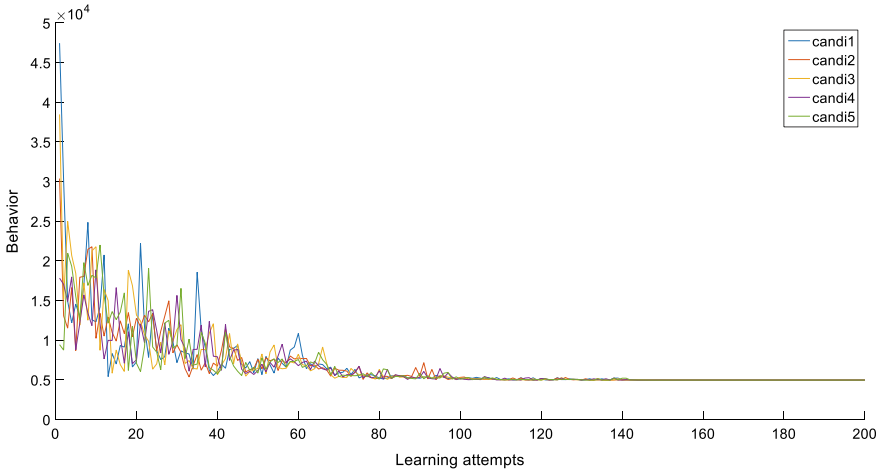


Fig. 5.4 Convergence trend of follow-best for solving six-bar truss problem

As compared to ABC (Sonmez 2011) and ADS (Hasançebi and Azad 2015) algorithms, the follow-best approach successfully solved Case 1 with a very small computational effort (refer to Table 5.2). The average count of function evaluations is 1855, standard deviation is 54.2289, average computational time required is 6.21 s. The function evaluations are very less as compared to other compared algorithms except ADS. This results to lower down CPU time as well. On the other hand, the

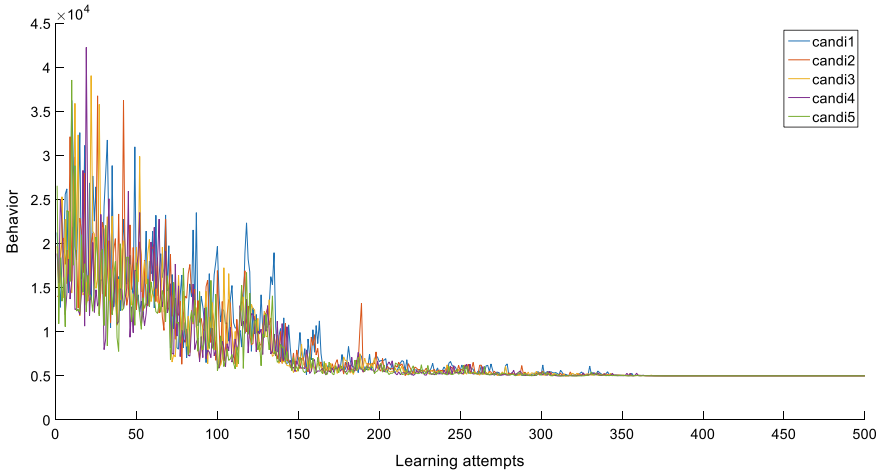


Fig. 5.5 Convergence trend of follow-better for solving six-bar truss problem

follow-better approach failed to obtain comparable solution. The convergence trend can be observed from Figs. 5.6 and 5.7, respectively.

It has been shown in Table 5.3 that the follow-best method of CI results is superior to PSO, PSOPC, and HPSO in solving case 2 (Li et al. 2009), marginally worse than CI-SAPF and CI-SAPF-CBO (Kale and Kulkarni, 2021) algorithms, and completely worse than PC (Kulkarni et al. 2016). The standard deviation with the follow-best approach was 21.5726, average function evaluation count is 2070, and average CPU time was 6.62 sec. The convergence trends for ten-bar Case 2 can be observed from Figs. 5.8 and 5.9, respectively.

5.5 Conclusions and Future Directions

Follow-best and follow-better versions of CI are successfully applied and validated for solving discrete variable truss structures with linear constraints in two cases of 6 bars and two cases of 10 bars. An integer sampling approach is used to handle discrete variables. In contrast, SAPF is used to manage the constraints associated with the problems. It must be noted that the CI variations doesn't require any preliminary trials as SAPF approach is self-supervised. The sampling space reduction factor is one of the solution driving factors; however, it is pre-defined within the range [0.95, 0.98] for these problems. The follow-best approach has obtained better results than follow-better approach due to the higher probability of following a good candidate/behavior from the set-in follow-best approach. There is a scope of following a worse solution in follow-better approach. We intend to apply this approach for complex 3-D spatial truss structure problems. The follow-best mechanism with SAPF approach could be

Table 5.2 Comparative analysis of optimizers for ten-bar case 1 truss structure

Design variables (in^2)	GA (Nanakorn and Meesomklin 2001)	ABC (Sonmez 2011)	ADS Hasançebi and Azad 2015	PC (Kulkarni et al. 2016a)	CI-SPF (Kale and Kulkarni 2018)	CI-SAPF (Kale and Kulkarni 2021)	CI-SAPF-CBO (Kale and Kulkarni 2021)	Follow-best	Follow-better
A_1	33.5	33.5	33.5	33.5	33.5	33.5	33.5	33.5	33.5
A_2	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62
A_3	22.9	22.9	22.9	22.9	22.9	22.9	22.9	22.9	22.9
A_4	15.5	14.2	14.2	14.2	14.2	13.9	13.9	13.9	15.5
A_5	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62
A_6	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62
A_7	7.22	7.97	7.97	7.97	7.97	7.97	7.97	7.97	7.22
A_8	22.9	22.9	22.9	22.9	22.9	22.9	22.9	22.9	22.9
A_9	22	22	22	22	22	22	22	22	22
A_{10}	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62
Truss weight W(lb)	5499.3000	5490.7400	5490.7400	5490.7378	5490.7378	5490.6020	5490.6020	5490.6021	5499.3258
Function evaluations	–	25,800	1000	1,852,059	19,250	16,940	14,160	1855	1900
Time	–	–	–	–	–	–	–	6.21	7.86

[NA] Not Available

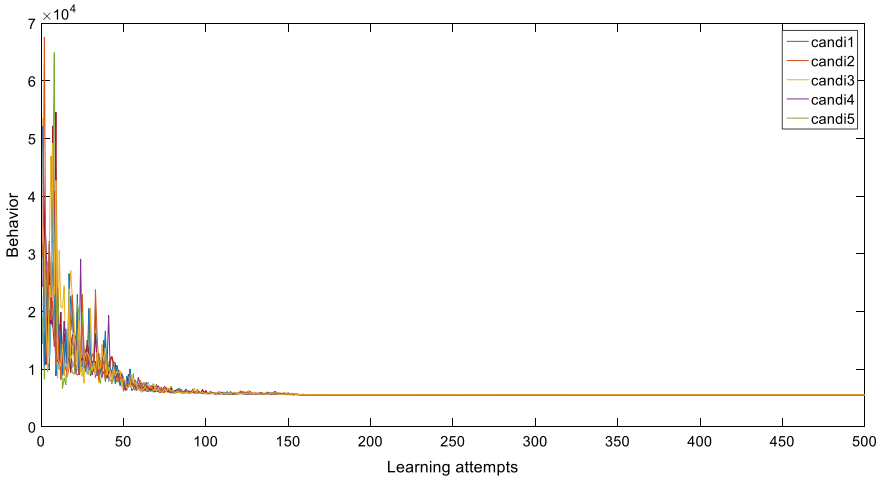


Fig. 5.6 Convergence trend of follow-best for case 1 of ten-bar truss problem

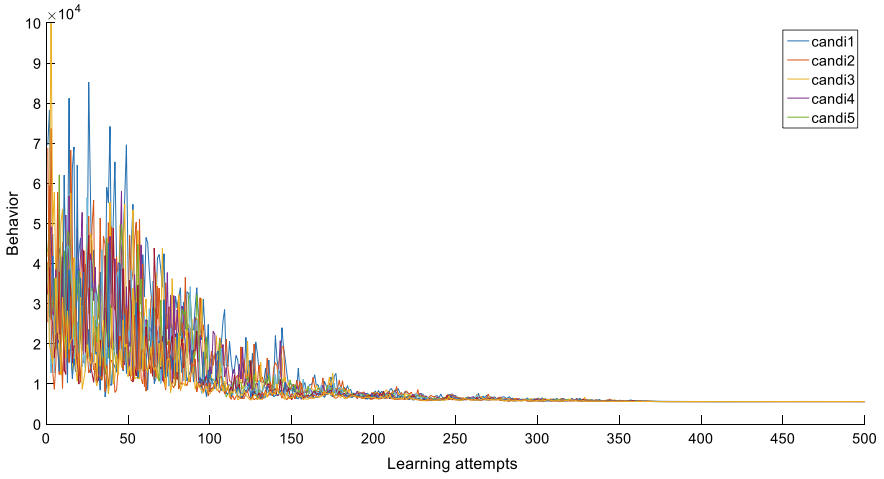


Fig. 5.7 Convergence trend of follow-better for case 1 of ten-bar truss problem

used to solve the scheduling and transportation problems as well as mixed variable design engineering problems.

Table 5.3 Comparative analysis of optimizers for ten-bar case 2 truss Structure

Design variables (in^2)	PSO (Li et al. 2009)	PSOPC (Li et al. 2009)	HPSO (Li et al. 2009)	MBA (Sadollah et al. 2012)	PC (Kulkarni et al. 2016a)	CI-SAPF (Kale and Kulkarni 2021)	CI-SAPF-CBO (Kale and Kulkarni 2021)	Follow-best	Follow-better
A_1	24.5	31.5	31.5	29.5	23.5	31	31	30.5	29.5
A_2	0.1	0.1	0.1	0.01	0.1	0.1	0.1	0.1	0.1
A_3	22.5	23.5	24.5	24	26	23	23	23	24.5
A_4	15.5	18.5	15.5	15	14	15	15	15.5	14
A_5	0.1	0.1	0.5	0.01	0.1	0.1	0.1	0.1	0.1
A_6	1.5	0.5	0.5	0.05	2	0.5	0.5	0.5	1
A_7	8.5	7.5	7.5	7.5	12.5	7.5	7.5	7.5	8
A_8	21.5	21.5	20.5	21.5	13	21	21	21	21.5
A_9	27.5	23.5	20.5	21.5	2	21.5	21.5	21.5	21.5
A_{10}	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Truss weight W(lb)	5243.7100	5133.1600	5073.5100	5067.3300	4686.7729	5061.7597	5061.7597	5062.5325	5096.7936
Function evaluations	–	–	–	–	2,363,380	9450	8400	2070	2100
CPU time	–	–	–	–	–	–	–	6.62	10.45

[NA] Not Available

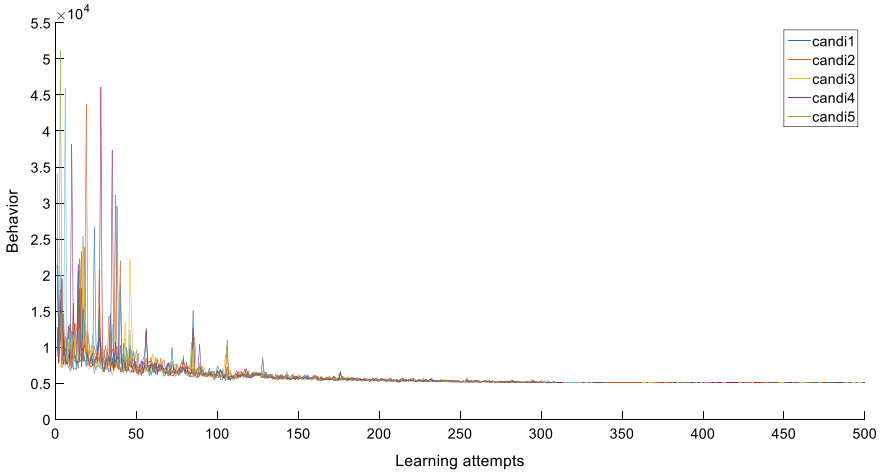


Fig. 5.8 Convergence trend of follow-best for case 2 ten-bar truss problem

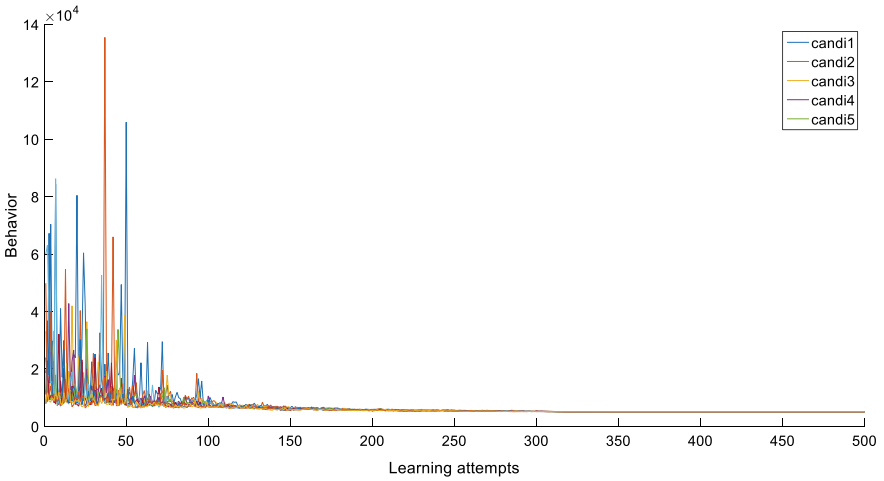


Fig. 5.9 Convergence trend of follow-better for ten-bar case 2 truss problem

References

Gandomi AH, Yang X-S, Alavi AH (2011) Mixed variable structural optimization using firefly algorithm. Comput Struct 89(23–24):2325–2336

Gulia V, Nargundkar A (2019) Optimization of process parameters of abrasive water jet machining using variations of cohort intelligence (CI). In: Applications of artificial intelligence techniques in engineering, Springer, Singapore, pp 467—474

Hasançebi O, Azad SK (2015) Adaptive dimensional search: a new metaheuristic algorithm for discrete truss sizing optimization. Comput Struct 154:1–16

Kale IR, Kulkarni AJ (2018) Cohort intelligence algorithm for discrete and mixed variable engineering problems. Int J Parallel Emergent Distrib Syst 33(6):627–662

Kale IR, Kulkarni AJ (2021) Cohort intelligence with self-adaptive penalty function approach hybridized with colliding bodies optimization algorithm for discrete and mixed variable constrained problems. Complex Intell Syst 7(3):1565–1596

Kulkarni AJ, Kale IR, Tai K (2016) Probability collectives for solving discrete and mixed variable problems. Int J Comput Aided Eng Technol 8(4):325–361

Lemonge ACC, Barbosa HJC (2004) An adaptive penalty scheme for genetic algorithms in structural optimization. Int J Numer Methods in Eng 59(5):703–736

Li LJ, Huang ZB, Liu FA (2009) Heuristic particle swarm optimization method for truss structures with discrete variables. Comput Struct 87(7–8):435–443

Nanakorn P, Meesomklin K (2001) An adaptive penalty function in genetic algorithms for structural design optimization. Comput Struct 79:2527–2539

Patankar NS, Kulkarni AJ (2018) Variations of cohort intelligence. Soft Comput 22(6):1731–1747

Rajeev S, Krishnamoorthy CS (1992) Discrete optimization of structures using genetic algorithm. J Struct Eng 118(5):1123–1250

Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2012) Mine blast algorithm for optimization of truss structures with discrete variables. Comput Struct 102:49–63

Sapre MS, Kulkarni AJ, Shinde SS (2019) Finite element mesh smoothing using cohort intelligence. In: Proceedings of the 2nd international conference on data engineering and communication technology, Springer, Singapore, pp 469–480

Sonmez M (2011) Artificial Bee Colony algorithm for optimization of truss structures. Appl Soft Comput 11(2):2406–2418

Wu SJ, Chow PT (1995) Steady-state genetic algorithms for discrete optimization of trusses. Comput Struct 56(6):979–991

Chapter 6
Hybridization of Cohort Intelligence and Fuzzy Logic (CIFL) for Truss Structure Problems

Saif Patel, Ishaan R. Kale, and Anand J. Kulkarni

Abstract Several nature-based optimization methods have been developed by the researchers to solve the real-world problems. There are certain characteristics of the inherent approaches associated with the algorithms which could be combined with other algorithm to enhance the exploration and exploitation quality of the algorithm. Cohort Intelligence (CI) is one of the socio-inspired optimization algorithms which is inspired from self-supervised learning of the candidates in a cohort. To further increase the performance of CI, it is hybridized with fuzzy logic (FL). FL is an approach that allows multiple possible truth values to be processed through variables. FL was used to solve problems with an open, imprecise data, and heuristics that make it possible to obtain accurate results. In this current work, a new combination of CI and FL named as CIFL is introduced for solving truss structure optimization problem. The validity of the algorithm is verified using two cases of three-bar truss design optimization problem. CIFL is applied to both discrete and continuous variable-constrained problems. The self-adaptive penalty function (SAPF) approach is used to handle the constraints. The results obtained from CIFL are compared with other nature-inspired optimization techniques and discussed in details.

Keywords Cohort intelligence · Self-adaptive penalty function · Fuzzy logic · Membership function · Truss structure

6.1 Introduction

Truss structure is one of the important real-world applications which typically used in bridges, towers, roofs, buildings, domes, various industrial sectors, etc. It is the most common lightweight structure used in practices. These systems are designed to meet

S. Patel · I. R. Kale (✉) · A. J. Kulkarni
Institute of Artificial Intelligence, Dr Vishwanath Karad MIT World Peace University, Pune 411038, India
e-mail: ishaan.kale@mitwpu.edu.in

A. J. Kulkarni
e-mail: anand.j.kulkarni@mitwpu.edu.in

I. R. Kale and A. Sadollah (eds.), *Optimization Methods for Structural Engineering*, Engineering Optimization: Methods and Applications,
https://doi.org/10.1007/978-981-99-2378-6_6

optimality criteria with the lowest weight, i.e., cost-effective with maximum reliability. These truss structures are associated with number of members which sustain load acting on the structure. In order to withstand for the long life, the mechanical conditions such as deflection of nodes and stress exerted in the member must be satisfied. In these truss structure problems, the number members are equal to number of design variables. The variables are of continuous and discrete types. Increasing the number of members, complexity of the problems increases which may not be handled using tradition gradient-based optimization techniques. There are several heuristic and metaheuristic techniques that have been proposed by the researchers and applied to solve these problems. Those are Bat Algorithm (BA) Yang and Gandomi (2012), Cuckoo Search Algorithm (CSA) (Gandomi et al. 2013), Mine Blast Algorithm (MBA) (Sadollah et al. 2013), Particle Swarm Optimization (PSO) (Li et al. 2009), Probability Collectives (PC) (Kulkarni et al. 2016), Cohort Intelligence (Kale and Kulkarni 2018, 2021). Furthermore, several hybrid optimization techniques have also been proposed and applied to solve these truss structure problems such as Particle Swarm Optimization and Genetic Algorithm (PSOGA) (Omidinasab and Goodarzimehr 2020), Cohort Intelligence with self-adaptive penalty function and Colliding Bodies Optimization (CI-SAPF-CBO) (Kale and Kulakrni 2021). In this work, the concept of fuzzy logic is incorporated in CI algorithm to solve the truss structure domain problems.

The concept of fuzzy is made from the things that are uncertain in nature. The theory of *absolute true* and *absolute false* doesn't exist in fuzzy logic. In last three decades, fuzzy set and fuzzy logic theory have been evolving and have been used in multiple engineering and natural socioeconomics sciences. Fuzzy logic model can replicate human way of thinking in complex situation and that's why it can be used as a tool examining natural complexity. Moreover, fuzzy logic can be exploited to predict chaotic behaviors. A fuzzy logic integrated with Genetic Programming (GP) is proposed by (Soh and Yang 2000) to increase the performance of the GP-based approach for structural optimization. Fuzzy set theory is employed to deal with imprecise and vague information, during structural design process. In Fuzzy Tuned Interactive Search Algorithm (FTISA) by (Mortazavi 2019) proposed mechanism evaluates agents via two predefined concepts named as Normalized Objective Function (NOF_i) and Normalized Members Density (NMD_i). The defined nine-rule fuzzy mechanism takes these values as input parameters and through fuzzification–defuzzification process returns a Topology/Size (TS) regulator value for each agent. The hybrid fuzzy genetic system for optimizing cabled-truss structures (Finotto et al. 2013) demonstrates an application of a hybrid fuzzy genetic system in the optimized lightweight structure is determined through a stochastic discrete topology and sizing optimization procedure that uses ground structure approach, nonlinear finite element analysis, genetic algorithm, and fuzzy logic. When performing optimization, the increase of the ground structure discretization led to a sharp increase of the search space. In addition, an increase in the number of evaluations of the FE model was also observed. This is because iterative procedures become part of the optimization problem when cable elements are used. For this reason, the effectiveness of the GA

can be compromised since a relatively high number of evaluations may lead to a prohibitive computational cost.

In this work, an intrinsic property of cohort intelligence and fuzzy logic is combined together to design the new hybridized version named as CIFL. In the last two decades, fuzzy theory has been applied to structural optimization. Their results show that use of fuzzy set can mitigate the shortcomings of the aforementioned approaches, the current study deals with putting forward a new approach that takes into account the search space of size and topology optimization using fuzzy logic. This strategy does not only remove randomization but also decreases the convergence rate. The proposed CIFL is examined on two cases of three-bar truss structural test problems. These problems are having continuous as well as discrete variable with nonlinear constraints. To handle the constraints a self-adaptive penalty function (SAPF).

The chapter is organized as follows: Sect. 6.2 describes concept of CI algorithm. Section 2.1 explains the constrained handling SAPF approach. Section 6.3 describes detailed architecture of FL algorithm. The framework of hybrid CIFL is presented in Sect. 6.4. In Sect. 6.5, two cases of three-bar truss structure problems are presented. In Sect. 6.6, the results are analyzed and discussed with other contemporary algorithm. Section 6.7 concludes the proposed work.

6.2 Cohort Intelligence (CI) Algorithm

The CI method is inspired from social tendency of learning by cohort candidates through interaction and competition with every other candidate (Kulkarni et al. 2013). The cohort candidates with certain qualities make a particular behavior. Every candidate in a cohort follows certain behavior and adopts the associated qualities which assists to improve the behavior of individual candidates. This makes every candidate learn from one another and helps to evolve the overall cohort behavior. The cohort behavior is considered saturated if for considerable number of learning attempts (iterations) the behavior of the candidates does not improve and becomes almost same. The characteristics of CI algorithm (Kale and Kulkarni 2021) are as follows:

1. CI models the social behavior of learning candidates having common aim to achieve the best behavior by improving their individual qualities.
2. For every learning attempt, the cohort candidates are keen to improve its individual behavior by observing self and other candidate' behavior in a cohort.
3. Every candidate of CI algorithm updates its search space for every learning attempt using sampling space reduction factor.
4. The CI algorithm has an ability to solve problems having more number variables and constraints.

6.2.1 Self-Adaptive Penalty Function (SAPF)

SAPF approach is adopted from Kale and Kulkarni (2021). The mechanism of SAPF is as follows:

The penalized or pseudo-objective function is written as

$$\phi\left(X^c\right) = f\left(X^c\right) + \text{SAPF}\left(X^c\right),$$

where $\text{SAPF}(X^c) = f(X^c) \times \left(\sum_{i=1}^{p} g_i(X^c) + \sum_{i=1}^{m} h_i(X^c)\right)$ is the self-adaptive penalty function;

$f(X^c)$ is the behavior of individual candidate.

The penalty parameter used in SAPF approach is itself a behavior of an individual candidate, i.e., $f(X^c)$. Every candidate has a different penalty parameter, and it updates iteratively as the CI algorithm progresses.

6.3 Fuzzy Logic

Fuzzy represents unclear/vague/absolute, e.g., ON–OFF, 0–1, High-Low, True–False, etc. However, there are several applications or the situations where these vague/absolute outputs would not work which requires the degree of truth. Lotfi Zadeh proposed the fuzzy logic (FL) in 1960 to represent the vague information (fuzzy sets) in the form of actual degree of truth (crisp value). FL is a concept associated with conventional logic which handles the information with partial truth (i.e., completely true or completely false). However, in the real-world applications or in day-to-day life activities it is necessary to analyze the exact degree of that partial truth. For that, FL method models the membership functions using certain rules based on the behavior of application. This rule-based system dives by an inference engine and provides the prescribed degree of truth. The FL methodology consists of four steps, such as (i) Fuzzification, (ii) Rule-base, (iii) Inference engine, and (iv) Defuzzification. The FL architecture is presented in Sect. 3.1.

6.3.1 Fuzzy Logic Architecture

The architecture of FL is divided into four parts as follows:

1. **Fuzzification**—It is process of converting the crisp value input (precise value) in fuzzy inputs using the membership function defined for that application. The membership function is considered for the optimization of truss structure best, mean, and worst function value.
2. **Rule Base**—In rule base, the defined *if–then* conditions are stored which is further used to control the decision-making system.

3. **Inference Engine**—The inference engine used to process fuzzy input. It evaluates the degree of match between fuzzy set and defined rules. Based on the percentage of degree of match, the rules are further modified and implemented to develop the control action.
4. **Defuzzification**—The processed fuzzy output generated by inference engine is then converted into crisp value using defuzzification step.

The general block diagram of FL architecture is presented in Fig. 6.1.

The term membership function used in fuzzification step specifies the degree of match of given input belonging to available sets. The degree of membership is represented between 0 and 1 which specifies the level of match of particular input belong to its set. This is also referred as the membership value of that variable. Different membership functions used to fuzzify a crisp (numerical) value are present in Fig. 6.2.

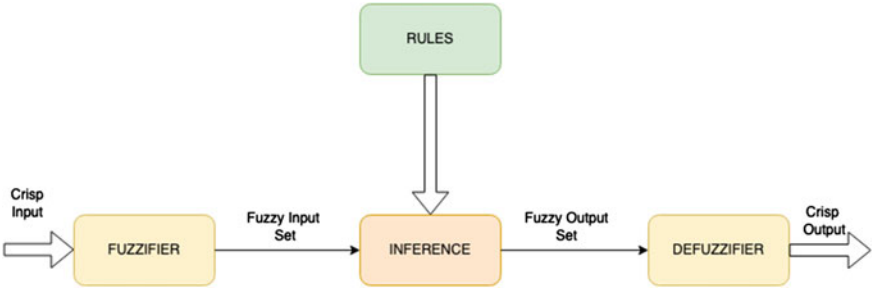


Fig. 6.1 Fuzzy logic architecture

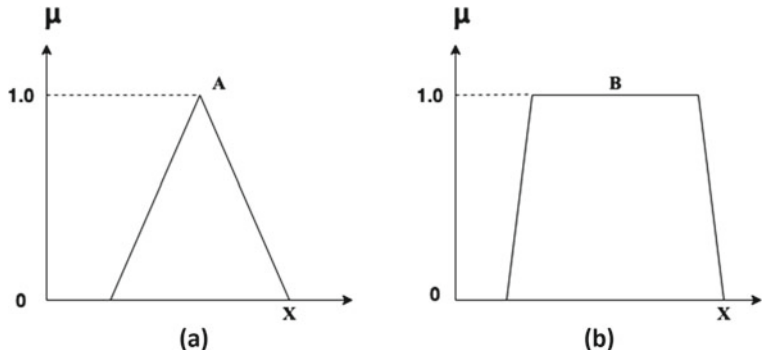


Fig. 6.2 Examples of triangular and trapezoidal membership function

6.4 Framework of CIFL

The proposed CIFL procedure is explained as follows:

Initialization of the number of cohort candidates C, variables t, sampling interval reduction factor R, and sampling interval Ψ.

$$\text{Minimize } f(x) = f(x_1, \ldots, x_i, \ldots, x_t) \tag{6.1}$$

$$\text{Subjected to } \Psi^{\text{lower}} \leq x_i \leq \Psi^{\text{upper}}, \quad i = 1, \ldots, t$$

Calculate probability of candidates based on objective function

$$p^c = \frac{1/f(x^c)}{\sum_{c=1}^{c} 1/f(x^c)}, \quad (c = 1, \ldots, C) \tag{6.2}$$

By using roulette wheel approach, it can decide which candidate to follow. The candidate that is being followed has produced the best results. This approach gives the candidates a choice to follow better behavior that their existing one.

Every candidate C shrinks the sampling interval Ψ associated with each variable t based on whether condition is saturated or not. The cohort behavior is considered to be saturated if there are no further improvements, the results are observed.

After that every candidate forms a behavior by sampling qualities from within the updated sampling intervals. Then it uses the updated sampling intervals for the membership function. Use updated sampling intervals as crisp input and convert them into fuzzy set. Create fuzzy rule base for the algorithm. After that convert the fuzzy set into crisp output. Check if the obtained solution is converged; if not, start the process again from calculating the objective function. A FL hybridized with CI to increase the performance of the CI algorithm for structural optimization. Fuzzy set theory is employed to deal with imprecise and vague information, during structural design process.

In CI algorithm, the solutions are randomly generated using uniformly distributed approach within its sampling interval. This range is iteratively modified using a sampling space reduction factor R. In the current work, the hybridization of FL with CI algorithm is demonstrated and it is referred as CIFL. In CI algorithm, modified sampling intervals for every candidate c associated with each variable t. Whereas, in CIFL these modified sampling intervals are treated as crisp input set which further utilized in the fuzzification process. In fuzzification, the crisp input is converted into fuzzy set which is defined as *best*, *medium,* and *worst,* respectively. Further inference engine helps to determine the degree of match of fuzzy input using *if–then* rule. As the algorithm progresses, the degree of match iteratively updated based on the modified sampling intervals. Next step is to defuzzify the fuzzy input into crisp values using Eq. (6.3).

$$x^* = \frac{\int \mu_{\tilde{c}}(x).x\mathrm{d}x}{\int \mu_{\tilde{c}}(x).\mathrm{d}x} \tag{6.3}$$

After the defuzzification, these crisp values are further used to evaluate the function. If after significant number of iterations, the solution is converged and the solution does not improve and also becomes the same accept it as the final solution and stop, else repeat the process from the objective function. The flowchart of the proposed CIFL is presented in Fig. 6.3.

The hybrid CIFL algorithm was coded in Python3 on Visual Studio Code Platform with an Apple M1 chip @3.2 GHz octa-core processor with 8 GB RAM at Institute of Artificial Intelligence, MIT World Peace University, Pune, India (Fig. 6.4).

6.5 Three-Bar Truss Structure Problems

There are many heuristic as well as metaheuristic techniques have been used to solve the three-bar truss design optimization problem such as Swarm Optimization Approach (SOA) (Ray and Saini 2001), Cuckoo Search Algorithm (CSA) (Gandomi et al. 2013), Bat Algorithm (BA) (Yang and Gandomi 2012), Mine Blast Algorithm (MBA) (Sadollah et al. 2013), Cricket Algorithm (CA) (Canayaz and Karci 2016), Artificial Atom Algorithm (A^3) (Yildirim and Karci 2018).

Case 1: Three-bar truss structure is shown in Fig. 6.5. The volume of the truss structure is to be minimized subject to stress constraints.

There are two design variables (x_1, x_2) and three nonlinear constraint functions in this problem. The problem is expressed mathematically as follows:

Objective function:

$$\text{Min } f(x) = \left(2\sqrt{2}x_1 + x_2\right) \times L \tag{6.4}$$

Constraints:

$$g_1 = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0 \tag{6.5}$$

$$g_2 = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0 \tag{6.6}$$

$$g_3 = \frac{1}{x_1 + \sqrt{2}x_2} P - \sigma \leq 0 \tag{6.7}$$

where $0 \leq x_1, x_2 \leq 1$. The constants are $L = 100$ cm, $P = 2$ KN/cm^2, and $\sigma = 2$ KN/cm^2

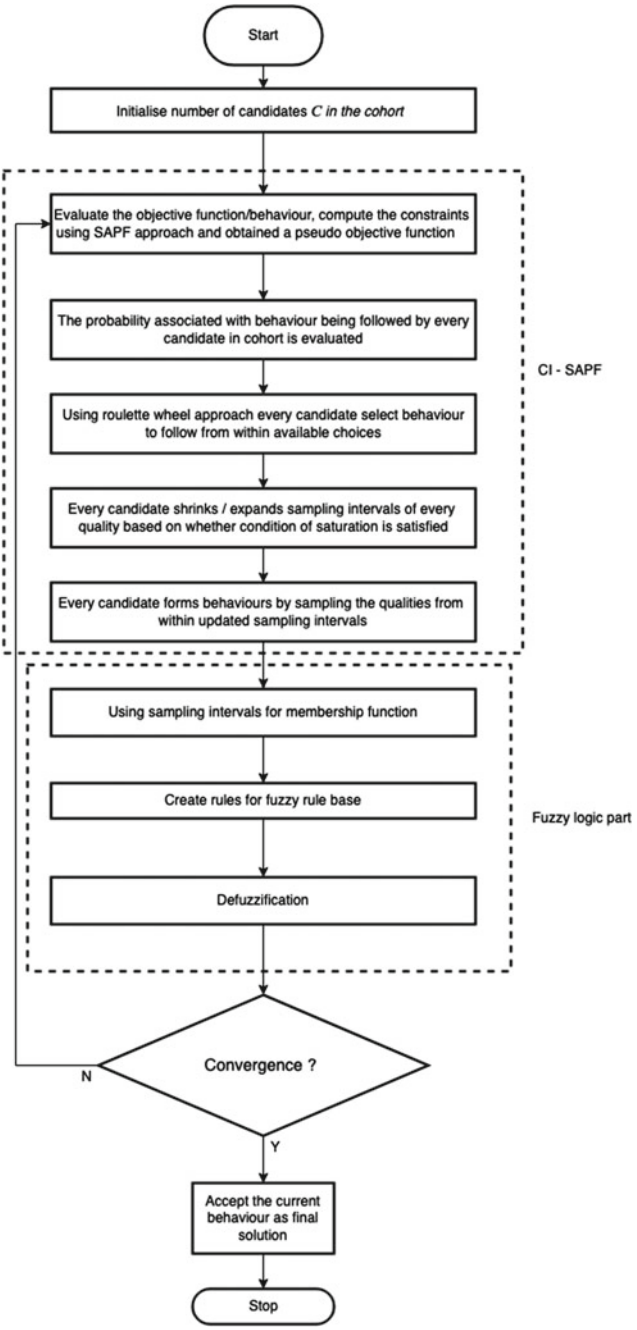


Fig. 6.3 Hybrid CIFL flowchart

C Number of cohort candidates ($c = 1, \ldots, C$)
Ψ Sampling interval
R Sampling interval reduction factor
P Penalty function
g Constraint violations

Initialize C, Ψ, R
For
1. The set of design variables ($\boldsymbol{X}$) are initialized using uniform distribution method within the defined sampling space Ψ.
2. Evaluate objective function value $f(X^c)$
3. SAPF approach applied to handle the constraints and generate a pseudo behavior:
   $\phi(X^c) = f(X^c) + P(X^c)$
   where $P(X^c) = f(X^c) \times \sum_{c=1}^{C} g\,(X^c)$
4. Calculate the probability p^c of every candidate behavior: $p^c = \frac{1/\phi^*(X^c)}{\sum_{c=1}^{C} 1/\phi^*(X^c)}$
5. Roulette wheel approach is used for every candidate c to select and follow a behavior from within C available choices.
6. Every candidate c updates its sampling interval $\boldsymbol{\Psi}^c$ using sampling space reduction parameter R and $\boldsymbol{X}^c$:
   $$\boldsymbol{\Psi}^c = [\boldsymbol{\Psi}^{c,lower}, \boldsymbol{\Psi}^{c,upper}] = [\boldsymbol{X}^c - \left\|\frac{\boldsymbol{\Psi}^{upper} - \boldsymbol{\Psi}^{lower}}{2}\right\| \times R, \boldsymbol{X}^c + \left\|\frac{\boldsymbol{\Psi}^{upper} - \boldsymbol{\Psi}^{lower}}{2}\right\| \times R$$
7. The updated sampling intervals $\boldsymbol{\Psi}^c$ are used as a range for membership function for every candidate.
8. Create *if-then* rules for fuzzy rule base.
9. Defuzzification: Centroid method: $x^* = \frac{\int \mu_{\tilde{c}}(x).\, x\, dx}{\int \mu_{\tilde{c}}(x).\, dx}$
10. Obtained solution
11. **If**: The solution $\phi(X^c)$ saturates and no further improvement in it.
    Update the sampling space is set $\boldsymbol{\Psi}^c$ as original $\boldsymbol{\Psi}$
    Accept the solutions.
    Else:
    Continue to step 2

Fig. 6.4 Hybrid CIFL pseudo-code

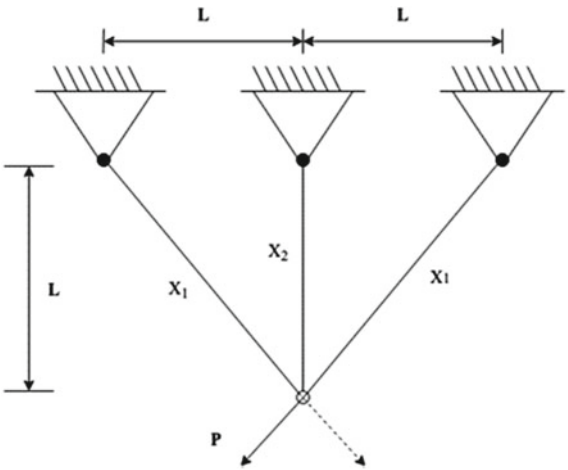


Fig. 6.5 Three-bar truss design Case 1 (Ayse and Karci 2018)

Case 2: The indeterminate three-bar truss structure (refer Fig. 6.6) is subject to vertical and horizontal forces at a single node which is an intersection of all the three members. The aim is to minimize the structural weight W and is minimized under the constraint that the stress in all members should be smaller than allowable stress σ_0 in absolute magnitude. After nondimensionalization of the objective function and variables, $F = \sigma_0 W/Ppl$ and $x_i = a_i \sigma_0/P$. This problem is previously

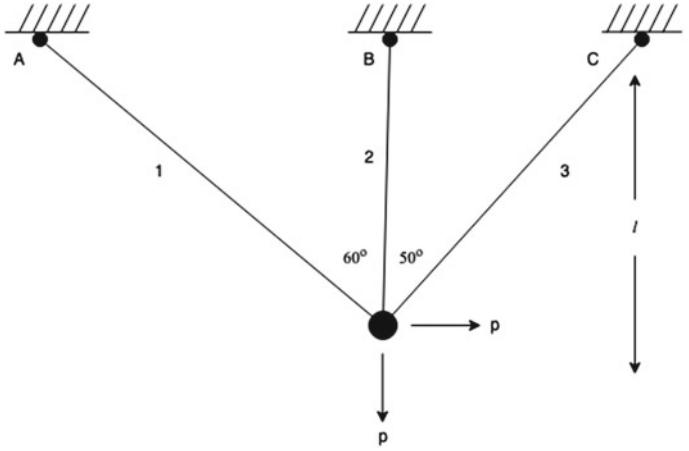


Fig. 6.6 Three-bar truss design Case 2 (Shin et al. 1990)

solved by using CI-SPF (Kale and Kulkarni 2018), Multi-Random Start Local Search (MRSLS), CI-SAPF, and CI-SAPF-CBO (Kale and Kulkarni 2021).

$$\text{Objective function Min: } f(x) = 2x_1 + x_2 + \sqrt{2x_3} \tag{6.8}$$

Subject to:

$$g_1 = 1 - \frac{\sqrt{3}x_3 + 1.932x_3}{1.5x_1x_2 + \sqrt{2}x_2x_3 + 1.319x_3} \geq 0 \tag{6.9}$$

$$g_2 = 1 - \frac{0.634x_1 + 2.828x_3}{1.5x_1x_2 + \sqrt{2}x_2x_3 + 1.319x_3} \geq 0 \tag{6.10}$$

$$g_3 = 1 - \frac{0.5x_1 + 2x_2}{1.5x_1x_2 + \sqrt{2}x_2x_3 + 1.319x_3} \geq 0 \tag{6.11}$$

$$g_4 = 1 + \frac{0.5x_1 - 2x_2}{1.5x_1x_2 + \sqrt{2}x_2x_3 + 1.319x_3} \geq 0 \tag{6.12}$$

$$x_i = \{0.1,\ 0.2,\ 0.3,\ 0.5,\ 0.8,\ 1.0,\ 1.2\}, \quad i = 1, 2, 3$$

6.6 Results Analysis and Discussion

The algorithm is validated by solving two cases of three-bar truss structure-constrained optimization problems for the minimization of weight. The solutions obtained from proposed CIFL algorithm for three-bar truss structure Case 1 problem are compared with Swarm Optimization Algorithm (SOA), Cuckoo Search Algorithm (CSA), Bat Algorithm (BA), Mine Blast Algorithm (MBA), Cricket Algorithm (CA), and Artificial Atom Algorithm (A^3) presented in Table 6.1. The statistical results such as best, mean, and worst function values, standard deviation, average CPU time and average function evaluations are obtained from 30 independent trails. From the result comparison, it shows that the proposed CIFL algorithm yielded satisfactory results compared with other algorithms (refer Table 6.1). It is observed that the solution precisely worst by 0.19% as compared to the latest solution obtained by A^3 (Yildirim and Karci 2018) and CA (Canayaz and Karci 2016). The convergence plot of Case 1 of three-bar truss structure problem is presented in Fig. 6.7.

In CSA (Gandomi et al. 2013), Lévy's flight approach was employed with three key rules such as selection of the best, exploitation by local random walk, and exploration by randomization. The performance of the CSA is dependent on the parameter which needs to be fine-tuned. It may require certain preliminary trail to set an appropriate value. The CSA adopts Lévy's flight strategy so that only best solution can be obtained which is close to optimal value. Like CSA, Bat Algorithm (BA) (Yang and Gandomi 2012) is also required fine-tuning of two computational parameters which directly affect the convergence of BA. The Bat Algorithm (BA) models the foraging behavior of bats. Bat uses echolocation to sense the distance food and background barrier. The Cricket Algorithm (CA) (Canayaz and Karci 2016) which models the behavior of cricket insect. These cricket insects intercommunicate with their peers through the sound in nature. They generate the sound by chirping of their wings based on the atmospheric temperature. An A^3 (Yildirim and Karci 2018) referred to

Table 6.1 Comparison of results for three-bar truss structure optimization problem Case 1

Algorithm	x_1	x_2	$f(x)$
SOA (Ray and Saini 2001)	0.79500	0.39500	264.3000
CSA (Gandomi et al. 2013)	0.78867	0.40902	263.9716
BA (Yang and Gandomi 2012)	0.78863	0.43838	263.8962
MBA (Sadollah et al. 2013)	0.7885650	0.4082482	263.8958
CA (Canayaz and Karci 2016)	0.78863	0.408368	263.8958
A^3 (Yildirim and Karci 2018)	0.7887357	0.408078	263.8958
CIFL	0.7049	0.5800	264.4031

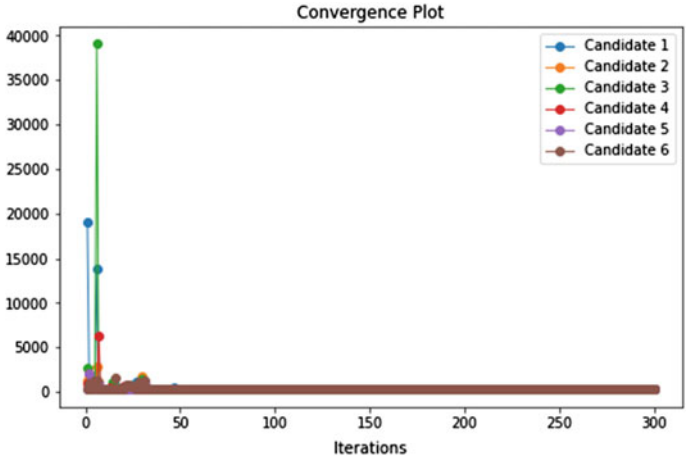


Fig. 6.7 Conversion plot for three-bar truss structure optimization problem Case 1

as Artificial Atom Algorithm is based on chemical compounding processes. There are two operators in A^3 which are iconic bond and covalent bond. It is associated with the conceptual strategy of an electron, atom, and atom set. Number of design variables are considered while determining number of electrons. The atom is formed randomly according to constraint conditions (Table 6.2).

For three-bar Case 2 truss structure problem, CIFL is compared with CI-SPF, CI-SAPF, MRSLS, CI-SAPF-CBO algorithm presented in Table 6.3. The statistical results obtained from 30 trials are presented in Table 6.4. It is observed that the CIFL algorithm has obtained same function value. The convergence plot of Case 2 of three-bar truss structure problem is presented in Fig. 6.8. The average computational time obtained from CIFL for Case 1 and Case 2 is 7.04 and 7.55 s, respectively. The constraints associated with these problems are very much crucial task. Here, in CIFL algorithm SAPF approach is incorporated to deal with the constraints. This is important to note that it does not need to set a penalty parameter. It is adaptively set as the algorithm progresses. This SAPF approach is also used in CI-SAPF, CI-SAPF-CBO, and MRSLS. In CI-SPF, the static penalty function is incorporated to deal with the constraints; however, it requires to fine-tuning to set an appropriate value of penalty parameter.

Table 6.2 Statistical results obtained from CIFL for three-bar truss design optimization problem Case 1

Best	264.4031
Mean	273.9651
Worst	286.3454
Standard deviation	6.249121
Avg. function evaluations	821
Avg. CPU time (s)	7.04

Table 6.3 Comparison of results for three-bar truss design optimization problem Case 2

Algorithms	Function value	Optimum variables
NEWSUMT-A (Shin et al. 1990)	3.0414	[1.2, 0.5, 0.1]
CI-SPF (Kale and Kulkarni 2018)	3.0414	[1.2, 0.5, 0.1]
MRSLS (Kale and Kulkarni 2021)	3.0414	[1.2, 0.5, 0.1]
CI-SAPF (Kale and Kulkarni 2021)	3.0414	[1.2, 0.5, 0.1]
CI-SAPF-CBO (Kale and Kulkarni 2021)	3.0414	[1.2, 0.5, 0.1]
CBO (Kale and Kulkarni 2021)	3.0414	[1.2, 0.5, 0.1]
CIFL	3.0414	[1.2, 0.5, 0.1]

Table 6.4 Statistical results for three-bar truss design optimization problem Case 2

Best	3.0414
Mean	3.5050
Worst	3.7071
Standard deviation	0.1511
Avg. function evaluation	250
Avg. CPU time (s)	7.55

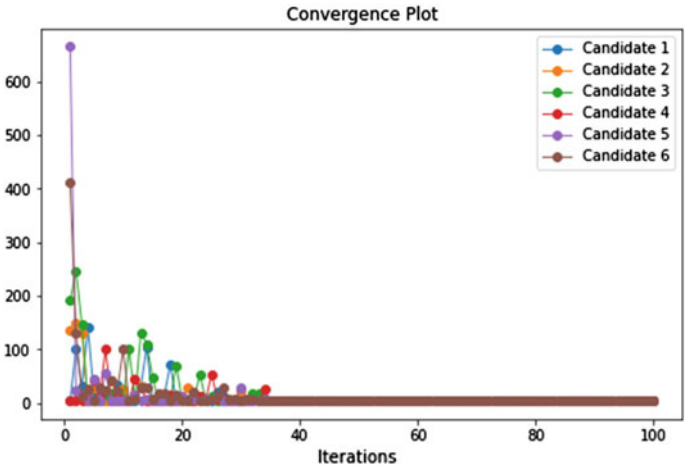


Fig. 6.8 Conversion plot for three-bar truss design optimization problem Case 2

In CIFL as well every candidate updates its sampling interval for every iteration using a sampling space reduction factor. It also requires certain preliminary trials however, according to the analysis conducted in Kale and Kulkarni (2018) the value of R can be set between 0.95 and 0.97. This eliminates the tuning of parameter R. The role of FL in CIFL is to nullify the randomly regeneration of variable values. The sampling space updates in every iteration are considered for the membership function. Then using a rule base condition, the fuzzy system provides the crisp values which are further used as a set of variables.

6.7 Conclusions

The CIFL algorithm is successfully applied to solve the two cases of constrained three-bar truss structure problem. These problems are associated with discrete as well as continuous variables and have nonlinear constraints. For discrete variables, a round-off integer sampling is incorporated. The performance of CIFL is observed to be precisely similar as compared to other metaheuristic algorithms. The hybrid version of CIFL algorithm eliminates the randomly generated solutions which in CI algorithm for every learning attempt. The SAPF approach is incorporated to handle the constraints associated with the problems. After the extensive comparative study, it is observed that CIFL obtained satisfactory results as compared to other contemporary metaheuristic algorithms. The CIFL algorithm can be used to solve more design engineering problems as well as structural engineering problems.

References

Canayaz M, Karci A (2016) Cricket behaviour-based evolutionary computation technique in solving engineering optimization problems. Appl Intell 44(2):362–376

Finotto VC, da Silva WR, Valášek M, Štemberk P (2013) Hybrid fuzzy-genetic system for optimising cabled-truss structures. Adv Eng Softw 62:85–96

Gandomi AH, Yang XS, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29(1):17–35

Kale IR, Kulkarni AJ (2021) Cohort intelligence with self-adaptive penalty function approach hybridized with colliding bodies optimization algorithm for discrete and mixed variable constrained problems. Complex Intell Syst 7(3):1565–1596

Kale IR, Kulkarni AJ (2018) Cohort intelligence algorithm for discrete and mixed variable engineering problems. Int J Parallel Emergent Distrib Syst 33(6):627–662

Li LJ, Huang ZB, Liu F (2009) A heuristic particle swarm optimization method for truss structures with discrete variables. Comput Struct 87(7–8):435–443

Mortazavi A (2019) Interactive fuzzy search algorithm: a new self-adaptive hybrid optimization algorithm. Eng Appl Artif Intell 81:270–282

Omidinasab F, Goodarzimehr V (2020) A hybrid particle swarm optimization and genetic algorithm for truss structures with discrete variables. J Appl Comput Mech 6(3):593–604

Ray T, Saini P (2001) Engineering design optimization using a swarm with an intelligent information sharing among individuals. Eng Optim 33(6):735–748

Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput 13(5):2592–2612

Shin DK, Gurdal Z, Griffin OH (1990) A penalty approach for nonlinear optimization with discrete design variables. Eng Optim 16(1):29–42

Yang XS, Gandomi AH (2012) Bat algorithm: a novel approach for global engineering optimization. Eng Comput

Yildirim AE, Karci A (2018) Application of three bar truss problem among engineering design optimization problems using artificial atom algorithm. In: 2018 international conference on artificial intelligence and data processing (IDAP), pp 1–5

Yang Y, Soh CK (2000) Fuzzy logic integrated genetic programming for optimization and design. J Comput Civ Eng 14(4):249–254

Chapter 7
Optimum Design of BRB Frame Based on Drift Uniformity, Structure Weight, and Seismic Parameters Using Nonlinear Time History Analysis

S. Ali Razavi and Rouhollah Shirjani

Abstract Buckling-restrained braces (BRBs) are one of the popular seismic-resistant structural systems. The cross-sectional area and length of the BRB is one of the most important characteristics of these braces that directly affect their cost. Since columns, beams, and connections are designed for the maximum force delivered by the brace, the decrease in cross-sectional area of the BRB causes a decrease in dimensions of the columns and beams. On the other hand, drift uniformity over the height of the structure is accounted as a structural health index and would lead in efficiency of BRB system in a seismic event. The aim of this study is then to optimize three objectives including weight of the BRB, weight of the structure, and uniformity of the drift profile over the height of structure by changing the cross-sectional area and the length of the BRB at the height of the structure using genetic algorithms and other multi-objective optimization algorithms. Optimization is based on the results of nonlinear time history analysis of 2D frames. Seven earthquake records are selected to conduct nonlinear time history analysis using OpenSees software. To this end, the desired functions and constraints were defined in the genetic algorithms, i.e., NSGA_II, MOPSO, MOEA_D, PESA_II, SPEA_II, and the initial created population was entered as the initial cross-sectional area and length of the braces in the OpenSees software. The optimization results showed that for all three objective functions, the weight of the structure, the weight of the BRB brace, and the uniformity of drift in the height of the structure can be optimized largely using a nonlinear time history analysis.

Keywords BRB · Multi-objective optimization · Time history analysis · Uniform drift

S. Ali Razavi (✉) · R. Shirjani
Department of Civil Engineering, University of Science and Culture, Tehran, Iran
e-mail: arazavi@usc.ac.ir

I. R. Kale and A. Sadollah (eds.), *Optimization Methods for Structural Engineering*,
Engineering Optimization: Methods and Applications,
https://doi.org/10.1007/978-981-99-2378-6_7

7.1 Introduction

Optimization is an important and decisive activity in design of structures. Designers will only be able to design better models if they use optimization methods to save time and design expenses. Many optimization problems in engineering are naturally more complex and difficult than they could be solved by conventional optimization methods, such as linear programming methods and the like. One of the solutions to deal with such problems is the use of evolutionary algorithms. In addition, the goal of optimization is to find the best acceptable solution with regard to the limitations of the problem. There may be different solutions for a problem, and in order to compare them and select the optimal solution, a function called the objective function is defined.

Currently, in order to waste earthquake energy, the use of energy dampers in structures has been considered. Conventional braces normally dissipate energy while they are loaded in tension. In compression, the occurrence of buckling phenomenon before yielding results in less energy dissipation, reduced lateral stiffness of the frame, reduced closed area of the hysteresis loops, and instability in one story or the whole structure (Ali Razavi 2011; Uang et al. 2003; Clark et al. 1999).

BRBs are a relatively new and improved type of concentric braced frames, whose performance is almost identical both in tension and compression (Fig. 7.1). In these braces, the axial stresses are tolerated by a steel core. The buckling resistance of the brace is provided by an external encasing made of concrete, steel, or any other combination of steel concrete material. While the encasing prohibits the brace from global buckling, the steel core only withstands uniform axial strains both in compression and tension (Lopez and Sabelli 2004). The aim of this research is therefore to achieve an optimal design for the cross section and length of BRBs at the height of the structure while providing the desired constraints. This aim has been accomplished using the genetic algorithm and the multi-objective algorithm.

Genetic algorithm is an optimization method inspired by the living nature (living organisms) that can be considered as an evolutionary algorithm in the classification of optimization methods from among a set of randomly guided search algorithms. This is an iteration-based algorithm, and its basic principles are adopted from genetics.

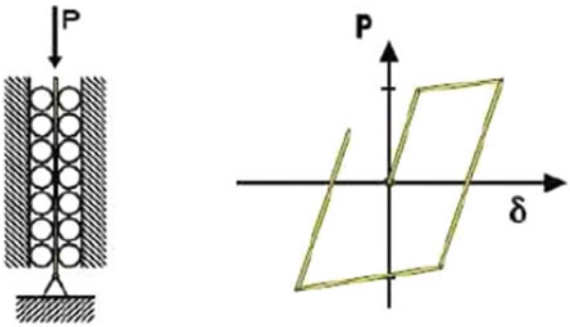


Fig. 7.1 Schematic behavior of BRB (Lopez and Sabelli 2004)

Genetics is the scientific study of how biological traits are inherited and passed from one generation to the next. Chromosomes and genes are the main factors in the transfer of biological traits in living organisms, and the way they work is such that eventually the superior and stronger genes and chromosomes survive and the weaker ones are destroyed. In other words, the result of the interaction of genes and chromosomes is the survival of the fittest. The genetic algorithm likewise finds the best optimization solution accordingly. In addition, in this research, multi-objective algorithms such as NSGA_II, MOPSO, MOEA_D, PESA_II, SPEA2 have been used.

Since the cost-controlling factor in BRB depends on their core's cross section and their length, the main purpose of this study is to optimize the core's cross section of these braces and their length in the height of 2D frames. One of the constraints is that in all stories, the story drift satisfies the allowable limit. Three objective functions are defined including bracing weight, total weight of the structure (without bracing weight), and deviation from the uniform drift over the height of the structure. In the optimization process, another constraint was controlled which addresses the low-cycle fatigue to ensure that the braces do not rupture during seismic event.

In order to achieve the optimal distribution, meta-heuristic algorithms were applied in MATLAB and a nonlinear analysis was conducted under seven ground motions using OpenSees software. At a part of research implementation, a bilateral connection was established between OpenSees and MATLAB software.

7.2 Multi-objective Optimization

In the single-objective optimization, the algorithm ends by optimizing the objective function. However, in multi-objective problems, optimizing several objective functions at the same time is complicated and time-consuming. Furthermore, in most of the problems, a number of acceptable solutions are obtained based on unfavorable criteria. Thus, the final solution is in the form of a set of solutions that is indicative of a balanced representation of the various objective functions of the problem. Finally, one of the solutions is selected as the reference solution by the decision maker. A general multi-objective optimization problem can be defined as Eq. (7.1):

$$\begin{array}{l} f(x) = \left[f_1(x), f_2(x), \ldots, f_Q(x)\right] \\ \text{Subject} : x \in X \end{array}, \tag{7.1}$$

where $X \subseteq R^Q$ is the problem-solving space and $x = \left\{x_1, x_2, \ldots, x_p\right\}$ is the set of decision variables in the next p-space. Among this set of finite solutions, the appropriate solution will be the answers that have acceptable performance with regard to all goals. Solving multi-objective problems using the beam approach is among more complex problems. This is because there is usually no specific optimal solution for these methods (Deb 2001).

To make appropriate comparisons in multi-objective optimization, the concept of dominance is used, assuming that F is the total space of the problem and $x_1, x_2 \in F$ are two answers of this problem. x_1 dominates x_2 (or x_2 is defeated by x_1) if and only, x_1 is not worse than x_2 in neither of the objectives ($f_i(x_1) \leq f_i(x_2) \forall i \in \{1, 2, \ldots, m\}$) and x_1 is definitely better than x_2 in at least one of the objectives ($f_i(x_1) < f_i(x_2)$). In other words, unresolved solutions are solutions that cover other solutions, but are not themselves covered by other solutions. Now, according to this concept, two operators have been added to single-objective algorithms and are known as multi-objective algorithms. These two operators are fast non-dominated sorting (FNDS) and crowding distance (CD) (Deb 2001).

7.2.1 Structural Geometry

The structure presented by Hosseini Hashemi et al. (2016) is considered as a benchmark. The aforementioned structure is located in Tehran with soil type II. The structure is seismically designed using the Iranian seismic regulations according to Standard 2800 (BHRC 2014). Figure 7.2 shows the *A*-axis frame which is used for nonlinear modeling and optimization of the braces. The optimization of the two structures, a 6-story building and a 10-story building, will be performed in two dimensions. The elevations of these structures are shown in Fig. 7.3a and b, respectively. Table 7.1 shows the seismic load-resisting system parameters.

7.2.2 Nonlinear Structural Modeling

OpenSees, which is an open-source software, was used for modeling and nonlinear analysis of the structure. The braces are modeled using a nonlinear beam–column element and a fiber-based cross section. In this method, the desired cross-sectional area is divided into small elements, and by assigning the desired material to each of the elements, instead of assuming the process of plasticization in certain parts of the structure (such as assuming a plastic hinge in the middle or two ends of the beam), plasticization can be considered as distributed along the entire length of the element, which increases the accuracy of the modeling process (Tauer et al. 1991).

As explained in the previous sections, the columns and beams of the BRBs frame must be strong enough to remain elastic during the earthquake and not to enter the inelastic region. Therefore, assigning the elastic beam–column element to the columns and beams does suffice and the speed of nonlinear analysis is reduced. However, in order to control the behavior of these elements and to know whether they have entered the nonlinear region or not, the nonlinear beam–column element has been used. In both structures, the *A*-axis frame is considered for nonlinear 2D analysis and optimization.

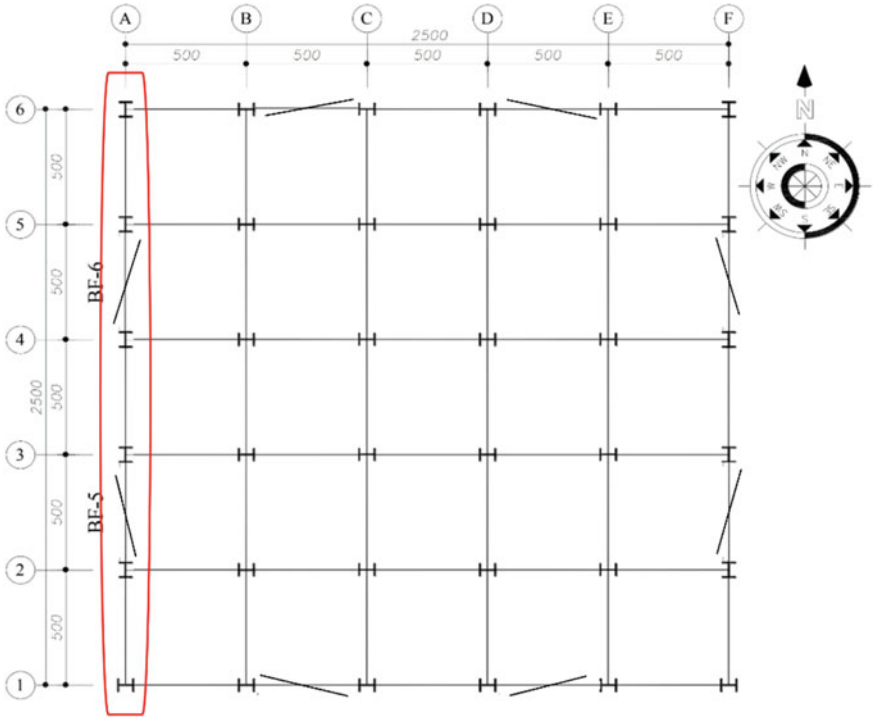
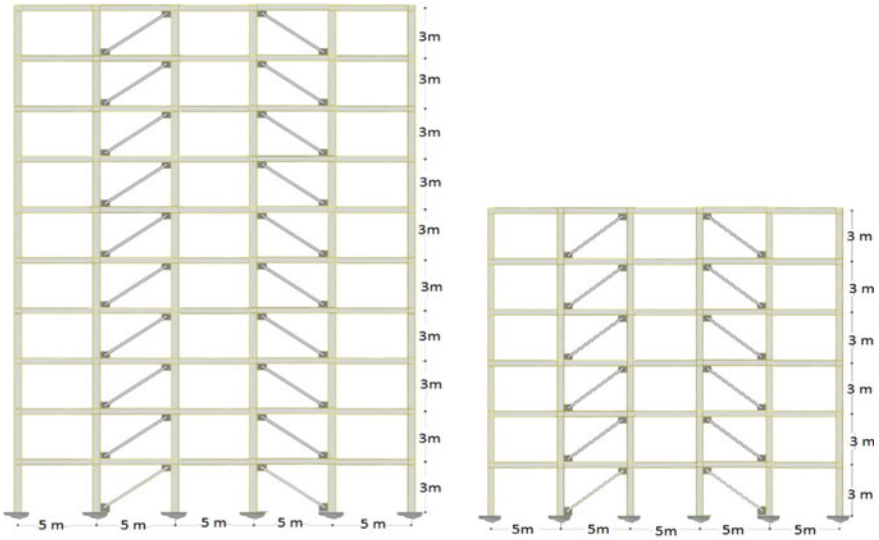


Fig. 7.2 Braced frame plan (Hosseini Hashemi et al. 2016)

The stress–strain material model of steel sections is simulated using Steel02 materials in OpenSees software. These materials approximate the cyclic behavior well and consider the strain stiffness kinematically.

The parameters required to determine the behavior of Steel02 materials are as follows: F_y is yield stress, E is initial elastic stiffness, b is strain stiffness ratio, and R is the degree of curvature at the intersection of the two lines of the diagram. The yield stresses of the steel core of the BRBs and the columns and beams are 2620 and 2400 kg/cm^2, respectively. Steel02 was calibrated according to the test data obtained from Ali Razavi et al. (2018).

As a means to estimate the of low-cycle fatigue status of BRBs modeled in OpenSees, fatigue material has been used in order to control the damage criterion. This material is defined by Uriz (2005), which considers the effect of low-cycle fatigue on the model. Based on the cumulative damage of Miner (1945) and the Coffin-Manson equation (Stephens et al. 2000), this material determines the damage criterion in the braces that are allocated to Steel02 material. The parameters required to define this material are the yield stress of the cross-sectional steel as well as the two values m and E0, which indicate the slope of the Coffin-Manson curve in the logarithmic space and the amount of strain in the loops leading to rupture, respectively.



(a) (b)

Fig. 7.3 Frame elevation of 6 and 10 floors

Table 7.1 Seismic load-resisting system parameters

Parameter	Value	Regulations
Structure height	18.30 m	–
Location of the structure	Tehran	–
Design basis acceleration (a)	0.35	Standard No. 2800
Soil type	II	Standard No. 2800
User	Residential and office	–
Significance factor	1	Standard No. 2800
Seismic-bearing system	BRBF with simple beam–column connections	–
Behavior coefficient (R)	7	Standard No. 2800
C_d	5.5	Standard No. 2800

7.2.3 Time History Analysis

Nonlinear dynamic analysis was performed on the frame in 6- and 10-story structures. According to scaling guidelines of Standard No. 2800, the selected ground motions should preferably reflect the actual motion of the ground at the construction site during an earthquake. To reach this goal, at least seven pairs of horizontal ground motions are required. Therefore, seven pairs of accelerograms have been used in this research in order to use the average of their responses in the optimization process.

7.2.4 Ground Motion Records

In order to perform nonlinear dynamic analyses, seven records including Nahanni, Loma Prieta, Cape Mendecino, Northrukidge, Chichi, Irpinia, and Loma Prieta were used to calculate the responses, so that their average responses could be used. Records were selected from the set of records provided by PEER. Table 7.2 shows the specifications of the selected records, and Fig. 7.4 shows the chart of time history of the acceleration of the selected records.

Figure 7.5 shows the average of acceleration response spectrum of the selected records.

7.3 Optimization Process

The purpose of optimization in this study is to minimize the length and cross-sectional area of the BRBs by observing the defined constraints. To calculate and control the constraints, the average of the results obtained from the nonlinear analysis of frames under seven earthquakes by OpenSees software in the optimization algorithm coded in MATLAB has been used. For this purpose, it was necessary to establish a connection between OpenSees and MATLAB software, so that the results of nonlinear analysis in OpenSees could be used as the input of the optimization algorithm and vice versa. In other words, the sections generated by the optimization algorithm could be used in OpenSees. The process is presented in the following section.

Table 7.2 Specifications of earthquake records (http://peer.berkeley.edu/nga)

Earthquake name	Recording station	Year	Magnitude	Mechanism	Epicentral dist. (Km) (Rjb)	Soil type
Nahanni_Canada	Site 1	1985	6.76	Reverse	2.48	II
Loma Prieta	BRAN	1989	6.93	Reverse oblique	3.85	II
Cape Mendocino	Cape Mendocino	1992	7.01	Reverse	0	II
Northridge-01	LA—Sepulveda VA Hospital	1994	6.69	Reverse	0	II
Chi-Chi_Taiwan	TCU084	1999	7.62	Reverse oblique	0	II
Irpinia_Italy-01	Sturno (STN)	1980	6.9	Normal	6.78	II
Loma Prieta	Saratoga—Aloha Ave	1989	6.93	Reverse oblique	7.58	II

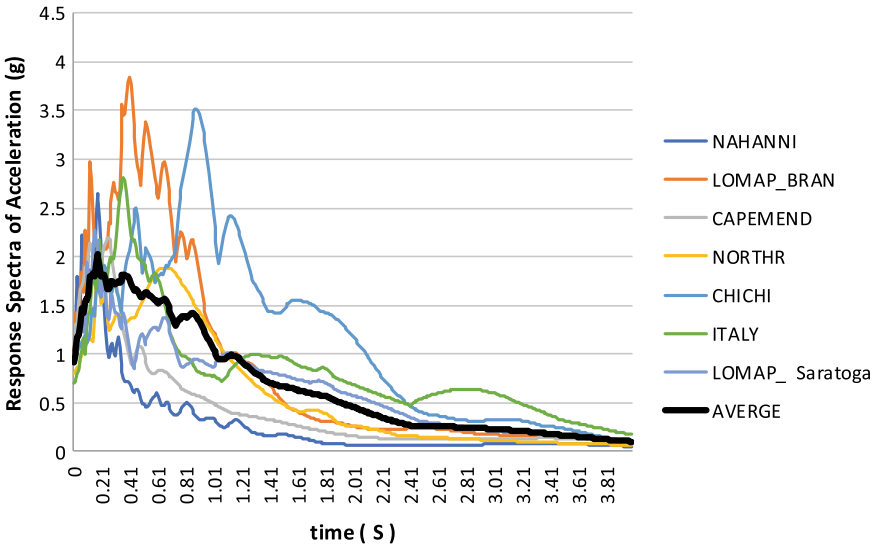


Fig. 7.4 Average of the response spectrum of the acceleration records

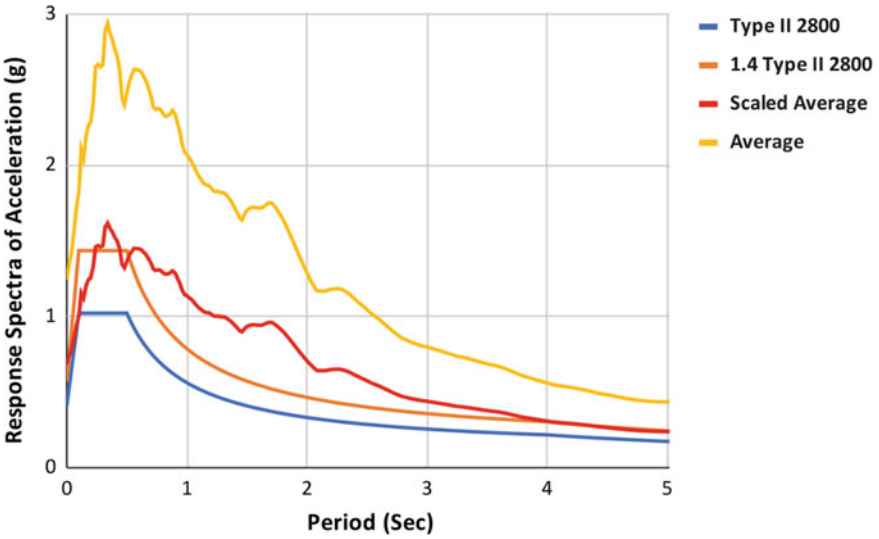


Fig. 7.5 Scaling the average response spectrum of earthquakes with the soil response spectrum II

7.3.1 Formulation of the Optimization Problem

Optimization searches for the optimal values of design variables, so that the best output could be given to the objective function and could meet the criteria of the

Table 7.3 Parameters used in the algorithm for optimization

2D Frame	Algorithm	Number of design variables	Population size	Number of optimization steps (generation)
6 Story	NSGA_II	12	30	100
6 Story	SPEA2	12	50	100
6 Story	PESA_II	12	25	100
6 Story	MOPSO	12	100	100
6 Story	MOEA_D	12	25	100
10 Story	NSGA_II	24	30	100
10 Story	SPEA2	24	50	100
10 Story	PESA_II	24	25	100
10 Story	MOPSO	24	300	100
10 Story	MOEA_D	24	25	100

regulations and the designer's objective (constraints). The optimal value can be the minimum or maximum value of the desired function. In this research, the minimum value of the objective function is the answer to the optimization problem.

In order to use the multi-objective algorithms such as NSGA_II MOPSO, MOEA_D, PESA_II, SPEA2 in the optimization process, the parameters required in this algorithm are shown in Table 7.3.

7.3.2 Design Variables

During a seismic event, the BRBs effectively dissipate the input energy both in tension and compression. Changes in the cross-sectional area, length and characteristics of the material used in the bracing core, and its installation location in the structure affect the yield of the bracing core. The total weight of the braces, the total weight of the structure excluding the weight of the braces, and the drift uniformity are selected as the target functions. The thickness of the brace sections is considered constant value of 30 mm and the design variable; i.e., the width of the sections and the length of the brace are considered for optimization (Formulas (7.2) and (7.3)) (Ali et al. 2014).

$$X = (b_1, b_2, b_3, \ldots, b_n) \tag{7.2}$$

$$L = (l_1, l_2, l_3, \ldots, l_n) \tag{7.3}$$

where n is the number of the stories and l_i is the width of the section and b_i is the length of the i brace. It should be noted that a common brace is considered for each floor.

7.3.3 Design Constraints

In structural design, variables cannot have any value and must be limited by a series of requirements and constraints called design constraints. The most important necessity in BRBFs after the earthquake is to minimize the residual deformations in the structure and the amount of damage to it. In this research, providing such a requirement is accomplished by satisfying the criteria of limiting the lateral displacement of the story to the allowable amount, i.e., Formula (7.4). In order to control the residual displacement of the structure, the amount addressed in Eq. (7.5), is considered as the upper permissible limit.

$$\text{Allowable Drift} = 0.02 \tag{7.4}$$

$$\text{Residual Displacement} = 0.005H, \tag{7.5}$$

where H is the height of the structure. In the above equations, allowable drift is the relative displacement of the floor, and residual displacement is the amount of residual displacement allowed. The amount of relative lateral displacement of the floors shall be limited to 2% according to ASCE 7–16 (American Society of Civil Engineers 2016).

The maximum brace strain is not the only criterion for the proper performance of BRB up to the end of loading, since according to the cumulative damage criterion, a set of low-cycle fatigue losses in different cycles should be considered in order to guarantee the BRB stable performance. Accordingly, the end of the performance of BRBs during seismic loads is the rupture due to low-cycle fatigue. It is noteworthy that the probability of this rupture is increased by reduction in the length of the braces. According to the explanations provided, a constraint has been considered to control the criterion of damage due to low-cycle fatigue. The criterion of damage for each brace is calculated during a chronological analysis. The criterion of fatigue damage (FDI index) of all braces during each of the seven chronological analyses must be less than one.

In addition, other lateral constraints such as the minimum and maximum amount for the width of the braces core section are equal to 1 cm and 20 cm, respectively. The minimum length of the BRB length is considered 35 cm both the six-story and ten-story frame.

Furthermore, since the number of analyses in optimizations is high and each of them is important, there is a limit to the adequacy of shear, flexural, and axial forces for beams and columns, meaning that in each analysis, individual beams and columns are examined. And their appropriate cross section will be selected from the list of prepared cross sections.

7.3.4 Objective Functions

The objective function, commonly known as a cost or performance criterion, is defined based on design variables and decision motivation. In optimal design, the best value of the objective function (minimum or maximum) is obtained, so that all constraints will be met. It is, therefore, important to select an appropriate objective function. In this research, a multi-objective optimization problem is solved using the objective function related to cost under seismic loads. The aim is to minimize the cross section and length of BRBs by three objective functions. The first objective function is to minimize the weight of BRBs (7.6), the other is to minimize the weight of the whole structure without BRBs (7.7), and the last is the uniform relative displacement (Drift) in the structure, Eq. (7.8).

$$\text{Minimize } F_1(x) = S_f\left(\rho\sum_{i=1}^{n}\frac{f_p}{f_d}\{L_i\cdot A_i\}+\rho\sum_{i=1}^{n}\frac{f_d}{f_p}\{L_i\cdot A_i\}\right), \tag{7.6}$$

where n is the number of braces, A is the core area of the brace, i and L are the length of the brace, and i and ρ are the specific gravity of the steel used for beams and columns. S_f is a coefficient to control low-cycle fatigue. If any of the braces buckle due to low-cycle fatigue, this coefficient sets the whole function of the BRB weight ($F_1(x)$) equal to a maximum value to prevent it from being percent in later generations. Only one of the terms of Eq. (7.6) will be calculated depending on which of the applied force or the axial strength of the brace is greater. This is because somehow less or more effect than the required cross-sectional capacity of the brace can be seen in optimization algorithms. Here S_f, f_p, and f_d play the role of the penalty function. In order to consider the effect of the constraints in determining the best population (minimum value for the objective function), a penalty function proportional to the distance of the constraints from the permissible space of the problem's decision is defined, which is then applied to the objective function. As the value of the target function of a population increases, the probability of selecting that population as the best solution decreases.

$$\text{Minimize } F_2(x) = \rho\sum_{j=1}^{n}\left\{L_j\cdot A_j\right\}, \tag{7.7}$$

where n is the number of beams and columns and A is the cross-sectional area of the beam and column i, and L is the length of the beam and column, i and ρ are the specific gravity of the beams and columns.

In Eq. (7.8), the first part of the equation is related to the objective function of uniform relative displacement itself, and the second and third parts are the functions of relative displacement penalty and residual displacement of the permissible values, respectively. This function is such that the less its value is, the better will be the uniformity of drift in the structure.

$$\text{Minimize } F_3(x) = \sum_{j=1}^{n} \left|\text{Avg}_{\text{Drift}} - \text{story}_{\text{Drift}}\right| + W_{\text{Drift}} \left(\sum_{j=1}^{n} \left|\text{Drift}_{\text{story}} - 0.02\right| \right) + W_{\text{Disp}} \left(\sum_{j=1}^{n} \left|\text{Disp}_{\text{story}} - 0.005H\right| \right) \tag{7.8}$$

where $\text{Avg}_{\text{Drift}}$ is the average relative displacement (Drift) of the whole structure, $\text{story}_{\text{Drift}}$ and $\text{Drift}_{\text{story}}$ is the relative displacement (Drift) of each floor, $\text{Disp}_{\text{story}}$ is the residual displacement in each floor, H is the height of the structure and W_{Drift} and W_{Disp} are the reduction coefficients that will be multiplied by the constraints, so that the values of the constraint terms do not dominate the objective function.

Obtaining the values of the objective function requires going through several steps, which include checking the shear flexural adequacy and axial force of the beams and columns. Moreover, the adequacy of the relative displacement in the height of the structure is examined. The steps are shown in Fig. 7.6. Most of the algorithms used in this research support the Pareto system, meaning that in addition to the response values, the response space is also examined. Consequently, infinite and empty answers are not acceptable and interfere with the optimization process. In case of the occurrence of changes in generations, unacceptable numbers in the form of structural geometry are not entered in the OpenSees program. These constraints are considered in the program to obtain acceptable and logical answers.

7.3.5 Evaluation of Optimization Results

The optimization will be done using NSGA_II, MOPSO, MOEA_D, PESA_II, SPEA2 multi-objective optimization algorithms, and the results will be shown by three-dimensional diagrams, each axis of which symbolizes one of the objective functions. The goal is considered simultaneously.

The general process for all optimization algorithms is that by creating an initial population and examining it and using the formulas and methods in each of the algorithms, the best solutions are selected. Then, the features and characteristics in each of the populations are used in the next generations to create the best new populations, and in all these cases, all the target functions are examined. The results shown in the diagrams represent the selected populations in the latest generations or the best solutions in all generations depending on the performance of the algorithm.

Optimization will be done for two 6- and 10-story structures. First, the results of the 6-story structure, then the results of the 10-story structure, and then the performance of both 6- and 10-story structures will be shown in a diagram for each of the algorithms. Finally, the optimization results for all algorithms will be shown simultaneously in a three-dimensional diagram for each of the 6- and 10-story structures.

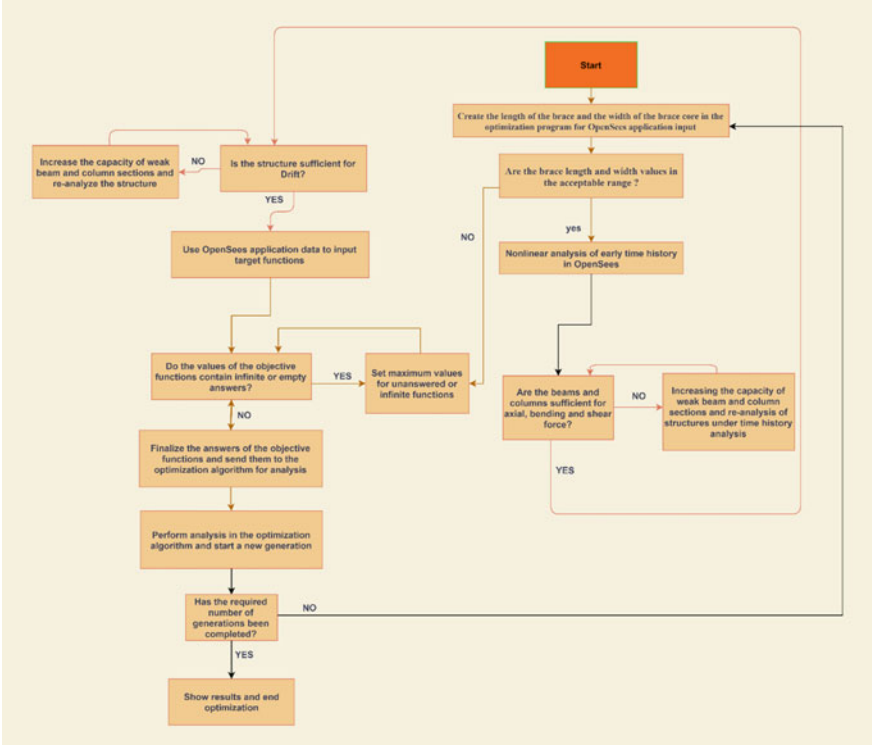


Fig. 7.6 Different steps of obtaining objective functions and the effect of some constraints in OpenSees

7.4 Results of Optimization Algorithms for 6-Story Structures

The optimization results for the 6-story structure under different algorithms are shown below. These algorithms have fine convergence that the values of drift uniformity increase with the increase in the weight of the structure. Furthermore, with the increase in the weight of the BRB, the amount of weight of the structure and the amount of drift uniformity decrease. Moreover, it has been attempted to find a variety of objective functions to achieve the necessary coherence and convergence in the solutions. Similar to a catalog, the obtained solutions are selected from the best solutions obtained in the thousands of analyses performed by the algorithm, and the respondent can choose and use one of the solutions depending on his needs and the values required for the objective functions. Although the range of solutions is different from each other, the algorithm has attempted to select the whole desired range and the best possible solutions and to show the best results in the end, as well as to create the necessary consistency between the solutions.

Figures 7.7, 7.8, 7.9, 7.10, 7.11 and 7.12 show the optimization results for the 6-story structure. It should be noted that the values of the objective functions are adjusted in such a way that lower values will be better solutions. However, given the fact that there are three objective functions and a change in one of the objective functions will cause a change in the others, the algorithm selects a set of solutions at different intervals so that the best solutions could be selected. Meanwhile, the solutions have even better values compared to the results obtained from the algorithm itself, and they are selected and superior solutions.

Considering the results of the diagrams, it can be inferred that with an increase in the weight of the structure, the values of the weight function of the BRB decrease. The population is at the lowest level for the weight of the brace.

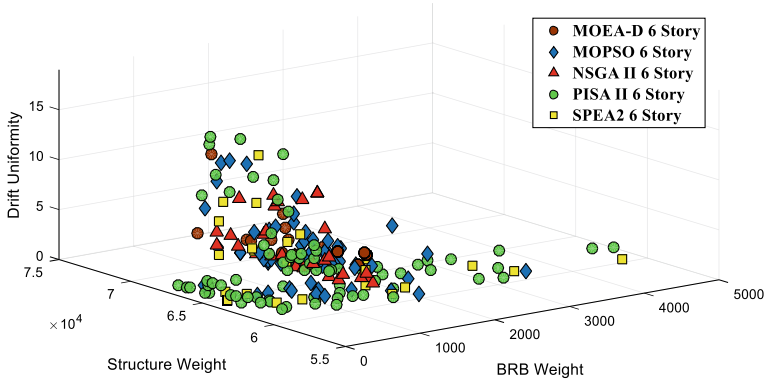


Fig. 7.7 Comparison of the results of all multi-objective optimization algorithms for 6-story structures

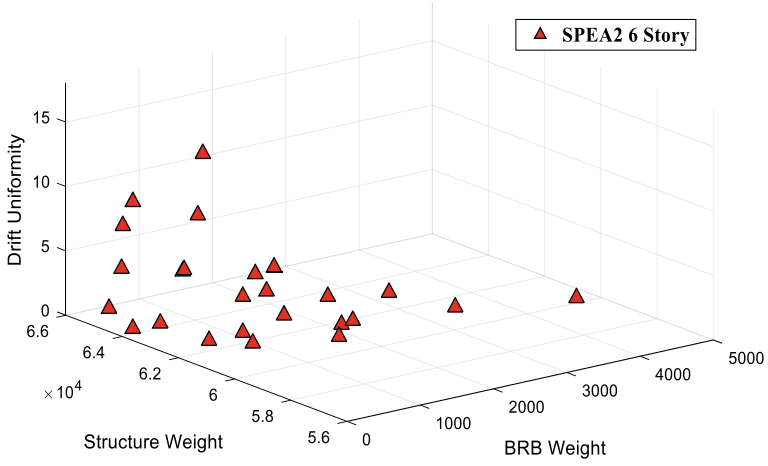


Fig. 7.8 Results of SPEA2 algorithm optimization for 6-story structures

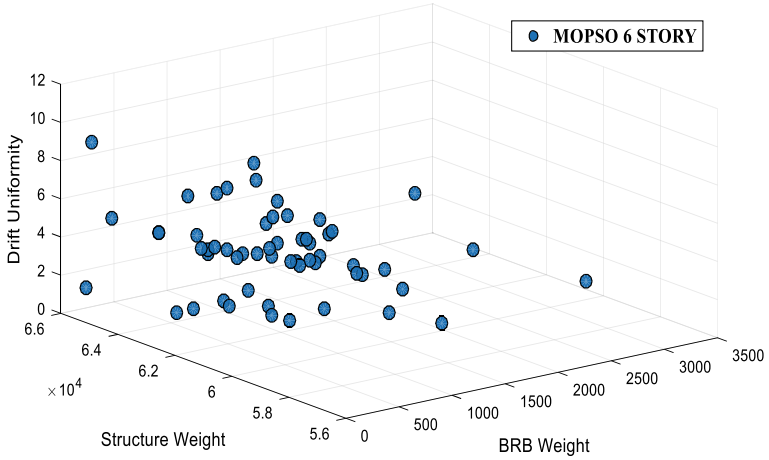


Fig. 7.9 Results of MOPSO algorithm optimization for 6-story structures

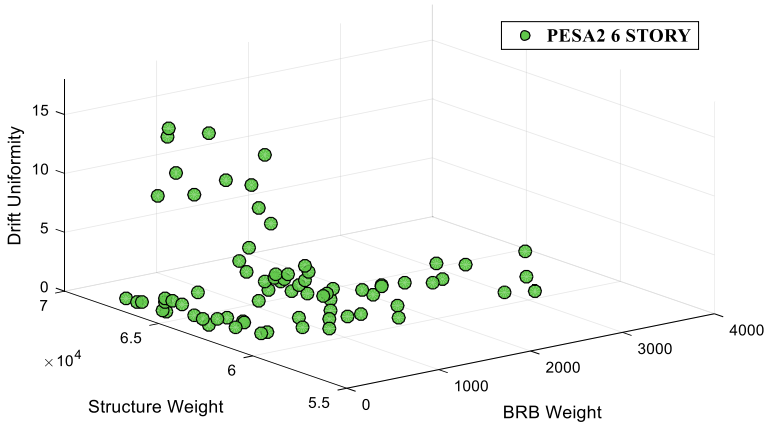


Fig. 7.10 Results of PESA2 algorithm optimization for 6-story structures

These results indicate the accuracy of the solutions obtained from different algorithms relative to each other; and the obtained solutions are in a significant range for the weight function of the BRB, which decreases with the increase in the values of the structure weight function and the relative displacement uniformity according to the diagram.

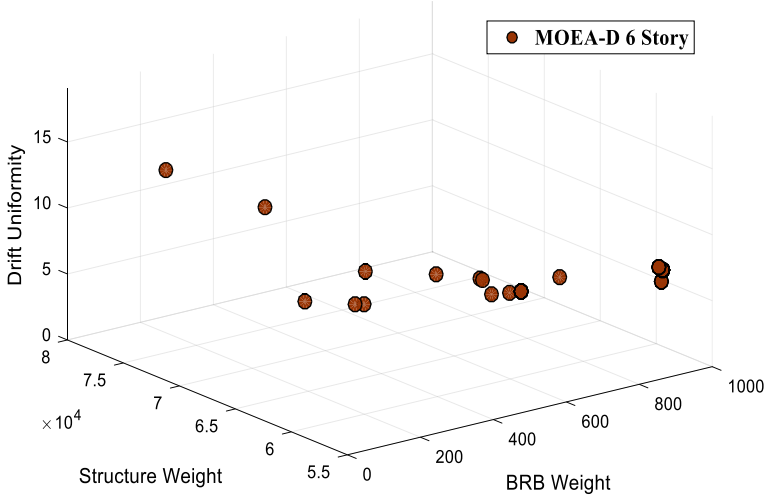


Fig. 7.11 Results of MOEA-D algorithm optimization for 6-story structures

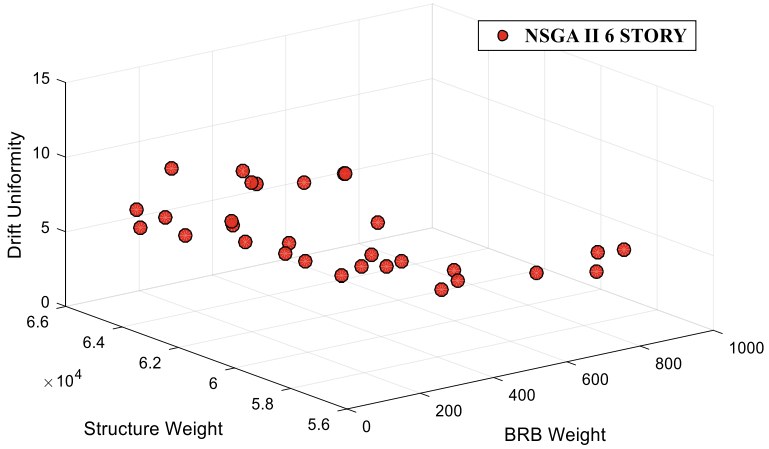


Fig. 7.12 Results of NSGA II algorithm optimization for 6-story structures

7.5 Results of Optimization Algorithms for 10-Story Structures

The solutions are scattered and the algorithm has attempted to achieve the necessary convergence by finding different solutions in different intervals. By decreasing the values of the weight function of the BRB, the values of the weight function of the structure increase. By decreasing the weight of the BRB, the values of the relative

displacement uniformity function also increase. Figures 7.13, 7.14, 7.15, 7.16, 7.17 and 7.18 show the optimization results for the 10-story structure.

Figure 7.13 shows the optimization results for a 10-story structure under all the algorithms used in this research. The results obtained from this diagram also reveal that the results of different algorithms are consistent with each other, indicating that the obtained solutions are in the acceptable range. Regarding the analysis of the diagram itself, it can be stated that with the decrease in the weight of the brace, the values of the structure weight, and the relative displacement of the floors increase, which is common for all the used algorithms. In addition to confirming the solutions obtained by other algorithms, it also shows the trend of the movement of the value

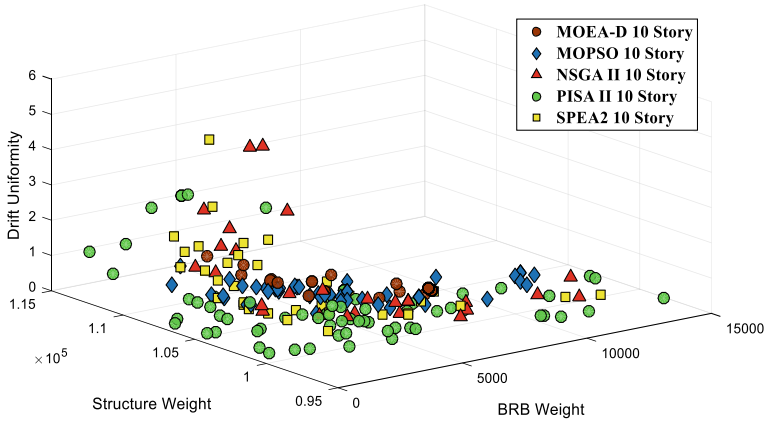


Fig. 7.13 Comparison of the results of all multi-objective optimization algorithms for 10-story structures

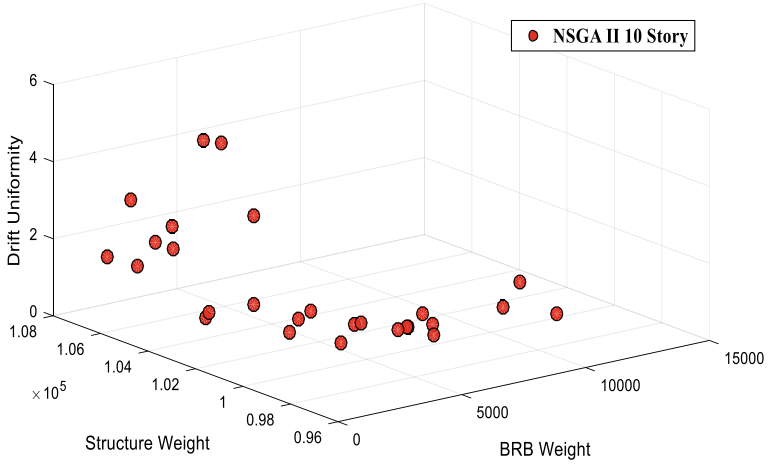


Fig. 7.14 Results of NSGA II algorithm optimization for 10-story structures

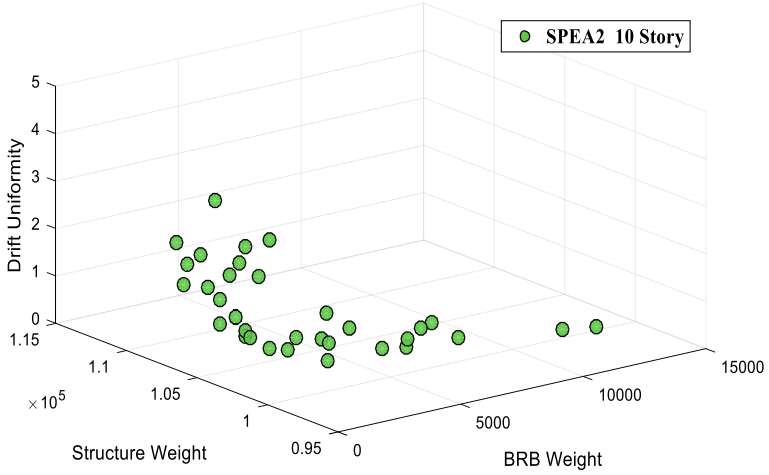


Fig. 7.15 Results of SPEA2 algorithm optimization for 10-story structures

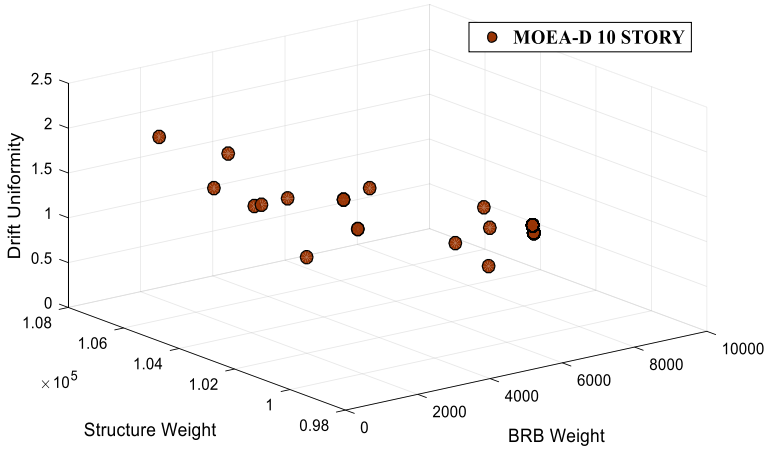


Fig. 7.16 Results of MOEA-D algorithm optimization for 10-story structures

of the objective function. Furthermore, the range of changing the solutions for the objective functions was relatively the same in different algorithms, which indicates the agreement of the solutions in different algorithms.

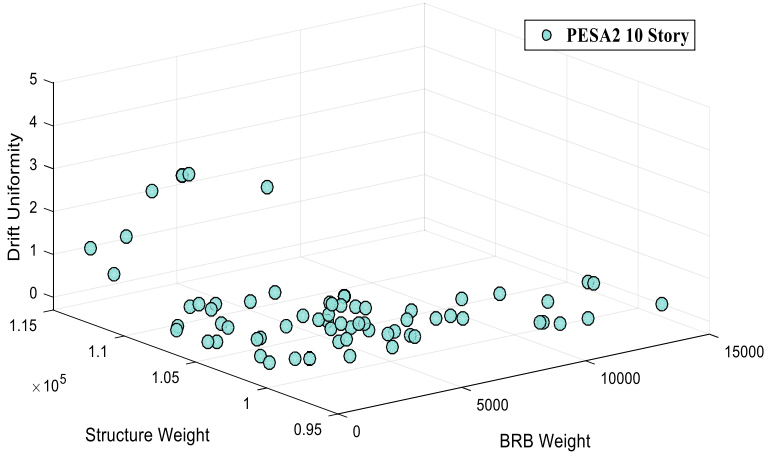


Fig. 7.17 Results of PESA2 algorithm optimization for 10-story structures

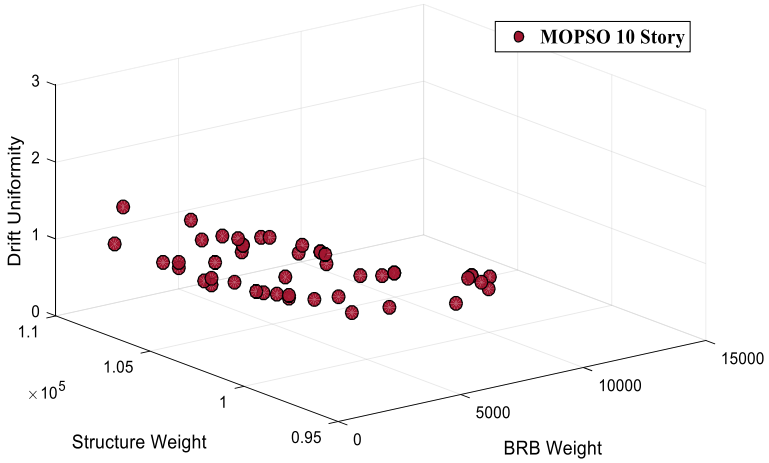


Fig. 7.18 Results of MOPSO algorithm optimization for 10-story structures

7.6 Comparison of the Results of Optimization Algorithms for 6- and 10-Story Structures

Figures 7.19, 7.20, 7.21, 7.22 and 7.23 show the comparison between the optimization of 6-story and 10-story structures. The value of the response intervals for the objective function of the weight of the brace in the 10-story structure is higher than in the 6-story structure, which is due to its higher number of floors. Moreover, the structure weight in a 10-story structure is more than that of a 6-story structure. However, the values of the uniform relative displacement function in a 10-story structure change

between numbers two and zero and in a 6-story structure between zero and eight. This indicates that the algorithm in the 10-story structure could better solution the objective of drift uniformity. Also, in the 6-story structure this value is several times greater than that of the 10-story structure, both of which are less than the amount of static analysis results that will be compared in the following sections.

The value of the BRB weight function is in relatively similar ranges, and it is because of the performance of this brace that the selected range in which the brace could move has been analyzed by every range. Therefore, the range of solutions

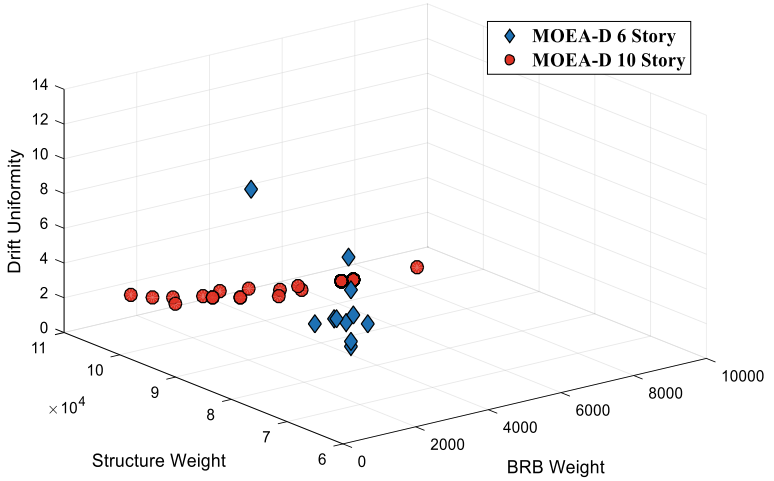


Fig. 7.19 Optimization results of MOEA-D algorithm for 6- and 10-story structures

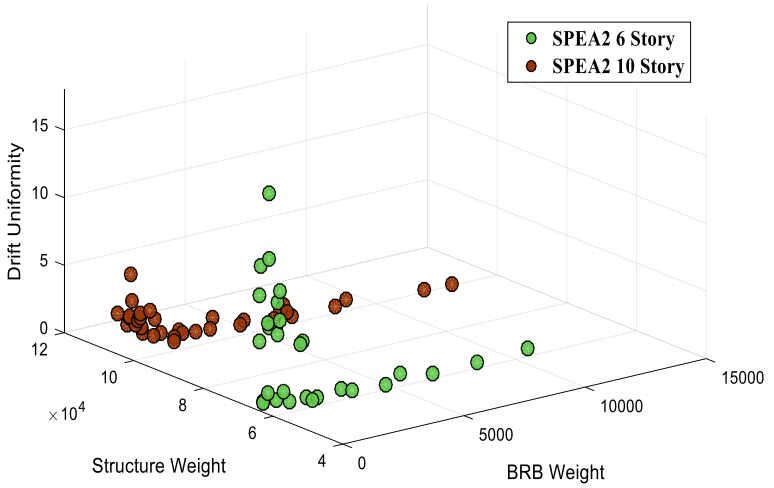


Fig. 7.20 Optimization results of SPEA2 algorithm for 6- and 10-story structures

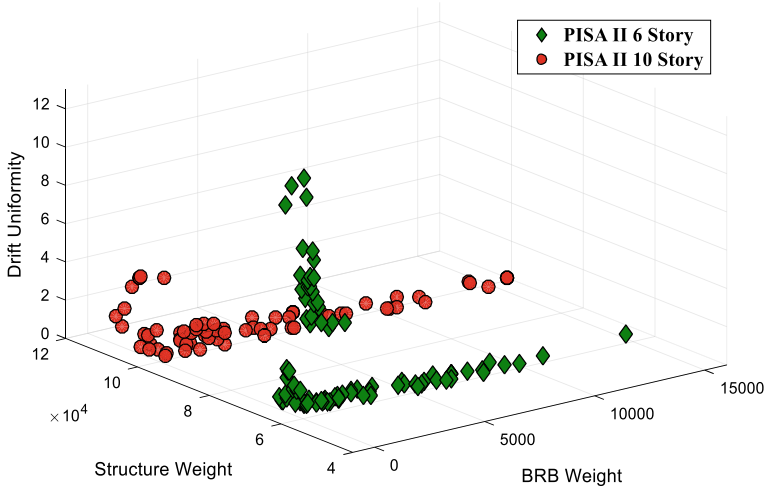


Fig. 7.21 Optimization results of PISA II algorithm for 6- and 10-story structures

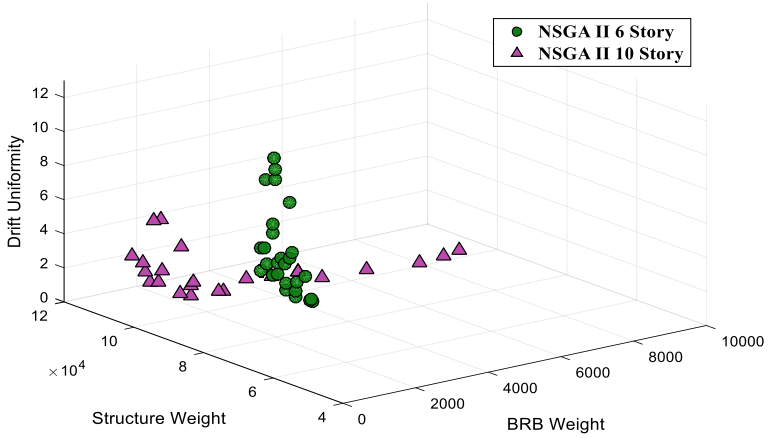


Fig. 7.22 Optimization results of NSGA II algorithm for 6- and 10-story structures

obtained for it is greater than other algorithms. However, the results of the uniform relative displacement function are much better for a 10-story structure than for a 6-story structure, which is similar to other algorithms and performs better in a 10-story structure.

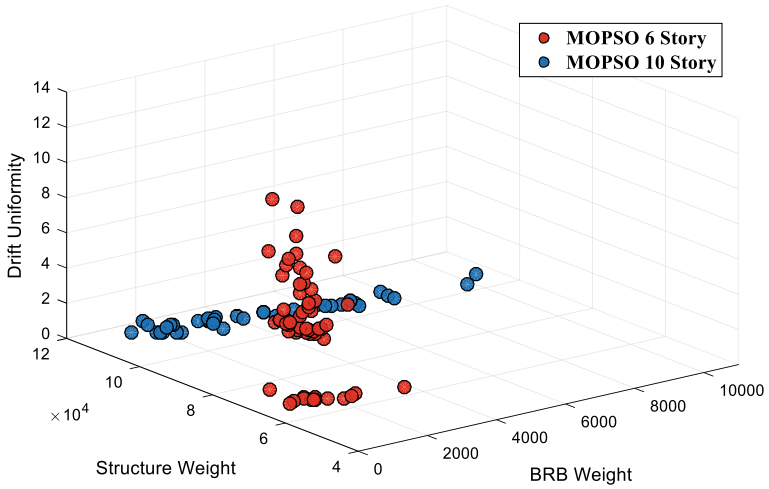


Fig. 7.23 Optimization results of MOPSO algorithm for 6- and 10-story structures

7.7 Comparison of the Results of All Multi-objective Optimization of 6- and 10-Story Structures with the Results of Static Linear Analysis

Due to the nature of the solutions in the multi-objective optimization algorithm, a set of solutions is obtained, each of which shows better performance in one of the objective functions or in several objective functions. Thus, it will not be possible to compare it with one of the results. Therefore, the performance of all algorithms as well as the range of solutions obtained shall be compared. The value of the three objective functions for the 6-story and 10-story structures under static linear analysis is shown in Tables 7.4 and 7.5.

Figures 7.7 and 7.13 show the optimization results of 6- and 10-story structures resulted from optimization algorithms. According to the solution range in 6-story structure for the BRB weight function, the majority of optimization results are less than 50% that of the linear elastic analysis. Moreover, all the solutions obtained

Table 7.4 Values of objective functions for 6-story structures under static analysis

Drift function	Structure weight function	Braces weight function
8.1	81,910.4	2715.6

Table 7.5 Values of objective functions for 10-story structures under static analysis

Drift function	Structure weight function	Braces weight function
7.2	134,288.1	8821.2

from the optimization algorithms are less than the static linear analysis in terms of structure weight. The majority of the results are less than the values corresponding to that of static linear analysis in terms of drift uniformity function.

Considering the range of solutions in the 10-story structure for the BRB weight function, it is shown that the majority of the solutions are in the range of 2000–7000, which is less than those of the 6-story structure for the BRB weight function, as well as all the solutions obtained in the section. The weight of the structure is less than the weight of the structure in linear static analysis. Also, the points obtained for the uniform drift function are all less than the amount of the linear static, indicating better optimization performance for the 10-story structure in the drift uniformity function.

The comparison of the objective functions with the results obtained in the linear static analysis shows that the optimization has been able to show quite significant percentages of reduction of values for each of the objective functions, which shows the importance and efficiency of optimization in the designs. Also, due to the presence of control coefficients in the values of the objective functions, a specific unit cannot be considered for the objective functions, including the results of the linear static analysis.

7.8 Conclusion

In the present research, meta-heuristic algorithms such as NSGA_II, MOPSO, MOEA_D, PESA_II SPEA2 were used in MATLAB software to search for the best solution in a set of possible solutions for BRB cross sections and lengths. Moreover, in order to consider the actual behavior of the structures and to use the maximum capacity of the braces, the structures were analyzed under seven earthquake records in OpenSees software using nonlinear dynamic procedure.

By applying constraints to the optimization problem, a set of possible solutions was generated, and during several optimization steps, an attempt was made to select the solution that results in the least value for the three objective functions of BRB weight, total weight of the structure without a brace and drift uniformity. The constraints considered in this study were the allowable amount of lateral displacement (Drift) in all stories, control of residual displacement, control of flexural and shear forces of beams and columns, and control of failure of the BRB cores due to low-cycle fatigue. Based on the extent each of the constraints were exceeded, a penalty function was defined for each individual in the population, which correspondingly reduced or increased the likelihood of that member being selected.

The frames were modeled in OpenSees software. After creating the length and cross-sectional area using optimization algorithm, the braces entered the OpenSees environment. After examining the structure for satisfying code criteria, the results of nonlinear time history analysis were used to get the output to obtain the values of the objective functions.

Since the use of optimization algorithm and nonlinear analysis are all effective in reducing the cross section and length of BRB, their effect was investigated separately

by the three objective functions, i.e., BRB weight, total weight of bracing structure and relative displacement uniformity in structure height. Using the optimization algorithm, the solutions were directed so that the minimum solutions were selected for the objective functions, while the interaction of these three objective functions is reflected in the generated diagrams.

According to the results of the performed algorithms, it can be concluded that reducing the value of the weight function of the BRB increases the total weight of the structure and increases the value of the drift uniformity function.

Moreover, by comparing the optimization results with the results of linear static analysis, it was found that the values obtained in optimization process reduce the values of bracing weight, structure weight and drift uniformity with quite significant percentages. This further shows the importance of optimization in designs.

These diagrams generated in this research act as a catalog showing the effect of each of the objective functions on each other. One can select this structure from the diagram and take the length of the brace and area of the core to gain an optimum design.

References

Ali Razavi S, Shemshadian ME, Mirghaderi SR, Ahlehagh S (2011) Seismic design of buckling restrained braced frames with reduced core length. In: Proceeding of the structural engineers world congress. Italy

Ali RS, Mirghaderi SR, Hosseini A (2014) Experimental and numerical developing of reduced length buckling-restrained braces. Eng Struct 77:143–160

Ali Razavi S, Kianmehr A, Hosseini A, Rasoul Mirghaderi S (2018) Buckling-restrained brace with CFRP encasing: mechanical behavior and cyclic response. Steel Compos Struct 27:675–689

American Society of Civil Engineers (2016) ASCE/SEI 7–10, Minimum design loads for buildings and other structures. Author, Reston, VA

BHRC—Standard 2800 [2014] Iranian code of practice for seismic resistant design of building. Iranian building codes and standards, fourth revision, (in Persian)

Clark P, Aiken L, Kasai K, Ko E, Kimura L (1999) Design procedures for buildings incorporating hysteretic damping devices. In: Proceedings of the 68th annual convention, structural engineers, pp 355–371

Deb K (2001) Multi-objective optimization using evolutionary algorithms. Chichester, UK

Hosseini Hashemi B, Alirezaei M, Ahmadi H (2016) Analysis and design of steel structures: with emphasis on the limit state method, design principles with practical examples. Danesh Atrak, Bojnourd

http://peer.berkeley.edu/nga

Lopez W, Sabelli R (2004) Seismic design of buckling-restrained. Braced Frames. Steel Tips

Miner MA (1945) Cumulative damage in fatigue. J Appl Mech 12:159–164

Stephens R, Fuchs H, Fatemi A, Stephens R (2000) Metal fatigue in engineering, 2nd edn. Wiley-Interscience

Tauer F, Spacone E, Filippou F (1991) A fiber beam-column element for seismic response analysis of reinforced concrete structures. Earthq Eng Res Center Eng UCBEERC-91/17

Uang C, Nakashima M, Bozorgnia Y, Bertero V (2003) Steel buckling restrained braced frames. CRC Press, p chapter 16

Uriz P (2005) Towards earthquake resistant design of concentrically braced steel buildings. PhD Thesis, Department of Civil and Environmental Engineering: University of California, Berkeley, California

Part II
Topology Optimization and Design of Structures Under Dynamic Conditions Using Finite Element Methods

Chapter 8
Topology Optimization in Linear Elasticity, Plasticity and Fracture Mechanics

J. Desai

Abstract This chapter presents the theoretical and the numerical study of shape and topology optimization for linear elasticity, plasticity and fracture using the level-set method. The governing equation of linear elasticity is theoretically shape-differentiable, while that of plasticity and damage is not shape-differentiable. To overcome the non-differentiability issue for the latter, we construct an approximation by penalization and regularization. For the three physics, the shape sensitivity analysis is performed using Céa's technique. The shape optimization is implemented numerically using a level-set method with body-fitted remeshing, which captures the boundary of the shapes while allowing for topology changes. Numerical results are presented in 2D and 3D.

Keywords Shape and topology optimization · Level-set method · Damage model · Fracture mechanics · Plasticity with isotropic and kinematic hardening

8.1 Introduction

Structural optimization aims at determining mechanical structures with desired properties. The industrial goal of this optimization is to design structures that are stiff, resistant to external loads and lightweight, hence cheaper or less energy-consuming. Depending on the number of parameters involved in the structural optimization, it can be broadly divided in three categories,

1. Parametric optimization:
   The shape is parametrized using a limited number of variables, for instance, the "thickness" of a structure (Pedersen 1991), or the truss configuration (Gomes 2011). Given the simplicity of parametric optimization, it has a wide variety of industrial applications

J. Desai (✉)
CNRS, Université de Paris and Sorbonne Université, LJLL, Paris 75006, France
e-mail: jeet@3dware.co.in

I. R. Kale and A. Sadollah (eds.), *Optimization Methods for Structural Engineering*,
Engineering Optimization: Methods and Applications,
https://doi.org/10.1007/978-981-99-2378-6_8

2. Geometric shape optimization:
   The shape is parametrized using the boundary of the structure. This parametrization can be brought about using the mesh boundary (Allaire and Pantz 2006), splines (Seo et al. 2010), NURBS (non-uniform rational basis splines) (Costa et al. 2019), etc. This method is very robust, except that it cannot introduce topological changes.
3. Topology optimization:
   By topology of a structure, we mean the number of holes in the structure in two dimensions and the Euler characteristic in three dimensions. Two structures have the same topology, if one structure can be warped smoothly into another without changing its number of holes. In topology optimization, the topology is defined by a density function (Bendsoe and Sigmund 2004), a level-set (Allaire et al. 2020) or a characteristic function. The function is then evolved using a heuristic approach or a shape derivative, and the topology of the structure is optimized.

Evidently, there is gradual rise in the number of optimization parameters as one moves from parametric optimization to topology optimization. Topology optimization is what we deal with in this chapter.

8.2 Existing Methods

A holistic review on the several topology optimization methods can be found in Sigmund and Maute (2013). Any topology optimization method must facilitate a "way" to define the topology. There are several ways, which exist in literature. The oldest known method is homogenization (Allaire 2012; Bendsøe and Kikuchi 1988; Hassani and Hinton 1998a, b). This method results in an optimal shape, that is a density function taking values between zero and one. For instance, a 2D cantilever beam (optimized for compliance) obtained via this method is plotted in Fig. 8.1. As can be seen from the image, there is no clear description of a boundary and there are large gray zones (the region where the optimized density function takes a value between 0 and 1).

In order to overcome this drawback of homogenization, one resorts to a penalization approach, where the density function (representing the shape) is forced to takes binary values, 0 or 1. One of the most successful penalization approaches is the so-called solid isotropic material with penalization (SIMP) approach (Bendsøe and Sigmund 1999; Bendsoe and Sigmund 2004). In this approach, the density of the structure is raised to an exponent p. The value of this exponent is chosen heuristically, typically 3. The previous 2D cantilever beam obtained using SIMP is plotted in Fig. 8.1. As seen in the figure, the density function seems to have lesser gray zones. This marks the success of SIMP, making it one of the most widely used methods in literature and in commercial packages like Optistruct and ANSYS. However, one very well-known drawback of SIMP is the fact that there are still a few gray zones in the optimized density function. These gray zones are hard to interpret from the

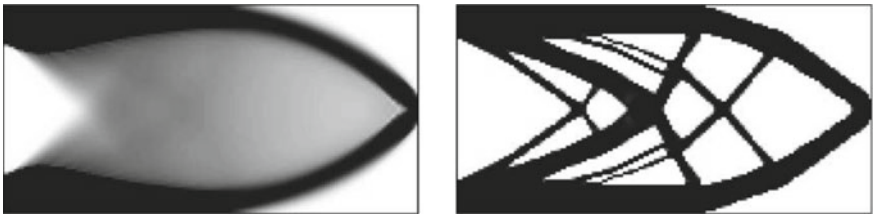


Fig. 8.1 Cantilever computed with the homogenization method. Composite (left) and penalized solution (right), taken from Allaire et al. (2002)

fabrication point of view. Incorrect interpretation can result in loss of optimality of the optimized density function. There are several ways to avoid these gray zones, the most famous being the density-filter developed by Sigmund (2007). Another possibility is to project the density function and eliminate the gray zones (Wang et al. 2011). There are several approaches that do not rely on a density function. Evolutionary structural optimization (ESO) is a heuristic method, where one removes material from low stressed regions in the structure (Xie and Steven 1993). Bi-evolutionary structural optimization (BESO) is a modification of ESO, where one can in addition add material wherever needed (Huang and Xie 2009) in a fixed design space. Here, the boundary of the structure is clearly defined as there are no density functions. These two methods are very easy to implement and known to work well for compliance minimization. These methods can be combined with extended finite element method (XFEM) approach to remesh the structure at every shape optimization iteration (Liu et al. 2016; Martínez-Frutos and Herrero-Pérez 2018).

One strength of ESO/BESO is the fact that it can introduce holes in the structure and hence bring about topological changes quite easily. But the two methods are heuristic. A mathematically rigorous way of introducing holes in the structure is using the so-called topological derivative (Sokolowski and Zochowski 1999). In this approach, one computes the topological derivative everywhere in the structure and determines the most feasible location of adding an infinitesimal hole. The topological derivative can then be considered in conjunction with the shape derivative of the structure to optimize the topology of the structure (Allaire et al. 2011).

Another method to deal with topological changes is using the level-set function. The level-set function is a continuous function, whose zero isovalue defines the shape (see Eq. (8.18) for a precise definition). One can then work with this level-set function, instead of dealing only with the boundary of the shape like in typical shape optimization approaches. This facilitates changes in topology. The level-set method was integrated into the shape and topology optimization framework in Sethian and Wiegmann (2000), Osher and Santosa (2001), Allaire et al. (2002) and Wang et al. (2003), and developed over the years, finding several applications such as robust design (Gournay et al. 2008; Martínez-Frutos et al. 2016), manufacturing constraints (Allaire et al. 2016), composite design (Allaire and Delgado 2016). In the method proposed by Allaire et al. (2002) and Wang (2003), the evolution of the level-set

function is brought about by solving the transport equation (see Eq. 8.19). By doing so, the zero isovalue of the level-set function (representing the boundary of the structure) moves, and the creation of holes is facilitated. Another way to bring about the evolution of the level-set function is by solving the convection-diffusion equation, proposed by Yamada (2010). Using this approach, one can create holes inside the structure without waiting for the boundary to move and collapse, unlike in the case of transport equation.

In most level-set-based approaches, the magnitude of velocity of advection is taken equal to the shape derivative and not the topological derivative (which is cumbersome to calculate). Hence, we study the shape optimization problem in the framework of the level-set method for topology optimization. In the next section, we discuss about the mathematical formulation of the shape optimization problem for linear elasticity.

8.3 Mathematical Framework of Shape Optimization

The shape or the structure we deal with in this chapter is represented by an open set $\Omega \subset \mathbb{R}^d$ ($d = 2$ or 3 being the dimension of the problem). The structure has a boundary $\partial\Omega = \Gamma_N \cup \Gamma_D \cup \Gamma$, such that it is fixed on Γ_D and loaded on Γ_N as shown in Fig. 8.2.

8.3.1 Problem Formulation

The open set $\Omega \in \mathcal{U}_{\text{ad}}$, which is a set of admissible shapes (typically defined by the user, see Definition (8.4) for an example). As stated earlier, via structural optimization, we wish to determine structures with desired mechanical properties. These mechanical properties can be viewed as an objective function denoted by $J(\Omega)$, that

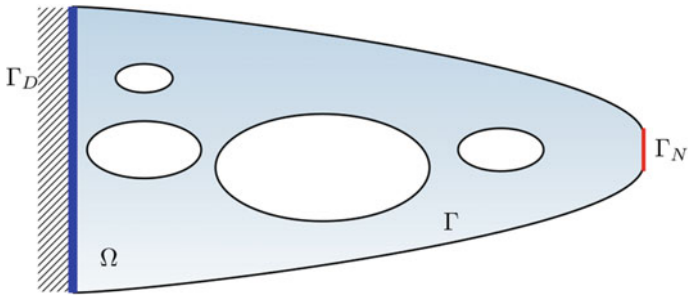


Fig. 8.2 Shape Ω and boundary conditions

ought to be minimized. A general mathematical formulation of a shape optimization problem then reads

$$\inf_{\Omega \in \mathcal{U}_{\text{ad}}} J(\Omega). \tag{8.1}$$

One particular form of $J(\Omega)$ that one typically considers is

$$J(\Omega) = \int_{\Omega} m(\boldsymbol{u}(\Omega))\, \mathrm{d}x + \int_{\partial\Omega} p(\boldsymbol{u}(\Omega))\mathrm{d}s, \tag{8.2}$$

where $\boldsymbol{u}(\Omega)$ is solution to a partial differential equation (e.g., solution to equation (8.9)), and $m(\cdot)$, $p(\cdot)$ are functions that are assumed to be at least $\mathcal{C}^1$ smooth such that m', p' have a linear growth rate. These regularity assumptions are made on $m(\cdot)$ and $p(\cdot)$ so that the objective function is well-defined.

The above expression of the objective $J(\Omega)$ can represent several mechanical properties such as the total compliance, elastic energy, stress as well as geometric properties such as volume and perimeter. One objective function that almost every researcher working on shape optimization deals with is compliance, given by

$$J(\Omega) = \int_{\Omega} \boldsymbol{f} \cdot \boldsymbol{u}(\Omega)\mathrm{d}x + \int_{\Gamma_N} g \cdot \boldsymbol{u}(\Omega)\ \mathrm{d}s, \tag{8.3}$$

where $\boldsymbol{f}$ is the body force and g, the surface force applied on the structure. Compliance is the work done by the external force. Lesser compliance implies a lesser deformation at the zone of application of the force (Γ_N) and in the shape Ω and hence results in a lesser energy transferred to the structure from the external force. In the framework of linear elasticity, compliance is directly proportional to the inverse of rigidity of the structure. Thus, minimizing compliance implies maximizing rigidity (only in the case of linear elasticity).

Typically, the structure or the shape Ω is conceived inside a fixed design space $D \subset \mathbb{R}^d$ (e.g., see Fig. 8.4). The set of admissible shapes $\mathcal{U}_{\text{ad}}$ is user-defined. Here, it is defined as

$$\mathcal{U}_{\text{ad}} = \left\{ \Omega \subset D : \int_{\Omega} \mathrm{d}x = V_t \right\}, \tag{8.4}$$

where V_t is a target volume. The existence of optimal shapes is subject to the choice of $\mathcal{U}_{\text{ad}}$. For instance, if one puts a perimeter constraint on Ω in (8.4), one can show the existence of a global minimum to the problem (8.1). In general, the question of existence of optimal shapes Ω is theoretically involved and we shall not dwell into it (see Henrot and Pierre (2018) and Chenais (1975) for more details). Rather, we content ourselves with computing numerical minimizers, using a gradient-descent method. A priori, this results in the determination of a local minimum.

8.3.2 Hadamard's Approach

Gradient-based optimization methods demand an explicit expression of the derivative of the objective function $J(\Omega)$. The gradient in the context of shape optimization is based on the notion of the Hadamard shape derivative (Allaire 2007; Allaire et al. 2020; Henrot and Pierre 2018; Sokolowski and Zolésio 1992). As shown in Fig. 8.3, starting from a smooth domain Ω, the perturbation of the domain is expressed as

$$\Omega_{\boldsymbol{\theta}} = (I_d + \boldsymbol{\theta})(\Omega),$$

where $\boldsymbol{\theta} \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ and I_d is the identity map. It is well-known that when the $W^{1,\infty}$ norm of $\boldsymbol{\theta}$ is sufficiently small, the map $I_d + \boldsymbol{\theta}$ is a diffeomorphism in $\mathbb{R}^d$. With this perturbation of the domain, one can define the notion of a Fréchet derivative for the function $J(\Omega)$.

Definition 1 The shape derivative of $J(\Omega)$ at Ω is defined as the Fréchet derivative in $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ evaluated at 0 for the mapping $\boldsymbol{\theta} \mapsto J((I_d + \boldsymbol{\theta})(\Omega))$ i.e.,

$$J((I_d + \boldsymbol{\theta})(\Omega)) = J(\Omega) + J'(\Omega)(\boldsymbol{\theta}) + o(\boldsymbol{\theta}) \quad \text{with} \quad \lim_{\boldsymbol{\theta} \to 0} \frac{o(\boldsymbol{\theta})}{\|\boldsymbol{\theta}\|_{W^{1,\infty}}} = 0,$$

where $J'(\Omega)$ is a continuous linear form on $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$.

For the objective function (8.2), the shape derivative $J'(\Omega)$ can be written using a function $j(\Omega)$ such that

$$J'(\Omega)(\boldsymbol{\theta}) = \int_{\partial\Omega} j(\Omega)\boldsymbol{\theta} \cdot \boldsymbol{n} \, \mathrm{d}x$$

($\boldsymbol{n}$ being the outward normal to the Ω). One can then set the descent direction to $\boldsymbol{\theta} = -t\, j(\Omega)\boldsymbol{n}$, where $t \in \mathbb{R}^+$ is the descent step. For the new shape $\Omega_t = (I_d + \boldsymbol{\theta})\Omega$, one can formally write

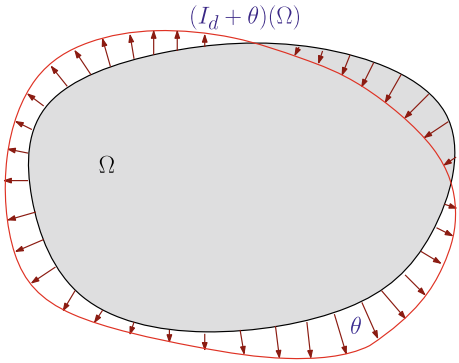


Fig. 8.3 Perturbation of Ω using Hadamard's method

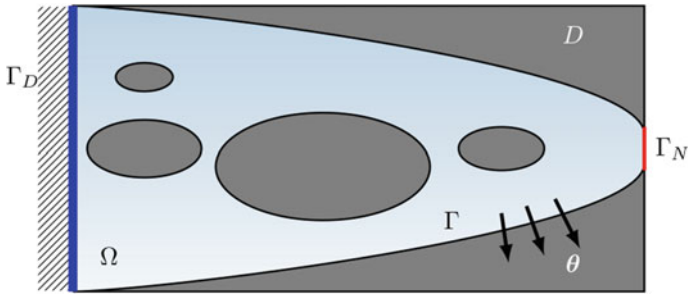


Fig. 8.4 Design domain D and the shape Ω

$$J(\Omega_t) = J(\Omega) - t \int_{\partial\Omega} j(\Omega)^2 \mathrm{d}s + \mathcal{O}(t^2).$$

Choosing a descent step t sufficiently small, one can show that the left hand side of the above is smaller than $J(\Omega)$, which guarantees a descent. Given an initial shape Ω, one can then apply the above gradient $J'(\Omega)$, and move the shape iteratively, minimizing the objective. In general, nothing ensures that our iterations would converge. Moreover, even in the case of convergence, one ends up in a final shape, which is often a local minimum, depending on the choice of the initial shape.

As shown in Fig. 8.4, the blue region represents the shape Ω, and the blue and gray area represent the design space D. Typically when a structure is designed, the clamped and the forced boundaries are assumed to be non-optimizable. Hence in all our optimizations, only Γ is allowed to move along $\boldsymbol{\theta}$ as shown in Fig. 8.4. This constraint can be incorporated by introducing the space

$$W_0^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d) = \{\boldsymbol{\theta} \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d) : \ \boldsymbol{\theta} = \mathbf{0} \text{ on } \Gamma_N \cup \Gamma_D\} \tag{8.5}$$

and state a classical lemma, which shall use for the derivation of shape derivative.

Lemma 1 *Let Ω be a smooth bounded open set and $\varphi, \psi, \xi \in W^{1,1}(\mathbb{R}^d, \mathbb{R})$. Define $J_V(\Omega)$, $J_S(\Omega)$ and $J_N(\Omega)$ by*

$$J_V(\Omega) = \int_{\Omega} \varphi(x) dx \qquad J_S(\Omega) = \int_{\Gamma} \xi(x)\, ds, \qquad J_N(\Omega) = \int_{\Gamma_N} \psi(x)\, ds$$

then the functions $J_V(\Omega)$, $J_S(\Omega)$ and $J_N(\Omega)$ are differentiable at Ω in the direction $\boldsymbol{\theta} \in W_0^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ with the derivative

$$J_V'(\Omega)(\boldsymbol{\theta}) = \int_{\Gamma} \boldsymbol{\theta} \cdot \boldsymbol{n} \varphi \, ds, \quad J_S'(\Omega)(\boldsymbol{\theta}) = \int_{\Gamma} \boldsymbol{\theta} \cdot \boldsymbol{n} \left(\frac{\partial \xi}{\partial n} + H\xi \right) ds, \quad J_N'(\Omega)(\boldsymbol{\theta}) = 0,$$

where H is the mean curvature of $\partial\Omega$ (assumed to be smooth).

In Lemma 1, one differentiates a functional whose integrand functions are independent of Ω. To determine a shape derivative, one ought to differentiate (8.2), whose integrand functions ($m(\boldsymbol{u}(\Omega))$ and $p(\boldsymbol{u}(\Omega))$) are dependent on Ω. This dependency arises from the fact that $\boldsymbol{u}(\Omega)$ is solution to the partial differential equation defined on Ω. One thus needs to determine the derivative of $\boldsymbol{u}(\Omega)$ with respect to Ω. In this regard, there exist two notions of derivatives: Eulerian and Lagrangian. We do not expound on the Eulerian and Lagrangian derivatives (see Allaire (2007) for more details). Instead, we use the "fast derivation" method, proposed by Céa (1986). This method is presented for linear elasticity in Sect. 8.3.3, plasticity in Sect. 8.6.4 and fracture in Sect. 8.8.3.

8.3.3 Shape Derivative Computation for Linear Elasticity

Structural optimization is often performed assuming the structure to be linear with respect to external forcing. Linear elasticity is a mathematical model that assumes all mechanical deformations to be linear with respect to the external force. Throughout this chapter, we use the convention of representing all scalar mathematical entities by lowercase italic alphabets. The structure is represented by Ω as shown in Fig. 8.2. Let $\mathcal{M}_s^d$ represent the set of symmetric $d \times d$ matrices, $\mathbb{I}$ represent the fourth-order identity tensor of dimension d, $\boldsymbol{u} : \Omega \to \mathbb{R}^d$ denote the displacement field, $\boldsymbol{\sigma}$ denote the second order stress tensor, $\boldsymbol{\varepsilon}(\boldsymbol{u})$ denote the second order strain tensor, and $\boldsymbol{n}$ denote the outward normal to $\partial\Omega$. $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}(\boldsymbol{u})$ are second order tensors of dimension d. The structure when subjected to a body force $\boldsymbol{f} : \Omega \to \mathbb{R}^d$ and a surface force $g : \Gamma_N \to \mathbb{R}^d$ respects the momentum balance equation:

$$\begin{aligned} \operatorname{div}(\boldsymbol{\sigma}) + \boldsymbol{f} &= \boldsymbol{0} && \text{in } \Omega, && \text{(8.6a)} \\ \boldsymbol{\sigma} \cdot \boldsymbol{n} &= g && \text{on } \Gamma_N, && \text{(8.6b)} \\ \boldsymbol{\sigma} \cdot \boldsymbol{n} &= \boldsymbol{0} && \text{on } \Gamma, && \text{(8.6c)} \\ \boldsymbol{u} &= \boldsymbol{0} && \text{on } \Gamma_D. && \text{(8.6d)} \end{aligned}$$

In the framework of linear elasticity, the mechanical displacement is assumed to be small. In this case, the strain tensor $\boldsymbol{\varepsilon}(\boldsymbol{u})$ is given by

$$\boldsymbol{\varepsilon}(\boldsymbol{u}) = \frac{1}{2}\left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T\right),$$

and the constitutive law is given by

$$\boldsymbol{\sigma} = \mathbb{C}\boldsymbol{\varepsilon}(\boldsymbol{u}), \tag{8.7}$$

where $\mathbb{C}$ is the Hooke's tensor. The Hooke's tensor $\mathbb{C}$ is a fourth-order tensor of dimension d and for a structure composed of an isotropic material, given by

$$\mathbb{C} = \lambda \mathbf{1} \otimes \mathbf{1} + 2\mu \mathbb{I},$$

where λ, μ are the Lamé coefficients and $\mathbf{1}$ is second order identity tensor of dimension d. For most materials, the tensor $\mathbb{C}$ is coercive, i.e., there exists $c_0 > 0$ such that,

$$\mathbb{C}\boldsymbol{\xi} : \boldsymbol{\xi} \geq c_0 |\boldsymbol{\xi}|^2 \qquad \forall \boldsymbol{\xi} \in \mathcal{M}_s^d. \tag{8.8}$$

Then using Eqs. (8.6) and the constitutive equation (8.7), the linear elasticity model is given by

$$\begin{aligned} \operatorname{div}(\mathbb{C}\boldsymbol{\varepsilon}(\boldsymbol{u})) + \boldsymbol{f} &= \mathbf{0} && \text{in } \Omega, && (8.9a)\\ \mathbb{C}\boldsymbol{\varepsilon}(\boldsymbol{u}) \cdot \boldsymbol{n} &= g && \text{on } \Gamma_N, && (8.9b)\\ \mathbb{C}\boldsymbol{\varepsilon}(\boldsymbol{u}) \cdot \boldsymbol{n} &= \mathbf{0} && \text{on } \Gamma, && (8.9c)\\ \boldsymbol{u} &= \mathbf{0} && \text{on } \Gamma_D. && (8.9d) \end{aligned}$$

For applying Céa's method (of computing the shape derivative), the above must be converted to its weak form. Multiplying the above by a smooth test function $\boldsymbol{v}$, integrating it by parts, and using the boundary conditions prescribed in (8.9), we get the weak form

$$\int_\Omega \mathbb{C}\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v}) \, \mathrm{d}x = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} \, \mathrm{d}x + \int_{\Gamma_N} g \cdot \boldsymbol{v} \, \mathrm{d}s \tag{8.10}$$

In the context of shape optimization for linear elasticity, the above weak form is called as the state equation and the solution $\boldsymbol{u}$ to it, the state solution. In order to find the state solution, we define the displacement space

$$V = \{\boldsymbol{u} \in H^1(\Omega)^d \text{ such that } \boldsymbol{u} = \mathbf{0} \text{ on } \Gamma_D\}. \tag{8.11}$$

This space is complete with a norm $\|\cdot\|_V$ defined as

$$\|\boldsymbol{u}\|_V^2 = \int_\Omega \boldsymbol{u} \cdot \boldsymbol{u} \, \mathrm{d}x + \int_\Omega \nabla \boldsymbol{u} : \nabla \boldsymbol{u} \, \mathrm{d}x.$$

Let $a : V \times V \to \mathbb{R}$ be a bilinear form, defined as

$$a(\boldsymbol{u}, \boldsymbol{\varphi}) = \int_\Omega \mathbb{C}\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{\varphi}) \mathrm{d}x. \tag{8.12}$$

and $l : V \to \mathbb{R}$ be a linear form given by

$$l(\boldsymbol{\varphi}) = \int_{\Omega} \boldsymbol{f} \cdot \boldsymbol{\varphi} \, \mathrm{d}x + \int_{\Gamma_N} g \cdot \boldsymbol{\varphi} \, \mathrm{d}s.$$

Then using the bilinear form $a(\cdot, \cdot)$ and the linear form $l(\cdot)$, and choosing the test function $\boldsymbol{v} \in V$, the weak form (8.10) can be written in a compact manner as: find $\boldsymbol{u} \in V$ such that

$$a(\boldsymbol{u}, \boldsymbol{v}) = l(\boldsymbol{v}) \qquad \forall \boldsymbol{v} \in V. \tag{8.13}$$

Given the coercivity of the Hooke's tensor and the Korn's inequality, one can show that

$$\exists \alpha > 0 \quad \text{such that} \quad \alpha \, \|\boldsymbol{v}\|^2_{H^1(\Omega)} \leq a(\boldsymbol{v}, \boldsymbol{v}).$$

Thus, the bilinear form $a(\cdot, \cdot)$ is coercive, or stable. If $\boldsymbol{f}, g \in V'$ (the dual space of V), one can show that the linear form $l(\cdot)$ is continuous. The coercivity of $a(\cdot, \cdot)$ and the continuity of $l(\cdot)$ allows us to apply Lax-Milgram's Lemma to (8.13) leading to the existence of a unique solution $\boldsymbol{u} \in V$.

We now consider the objective function (8.2) [where $\boldsymbol{u}(\Omega)$ is the solution to (8.13)]. The shape derivative for this objective is determined using Céa's method. Explained in one line, this method is a Lagrange multiplier method where the objective (8.2) is minimized with the differential equation (8.9) viewed as a constraint. In this method, one constructs a Lagrangian by assuming the variable $\boldsymbol{u}$ to be independent of Ω, and by introducing a Lagrange multiplier $\boldsymbol{v}$ (a variable defined in $\mathbb{R}^d$). One then proceeds to apply the optimality condition on the Lagrangian with respect to its variables, $\boldsymbol{u}$, $\boldsymbol{v}$, Ω. The optimality condition on the resulting Lagrangian results in an adjoint equation (a partial differential equation to determine $\boldsymbol{v}$). For this optimality condition to hold, the adjoint equation must be well-posed. Furthermore, to determine the shape derivative, one must differentiate the Lagrangian with respect to Ω at $\boldsymbol{u}(\Omega)$. This differentiation is justified if $\boldsymbol{u}(\Omega)$ is differentiable with respect to the shape Ω. This is a strong requirement and often taken for granted. Hence, Céa's method can be used to determine the shape derivative if the adjoint equation is well-posed and $\boldsymbol{u}(\Omega)$ is differentiable respect to Ω.

We now demonstrate the computation of shape derivative for linear elasticity using Céa's method.

Theorem 1 *Let $\Omega \subset \mathbb{R}^d$ be a smooth bounded open set. Let $\boldsymbol{f} \in H^1(\mathbb{R}^d)^d$, $g \in H^2(\mathbb{R}^d)^d$ and $\boldsymbol{u}(\Omega) \in V$ the solution to (8.13). Then the shape derivative of $J(\Omega)$ along $\boldsymbol{\theta} \in W_0^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$, $J'(\Omega)(\boldsymbol{\theta})$ is given by*

$$J'(\Omega)(\boldsymbol{\theta}) = \int_{\Gamma} \boldsymbol{\theta} \cdot \boldsymbol{n} \Big(m(\boldsymbol{u}) + \mathbb{C}\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v}) - \boldsymbol{f} \cdot \boldsymbol{v} \Big) ds \tag{8.14}$$

where $\boldsymbol{v}(\Omega) \in V$ is the solution to the adjoint problem,

$$a(\boldsymbol{v}, \boldsymbol{\varphi}) = -\int_{\Omega} m'(\boldsymbol{u}) \cdot \boldsymbol{\varphi}\, dx - \int_{\Gamma_N} p'(\boldsymbol{u}) \cdot \boldsymbol{\varphi}\, ds \quad \forall \boldsymbol{\varphi} \in V, \tag{8.15}$$

which can be easily shown to be well-posed.

Proof Define space $\tilde{V}$ (analog of the space V given in (8.11))

$$\tilde{V} = \{\boldsymbol{u} \in H^1(\mathbb{R}^d)^d \text{ such that } \boldsymbol{u} = 0 \text{ on } \Gamma_D\}. \tag{8.16}$$

As per Céa's technique (J. Céa 1986), for $\tilde{\boldsymbol{u}},\ \tilde{\boldsymbol{v}} \in \tilde{V}$, a Lagrangian is constructed

$$\mathcal{L}(\Omega, \tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}) = \int_{\Omega} m(\tilde{\boldsymbol{u}})\, \mathrm{d}x + \int_{\Gamma_N} p(\tilde{\boldsymbol{u}})\, \mathrm{d}s + a(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}) - \int_{\Omega} \boldsymbol{f} \cdot \tilde{\boldsymbol{v}}\, \mathrm{d}x - \int_{\Gamma_N} g \cdot \tilde{\boldsymbol{v}}\, \mathrm{d}s. \tag{8.17}$$

The variables $\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}$ are defined on the full space $\mathbb{R}^d$ and are thus independent of Ω. Although $\tilde{\boldsymbol{u}}$ and $\tilde{\boldsymbol{v}}$ are chosen such that they vanish on Γ_D, they do not depend on Ω since Γ_D is a fixed boundary.

Given the construction of $\mathcal{L}$ (8.17), applying the optimality condition on $\mathcal{L}$ with respect to the independent variables $\boldsymbol{u}$, $\boldsymbol{v}$ and Ω yields the state equation, the adjoint equation and the shape derivative, respectively.

The optimality condition on $\mathcal{L}$ with respect to $\tilde{\boldsymbol{v}}$ amounts to differentiating it with respect to the adjoint variable $\tilde{\boldsymbol{v}}$ along $\boldsymbol{\varphi} \in \tilde{V}$, equating it to zero, followed by the substitution $\tilde{\boldsymbol{u}} = \boldsymbol{u}$, resulting in

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{v}}(\boldsymbol{\varphi}) = a(\boldsymbol{u}, \boldsymbol{\varphi}) - \int_{\Omega} \boldsymbol{f} \cdot \boldsymbol{\varphi}\, \mathrm{d}x - \int_{\Gamma_N} g \cdot \boldsymbol{\varphi}\, ds = 0 \qquad \forall \boldsymbol{\varphi} \in \tilde{V}.$$

The bilinear form $a(\cdot, \cdot)$ in the above being defined only on Ω, we can replace $\tilde{V}$ by V. We thus recover the state equation (8.13). In a similar way, we apply the optimality condition on $\mathcal{L}$ with respect to $\tilde{\boldsymbol{u}}$ at $\tilde{\boldsymbol{u}} = \boldsymbol{u},\ \tilde{\boldsymbol{v}} = \boldsymbol{v}$,

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}}(\boldsymbol{\varphi}) = \int_{\Omega} m'(\boldsymbol{u}) \cdot \boldsymbol{\varphi}\, \mathrm{d}x + \int_{\Gamma_N} p'(\boldsymbol{u}) \cdot \boldsymbol{\varphi}\, \mathrm{d}s + a(\boldsymbol{\varphi}, \boldsymbol{v}) = 0 \qquad \forall \boldsymbol{\varphi} \in \tilde{V}.$$

Since all integrals in the above are defined only on Ω, we can replace $\tilde{V}$ by V. We thus obtain the following adjoint equation:

$$a(\boldsymbol{v}, \boldsymbol{\varphi}) = -\int_{\Omega} m'(\boldsymbol{u}) \cdot \boldsymbol{\varphi}\, \mathrm{d}x - \int_{\Gamma_N} p'(\boldsymbol{u}) \cdot \boldsymbol{\varphi}\, \mathrm{d}s \quad \forall \boldsymbol{\varphi} \in V.$$

Finally, using the relation $J(\Omega) = \mathcal{L}(\Omega, \boldsymbol{u}, \tilde{\boldsymbol{v}})$, we determine the shape derivative $J'(\Omega)(\boldsymbol{\theta})$ for any $\boldsymbol{\theta} \in W_0^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ by chain differentiation rule

$$J'(\Omega)(\boldsymbol{\theta}) = \frac{\partial \mathcal{L}}{\partial \Omega}(\boldsymbol{\theta}) + \frac{\partial \mathcal{L}}{\partial \boldsymbol{u}}\left(\frac{\partial \boldsymbol{u}}{\partial \Omega}(\boldsymbol{\theta})\right).$$

Now, using the adjoint equation, the last term annihilates in the above and we get

$$J'(\Omega)(\boldsymbol{\theta}) = \frac{\partial \mathcal{L}}{\partial \Omega}(\boldsymbol{\theta}).$$

Formula (8.14) is deduced by straightforward application of Lemma 1. □

As mentioned earlier, the Céa's method in the above was applied assuming the adjoint equation to be well-posed. Fortunately, the adjoint equation (8.15) turned out to be well-posed. This is because the constraint in the method was the elasticity equation (8.9), which is linear and well-posed. However, if instead we had a nonlinear equation or an inequation as constraint, nothing would guarantee that the adjoint equation shall be well-posed. In such a case, we apply the Céa's method assuming the adjoint equation to be well-posed. And after deriving the shape derivative, we check if the adjoint equation is well-posed or not. This shall be our approach in plasticity and fracture (Sects. 8.6.4 and 8.8.3).

8.4 Level-Set Method

Any topology optimization method necessitates a way to define and alter a topology of the shape. The method must ensure that topology changes are captured. One obvious possibility is to parametrize the boundary Γ of the shape Ω and move the boundary along its normal $\boldsymbol{n}$ by the magnitude of the shape derivative. This approach cannot capture topology changes.

The level-set method was introduced by Osher and Sethian (2006) to capture the flame front in combustion. They proposed to use the zero isovalue of a continuous function to define the boundary $\partial\Omega$ of a shape Ω. The function defining a shape is such a manner is called a level-set function as each level or value represents a boundary and thus a shape. The mathematical formulation being, for $\Omega \in \mathbb{R}^d$, the level-set function $\phi : D \to \mathbb{R}$ is defined as

$$\begin{cases} \phi(\boldsymbol{x}) < 0 \text{ if } \boldsymbol{x} \in \Omega, \\ \phi(\boldsymbol{x}) = 0 \text{ if } \boldsymbol{x} \in \Gamma, \\ \phi(\boldsymbol{x}) > 0 \text{ if } \boldsymbol{x} \in \overline{\Omega}^c \end{cases} \tag{8.18}$$

where Γ is the movable part of the boundary $\partial\Omega$ and D is the design space as shown in Fig. 8.4. This level-set method was adapted to the shape optimization framework in Allaire et al. (2002), Sethian and Wiegmann (2000) and Wang et al. (2003). The crux of the method lies in letting the shape deform along a velocity field $\boldsymbol{\theta} : D \to \mathbb{R}^d$. The evolution of the shape is governed by the transport equation

$$\frac{\partial \phi}{\partial t} + \boldsymbol{\theta} \cdot \nabla \phi = 0. \tag{8.19}$$

Very often, the velocity field is oriented along the normal, namely $\boldsymbol{\theta} = \theta \boldsymbol{n}$ where $\boldsymbol{n} = \nabla\phi/|\nabla\phi|$ and the scalar function θ is the normal velocity. In such a case, (8.19) can be re-written as a Hamilton–Jacobi equation

$$\frac{\partial \phi}{\partial t} + \theta |\nabla \phi| = 0. \tag{8.20}$$

In the framework of shape optimization, the normal velocity θ is chosen to be $j(\Omega)$ [the integrand in (8.14)]. In the Hamilton–Jacobi equation above, if the initial level-set ϕ is such that $|\nabla\phi(\boldsymbol{x})| = 1$, every point $\boldsymbol{x} \in \Gamma$ shall move along the normal $\boldsymbol{n}$ by a value that is exactly equal to the magnitude of the shape derivative. Thus, it is preferable to construct a level-set such that $|\nabla\phi(\boldsymbol{x})| = 1$ on Γ. One possible construction is the signed distance function (Osher and Fedkiw 2003), d_Ω is defined such that

$$\forall \boldsymbol{x} \in D,\ d_\Omega(\boldsymbol{x}) = \begin{cases} \min |P_\Gamma(\boldsymbol{x}) - \boldsymbol{x}| & \text{if } \boldsymbol{x} \in \Omega, \\ 0 & \text{if } \boldsymbol{x} \in \Gamma, \\ -\min |P_\Gamma(\boldsymbol{x}) - \boldsymbol{x}| & \text{if } \boldsymbol{x} \in \overline{\Omega}^c, \end{cases} \tag{8.21}$$

where P_Γ is the orthogonal projection of $\boldsymbol{x}$ on Γ. Since Ω is not necessarily convex, the orthogonal projection $P_\Gamma(\boldsymbol{x})$ is not necessarily unique. In order to define $d_\Omega(\boldsymbol{x})$ uniquely, we take the min of all possible projections $P_\Gamma(\boldsymbol{x})$ in the definition (8.21).

As an example, we plot the signed distance level-set function d_Ω in Fig. 8.5, where Ω is the statue of liberty. When one solves the transport equation (8.19), the

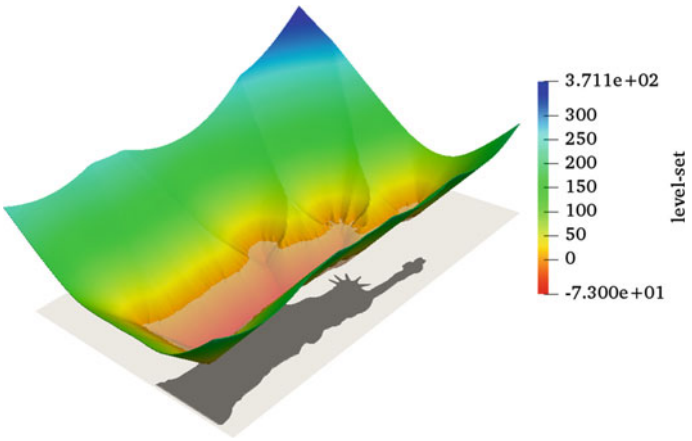


Fig. 8.5 Signed distance level-set function ϕ representing statue of liberty (in black)

level-set function ϕ evolves and its corresponding shape. Since the shape evolves, the shape derivative evolves as well. The descent direction $\boldsymbol{\theta}$, that depends on the shape derivative must evolve, but is assumed to be constant. Since the transport equation (8.19) is solved for a very small time step (8.30), this assumption on $\boldsymbol{\theta}$ turns out to be fairly robust.

Moreover, descent direction $\boldsymbol{\theta}$ is defined using the shape derivative and not the topological derivative. Yet resolution of the transport equation results in topological changes in the shape. When the topological changes occur, there is no guarantee that the objective function shall diminish. Nevertheless, we use the shape derivative to perform topology optimization.

In our numerical setting, we work with the linear transport equation (8.19) because we use non-cartesian meshes. The level-set function is a $\mathbb{P}^1$ function on a simplicial mesh. The initial level-set function that is given as input to Eq. (8.19) is chosen as $\phi = d_\Omega$ (8.21). We rely on the library `advect` (Bui et al. 2012) which solves (8.19) by the method of characteristics, known to be unconditionally stable. The equation (8.19) is solved for a small time interval δt (specified in Algorithm 1). The result is a new level-set function representing the updated shape. However, the new level-set function is no longer a signed distance function. Hence, the level-set function ought to be re-initialized. This is brought about by using a library `mshdist` (Dapogny and Frey 2012). The library `mshdist` re-initialize by solving the Eikonal equation.

8.4.1 Ersatz Material Approach

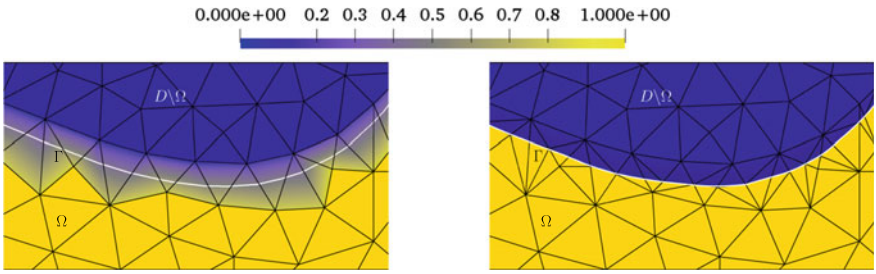
In the original level-set method (Allaire et al. 2002; Sethian and Wiegmann 2000; Wang et al. 2003), the mesh for the design space D (see Fig. 8.4) is kept fixed for all the shape optimization iterations. The computation of the state equation (8.13) and the adjoint equation (8.15), is brought about in the entire design space D (instead of Ω). This is done by assuming that the "void" space $D\backslash\Omega$ has an ersatz material (having a very small stiffness). The Hooke's tensor $\mathbb{C}$ is modified to $\mathbb{C}_\epsilon(\Omega)$, using a density function $\rho_\epsilon(\Omega)$

$$\rho_\epsilon(\Omega) = (\chi_\Omega + (1 - \chi_\Omega)\epsilon)\,, \qquad \mathbb{C}_\epsilon(\Omega) = \rho_\epsilon(\Omega)\mathbb{C}, \tag{8.22}$$

where χ_Ω is the characteristic function of the domain Ω and ϵ is the residual stiffness. The displacement solution now becomes a function of the ersatz material stiffness, denoted by $\boldsymbol{u}_\epsilon(\Omega)$ and is slightly different from the solution obtained without ersatz, $\boldsymbol{u}(\Omega)$. This is difference is proportional to ϵ and is given by the estimate (Dambrine and Kateb 2010)

$$\|\boldsymbol{u} - \boldsymbol{u}_\epsilon\|_{H^1(\Omega)} \leq C(\Omega, D, \boldsymbol{f}, g)\epsilon$$

where C is a constant. In practice, ϵ is typically taken to be 10^{-3}. The characteristic function χ_Ω is $\mathbb{P}^1$ on a simplicial mesh. Using this stiffness the variational formulation (8.13) is defined on the entire design space D. The advantage of the ersatz approach



(a) Fixed mesh (b) XFEM mesh

Fig. 8.6 Ersatz material density $\rho_\epsilon(\Omega)$

is the fact that no remeshing is needed, and all the calculations can be performed on the same mesh. The topological changes are easily captured on the same mesh. However, there are two drawbacks:

1. The computation time of state and adjoint equations on D is higher compared to the same computation on Ω only
2. There are intermediate densities in the mesh elements containing $\partial\Omega$ as seen in Fig. 8.6a

In order to deal with the first drawback, one may consider eliminating the entries in the stiffness matrix corresponding to the ersatz material, to end up with a stiffness matrix that is relatively smaller. However, in this approach, the intermediate densities still remain. To deal with the second drawback, there are two solutions:

1. XFEM (Villanueva and Maute 2014)
   This involves eliminating mesh elements consisting of ersatz and cutting the mesh elements on boundary Γ, so as to capture the shape Ω as shown in Fig. 8.6b. The issue with this approach being, one might end up with heavily distorted boundary elements. This leads to an ill-conditioned stiffness matrix having a very small coercivity constant.
2. CutFEM (Burman et al. 2015)
   This involves eliminating mesh elements consisting of ersatz and enriching the finite element basis function of the interface zone (the elements containing the boundary Γ). This along with a ghost penalty term on the interface zone can ensure that the coercivity constant is controlled and the stiffness matrix remains well-conditioned.

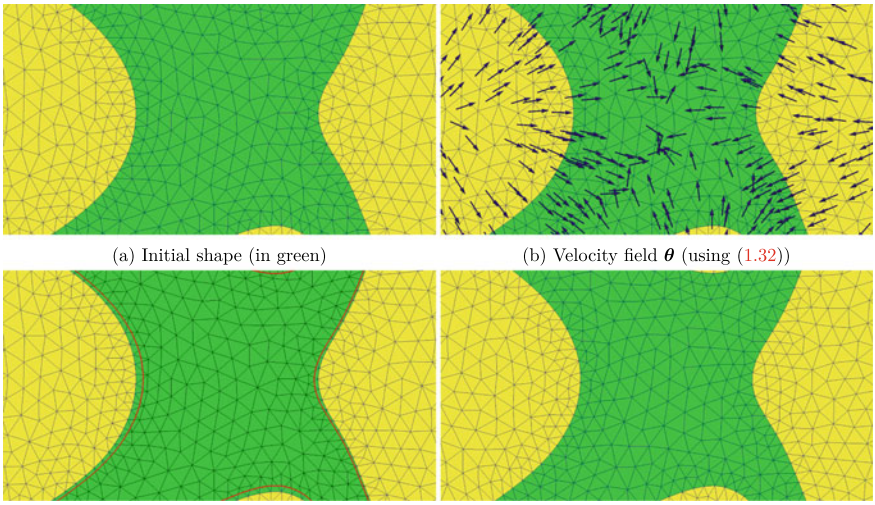
An alternative to the XFEM and the CutFEM approach is remeshing the whole domain D, explained in the next subsection and considered in the rest of the chapter.

8.4.2 Remeshing

One method to avoid intermediate densities on the boundary Γ (as seen in Fig. 8.6a) is remeshing. By remeshing, we mean construction of a body-fitted mesh that captures the boundary of the shape Ω. In this remeshing approach, the mesh quality can be controlled, resulting in stiffness matrix, that is well-conditioned compared to the XFEM approach. Moreover, no additional penalty terms are needed as in CutFEM approach. The remeshing approach (using the level-set method) was first adapted to topology optimization by Dapogny (2013), Dapogny et al. (2014), Bui et al. (2012) and Allaire et al. (2014)). Later, several applications were developed, for instance, for fluid-structure interaction (Feppon et al. 2019), heat exchangers (Feppon et al. 2020), etc. The only downside of this approach is the remeshing computing time.

The application of remeshing in the context of shape optimization is explained with the following example. As shown in Fig. 8.7a, consider an initial shape and its corresponding mesh (in green). For this shape, the velocity for advection is computed using the formula (8.32), resulting in $\boldsymbol{\theta}$ as shown in Fig. 8.7b. Using this velocity field, the transport equation (8.19) is solved for a time step τ [given in (8.30)] using `advect` (Bui et al. 2012), resulting in the new shape in Fig. 8.7c (marked with red lines). This new shape is then remeshed (using a tool `MMG`, explained below) to obtain the body-fitted mesh (8.7d). Finally, for the new shape, the signed distance level-set function is computed using `mshdist` (Dapogny and Frey 2012).

Body-fitted remeshing can be carried out using an open-source library `MMG 5.5.2` (Dapogny et al. 2014). `MMG` is a remeshing tool that can not only remesh to capture a shape (defined by a level-set function) but also to improve the quality



(a) Initial shape (in green) (b) Velocity field $\boldsymbol{\theta}$ (using (1.32))

(c) New shape after solving transport Eqn.1.19 (d) New shape captured using body-fitted mesh

Fig. 8.7 Shape capturing using advection and remeshing

of the mesh. The documentation of MMG can be found online[1]. The program has to be launched via command line. In order to launch it, the user has to give several parameters (a few optional ones) as input. The parameters that ought to be defined for our application are

1. hmax: the prescribed maximal mesh size
2. hmin: the prescribed minimal mesh size
3. hausd: the Hausdorff metric (to refine mesh according to the curvature of $\partial\Omega$)
4. rmc: the volume fraction of small floating island inside or outside the domain Ω to be removed, as islands outside the domain Ω may make the stiffness matrix non-invertible
5. nr: a flag to avert MMG about detecting edges and corners of Ω
6. sol: a flag to give the level-set function as input

hmin is the prescribed lower bound on the minimal mesh size and not the actual minimal mesh size. Hence, the minimal mesh size of the mesh generated by MMG can be very well bigger than hmin. In the same way hmax is the prescribed upper bound on the maximal mesh size and not the actual maximal mesh size. The parameter that takes precedence over hmin and hmax is hausd. This parameter is a measure of mesh refinement, lesser the value, more the refinement. The mesh refinement is by default according to the curvature of the boundary $\partial\Omega$,, but it can also be according to a predefined metric function. If hausd is too small, the minimal mesh size may be lesser than hmin as hausd is given preference. The parameters hmin and hmax are thus not always respected in order to create a good quality mesh.

For any test case in this entire thesis, we first choose $h_{\min}$ in a heuristic manner. Using this, we define the above parameters as for 2D geometries as

$$\texttt{hmax} = 2\,\texttt{hmin},\ \ \texttt{hausd} = 0.1\,\texttt{hmin},\ \ \texttt{rmc} = 20\,\texttt{hmin}^2, \tag{8.23}$$

and for 3D geometries as

$$\texttt{hmax} = 2\,\texttt{hmin},\ \ \texttt{hausd} = 0.5\,\texttt{hmin},\ \ \texttt{rmc} = 10\,\texttt{hmin}^3. \tag{8.24}$$

This choice of these parameters is very much based out of heuristics.

The mesh file given as input to MMG to construct a body-fitted mesh must contain the list of corners and edge vertices of the design space D. If the mesh file does not contain this list, while MMG remeshes, the corners and edges might be smoothened, leading to a filet. In order to avert MMG about the corners and edges, one needs to remesh the first mesh of the shape optimization using MMG without the flag nr. An example of the command that ought to be given in the command line is

```
mmg2d_03 –rmc 0.0005 –hmax 0.01 –hmin 0.005 –hausd 0.0005
levelsetmesh.mesh –sol levelsetmesh.sol
```

[1] https://www.mmgtools.org/mmg-remesher-try-mmg/mmg-remesher-options.

This creates a list of corner and edge vertices in the output file. For the succeeding iterations, the flag `nr` need to be given and the command to remesh reads

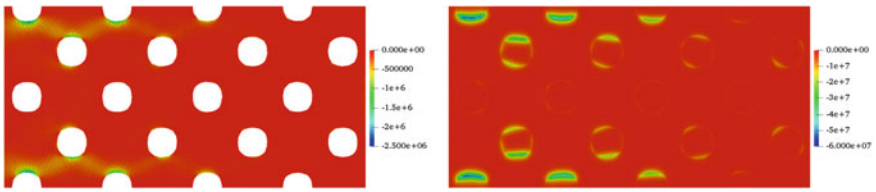
```
mmg2d_03 -rmc 0.0005 -hmax 0.01 -hmin 0.005 -nr -hausd 0.0001
levelsetmesh.mesh -sol levelsetmesh.sol
```

for the remaining iterations.

8.4.3 *Regularization and Extension of the Shape Derivative (8.14)*

In a typical topology optimization algorithm, the shape derivative (also known as sensitivity) must be smoothed. This is typically done by solving a partial differential equation. In the density-based methods, such a smoothing algorithm is known as a sensitivity filter (Bendsoe and Sigmund 2004; Lazarov and Sigmund 2011) and plays a pivotal role in avoiding numerical instabilities. In level-set-based methods, smoothing (or regularization) plays a crucial role in ensuring the smooth evolution of the topology (Allaire et al. 2002; Burger 2003; De Gournay 2006). Such a smoothing is very classical and well-known in geometric optimization for a long time (Mohammadi and Pironneau 2001).

The shape derivative computed numerically using the expression (8.14) is typically not very smooth. There are two factors that contribute to this lack of smoothness. First, during the shape optimization process, the intermediate shapes may have sharp boundaries. So the shape derivative may not be rigorously defined on the optimizible boundary Γ. Second, when $\boldsymbol{u}$ is $\mathbb{P}^1$ smooth, the shape derivative (8.14) is $\mathbb{P}^0$ smooth (an example plotted in Fig. 8.8a). Given the lack of smoothness, it is imperative to regularize the shape derivative (Burger 2003; De Gournay 2006) in such a way that it is still a descent direction. One possibility is to consider the H^1 scalar product instead of the L^2 scalar product by finding a function $dj_\alpha(\Omega) \in H^1(D)$. We introduce a regularization parameter $0 < \alpha \ll 1$ and solve the variational problem: find $\mathrm{d}j_\alpha(\Omega) \in H^1(\Omega)$ such that



(a) Shape derivative $j(\Omega)$(1.14) extended to Ω (b) Regularized shape derivative $dj_\alpha(\Omega)$ (using (1.25))

Fig. 8.8 Shape derivative plot

$$\int_D (\alpha^2 \nabla \mathrm{d}j_\alpha(\Omega) \cdot \nabla\varphi + \mathrm{d}j_\alpha(\Omega)\varphi)\mathrm{d}x = \int_\Gamma j(\Omega)\varphi \, \mathrm{d}s \qquad \forall \varphi \in H^1(\Omega). \tag{8.25}$$

Typically, α is taken to be $h_{\min}$ or $h_{\max}$ (the minimal or maximal mesh size, respectively). Larger the parameter, greater the regularization. One can quite easily demonstrate that the above problem (8.25) is well-posed and admits a unique solution $dj_\alpha(\Omega) \in H^1(\Omega)$.

8.4.4 Imposing a Non-optimizable Domain

Having regularized the shape derivative in the previous subsection, we can easily perform the advection by solving (8.19) [using `advect` library (Bui et al. 2012)]. After advection, we might wish to impose a non-optimizable domain Ω^{non}. Typical non-optimizable domains are around the zone of application of force Γ_N, or around the Dirichlet boundary, an example is given in Fig. 8.9. To do this, assume that after the advection and the volume correction at optimization iteration i (the 8-th step of the Algorithm 1), we determine the level-set ϕ_{i+1} and a shape Ω_{i+1}, which must contain a non-optimizable domain. Let ϕ^{non} be the signed distance function corresponding to Ω^{non}. Then in order to impose this domain, we simply modify ϕ_{i+1} using

$$\phi_{i+1}^{\text{non}} = \min(\phi_{i+1}, \phi^{\text{non}}), \tag{8.26}$$

just before remeshing at the end of optimization iteration i. The level-set function ϕ_{i+1}^{non} is later re-initialized using `mshdist`. As an example, we show Fig. 8.9, where a non-optimizable domain has been imposed. Another possibility of imposing a non-optimizable domain is via solving (8.25) with a homogeneous Dirichlet boundary condition imposed on $dj(\Omega)$ in Ω^{non}. This possibility is not explored here as it might result in artificial changes in the shape derivative around Ω^{non}.

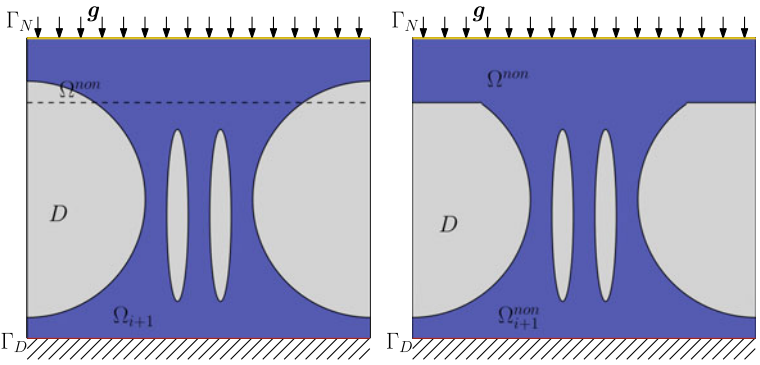


Fig. 8.9 The shape Ω_{i+1} after the application of the non-optimizable domain using (8.26)

8.4.5 Shape Optimization Algorithm

Assume that the design space D contains a non-optimizable domain Ω^{non} and its complementary set, the optimizable domain, $D \backslash \Omega^{\text{non}}$. By volume of the domain, we mean the volume of the optimizable domain. The space of admissible spaces is thus redefined as

$$\mathcal{U}_{\text{ad}} = \left\{ \Omega \subset D : \; V_\Omega = \int\limits_{\Omega \backslash \Omega^{\text{non}}} \mathrm{d}x = V_t \right\}, \tag{8.27}$$

We consider the shape optimization problem

$$\min_{\Omega \in \mathcal{U}_{\text{ad}}} J(\Omega).$$

In order to ensure that the optimized shape satisfies the volume constraint in (8.27), we follow the following strategy (which is different from the strategy in Allaire et al. (2002)) by constructing the following Lagrangian

$$\mathcal{L}(\Omega, \lambda) = J(\Omega) + \frac{\lambda}{C_V} \left(V_\Omega - V_t \right), \tag{8.28}$$

where λ is the Lagrange multiplier for the volume constraint, V_{Ω^h} is the volume of the optimizable domain, and C_V is a normalization constant. The optimizable domain is D unless otherwise mentioned. Denoting by Ω_0 the initial shape, the constant C_V is defined by

$$C_V = \left| V_{\Omega_0} - V_f \right|. \tag{8.29}$$

We apply a standard gradient-based Uzawa-type algorithm to the Lagrangian (8.28). Let $I_{\max} = 200$ be the maximal number of shape optimization iterations. The iteration number is denoted by i with $1 \leq i \leq I_{\max}$. At each iteration i, once the shape derivative $dj_\alpha(\Omega_i)$ is evaluated by (8.25), a pseudo-time step (or descent step) τ is defined by

$$\tau = \frac{h_{\min}}{2C_i}, \tag{8.30}$$

where $h_{\min}$ is the minimal mesh size of the first iteration and C_i is a normalization constant, given by

$$C_i = \int\limits_{\partial \Omega_i} |\mathrm{d}j_\alpha(\Omega_i)| \, \mathrm{d}x.$$

Updating the constant C_i at every iteration of the optimization process ensures a control of the descent step τ.

For every optimization iteration, the simplest choice of the multiplier λ is the one that ensures that the volume constraint is satisfied. If the initial volume is much larger

than the target volume, we might deflate the holes too much and break the structure Ω. We thus prefer not to satisfy the volume constraint at all iterations, instead update the multiplier λ at each iteration by

$$\lambda_{i+1} = \lambda_i + \frac{C_i \tau}{C_V} \left(V_{\Omega_i} - V_f \right), \tag{8.31}$$

ensuring that the volume will converge (slowly) to the target volume. Then, for the descent step τ, the transport equation (8.19) is solved with a velocity θ_i, given by

$$\boldsymbol{\theta}_i = \left(d j_\alpha(\Omega_i) + \frac{\lambda_{i+1}}{C_V} \right) \boldsymbol{n}, \tag{8.32}$$

where $\boldsymbol{n} = \nabla \phi_i$ is the normal to the level-set function associated to the shape Ω_i.

To improve the satisfaction of the volume constraint, we apply the following trick. As soon as the volume is close to the volume target, namely $|V_{i+1} - V_f| \leq 10^{-1} V_f$, we apply a projection algorithm to satisfy the target volume exactly. More precisely, the level-set ϕ_{i+1} is iteratively updated by

$$\phi_{i+1} = \phi_{i+1} + \frac{V_{\Omega_{i+1}} - V_f}{\int_{\partial \Omega_{i+1}} \mathrm{d}s}, \tag{8.33}$$

until $|V_{\Omega_{i+1}} - V_f| \leq 10^{-4} V_f$. The newly obtained shape $V_{\Omega_{i+1}}$ is remeshed with `MMG` (Dapogny et al. 2014). Eventually, the objective function $J(\Omega_{i+1})$ is evaluated but is not compared to the previous value $J(\Omega_i)$. Summing up this subsection, we basically implement Algorithm 1. The shape optimization algorithm used in the succeeding sections is very similar to the Algorithm 1, modulo a few differences.

Algorithm 1 Repeat over $i = 1, \ldots, I_{\max}$

1. Solve (8.13) for the state solution $\boldsymbol{u}$ on the mesh of Ω_i
2. Solve (8.15) for the adjoint solution $\boldsymbol{v}$ on the mesh of Ω_i
3. Compute the shape derivative $J'(\Omega_i)$ using (8.14) and regularize it by solving (8.25) for $\mathrm{d} j_\alpha(\Omega_i)$
4. Update the multiplier λ_{i+1} using (8.31)
5. Solve the transport equation (8.19) with the velocity given by (8.32) for the pseudo-time step τ given by (8.30) to obtain the new level-set function $\tilde{\phi}_{i+1}$
6. Re-initialize $\tilde{\phi}_{i+1}$ to the signed distance function ϕ_{i+1} (defining a new shape Ω_{i+1})
7. Compute the volume $V_{\Omega_{i+1}}$. If it is close to the volume target, apply the projection algorithm (8.33) to satisfy exactly the volume constraint.
8. Impose non-optimizable domain using (8.26)
9. Remesh the box D using `MMG` (Dapogny et al. 2014) to obtain the body-fitted mesh of the new shape Ω_{i+1}

8.5 Numerical Results: Linear Elasticity

In this section, we shall study briefly, a few well-known test cases (cantilever, L-beam, etc.). The material properties are taken to be $E = 210$ GPa and $\nu = 0.3$. The finite element computations are brought about on an open-source software `FreeFEM` (Hecht 2012). In all the test cases in this section, only surface forces are applied and body forces are not applied. The finite element mesh consists of tetrahedral elements and the displacement solution $\boldsymbol{u} \in \mathbb{P}^1(\Omega)^3$.

8.5.1 3D Cantilever

We consider a cantilever beam of dimensions $5\,\text{m} \times 2.4\,\text{m} \times 3\,\text{m}$, as shown in the Fig. 8.10. The cantilever beam is fixed on its leftmost plane, forced downwards on a circular region of radius 0.1 m on its rightmost plane (in chrome) with $g = (0, 0, -1000)$ MN/m. The compliance (8.3) is minimized. A target volume $V_t = 6\,\text{m}^3$ is imposed. We show the initial shape and the final shape in Fig. 8.11. The corresponding convergence histories are plotted in Fig. 8.12. The optimal shape we have obtained in Fig. 8.11 seems to have a fairly simple topology, with a plate-like structure at the center. The shape resembles an I-beam with varying length. This is quite expected from an engineering perspective as I-beams are known to be extremely stiff structures. The same 3D cantilever beam optimized using the ersatz material approach might result in an optimal shape with several bars (see Fig. 8.11c) instead of plate-like structures. Moreover, in the ersatz material approach, if the target volume fraction is decreased, the chances of seeing bar-like structures is higher rather than seeing plate-like structures, like in the remeshing approach. This is because remeshing can capture thin structures precisely, leading to a more accurate shape derivative and a higher tendency of creating plates rather than bars.

In Fig. 8.12, we observe that the compliance grows initially. This is because the optimization Algorithm 1 seems to favor the minimization of volume at the start.

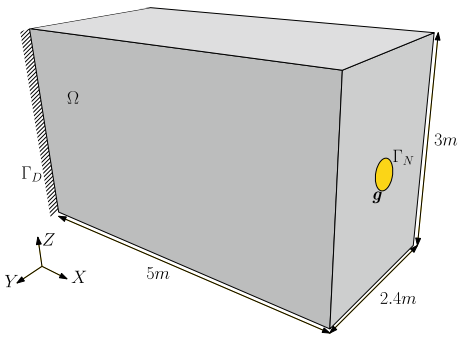
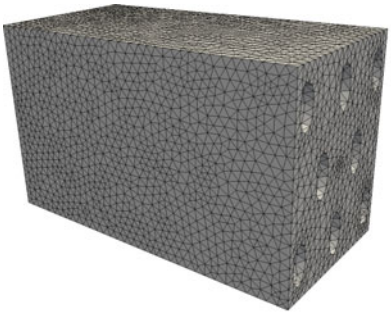
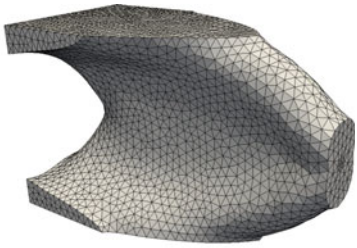


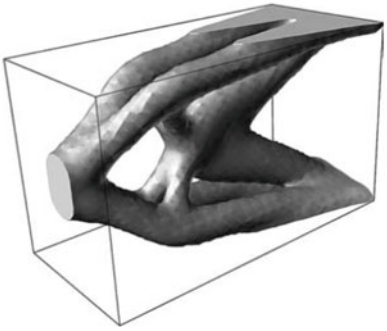
Fig. 8.10 Boundary condition applied on the 3D cantilever beam



(a) Initial shape

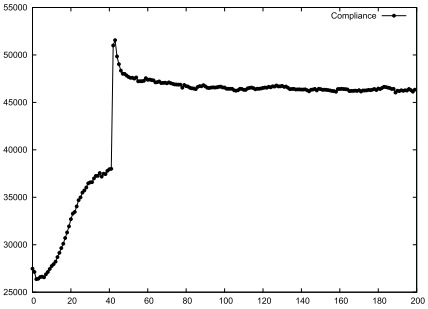


(b) Final shape

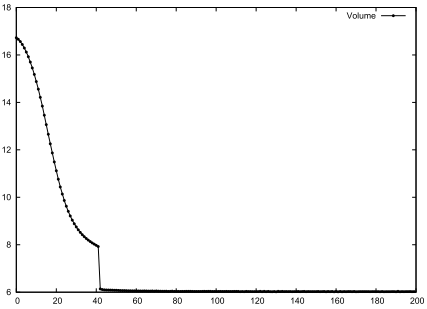


(c) Optimal shape using ersatz approach (taken from [9])

Fig. 8.11 Cantilever beam, minimization of compliance (8.3)



(a) Compliance (1.3) vs iterations



(b) Volume vs iterations

Fig. 8.12 Convergence history for the shape in Fig. 8.11

Then at iteration 44, the level-set is projected [as explained in (1] so as to satisfy the target volume V_t exactly. This results in a spike in the compliance. After this iteration, the volume remains constant (because of the projection of the level-set), whereas the compliance effectively decreases. Also, we note that there are small oscillations in the compliance in Fig. 8.12. This is because of the fact that we compute the continuous shape derivative and use it for the discretized problem, rather than finding the shape derivative for the discretized problem. While remeshing helps to capture the shape Ω very well, resulting in a better calculation of the solution $\boldsymbol{u}$ and the shape derivative (compared to ersatz material approach in 8.4.1), it adds to a slight fluctuation in the objective function. This is because the change in mesh is not taken into account in the continuous shape derivative.

8.5.2 3D L-beam

We study an L-beam as shown in Fig. 8.13. The L-beam is fixed on the topmost surface (in red) and forced on a small circular part on the left (in gold). The circular region has a radius of 0.1m. The surface force reads $g = (0, -1, 0)$ MN/m. Here again the compliance (8.3) is minimized. We show the initial meshed shape and the final meshed shape for linear elasticity in Fig. 8.14 and the corresponding convergence histories are plotted in Fig. 8.15. Like in the cantilever test case, the optimal shape consists of a plate-like structure at the center. As seen in Fig. 8.15, the compliance decreases well after the volume constraint is satisfied at the 15-th iteration. In the convergence for compliance seems to be smoother than the same convergence for the cantilever beam in the previous section. This is because the mesh considered for the L-beam optimization is more refined than the one used for cantilever beam. A refined mesh results in a better computation of the shape derivative.

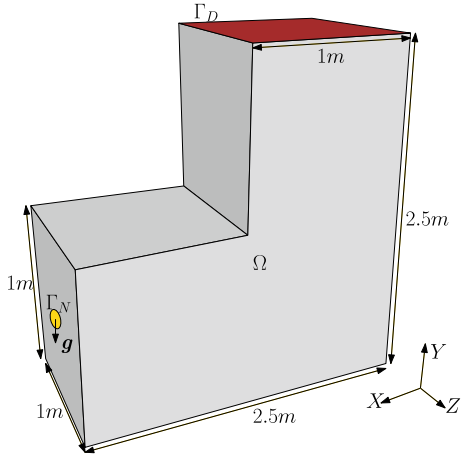
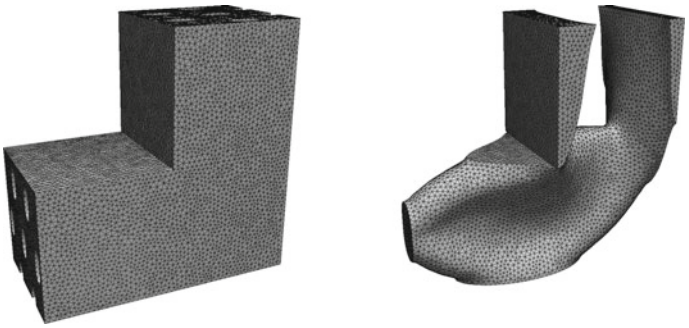
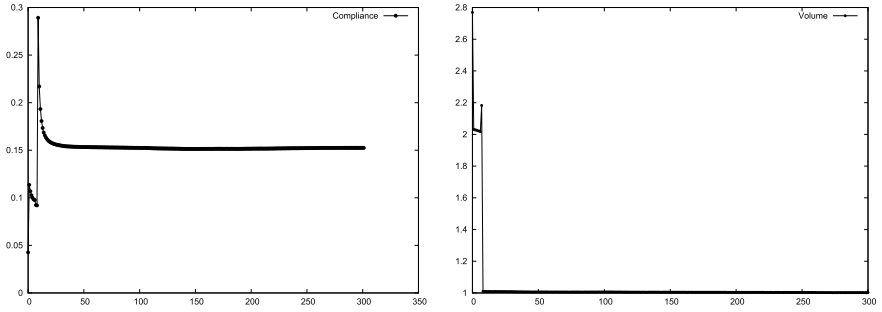


Fig. 8.13 3D L-beam boundary conditions



(a) Initial shape (b) Final shape

Fig. 8.14 Meshed L-beam, minimization of compliance (8.3)



(a) Compliance (1.3) vs iterations (b) Volume vs iterations

Fig. 8.15 Convergence history for the shape in Fig. 8.14

8.6 TO in Plasticity with Linear Kinematic and Linear Isotropic Hardening

We now discuss and adaptation of topology optimization to plasticity model. Most of the content in this section can be found in a recent publication (Desai et al. 2021). In design of structures involving steel, a plasticity model is often used in order to determine the plastic strain, or the permanent deformation, which occurs in the structure when it undergoes a stress that exceeds a value known as the yield strength (Han and Reddy 2013). As the time-dependent force evolves, if the yield strength remains constant everywhere in the structure, the resulting phenomenon is called perfect plasticity; otherwise, it is called plasticity with hardening. Using a hardening law, one can determine the shift in the yield strength and measure how ductile the material is. Plasticity modeling has been developed significantly since the 1960s. At the heart of the model lies the Hill's principle and its equivalent Drucker–Illyushin principle (Marigo 2000).

Very often, the plasticity model is simplified by assuming that the evolution of the force is slow. This assumption results in a quasi-static plasticity model, which has been largely studied theoretically. The model, when written in its variational formulation, is an inequality, which can be expressed either in a dual form or in a primal form. The primal form is relevant as it can be shown to be a well-posed problem (Han et al. 1997). While the primal formulation illuminates the theoretical properties of the solution to the plasticity problem, it is very hard to resolve numerically. Therefore, one instead resorts to the radial return algorithm (Simo and Hughes 2006; Simo and Taylor 1986) which involves discretizing the governing equations of (hardening-based) plasticity using an implicit Euler scheme.

Most of the shape and topology optimization algorithms are developed for linearized elasticity problems while less focus is given to the nonlinear ones. Nonlinearities can arise due to material properties (plasticity, damage), contact boundary conditions, hyper-elasticity (Klarbring and Strömberg 2013), large displacement (large strain Buhl et al. 2000; Wallin et al. 2015, finite strain Wallin et al. 2016) and structural buckling (Lindgaard and Dahl 2013). Topology optimization using density approaches or solid isotropic material with penalization (SIMP) was applied to elasto-plastic problems (Bogomolny and Amir 2012; Kato et al. 2015; Li et al. 2017; Maute et al. 1998; Wallin et al. 2016), visco-elastic problems (James and Waisman 2015b) and visco-elasto-plastic problems (Nakshatrala and Tortorelli 2016).

A common feature in all the works involving density-based approach is the determination of a design gradient by differentiating the space and time discretized schemes of the plasticity models, which are approximated using a fictitious material density. The material properties like the Young's modulus and the hardening coefficients are modified using this material density raised to a certain exponent. This exponent is different for every mechanical property and ought to be chosen in an ad hoc manner, ensuring numerical stability. The optimized shape obtained has intermediate densities undergoing a plastic flow, which might actually be artificial. In the level-set framework, since the material properties are not approximated using material densities, such artificial plastic zones are avoided.

The level-set method for topology optimization has been applied to a simplified version of perfect plasticity in Maury et al. (2018). There, the first time step of time discretized perfect plasticity (also known as the Hencky's model) was approximated using Perzyna penalization, and the resulting approximation was shown to be well-posed. The model did not take into account hardening laws, the time dependence or the irreversibility of the plasticity problem. As soon as one incorporates the irreversibility of the plastic flow and hardening laws, one ends up with a variational inequation with a complex theoretical and numerical treatment.

In this section, we present the level-set method applied to quasi-static plasticity with linear kinematic and linear isotropic hardening. Unlike the previous works, the quasi-static plasticity is considered in its primal form and the shape derivative is determined for the continuous problem. The primal form being non-smooth is not differentiable. Nevertheless, we construct an approximate problem that is differentiable using a penalization and regularization technique. The approximate problem is well-posed, and the corresponding solution converges to the actual solution.

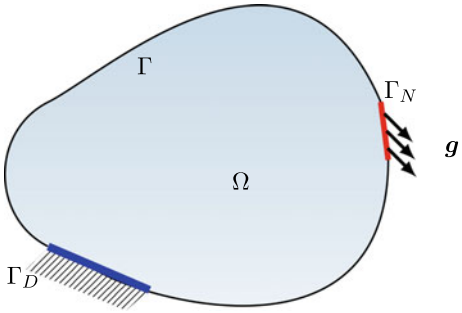


Fig. 8.16 Boundary conditions on the structure Ω

Then, we compute the shape derivative for minimizing an objective function with the approximate problem as a constraint. As usual, this shape derivative involves an adjoint problem. Thanks to the proposed penalization and regularization, the time discretized version of this adjoint problem is proved to be well-posed. It is well known that the original primal problem is not differentiable in the usual sense but admits only a so-called conical derivative (see Maury et al. 2018; Mignot 1976; Sokolowski and Zolésio 1992). The efficiency of the proposed shape derivative (obtained with penalization and regularization process) is assessed by optimizing some numerical examples in 2D and 3D. The plasticity problem is numerically solved using the radial return algorithm.

We first present the laws governing plasticity with linear kinematic and isotropic hardening in Sect. 8.6.1. Using these laws, we derive the primal variational formulation in Sect. 8.6.2. This formulation is further subjected to penalization and regularization in order to make it differentiable in Sect. 8.6.3. This section closes with some statements about the well-posedness and the convergence of the solution of the penalized-regularized plasticity model toward the actual solution.

8.6.1 Governing Equations

We consider a structure represented by a smooth bounded open set $\Omega \subset \mathbb{R}^d$, $d = 2$ or 3 and a bounded time interval $[0, T]$. Let $\mathcal{M}_s^d$ denote the set of symmetric $d \times d$ matrices and $\mathbb{I}$ represent the fourth-order identity tensor of dimension d. The structure, having a boundary $\partial\Omega = \Gamma_N \cup \Gamma_D \cup \Gamma$, is fixed on Γ_D and loaded on Γ_N as shown in Fig. 8.16. Plasticity is a quasi-static process as we now describe (see Han and Reddy (2013) for more details). Let $\boldsymbol{u} : \Omega \times [0, T] \to \mathbb{R}^d$ denote the displacement field, $\boldsymbol{\sigma} : \Omega \times [0, T] \to \mathcal{M}_s^d$ denote the stress tensor, $\boldsymbol{n}$ denote the outward normal to $\partial\Omega$. The structure when subjected to a time-dependent body force $\boldsymbol{f} : \Omega \times [0, T] \to \mathbb{R}^d$ and a surface force $g : \Gamma_N \times [0, T] \to \mathbb{R}^d$, satisfies the equilibrium equation:

$$\operatorname{div}(\boldsymbol{\sigma}) + \boldsymbol{f} = \mathbf{0} \quad \text{in } \Omega \times (0, T], \tag{8.34a}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = g \quad \text{on } \Gamma_N \times (0, T], \tag{8.34b}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \mathbf{0} \quad \text{on } \Gamma \times (0, T], \tag{8.34c}$$

$$\boldsymbol{u} = \mathbf{0} \quad \text{on } \Gamma_D \times (0, T]. \tag{8.34d}$$

The total strain tensor of the structure $\boldsymbol{\varepsilon} : \Omega \times [0, T] \to \mathcal{M}_s^d$, expressed in terms of $\boldsymbol{u}$, $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}(\boldsymbol{u}) = (\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T)/2$ can be decomposed as

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_e + \boldsymbol{\varepsilon}_p,$$

where $\boldsymbol{\varepsilon}_e$ denotes the elastic strain and $\boldsymbol{\varepsilon}_p$, the plastic strain. Plasticity occurs when the magnitude of $\boldsymbol{\sigma}$ exceeds the yield strength, a material parameter determined experimentally. Hardening occurs when the plastic flow is followed by a change in yield strength. The hardening is modeled by a stress-like hardening tensor $\boldsymbol{q} : \Omega \times [0, T] \to \mathcal{M}_s^d$, a scalar force $g : \Omega \times [0, T] \to \mathbb{R}$, and the corresponding internal variable, $\boldsymbol{r} : \Omega \times [0, T] \to \mathcal{M}_s^d$, $\gamma : \Omega \times [0, T] \to \mathbb{R}$,, respectively. To define the structure's elastic limit, we consider the von Mises yield criterion (Simo and Hughes 2006)

$$f(\boldsymbol{\sigma}, \boldsymbol{q}, g) = |\boldsymbol{\sigma}^D - \boldsymbol{q}^D| + \sqrt{\frac{2}{3}}\,(g - \sigma_Y) \leq 0, \tag{8.35}$$

where the superscript D denotes the deviatoric part of a tensor and $\sigma_Y \in \mathbb{R}^+$ is the yield strength. This criterion defines the elastic domain

$$\mathbb{E} = \{(\boldsymbol{\sigma}, \boldsymbol{q}, g) : f(\boldsymbol{\sigma}, \boldsymbol{q}, g) \leq 0\},$$

which, by definition, is convex. The structure is made of an isotropic material, with Hooke's tensor given by

$$\mathbb{C} = \lambda \mathbf{1} \otimes \mathbf{1} + 2\mu \mathbb{I},$$

where λ, μ are Lamé constants. We place ourselves in the framework of associated plasticity, namely the plastic flow rate is proportional to the normal of the elastic domain. We first state the second law of thermodynamics

$$\boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}} - \dot{\boldsymbol{\psi}} \geq 0, \tag{8.36}$$

where the overdot denotes differentiation with respect to time and $\boldsymbol{\psi}$ is the Helmholtz free energy, given by the sum

$$\boldsymbol{\psi} = \boldsymbol{\psi}(\boldsymbol{\varepsilon}_e, \boldsymbol{r}, \gamma) = \hat{\boldsymbol{\psi}}_e(\boldsymbol{\varepsilon}_e) + \hat{\boldsymbol{\psi}}_p(\boldsymbol{r}, \gamma),$$

where the elastic and plastic energies are, respectively, defined as

$$\hat{\boldsymbol{\psi}}_e(\boldsymbol{\varepsilon}_e) = \frac{1}{2}\mathbb{C}\boldsymbol{\varepsilon}_e : \boldsymbol{\varepsilon}_e \quad \text{and} \quad \hat{\boldsymbol{\psi}}_p(\boldsymbol{r}, \gamma) = \frac{1}{2}\mathbb{H}\boldsymbol{r} : \boldsymbol{r} + \frac{1}{2}E_{\text{iso}}\gamma^2,$$

where $\mathbb{H}$ is the hardening tensor and $E_{\text{iso}} \geq 0$ is a material parameter. On the other hand, the stress is assumed to be $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{\varepsilon}_e)$. Using these definitions, the second law (8.36) is re-written as

$$\left(\boldsymbol{\sigma} - \frac{\partial \hat{\boldsymbol{\psi}}_e}{\partial \boldsymbol{\varepsilon}_e}\right) : \dot{\boldsymbol{\varepsilon}} + \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}_p - \frac{\partial \hat{\boldsymbol{\psi}}_p}{\partial \boldsymbol{\varepsilon}_p} : \dot{\boldsymbol{\varepsilon}}_p - \frac{\partial \hat{\boldsymbol{\psi}}_p}{\partial \boldsymbol{r}} : \dot{\boldsymbol{r}} - \frac{\partial \hat{\boldsymbol{\psi}}_p}{\partial \gamma}\dot{\gamma} \geq 0. \tag{8.37}$$

Using Coleman–Noll arguments (Coleman and Gurtin 1967), we deduce

$$\boldsymbol{\sigma} = \frac{\partial \hat{\boldsymbol{\psi}}_e}{\partial \boldsymbol{\varepsilon}_e} = \mathbb{C}\boldsymbol{\varepsilon}_e = \mathbb{C}(\boldsymbol{\varepsilon}(\boldsymbol{u}) - \boldsymbol{\varepsilon}_p). \tag{8.38}$$

Now, the power dissipation function $\mathcal{D}$ is introduced as the difference between the external power and the rate of change of Helmholtz free energy

$$\mathcal{D} = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}_p + \boldsymbol{q} : \dot{\boldsymbol{r}} + g\dot{\gamma}$$

where

$$\boldsymbol{q} = -\frac{\partial \hat{\boldsymbol{\psi}}_p}{\partial \boldsymbol{r}} = -\mathbb{H}\boldsymbol{r} \quad \text{and} \quad g = -\frac{\partial \hat{\boldsymbol{\psi}}_p}{\partial \gamma} = -E_{\text{iso}}\gamma. \tag{8.39}$$

Substituting $\mathcal{D}$ in (8.37), we get

$$\mathcal{D} \geq 0.$$

This is exactly Hill's principle (or second law of thermodynamics) and is equivalent to the Drucker–Illyushin's principle of maximum work which states that for any stress state $(\boldsymbol{\sigma}, \boldsymbol{q}, g)$ in $\mathbb{E}$, the plastic flow variables $(\dot{\boldsymbol{\varepsilon}}_p, \dot{\boldsymbol{r}}, \dot{\gamma})$ must satisfy

$$\boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}_p + \boldsymbol{q} : \dot{\boldsymbol{r}} + g\dot{\gamma} \geq \boldsymbol{\tau} : \dot{\boldsymbol{\varepsilon}}_p + \boldsymbol{p} : \dot{\boldsymbol{r}} + k\dot{\gamma} \qquad \forall (\boldsymbol{\tau}, \boldsymbol{p}, k) \in \mathbb{E}. \tag{8.40}$$

Since the set $\mathbb{E}$ is invariant by addition of a multiple of the identity tensor to $\boldsymbol{\sigma}$ and $\boldsymbol{q}$, (8.40) implies that necessarily the trace of $\dot{\boldsymbol{\varepsilon}}_p + \dot{\boldsymbol{r}}$ vanishes. Furthermore, (8.40) yields the following characterization of $\mathcal{D}$

$$\mathcal{D}(\dot{\boldsymbol{\varepsilon}}_p, \dot{\boldsymbol{r}}, \dot{\gamma}) = \sup_{(\boldsymbol{\tau}, \boldsymbol{p}, k) \in \mathbb{E}} \left(\boldsymbol{\tau} : \dot{\boldsymbol{\varepsilon}}_p + \boldsymbol{p} : \dot{\boldsymbol{r}} + k\dot{\gamma}\right), \tag{8.41}$$

where the supremum is attained at $(\boldsymbol{\sigma}, \boldsymbol{q}, g)$. This maximization ensures that the normality law is satisfied (Han and Reddy 2013)

$$f(\boldsymbol{\sigma}, \boldsymbol{q}, g) < 0 \implies \dot{\boldsymbol{\varepsilon}}_p = \mathbf{0},\ \dot{\boldsymbol{r}} = \mathbf{0},\ \dot{\gamma} = 0$$
$$f(\sigma, \boldsymbol{q}, g) = 0 \implies \dot{\boldsymbol{\varepsilon}}_p = \zeta\, \partial_{\boldsymbol{\sigma}} f,\ \dot{\boldsymbol{r}} = \zeta\, \partial_{\boldsymbol{q}} f,\ \dot{\gamma} = \zeta\, \partial_g f, \tag{8.42}$$

where ζ is a Lagrange multiplier satisfying

$$\zeta \geq 0 \quad \text{and} \quad \zeta f(\boldsymbol{\sigma}, \boldsymbol{q}, g) = 0.$$

The derivatives of f (normal to the elastic domain) are given by

$$\partial_{\boldsymbol{\sigma}} f = \frac{\boldsymbol{\sigma}^D - \boldsymbol{q}^D}{|\boldsymbol{\sigma}^D - \boldsymbol{q}^D|}, \quad \partial_{\boldsymbol{q}} f = -\frac{\boldsymbol{\sigma}^D - \boldsymbol{q}^D}{|\boldsymbol{\sigma}^D - \boldsymbol{q}^D|} \quad \text{and} \quad \partial_g f = \sqrt{\frac{2}{3}}.$$

The multiplier ζ is determined by imposing the consistency condition $\dot{f} = 0$ (Simo and Hughes 2006) and in our case (of linear isotropic and kinematic hardening) an analytic formula is available in the plastic zone (where $f = 0$)

$$\zeta = \frac{\partial_{\boldsymbol{\sigma}} f : \dot{\boldsymbol{\sigma}}}{\sqrt{\frac{2}{3}} E_{\text{iso}} + \mathbb{H} \partial_{\boldsymbol{\sigma}} f : \partial_{\boldsymbol{\sigma}} f}.$$

From (8.42), we get $\dot{\boldsymbol{\varepsilon}}_p = -\dot{\boldsymbol{r}}$. Assuming that the plastic variables $\boldsymbol{\varepsilon}_p$ and $\boldsymbol{r}$ are zero at the initial time instant, we deduce $\boldsymbol{\varepsilon}_p = -\boldsymbol{r}$ for all time t. The internal variable $\boldsymbol{r}$ has thus been characterized and $\mathcal{D}(\dot{\boldsymbol{\varepsilon}}_p, \dot{\boldsymbol{r}}, \dot{\gamma}) = \mathcal{D}(\dot{\boldsymbol{\varepsilon}}_p, \dot{\gamma})$. Using the definition of a sub-differential, the maximization (8.41) can then be written as

$$\mathcal{D}(\boldsymbol{\varepsilon}_q, \mu) \geq \mathcal{D}(\dot{\boldsymbol{\varepsilon}}_p, \dot{\gamma}) + (\boldsymbol{\sigma} - \boldsymbol{q}) : (\boldsymbol{\varepsilon}_q - \dot{\boldsymbol{\varepsilon}}_p) + g(\mu - \dot{\gamma}) \quad \forall \boldsymbol{\varepsilon}_q \in \mathcal{M}_s^d,\ \mu \in \mathbb{R}. \tag{8.43}$$

The primal variables are $(\boldsymbol{u}, \boldsymbol{\varepsilon}_p, \gamma)$. We wish to work with a primal formulation, and hence, we need an expression of $\mathcal{D}(\dot{\boldsymbol{\varepsilon}}_p, \dot{\gamma})$ in terms of the primal variables. The dissipation function $\mathcal{D}$ satisfies (Reddy and Martin 1994)

$$\mathcal{D}(\dot{\boldsymbol{\varepsilon}}_p, \dot{\gamma}) = \begin{cases} \sqrt{\frac{2}{3}} \sigma_Y |\dot{\boldsymbol{\varepsilon}}_p| & \text{if } \sqrt{\frac{2}{3}} |\dot{\boldsymbol{\varepsilon}}_p| \leq \dot{\gamma}, \\ \infty & \text{if } \sqrt{\frac{2}{3}} |\dot{\boldsymbol{\varepsilon}}_p| > \dot{\gamma}. \end{cases} \tag{8.44}$$

The above expression is obtained by substituting $f = 0$ in (8.41) and performing simple algebra to determine the variables $(\boldsymbol{\sigma}, \boldsymbol{q}, g)$, which maximize $\mathcal{D}$. The first expression in (8.44) also follows from a simple substitution of (8.42) in (8.41). As a consequence, the domain of $\mathcal{D}$ is defined by

$$\text{dom}\mathcal{D} = \left\{ (\dot{\boldsymbol{\varepsilon}}_p, \dot{\gamma}),\ \sqrt{\frac{2}{3}} |\dot{\boldsymbol{\varepsilon}}_p| \leq \dot{\gamma} \text{ a.e. in } \Omega \right\}. \tag{8.45}$$

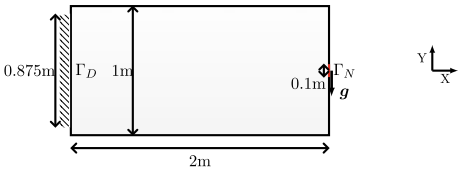


Fig. 8.17 2D Cantilever boundary conditions

Eventually, the plasticity model used in this chapter is:

$$\begin{aligned}
\boldsymbol{\sigma} &= \mathbb{C}(\boldsymbol{\varepsilon}(\boldsymbol{u}) - \boldsymbol{\varepsilon}_p) && \text{in } \Omega \times (0,T], \qquad (8.46)\\
\operatorname{div}(\boldsymbol{\sigma}) + \boldsymbol{f} &= \mathbf{0} && \text{in } \Omega \times (0,T],\\
\boldsymbol{\sigma}\cdot\boldsymbol{n} &= g && \text{on } \Gamma_N \times (0,T],\\
\boldsymbol{\sigma}\cdot\boldsymbol{n} &= \mathbf{0} && \text{on } \Gamma \times (0,T],\\
\boldsymbol{u} &= \mathbf{0} && \text{on } \Gamma_D \times (0,T],\\
\sqrt{\frac{2}{3}}\sigma_Y|\boldsymbol{\varepsilon}_q| &\geq \sqrt{\frac{2}{3}}\sigma_Y|\dot{\boldsymbol{\varepsilon}}_p| + (\boldsymbol{\sigma} - \mathbb{H}\boldsymbol{\varepsilon}_p) : (\boldsymbol{\varepsilon}_q - \dot{\boldsymbol{\varepsilon}}_p)\\
&\quad - E_{\text{iso}}\gamma(\mu - \dot{\gamma}) \quad \forall(\boldsymbol{\varepsilon}_q, \mu) \in \operatorname{dom}\mathcal{D} && \text{on } \Omega \times (0,T]. \qquad (8.47)
\end{aligned}$$

The inequality (8.47) is obtained by injecting (8.39) and (8.44) in (8.43).

Very often, the partial differential equations (8.46) are solved in conjunction with the ordinary differential equations (8.42). But here, we solve (8.46) coupled to the inequation (8.47). This coupling, which is purely in terms of the variables $(\boldsymbol{u}, \boldsymbol{\varepsilon}_p, \gamma)$ results in the so-called primal formulation.

If the dissipation function $\mathcal{D}$ is expressed in terms of the stress variables, the plasticity problem is formulated in terms of $(\boldsymbol{u}, \boldsymbol{\sigma}, \boldsymbol{q}, g)$ resulting in the dual formulation. The analytical treatment of the primal formulation being much easier than that of the dual formulation, we have chosen the former.

8.6.2 Variational (Primal) Formulation

The material tensors $\mathbb{C}$ and $\mathbb{H}$ are assumed to be coercive, i.e., $\exists\ c_0 > 0,\ \exists\ h_0 > 0$ such that, $\forall \boldsymbol{\xi} \in \mathcal{M}_s^d$,

$$\mathbb{C}\boldsymbol{\xi} : \boldsymbol{\xi} \geq c_0|\boldsymbol{\xi}|^2 \quad \text{and} \quad \mathbb{H}\boldsymbol{\xi} : \boldsymbol{\xi} \geq h_0|\boldsymbol{\xi}|^2.$$

We define the displacement space

$$V = \{\boldsymbol{u} \in H^1(\Omega)^d,\ \boldsymbol{u} = \mathbf{0} \text{ on } \Gamma_D\}$$

and the space of plastic strain Q as

$$Q = \{\boldsymbol{\varepsilon}_q \in L^2(\Omega)^{d\times d},\ \boldsymbol{\varepsilon}_q = \boldsymbol{\varepsilon}_q^T,\ \mathrm{tr}(\boldsymbol{\varepsilon}_q) = \mathbf{0} \text{ a.e. in } \Omega\}.$$

We then define the product space

$$Z = V \times Q \times L^2(\Omega), \tag{8.48}$$

where we seek the solution $\boldsymbol{w} = (\boldsymbol{u}, \boldsymbol{\varepsilon}_p, \gamma)$. The space Z is a Hilbert space equipped with the scalar product, for $\boldsymbol{w} = (\boldsymbol{u}, \boldsymbol{\varepsilon}_p, \gamma)$ and $\boldsymbol{z} = (\boldsymbol{v}, \boldsymbol{\varepsilon}_q, \mu)$,

$$\langle \boldsymbol{w}, \boldsymbol{z}\rangle = \int_\Omega \boldsymbol{u}\cdot\boldsymbol{v}\ \mathrm{d}x + \int_\Omega \boldsymbol{\varepsilon}_p : \boldsymbol{\varepsilon}_q\ \mathrm{d}x + \int_\Omega \gamma\mu\ \mathrm{d}x. \tag{8.49}$$

Let Z^* be the dual space of Z. The forces are assumed to be smooth as

$$\boldsymbol{f} \in H^1([0,T], L^2(\Omega)^d) \quad \text{and} \quad g \in H^1([0,T], L^2(\Gamma_N)^d).$$

Indeed, since $H^1([0,T], H) \subset \mathcal{C}^0([0,T], H)$ for any Hilbert space H, at any time t the forces $\boldsymbol{f}(t)$ and $g(t)$ are well defined. We introduce a bilinear form $a : Z \times Z \to \mathbb{R}$,

$$a(\boldsymbol{w}, \boldsymbol{z}) = \int_\Omega \left(\mathbb{C}(\boldsymbol{\varepsilon}(\boldsymbol{u}) - \boldsymbol{\varepsilon}_p) : (\boldsymbol{\varepsilon}(\boldsymbol{v}) - \boldsymbol{\varepsilon}_q) + \boldsymbol{\varepsilon}_q : \mathbb{H}\boldsymbol{\varepsilon}_p + E_{\mathrm{iso}}\gamma\mu\right)\ \mathrm{d}x, \tag{8.50}$$

and a linear form $l_t : Z \to \mathbb{R}$ such that

$$l_t(z) = \int_\Omega \boldsymbol{f}(t)\cdot\boldsymbol{v}\ \mathrm{d}x + \int_{\Gamma_N} g(t)\cdot\boldsymbol{v}\ \mathrm{d}s, \tag{8.51}$$

with the forces $\boldsymbol{f}(t) \in L^2(\Omega)^d$, $g(t) \in L^2(\Gamma_N)^d$ and a nonlinear convex functional $j : Z \to \mathbb{R}$ such that

$$j(\boldsymbol{z}) = \int_\Omega \mathcal{D}(\boldsymbol{\varepsilon}_q, \mu)\ \mathrm{d}x, \tag{8.52}$$

where $\mathcal{D}(\boldsymbol{\varepsilon}_q, \mu)$ is defined by (8.44). This functional $j(\cdot)$ is convex and lower semi-continuous on Z and it is Lipschitz continuous on the convex set $K \subset Z$ defined as

$$K = V \times \mathrm{dom}\mathcal{D},$$

where $\mathrm{dom}\mathcal{D}$ is defined by (8.45). The admissible plastic flow rates $\dot{\boldsymbol{\varepsilon}}_p,\ \dot{\gamma}$ belong to the convex set $\mathrm{dom}\mathcal{D}$.

In order to obtain the primal formulation of (8.46) and (8.47), we multiply (8.34) by $\boldsymbol{v} - \dot{\boldsymbol{u}}$, use (8.38) and integrate the product over Ω by parts to obtain

$$\int_\Omega \mathbb{C}(\boldsymbol{\varepsilon}(\boldsymbol{u}) - \boldsymbol{\varepsilon}_p) : (\boldsymbol{\varepsilon}(\boldsymbol{v}) - \boldsymbol{\varepsilon}(\dot{\boldsymbol{u}}))\, \mathrm{d}x = \int_\Omega \boldsymbol{f}(t) \cdot (\boldsymbol{v} - \dot{\boldsymbol{u}})\, \mathrm{d}x + \int_{\Gamma_N} g(t) \cdot (\boldsymbol{v} - \dot{\boldsymbol{u}}) \mathrm{d}s \quad \forall \boldsymbol{v} \in V, \tag{8.53}$$

We then integrate (8.47) over Ω, add (8.53) to it and obtain the variational inequality, for any $\boldsymbol{z} \in K$,

$$\int_\Omega \sqrt{\frac{2}{3}}\sigma_Y |\boldsymbol{\varepsilon}_q|\, \mathrm{d}x \geq \int_\Omega \sqrt{\frac{2}{3}}\sigma_Y |\dot{\boldsymbol{\varepsilon}}_p|\, \mathrm{d}x + \int_\Omega \boldsymbol{f}(t) \cdot (\boldsymbol{v} - \dot{\boldsymbol{u}})\, \mathrm{d}x + \int_{\Gamma_N} g(t) \cdot (\boldsymbol{v} - \dot{\boldsymbol{u}})\, \mathrm{d}s$$
$$- \int_\Omega \left(\mathbb{C}(\boldsymbol{\varepsilon}(\boldsymbol{u}) - \boldsymbol{\varepsilon}_p) : (\boldsymbol{\varepsilon}(\boldsymbol{v}) - \boldsymbol{\varepsilon}_q - \boldsymbol{\varepsilon}(\dot{\boldsymbol{u}}) + \dot{\boldsymbol{\varepsilon}}_p) + \mathbb{H}\boldsymbol{\varepsilon}_p : (\boldsymbol{\varepsilon}_q - \dot{\boldsymbol{\varepsilon}}_p) + E_{\mathrm{iso}}\gamma(\mu - \dot{\gamma}) \right) \mathrm{d}x.$$

We complement this variational inequality with the following initial conditions

$$\boldsymbol{u}(0) = \boldsymbol{0}, \quad \boldsymbol{\varepsilon}_p(0) = \boldsymbol{0}, \quad \gamma(0) = 0 \qquad \text{in } \Omega.$$

To prove existence and uniqueness of a solution, we rely on Theorem 4.3 in (Han et al. 1997) which requires some additional regularity in time for the solution. Therefore, hence the forces are assumed to satisfy

$$\boldsymbol{f}(0) = \boldsymbol{0} \quad \text{in } \Omega \quad \text{and} \quad g(0) = \boldsymbol{0} \quad \text{on } \Gamma_N.$$

Using the linear forms and the nonlinear functional defined earlier, the primal form of the plasticity problem (8.46) and (8.47) is obtained: find $\boldsymbol{w}(t) = (\boldsymbol{u}, \boldsymbol{\varepsilon}_p, \gamma)(t)$ with $\boldsymbol{w}(0) = \boldsymbol{0}$ such that $\dot{\boldsymbol{w}}(t) \in K$ (for almost all $t \in (0, T)$) and

$$a(\boldsymbol{w}, \boldsymbol{z} - \dot{\boldsymbol{w}}) + j(\boldsymbol{z}) - j(\dot{\boldsymbol{w}}) \geq l_t(\boldsymbol{z} - \dot{\boldsymbol{w}}) \quad \forall \boldsymbol{z} \in K. \tag{8.54}$$

As a result of Theorem 4.3 in (Han et al. 1997), the variational inequality (8.54) is well-posed.

Theorem 2 *(Han et al. 1997) Let Z be a Hilbert space; $K \subset Z$ be a nonempty, closed, convex cone; $a : Z \times Z \to \mathbb{R}$ a continuous bilinear form that is symmetric and coercive; $j : K \to \mathbb{R}$ non-negative, convex, positively homogeneous, Lipschitz continuous form; $l_t \in H^1([0, T], Z^*)$ with $l_0(\cdot) = 0$. Then there exists a unique $\boldsymbol{w} \in H^1([0, T], Z)$ satisfying (8.54).*

Remark 1 In the absence of kinematic hardening or $h_0 = 0$, one cannot show the coercivity of $a(\cdot, \cdot)$ and thus the well-posedness of the problem (8.54).

Equation (8.54) is not shape-differentiable (Mignot 1976; Sokolowski and Zolésio 1992) in the classical sense and it ought to be approximated it by a smooth variational equation. The non-differentiability of (8.54) is due to $\mathcal{D}$, which is discontinuous

exactly where $\sqrt{\frac{2}{3}}|\dot{\boldsymbol{\varepsilon}}_p| = \dot{\gamma}$ (or equivalently, where $f = 0$). Thus, the function $\mathcal{D}$ admits only directional derivatives where the yield limit f is attained.

8.6.3 Penalization and Regularization

We approximate the problem (8.54) posed on the convex set K by a problem posed on the full vector space Z by penalizing the constraint $z(t) \in K$. The nonlinearity $j(z)$ being unbounded for $z \notin K$, (8.54) is not differentiable with respect to parameters like the shape of the domain (Mignot 1976; Sokolowski and Zolésio 1992). Moreover, $j(z)$ is non-smooth because of the norm of the plastic tensor. We therefore the nonlinearity $j(\cdot)$ must be regularized.

Let $0 \ll \epsilon < 1$ be the penalization parameter and $0 < \eta \ll 1$ be the regularization parameter. Then dissipation function is approximation as

$$\mathcal{D}_{\epsilon,\eta}(\dot{\boldsymbol{w}}) = \sqrt{\frac{2}{3}}\sigma_Y \left(N_\eta \left(\dot{\boldsymbol{\varepsilon}}_p \right) + \frac{1}{\epsilon} M_\eta \left(\sqrt{\frac{2}{3}} N_\eta(\dot{\boldsymbol{\varepsilon}}_p) - \dot{\gamma} \right) \right), \tag{8.55}$$

and we define $j_{\epsilon,\eta} : Z \to \mathbb{R}$ as,

$$j_{\epsilon,\eta}(\dot{\boldsymbol{w}}) = \int_\Omega \mathcal{D}_{\epsilon,\eta}(\dot{\boldsymbol{w}}) \, \mathrm{d}x.$$

We consider a new problem: find $\boldsymbol{w}_{\epsilon,\eta}(t) \in Z$ such that $\boldsymbol{w}_{\epsilon,\eta}(0) = \boldsymbol{0}$, $\dot{\boldsymbol{w}}_{\epsilon,\eta}(t) \in Z$ and

$$a(\boldsymbol{w}_{\epsilon,\eta}, z - \dot{\boldsymbol{w}}_{\epsilon,\eta}) + j_{\epsilon,\eta}(z) - j_{\epsilon,\eta}(\dot{\boldsymbol{w}}_{\epsilon,\eta}) \geq l_t(z - \dot{\boldsymbol{w}}_{\epsilon,\eta}) \qquad \forall z \in Z. \tag{8.56}$$

The above variational inequality can be shown to be well-posed (Desai et al. 2021).

Theorem 3 *The variational inequality (8.56) admits a unique solution* $\boldsymbol{w}_{\epsilon,\eta} \in H^1([0, T], Z)$.

One cannot apply directly Theorem 2 because the functional $j_{\epsilon,\eta}$ is not positively homogeneous. The variational inequation (8.56) can be converted into an equation. Since the function $\mathcal{D}_{\epsilon,\eta}$ is smooth, we can define its gradient

$$\nabla_Z \mathcal{D}_{\epsilon,\eta}(\boldsymbol{w}) = \left(\frac{\partial \mathcal{D}_{\epsilon,\eta}(\boldsymbol{w})}{\partial \boldsymbol{u}}, \frac{\partial \mathcal{D}_{\epsilon,\eta}(\boldsymbol{w})}{\partial \boldsymbol{\varepsilon}_p}, \frac{\partial \mathcal{D}_{\epsilon,\eta}(\boldsymbol{w})}{\partial \gamma} \right).$$

Lemma 2 *The variational inequality (8.56) is equivalent to the variational formulation: find* $\boldsymbol{w}_{\epsilon,\eta}(t) \in Z$ *such that* $\boldsymbol{w}_{\epsilon,\eta}(0) = \boldsymbol{0}$, $\dot{\boldsymbol{w}}_{\epsilon,\eta}(t) \in Z$ *and*

$$a(\boldsymbol{w}_{\epsilon,\eta}, z) + \langle \nabla_Z \mathcal{D}_{\epsilon,\eta}(\dot{\boldsymbol{w}}_{\epsilon,\eta}), z\rangle = l_t(z) \quad \forall z \in Z, \tag{8.57}$$

where $\langle , \rangle$ *is the scalar product defined by (8.49).*

Proof By definition of the convexity of $j_{\epsilon,\eta}$ we get

$$j_{\epsilon,\eta}(z) - j_{\epsilon,\eta}(\dot{\boldsymbol{w}}_{\epsilon,\eta}) \geq \langle \nabla_Z \mathcal{D}_{\epsilon,\eta}(\dot{\boldsymbol{w}}_{\epsilon,\eta}), z - \dot{\boldsymbol{w}}_{\epsilon,\eta}\rangle \quad \forall z \in Z.$$

The right hand side in the above is the tangent hyperplane to $j_{\epsilon,\eta}$ at $z = \dot{\boldsymbol{w}}_{\epsilon,\eta}$. On the other hand, (8.56) can be written as

$$j_{\epsilon,\eta}(z) - j_{\epsilon,\eta}(\dot{\boldsymbol{w}}) \geq a(\boldsymbol{w}_{\epsilon,\eta}, \dot{\boldsymbol{w}}_{\epsilon,\eta} - z) + l_t(z - \dot{\boldsymbol{w}}_{\epsilon,\eta}) \quad \forall z \in Z.$$

Again, the right hand side in the above is affine in z and it vanishes at $z = \dot{\boldsymbol{w}}_{\epsilon,\eta}$, implying that it is also tangent at $z = \dot{\boldsymbol{w}}_{\epsilon,\eta}$. Since $j_{\epsilon,\eta}$ is smooth, the two tangent hyperplanes must be equal

$$a(\boldsymbol{w}_{\epsilon,\eta}, \dot{\boldsymbol{w}}_{\epsilon,\eta} - z) + l_t(z - \dot{\boldsymbol{w}}_{\epsilon,\eta}) = \langle \nabla_Z \mathcal{D}_{\epsilon,\eta}(\dot{\boldsymbol{w}}_{\epsilon,\eta}), z - \dot{\boldsymbol{w}}_{\epsilon,\eta}\rangle \quad \forall z \in Z.$$

Replacing z in the above by $\dot{\boldsymbol{w}}_{\epsilon,\eta} + z \in Z$, we deduce (8.57). □

Equation (8.57) is our approximation of the plasticity problem (8.54) we treat for the computation of the shape derivative. In the context of plasticity, we call it the state equation and its solution, the state solution. As expected, for a fixed ϵ, one can prove the convergence of the sequence $\boldsymbol{w}_{\epsilon,\eta}$ of solutions to (8.56) to the solution $\boldsymbol{w}_\epsilon$ to (8.93) as $\eta \longrightarrow 0$, at least in the weak sense (Desai et al. 2021).

Theorem 4 *The sequence of solutions* $\boldsymbol{w}_{\epsilon,\eta}$ *to (8.56) satisfies*

$$\eta \longrightarrow 0, \quad \boldsymbol{w}_{\epsilon,\eta} \overset{*}{\rightharpoonup} \boldsymbol{w}_\epsilon \ \text{ in } L^\infty([0,T], Z) \ \text{ and } \ \dot{\boldsymbol{w}}_{\epsilon,\eta} \rightharpoonup \dot{\boldsymbol{w}}_\epsilon \ \text{ in } L^2([0,T], Z),$$

where $\boldsymbol{w}_\epsilon$ *is the solution to (8.93).*

8.6.4 Shape Derivative Computation

In this section, to simplify the notations, we drop the indices ϵ and η, and simply write $\boldsymbol{w}$ instead of $\boldsymbol{w}_{\epsilon,\eta}$. We minimize an objective function $J(\Omega)$ defined as

$$J(\Omega) = \int_0^T \left(\int_\Omega m(\boldsymbol{w}(\Omega))\, \mathrm{d}x + \int_{\Gamma_N} p(\boldsymbol{w}(\Omega)) \mathrm{d}s \right) \mathrm{d}t, \tag{8.58}$$

where $\boldsymbol{w}(\Omega)$ is solution to the state equation (8.57) and the integrands $m(\cdot)$ and $p(\cdot)$ are assumed to be smooth functions at least of class $\mathcal{C}^1$. In addition we assume a

growth condition on $m(\cdot)$ and $p(\cdot)$ such that the objective function is well-defined and the adjoint equation (8.61) is well-posed. This objective can represent a mechanical property such as the total compliance, total power, elastic energy, plastic energy as well as a geometric property such as the volume. An industrially relevant objective is the total compliance, given by

$$J(\Omega) = \int_0^T \int_{\Gamma_N} g \cdot \boldsymbol{u}(\Omega) \ \mathrm{d}s \ \mathrm{d}t. \tag{8.59}$$

In practice, the shape Ω is designed inside a pre-fixed design space $D \subset \mathbb{R}^d$. As shown in Fig. 8.4, the blue region represents the shape Ω, and the blue and gray area represent the design space D. The space of admissible shapes $\mathcal{U}_{\text{ad}}$ is given by the definition 8.4. The optimization problem then reads

$$\min_{\Omega \in \mathcal{U}_{\text{ad}}} J(\Omega).$$

Like in the case of linear elasticity (Sect. 8.3.3), we content ourselves with computing numerical minimizers, using a gradient-descent method. Again, the structure of the clamped and the forced boundaries is assumed to be non-optimizable. Hence in our optimization, we allow only Γ to move along $\boldsymbol{\theta}$ as shown in Fig. 8.29 and the space $W_0^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ (Def.8.5) is considered for the admissible directions $\boldsymbol{\theta}$. Since the regularized nonlinearity $j_{\epsilon,\eta}(\cdot)$ is $\mathcal{C}^\infty$, it is possible to compute the shape derivative of the objective function $J(\Omega)$ defined by (8.58).

Theorem 5 *Let $\Omega \subset \mathbb{R}^d$ be a smooth bounded open set. Let $\boldsymbol{f} \in \mathcal{C}^0([0,T], H^1(\mathbb{R}^d)^d)$, $g \in \mathcal{C}^0([0,T], H^2(\mathbb{R}^d)^d)$ and $\boldsymbol{w}(\Omega) \in H^1([0,T], Z)$ the solution to (8.57). Then the shape derivative of $J(\Omega)$ along $\boldsymbol{\theta} \in W_0^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$, $J'(\Omega)(\boldsymbol{\theta})$ is given by*

$$J'(\Omega)(\boldsymbol{\theta}) = \int_0^T \int_{\Gamma} \boldsymbol{\theta} \cdot \boldsymbol{n} \Big(m(\boldsymbol{w}) + \mathbb{C}(\boldsymbol{\varepsilon}(\boldsymbol{u}) - \boldsymbol{\varepsilon}_p) : (\boldsymbol{\varepsilon}(\boldsymbol{v}) - \boldsymbol{\varepsilon}_q) + \boldsymbol{\varepsilon}_q : \mathbb{H}\boldsymbol{\varepsilon}_p + E_{iso}\gamma\mu + \nabla_Z \mathcal{D}_{\epsilon,\eta}(\dot{\boldsymbol{w}}) \cdot \boldsymbol{z} - l_t(\boldsymbol{z}) \Big) ds\, dt, \tag{8.60}$$

where $\boldsymbol{z}(\Omega) \in H^1([0,T], Z)$ is the solution to the adjoint problem, with the final condition $\boldsymbol{z}(T) = \boldsymbol{0}$,

$$a(\boldsymbol{z}, \boldsymbol{\varphi}) - \left\langle \frac{d}{dt} \left(\nabla_Z^2 \mathcal{D}_{\epsilon,\eta}(\dot{\boldsymbol{w}}) z \right), \boldsymbol{\varphi} \right\rangle = -\langle \nabla_Z m(\boldsymbol{w}), \boldsymbol{\varphi} \rangle - \int_{\Gamma_N} \nabla_Z p(\boldsymbol{w}) \boldsymbol{\varphi} \, ds \qquad \forall t \in [0,T), \ \forall \boldsymbol{\varphi} \in Z, \tag{8.61}$$

which is assumed to be well-posed (recall that $\langle , \rangle$ is the scalar product defined by (8.49) in Z).

Proof The idea of the proof is classical and, assuming that the adjoint equation is well-posed, it relies on Céa's technique (1986). Define three spaces $\tilde{V}$, $\tilde{Q}$ and $\tilde{Z} = \tilde{V} \times \tilde{Q} \times L^2(\mathbb{R}^d)$ (which are similar to those in (8.48) except that Ω is replaced by $\mathbb{R}^d$) by

$$\tilde{V} = \{\boldsymbol{u} \in H^1(\mathbb{R}^d)^d,\ \boldsymbol{u} = \mathbf{0} \text{ on } \Gamma_D\} \quad \text{and} \quad \tilde{Q} = \{\boldsymbol{\varepsilon}_q \in L^2(\mathbb{R}^d)^{d\times d},\ \mathrm{tr}(\boldsymbol{\varepsilon}_q) = 0 \text{ a.e. in } \mathbb{R}^d\}. \tag{8.62}$$

For $\tilde{\boldsymbol{w}} = (\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{\varepsilon}}_p, \tilde{\gamma}) \in H^1([0,T], \tilde{Z}), \tilde{\boldsymbol{z}} = (\tilde{\boldsymbol{v}}, \tilde{\boldsymbol{\varepsilon}}_q, \tilde{\mu}) \in H^1([0,T], \tilde{Z})$ [the Lagrange multiplier for the state equation (8.57)] and $\tilde{\boldsymbol{\lambda}} \in L^2(\mathbb{R}^d)^d$ (the Lagrange multiplier for the initial condition $\tilde{\boldsymbol{w}}(0) = \mathbf{0}$), define a Lagrangian by

$$\begin{aligned}\mathcal{L}(\Omega, \tilde{\boldsymbol{w}}, \tilde{\boldsymbol{z}}, \tilde{\boldsymbol{\lambda}}) \ &= \int_0^T \left(\int_\Omega m(\tilde{\boldsymbol{w}})\, \mathrm{d}x + \int_{\Gamma_N} p(\tilde{\boldsymbol{w}}) \mathrm{d}s \right) \mathrm{d}t \\ &+ \int_0^T \left(a(\tilde{\boldsymbol{w}}, \tilde{\boldsymbol{z}}) - l_t(\tilde{\boldsymbol{z}}) + \langle \nabla_Z \mathcal{D}_{\epsilon,\eta}(\dot{\tilde{\boldsymbol{w}}}), \tilde{\boldsymbol{z}} \rangle \right) \mathrm{d}t + \int_\Omega \tilde{\boldsymbol{\lambda}} \cdot \tilde{\boldsymbol{w}}(0)\, \mathrm{d}x. \end{aligned} \tag{8.63}$$

We remark that here the variables $\tilde{\boldsymbol{w}}(t), \tilde{\boldsymbol{z}}(t)$ and $\tilde{\boldsymbol{\lambda}}$ are defined on the full space $\mathbb{R}^d$ and are thus independent of Ω. Although $\tilde{\boldsymbol{u}}(t)$ and $\tilde{\boldsymbol{v}}(t)$ are required to vanish on Γ_D, they do not depend on Ω since Γ_D is a fixed boundary. Therefore, writing the optimality conditions applied to the Lagrangian (8.63), namely that its partial derivatives with respect to the independent variables $(\Omega, \boldsymbol{w}, z, \boldsymbol{\lambda})$ vanishes, yields the state equation, the adjoint equation and the shape derivative.

When the Lagrangian (8.63) is differentiated with respect to the adjoint variable $\tilde{\boldsymbol{z}}$, along $\boldsymbol{\varphi} \in H^1([0,T], \tilde{Z})$, and equated to zero, followed by the substitution $\tilde{\boldsymbol{w}} = \boldsymbol{w}$, we get

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}}(\boldsymbol{\varphi}) = \int_0^T \left(a(\boldsymbol{w}, \boldsymbol{\varphi}) + \langle \nabla_Z \mathcal{D}_{\epsilon,\eta}(\dot{\boldsymbol{w}}), \boldsymbol{\varphi} \rangle - l_t(\boldsymbol{\varphi}) \right) \mathrm{d}t = 0 \qquad \forall \boldsymbol{\varphi} \in H^1([0,T], \tilde{Z}).$$

Since the bilinear form $a(\cdot,\cdot)$ and the linear forms in the above are defined only on Ω, we can replace $\tilde{Z}$ by Z. Differentiating (8.63) with respect to $\tilde{\boldsymbol{\lambda}}$ at $\tilde{\boldsymbol{w}} = \boldsymbol{w}$, equating it to zero, we deduce the initial condition $\boldsymbol{w}(0) = \mathbf{0}$ a.e. on Ω. We thus recover the state equation (8.57). Next, we differentiate the Lagrangian (8.63) with respect to $\tilde{\boldsymbol{w}}$ along $\boldsymbol{\varphi} \in H^1([0,T], \tilde{Z})$ and equate it to zero at $\tilde{\boldsymbol{w}} = \boldsymbol{w},\ \tilde{\boldsymbol{z}} = z,\ \tilde{\boldsymbol{\lambda}} = \boldsymbol{\lambda}$, to get

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}(\boldsymbol{\varphi}) = &\int_0^T \int_\Omega \nabla_Z m(\boldsymbol{w}) \boldsymbol{\varphi}\ \mathrm{d}x\ \mathrm{d}t + \int_0^T \int_{\Gamma_N} \nabla_Z p(\boldsymbol{w}) \boldsymbol{\varphi}\ \mathrm{d}s\ \mathrm{d}t \\ &+ \int_0^T a(\boldsymbol{\varphi}, z) \mathrm{d}t + \int_0^T \langle \nabla_Z^2 \mathcal{D}_{\epsilon,\eta}(\dot{\boldsymbol{w}}) \dot{\boldsymbol{\varphi}}, z \rangle \mathrm{d}t + \int_\Omega \boldsymbol{\lambda} \cdot \boldsymbol{\varphi}(0)\, \mathrm{d}x = 0 \qquad \forall \boldsymbol{\varphi} \in H^1([0,T], \tilde{Z}).\end{aligned}$$

Using the symmetry of the second derivative $\nabla_Z^2 \mathcal{D}_{\epsilon,\eta}(\dot{\boldsymbol{w}})$, and integrating by parts in time, we deduce

$$\int_0^T \int_\Omega \nabla_Z m(\boldsymbol{w})\boldsymbol{\varphi}\,\mathrm{d}x\,\mathrm{d}t + \int_0^T \int_{\Gamma_N} \nabla_Z p(\boldsymbol{w})\boldsymbol{\varphi}\,\mathrm{d}s\,\mathrm{d}t + \int_0^T a(\boldsymbol{\varphi}, z)\mathrm{d}t + \langle \boldsymbol{\varphi}, \nabla_Z^2 \mathcal{D}_{\epsilon,\eta}(\dot{\boldsymbol{w}})z\rangle\big|_{t=T}$$

$$- \langle \boldsymbol{\varphi}, \nabla_Z^2 \mathcal{D}_{\epsilon,\eta}(\dot{\boldsymbol{w}})z\rangle\big|_{t=0} - \int_0^T \left\langle \boldsymbol{\varphi}, \frac{\mathrm{d}}{\mathrm{d}t}(\nabla_Z^2 \mathcal{D}_{\epsilon,\eta}(\dot{\boldsymbol{w}})z)\right\rangle \mathrm{d}t + \int_\Omega \boldsymbol{\lambda}\cdot\boldsymbol{\varphi}(0)\,\mathrm{d}x = 0 \quad \forall \boldsymbol{\varphi} \in H^1([0,T],\tilde{Z}).$$

Since all integrals in the above are defined only on Ω, we can replace $\tilde{Z}$ by Z. Varying the test function $\boldsymbol{\varphi}$, we derive the following adjoint equation:

$$\boldsymbol{\lambda} = \nabla_Z^2 \mathcal{D}_{\epsilon,\eta}(\dot{\boldsymbol{w}})z\big|_{t=0}, \quad z(T) = \mathbf{0} \quad \text{and}$$

$$a(z,\boldsymbol{\varphi}) - \left\langle \frac{\mathrm{d}}{\mathrm{d}t}\left(\nabla_Z^2 \mathcal{D}_{\epsilon,\eta}(\dot{\boldsymbol{w}})z\right), \boldsymbol{\varphi}\right\rangle = -\langle \nabla_Z m(\boldsymbol{w}), \boldsymbol{\varphi}\rangle - \int_{\Gamma_N} \nabla_Z p(\boldsymbol{w})\boldsymbol{\varphi}\,\mathrm{d}s \qquad t \in [0,T),\ \forall\, \boldsymbol{\varphi} \in Z.$$

Finally, using the relation $J(\Omega) = \mathcal{L}(\Omega, \boldsymbol{w}, \tilde{z}, \tilde{\boldsymbol{\lambda}})$, we determine the shape derivative $J'(\Omega)(\boldsymbol{\theta})$ for any $\boldsymbol{\theta} \in W_0^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ by

$$J'(\Omega)(\boldsymbol{\theta}) = \frac{\partial \mathcal{L}}{\partial \Omega}(\boldsymbol{\theta}) + \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}\left(\frac{\partial \boldsymbol{w}}{\partial \Omega}(\boldsymbol{\theta})\right),$$

because $\tilde{z}$ and $\tilde{\boldsymbol{\lambda}}$ do not depend on Ω. Now, replacing them by their precise values z and $\boldsymbol{\lambda}$, given by the adjoint problem, the last term cancels to get

$$J'(\Omega)(\boldsymbol{\theta}) = \frac{\partial \mathcal{L}}{\partial \Omega}(\boldsymbol{\theta})$$

and formula (8.60) is deduced by application of Lemma 1. □

The time discretized version of the adjoint equation (8.61) can be shown to be well-posed (Desai et al. 2021).

8.6.5 *Numerical Implementation*

We first discuss the numerical resolution of the state equation (8.57), then that of the adjoint equation (8.61), and finally, we describe the shape optimization algorithm. The domain Ω is discretized using a simplicial unstructured mesh and the space Z, defined by (8.48), is discretized as Z^h, using the finite element framework

$$Z^h = \mathbb{P}^1(\Omega)^d \times \mathbb{P}^0(\Omega)^{d\times d} \times \mathbb{P}^0(\Omega). \tag{8.64}$$

The space K is discretized as K^h, defined by

$$K^h = \left\{ (\boldsymbol{u}, \boldsymbol{\varepsilon}_p, \gamma) \in Z^h,\ \sqrt{\frac{2}{3}} |\boldsymbol{\varepsilon}_p| \leq \gamma \text{ a.e. in } \Omega \right\}. \tag{8.65}$$

The maximal mesh size is denoted by $h_{\max}$, the minimal mesh size by $h_{\min}$ and the number of mesh vertices is N_v. We assume the mesh to be regular, or $h_{\max}$ and $h_{\min}$ to be of the same order. The space-time discretized state solution is $\tilde{\boldsymbol{w}}(t) \in Z^h$ and the space-time discretized adjoint solution is $\tilde{z}(t) \in Z^h$. The time interval $[0, T]$ is discretized in N intervals of length δt. We label the time at the end of n-th time interval as t_n, $n = 1, 2, \ldots, N$. All our numerical experiments are performed with the open-source software `FreeFEM++` (Hecht 2012).

8.6.5.1 Resolution of the Plasticity Formulation

The space discretized version of the problem (8.54) reads: find $\boldsymbol{w}^h(t) \in K^h$ such that

$$a(\boldsymbol{w}^h, z^h - \dot{\boldsymbol{w}}^h) + j(z^h) - j(\dot{\boldsymbol{w}}^h) \geq l_t(z^h - \dot{\boldsymbol{w}}^h) \qquad \forall z^h \in K^h. \tag{8.66}$$

The solution of the time discretized version of (8.66) is denoted by $\tilde{\boldsymbol{w}}(t) \in Z^h$. More precisely, it is defined by its values $\tilde{\boldsymbol{w}}_n = \tilde{\boldsymbol{w}}(t_n)$ at each time step and extended by affine interpolation as $\tilde{\boldsymbol{w}}(t) = \tilde{\boldsymbol{w}}_n + \delta\tilde{\boldsymbol{w}}_n(t - t_n)$ for $t \in [t_n, t_{n+1}]$, where $\delta\tilde{\boldsymbol{w}}_n = (\tilde{\boldsymbol{w}}_{n+1} - \tilde{\boldsymbol{w}}_n)/\delta t$ is the increment. Equation (8.66) could be regularized and penalized as before but we refrain ourselves from doing so and instead solve its time discretization via the radial return algorithm (Simo and Hughes 2006).

8.6.5.2 Resolution of Adjoint System

We denote by $\tilde{z}_n = \tilde{z}(t_n)$ the discrete values of the adjoint, which is linearly interpolated in time on each sub-interval. The space-time discretized adjoint problem is defined by: $\tilde{z}_N = 0$ and, for $n = N - 1, \ldots, 1, 0$, find the solution $\tilde{z}_n \in Z^h$ of

$$\begin{aligned} &\delta t \langle \nabla_Z m(\tilde{\boldsymbol{w}}_{n+1}), \boldsymbol{\varphi} \rangle + \delta t \int_{\Gamma_N} \nabla_Z p(\tilde{\boldsymbol{w}}_{n+1}) \boldsymbol{\varphi}\, \mathrm{d}s + \delta t a(\boldsymbol{\varphi}, \tilde{z}_n) \\ &\quad + \left\langle \nabla_Z^2 \mathcal{D}_{\epsilon,\eta}(\delta\tilde{\boldsymbol{w}}_n)\tilde{z}_n - \nabla_Z^2 \mathcal{D}_{\epsilon,\eta}(\delta\tilde{\boldsymbol{w}}_{n+1})\tilde{z}_{n+1}, \boldsymbol{\varphi} \right\rangle = 0 \qquad \forall \boldsymbol{\varphi} \in Z^h. \end{aligned} \tag{8.67}$$

This system is going backward in time. One ought to solve the state equation (8.66) until the last time step, store the solutions $\tilde{\boldsymbol{w}}_n$ for every time step and retrieve the solutions one by one starting from the last time step. This is thus quite heavy in terms of memory requirement for numerical simulations. Finally, the time discretized shape derivative reads

$$J'(\Omega)(\boldsymbol{\theta}) = \sum_{n=0}^{N-1} \delta t \int_{\Gamma} \boldsymbol{\theta} \cdot \boldsymbol{n} \Big(m(\tilde{\boldsymbol{w}}_n) + \mathbb{C}(\boldsymbol{\varepsilon}(\tilde{\boldsymbol{u}}_{n+1}) - \tilde{\boldsymbol{\varepsilon}}_{p,n+1}) : (\boldsymbol{\varepsilon}(\tilde{\boldsymbol{v}}_n) - \tilde{\boldsymbol{\varepsilon}}_{q,n}) + \mathbb{H}\tilde{\boldsymbol{\varepsilon}}_{p,n+1} : \tilde{\boldsymbol{\varepsilon}}_{q,n} + E_{\text{iso}}\tilde{\gamma}_{n+1}\tilde{\mu}_n + \nabla_Z \mathcal{D}_{\epsilon,\eta}\left(\frac{\tilde{\boldsymbol{w}}_{n+1} - \tilde{\boldsymbol{w}}_n}{\delta t}\right)\tilde{\boldsymbol{z}}_n - \boldsymbol{f}(t_n) \cdot \tilde{\boldsymbol{v}}_n \Big) \mathrm{d}s, \quad (8.68)$$

where $(\tilde{\boldsymbol{u}}_n, \tilde{\boldsymbol{\varepsilon}}_{p,n}, \tilde{\gamma}_n) = \tilde{\boldsymbol{w}}_n$ and $(\tilde{\boldsymbol{v}}_n, \tilde{\boldsymbol{\varepsilon}}_{q,n}, \tilde{\mu}_n) = \tilde{\boldsymbol{z}}_n$.

In numerical practice, denoting by L a characteristic length of the domain D, we choose the values of ϵ, η for penalization and regularization according to the following rule (see (Desai 2021) for more details)

$$\epsilon = \left(\frac{h_{\min}}{L}\right)^{1+d/2} \quad \text{and} \quad \eta = \epsilon^2. \quad (8.69)$$

Remark 2 For all of our numerical experiments, we replace $\tilde{\boldsymbol{w}}_n$ in the adjoint equation (8.67) and in the shape derivative (8.68) by the solution obtained via radial return, $\boldsymbol{w}_r(t_n) \in K^h$, which does not take into account the penalization and regularization. In formula (8.68) of the shape derivative, and more precisely in the term $\nabla_Z \mathcal{D}_{\epsilon,\eta}$, we neglect the contribution $\frac{1}{\epsilon} M'_\eta(\cdot)$. The reason for this is because we replace the penalized solution $\tilde{\boldsymbol{w}}(t)$ by the non-penalized one $\boldsymbol{w}_r(t)$. For the penalized solution, the contribution $\frac{1}{\epsilon} M'_\eta\left(\frac{\tilde{\boldsymbol{w}}_{n+1} - \tilde{\boldsymbol{w}}_n}{\delta t}\right)$ is of order $\mathcal{O}(1)$ since it satisfies the problem (8.57). However, the same term is of order $\mathcal{O}(1/\epsilon)$ for the non-penalized solution because the regularization $M_\eta(s)$ of $\max(0, s)$ is not exactly zero for negative values of s. To avoid this numerical artifact we found it more efficient to just cancel this term in (8.68).

8.6.6 Regularization and Extension of the Shape Derivative

During optimization the produced shapes may not have a smooth boundary so the shape derivative may have no rigorous meaning on the boundary Γ. In such a case, it is imperative to regularize the shape derivative (Allaire et al. 2021; Burger 2003; De Gournay 2006) in such a way that it is still a descent direction. One possibility is to consider the H^1 scalar product instead of the L^2 scalar product by finding a function $dj(\Omega) \in H^1(D)$ such that

$$\int_D \left(h_{\min}^2 \nabla \mathrm{d}j(\Omega) \cdot \nabla\varphi + \mathrm{d}j(\Omega)\varphi\right) \mathrm{d}x = \int_\Gamma j'(\Omega)\,\varphi \,\mathrm{d}x \qquad \forall \varphi \in H^1(D), \quad (8.70)$$

where $h_{\min}$ is the fixed minimal mesh size, and the function $j'(\Omega)$ is defined by formula (8.68) with

$$J'(\Omega)(\boldsymbol{\theta}) = \int_\Gamma \boldsymbol{\theta} \cdot \boldsymbol{n}\, j'(\Omega)\, \mathrm{d}s. \quad (8.71)$$

Since we have chosen $\mathbb{P}^1$ basis elements for the displacement vector and the plastic strain, the shape derivative in (8.68) is $\mathbb{P}^0$ smooth and so $j'(\Omega) \in \mathbb{P}^0(\Omega)$. Thus, it is enough to discretize (8.70) with $\mathbb{P}^1$ finite elements, so that $dj(\Omega) \in \mathbb{P}^1(D)$.

8.6.6.1 Shape Optimization Algorithm

We consider the shape optimization problem

$$\min_{\Omega \in \mathcal{U}_{\text{ad}}} J(\Omega),$$

where we remind the reader that $\mathcal{U}_{\text{ad}}$ is the space of admissible spaces inside the design space D (see Fig. 8.29). The optimization strategy is devised in the same manner in as in Sect. 8.4.5, resulting in the following algorithm. We thus perform the Algorithm 2.

Algorithm 2 Repeat over $i = 0, \ldots, I_{\max}$

1. Solve for $\tilde{\boldsymbol{w}}$ using the radial return algorithm on the mesh of Ω_i starting from t_1 until the last time t_N.
2. Solve for the adjoint $\tilde{z}$ using (8.67) on the mesh of Ω_i starting from the last time t_N until t_1.
3. Compute the regularized shape derivative $dj(\Omega_i)$ by solving (8.70) with the right hand side (8.68).
4. Apply a gradient ascent algorithm, with step $\tau = h_{min}/2$, to get
$$\lambda_{i+1} = \lambda_i + \frac{\tau}{C_2}\left(\int_{\Omega_i} \mathrm{d}x - V_f\right).$$
5. Set $\boldsymbol{n} = \nabla \phi_i$ (the level-set function for Ω_i) and solve (8.19) with the initial data ϕ_i and a velocity
$$\boldsymbol{\theta}_i = \left(\frac{dj(\Omega_i)}{C_1} + \frac{\lambda_{i+1}}{C_2}\right)\boldsymbol{n}$$
for a pseudo-time step τ to obtain $\tilde{\phi}_{i+1}$.
6. Re-initialize $\tilde{\phi}_{i+1}$ to the signed distance function, using `mshdist` (Dapogny and Frey 2012), to obtain ϕ_{i+1} corresponding to the new shape Ω_{i+1}.
7. Compute the volume V_{i+1}. If $|V_{i+1} - V_f| \leq 10^{-5} V_f$, then update the level-set ϕ_{i+1} by adding to it the constant $(V_{i+1} - V_f)/\int_{\partial\Omega_{i+1}} \mathrm{d}s$.
8. Remesh the box D using `MMG` (Dapogny et al. 2014) to obtain the body-fitted mesh of the new shape Ω_{i+1}

8.7 Numerical Results: Plasticity

This section displays 2D and 3D optimization results with three minimization criteria: total compliance (8.59), total energy (8.74) and plastic energy (8.75). In each case a volume constraint $|\Omega| = V_f$ is imposed and the optimization Algorithm 2 is applied. The structure is composed of mild steel with the properties: $E = 210\,GPa$, $\nu = 0.3$, $\sigma_Y = 279\,\text{MPa}$, $E_{\text{iso}} = 712\,\text{MPa}$. For all test cases in this subsection, we consider a force g that increases from zero to a final value in one second in a constant direction with a time step $\delta t = 0.05$. The time discretized adjoint equation (8.67) is solved using ϵ, η given in (8.69).

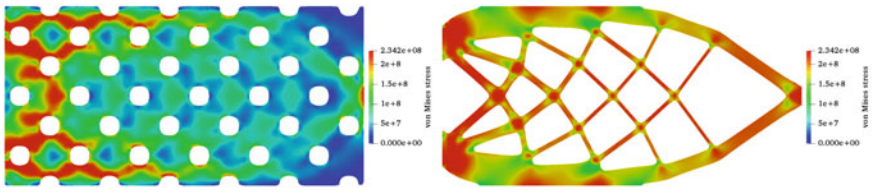
8.7.1 2D Cantilever

We study a $2\,\text{m} \times 1\,\text{m}$ 2D cantilever beam which is partially clamped on the left side (there is a small difference between the size of the Dirichlet boundary condition and the left edge of the beam), while a vertical concentrated force is applied at the middle of the right side of the beam (see Fig. 8.31). The reason to not completely clamp the left side of the cantilever beam is to allow the shape to move around Γ_D and to avoid potential plastic zone which often appears around the Dirichlet boundary condition. A target volume $V_f = 0.7\,\text{m}^2$ is imposed. Based on the quasi-static assumption, the rate of force increment has no impact on the solution at the final time instant $t = 1$. However, the rate does impact the objective function (8.59). If the force grows faster in the beginning and then slowly after the onset of plasticity, the objective function is influenced more by the plastic flow. To see a greater impact of the plastic flow on the shape derivative (and hence the shape), we choose

$$g = (0, 220\min(1.5t, 1))\text{MN/m},\ t \in [0, 1]s. \tag{8.72}$$

The parameters of the remeshing tool `MMG` are fixed to $h_{\min} = 0.01\,\text{m}$ (minimal mesh size), $h_{\max} = 0.02\,\text{m}$ (maximal mesh size). First, we minimize the total compliance (8.59). The initial shape and the final shapes for the linear elasticity and plasticity models are shown in Fig. 8.18. Let us first note that the presence, or not, of the hardening tensor $\mathbb{H}$ does not change much the resulting optimized shape in Figs. 8.18c, d. As can be seen on Fig. 8.18b, d, the optimized shapes for linear elasticity or plasticity are very similar. The only slight difference is near the Dirichlet boundary condition, where the bars are thicker for the plasticity case. It turns out that the displacement for linear elasticity is numerically very close to the one for plasticity. Although the plastic deformation $\boldsymbol{\varepsilon}_p$ does contribute to the shape derivative for the plasticity case, it does not induce a different topology, compared to the elasticity case. The convergence history for the total compliance is depicted in Fig. 8.19.

To quantitatively compare the two optimized shapes in Fig. 8.18b (elasticity) and Fig. 8.18d (plasticity), we perform a plasticity computation for both of them with



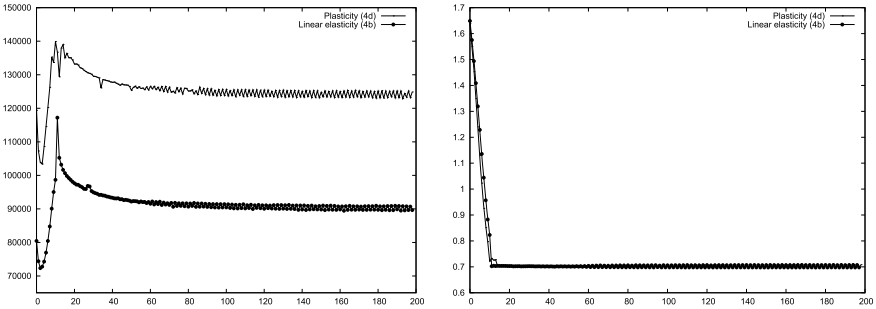
(a) Initialization with small holes placed periodically

(b) Optimized for (1.59), linear elasticity

(c) Optimized for (1.59), $E_{iso} = 712MPa$, $\mathbb{H} = 0$

(d) Optimized for (1.59), $E_{iso} = 712MPa$, $\mathbb{H} = 105\mathbb{I}MPa$

Fig. 8.18 Von Mises stress at $t = 1$ s corresponding to various shapes for a target volume $V_f = 0.7\,\mathrm{m}^2$ and force (8.72)



(a) Total compliance (1.59) v/s iterations

(b) Volume v/s iterations

Fig. 8.19 Convergence history corresponding to shapes (8.18b) and (8.18d)

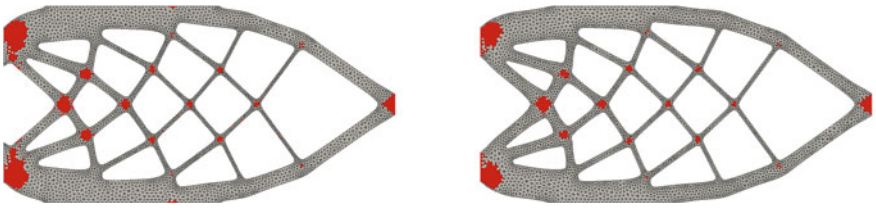


Fig. 8.20 Plastic zones ($\gamma > 0$) at $t = 1$s computed for shapes (8.18b) and (8.18d)

Table 8.1 2D Cantilever shape comparison for force (8.72)

	Shape (8.18b)	Shape (8.18d)
Total compliance (8.59) for linear elasticity	89,131	90,428
Total compliance (8.59) for plasticity ($E_{\text{iso}} = 712$ MPa, $\mathbb{H} = 105\mathbb{I}$MPa)	126,555	123,172

$E_{\text{iso}} = 712$ MPa, $\mathbb{H} = 105\mathbb{I}$MPa and the force (8.72). The plastic zones (where $\gamma > 0$) at time $t = 1$s along with the mesh are plotted in Fig. 8.20 and the total compliance (8.59) is noted in Table 8.1. In Fig. 8.20, we observe that the plastic zones are slightly smaller for (8.18b) compared to (8.18d). As seen in Table 8.1, the total compliance for the cantilever beam obtained for plasticity is 2.75% lesser than the one obtained for the linear elasticity case. While this improvement is pertinent, it is not very impressive. On the other hand, Table 8.1 confirms that Fig. 8.18b is (slightly) better than Fig. 8.18d for the linear elasticity.

Next, we investigate if a few parameters of the previous test case (external force, optimization criteria or initialization) result in a drastic change of the plastic zone. Specifically, we investigate three variations.

1. Increase the external force to

$$g = (0, 400 \min(1.5t, 1))\text{MN/m}, \ t \in [0, 1]s \tag{8.73}$$

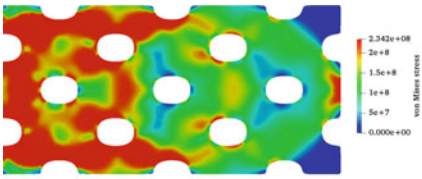
   such that the entire shape undergoes a plastic deformation.
2. Consider two new criteria for minimization: total energy

$$J(\Omega) = \int_0^T \int_\Omega \frac{1}{2} \left(\mathbb{C}\boldsymbol{\varepsilon}_e : \boldsymbol{\varepsilon}_e + \mathbb{H}\boldsymbol{\varepsilon}_p : \boldsymbol{\varepsilon}_p + E_{\text{iso}}\gamma^2 \right) \mathrm{d}x \tag{8.74}$$

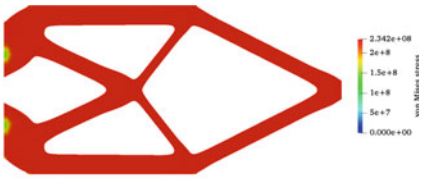
   and energy due to kinematic hardening

$$J(\Omega) = \int_0^T \int_\Omega \frac{1}{2} \mathbb{H}\boldsymbol{\varepsilon}_p : \boldsymbol{\varepsilon}_p \mathrm{d}x, \tag{8.75}$$

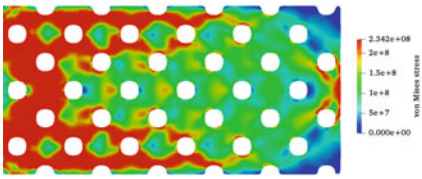
   in addition to the total compliance criterion (8.59).
3. Consider three different initializations (as shown in Fig. 8.21) for total compliance minimization.



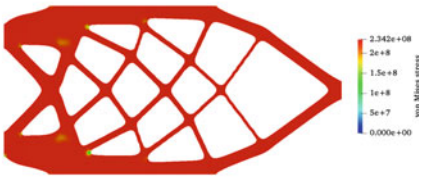
(a) Periodically distributed big holes



(b) Optimized shape, initialized from (1.21a)

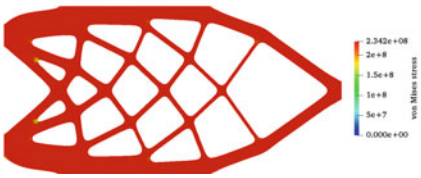


(c) Periodically distributed small holes



(d) Optimized shape, initialized from (1.21c)

(e) Shape obtained by minimizing compliance for linear elasticity



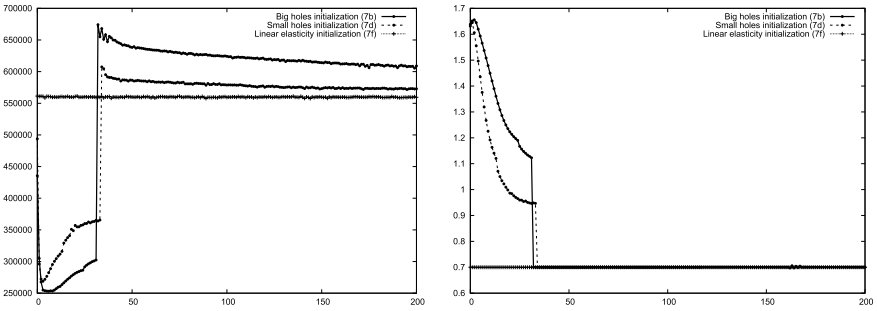
(f) Optimized shape, initialized from (1.21e)

Fig. 8.21 Von Mises stress at $t = 1$s for the initial shapes (on the left) and optimized shapes for total compliance (8.59) (on the right), with $V_t = 0.7\,\mathrm{m}^2$, $E_{\mathrm{iso}} = 712\,\mathrm{MPa}$, $\mathbb{H} = 105\mathbb{I}\mathrm{MPa}$ and force (8.73)

The shapes obtained for the three different initializations are plotted in Fig. 8.21, their corresponding compliances (8.59) are presented in Table 8.2, and their convergence histories are depicted in Fig. 8.22. As expected, we obtained three different topologies. In Fig. 8.21, we observe that plastic deformation occurs everywhere in the optimal shapes. This was not expected as yielding should have resulted in a high accumulated plastic deformation and hence a high total compliance. However, what actually happens is that, when the shapes reach the yield point, hardening occurs. Once the shape hardens, its load bearing capacity increases. Hence, the optimal shapes are the ones that struggle a balance between hardening and plastic deformation. Consequently, it is unrealistic to expect dramatic reduction in the size of plastic zones. As seen in Table 8.2, the cantilever beam is best optimized if initialized by the solution obtained for the linear elasticity case (Fig. 8.21c). In Fig. 8.22, we see almost no decrease in the objective function for the shape of Fig. 8.21f. This means that the shape obtained for the linear elasticity case is almost optimal for plasticity.

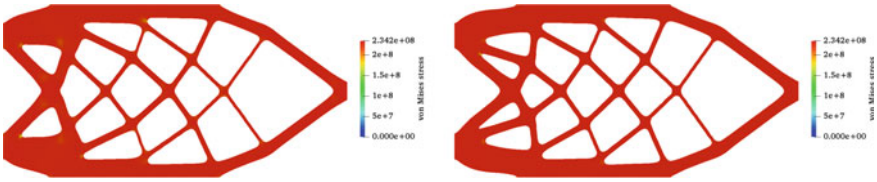
Table 8.2 2D Cantilever shape comparison for force (8.73)

	Shape (8.21b)	Shape (8.21d)	Shape (8.21f)
Total compliance (8.59) for linear elasticity	423,424	410,188	404,180
Total compliance (8.59) for plasticity ($E_{\text{iso}} = 712$ MPa, $\mathbb{H} = 105\mathbb{I}$MPa)	608,714	578,630	558,156



(a) Total compliance (1.59) v/s iterations (b) Volume v/s iterations

Fig. 8.22 Convergence history for the shapes (8.21b, d and f)



(a) Total energy (1.74) minimization (b) Plastic energy (1.75) minimization

Fig. 8.23 Von Mises stress at $t = 1$s for optimized shapes, initialized from (8.21c), with $V_t = 0.7\,\text{m}^2$, $E_{\text{iso}} = 712$ MPa, $\mathbb{H} = 105\mathbb{I}$MPa and force (8.73)

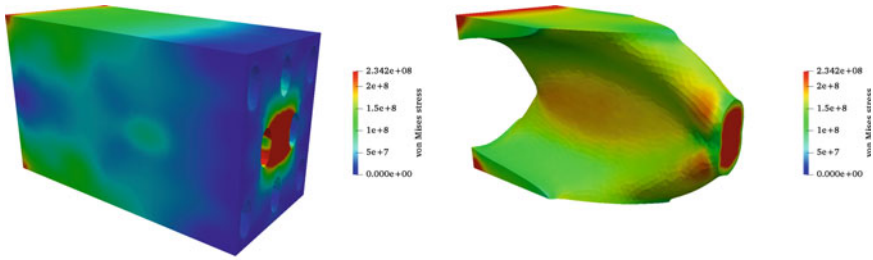
The shapes obtained for different objective functions, namely total energy (8.74) and plastic energy (8.75), are plotted in Fig. 8.23. The shapes Fig. 8.23a, b are similar to the previous shapes of Fig. 8.21d, f, respectively. In both cases, they were initialized with Fig. 8.21c. Again, the size of the plastic zone (where $\gamma > 0$) has not decreased. We believe it is because plastic zones are hardened zones and, as a result, are necessary for minimizing the total energy or the plastic energy.

8.7.2 3D Cantilever

We now consider the minimization of the total compliance (8.59) for a 3D cantilever beam of dimensions 5 m × 2.4 m × 3 m, as shown in Fig. 8.10. The cantilever beam is fixed on its leftmost side, loaded downwards on a circular region of radius $0.1m$ on its rightmost side with $g = (0, 5000t, 0)$MN/m where $t \in [0, 1]s$. For this test case, we consider combined hardening with $E_{\text{iso}} = 712$ MPa, $\mathbb{H} = 105\mathbb{I}$MPa and a target volume $V_f = 12\,\text{m}^3$. The parameters of `MMG` are set to $h_{\min} = 0.04$ m, and $h_{\max} = 0.12$ m. We initialize the shape optimization with a perforated shape as in Fig. 8.24a. Learning from the previous test cases, we also initialize with the shape obtained after minimizing compliance for linear elasticity (see Fig. 8.24c). The optimization from initialization in Fig. 8.24a is run for longer, 250 iterations instead of 200 iterations as in the other test cases. This is because an initialization with holes is far from the optimum and it take longer to converge to a form with plate-like features (which is known to be optimal for maximizing rigidity). As seen in Fig. 8.24, the two initializations result in the same shape (see Fig. 8.24b–d). Their corresponding convergence histories are plotted in Fig. 8.25. The shapes (Fig. 8.24b), (c) and (d) are compared quantitatively in Table 8.3. As seen in Fig. 8.25, it takes a long time for the shape (Fig. 8.24a) to converge, whereas the shape (Fig. 8.24c) converges in the first few iterations. Consequently, we conclude that it is often advantageous to first optimize the shape for linear elasticity and then use the optimized shape as initialization to minimize for plasticity.

8.7.3 3D Wedge

We now consider a 3D wedge of dimensions 1.2 m × 0.6 m × 0.6 m as shown in Fig. 8.26. The geometry is supported on four square surfaces each being 0.05 m × 0.05 m, three of which can be seen in the Fig. 8.26. The wedge is clamped along all the three axes on one surface and only along y-direction on the remaining three surfaces. The wedge is forced on a square surface on the topmost plane with $g = (0, -500t, 0)$MN/m where $t \in [0, 1]$s. The parameters of `MMG` are set to $h_{\min} = 0.012$ m, and $h_{\max} = 0.032$ m. We consider combined hardening with $E_{\text{iso}} = 712$ MPa, $\mathbb{H} = 105\mathbb{I}$MPa and impose a target volume of $V_f = 0.07\,\text{m}^3$. Optimized shapes for linear elasticity and plasticity are displayed in Fig. 8.27. Again, we consider two initializations: one with periodically distributed holes and one obtained by minimizing compliance for linear elasticity. It yields two topologically different optimized shapes as shown in Fig. 8.27d. As can be seen in Table 8.4, the shape (Fig. 8.27d) outperforms the shape (Fig. 8.27b) in terms of (8.59) in plasticity as well as in linear elasticity.



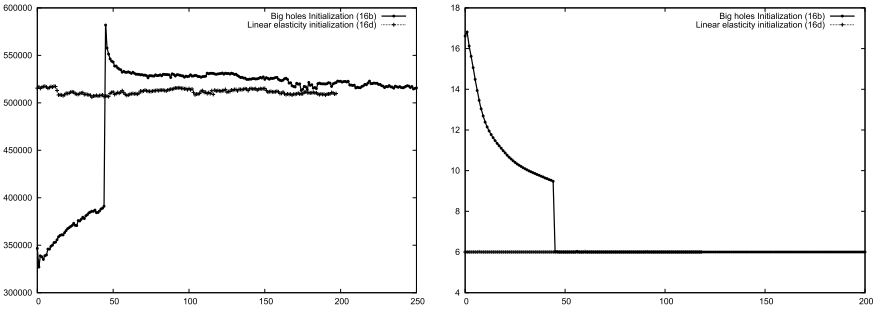
(a) Periodically distributed big holes

(b) Optimized shape, initialized from (1.24a)

(c) Shape obtained by minimizing compliance for linear elasticity

(d) Optimized shape, initialized from (1.24c)

Fig. 8.24 Von Mises stress at $t = 1$s for the initial shapes (on the left) and the optimized shapes for total compliance (8.59) (on the right), with $V_t = 12\,\mathrm{m}^3$, $E_{\mathrm{iso}} = 712\,\mathrm{MPa}$, $\mathbb{H} = 105\mathbb{I}\mathrm{MPa}$



(a) Total compliance (1.59) v/s iterations

(b) Volume v/s iterations

Fig. 8.25 Convergence history for shapes (8.24b) and (8.24d)

Table 8.3 3D Cantilever shape comparison

	Shape (8.24c)	Shape (8.24b)	Shape (8.24d)
Total compliance (8.59) in linear elasticity	453,774	456,363	451,848
Total compliance (8.59) in plasticity ($E_{\mathrm{iso}} = 712$ MPa, $\mathbb{H} = 105\mathbb{I}$MPa)	515,452	515,246	507,319

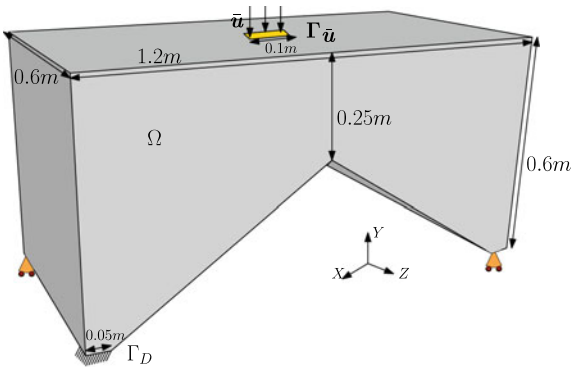
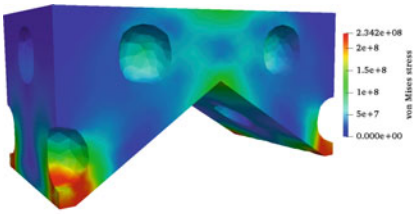
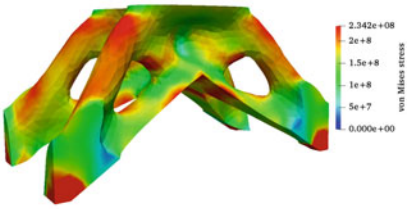


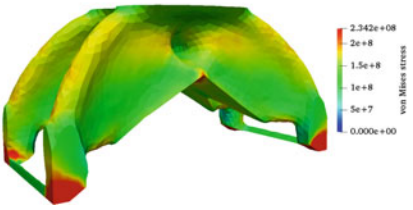
Fig. 8.26 3D wedge boundary conditions



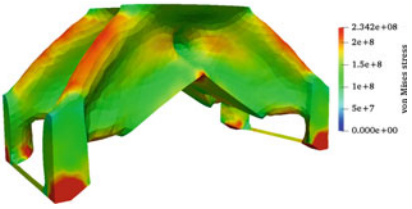
(a) Periodically distributed big holes



(b) Optimized shape, initialized from (1.27a)



(c) Shape obtained by minimizing compliance for linear elasticity



(d) Optimized shape, initialized from (1.27c)

Fig. 8.27 Von Mises stress at $t = 1$ s for the initial shapes (on the left) and the optimized shapes for total compliance (8.59) (on the right), with $V_t = 0.07\,\mathrm{m}^3$, $E_{\mathrm{iso}} = 712$ MPa, $\mathbb{H} = 105\mathbb{I}$MPa

Table 8.4 3D wedge shape comparison

	Shape (8.27b)	Shape (8.27d)
Total compliance (8.59) in linear elasticity	4843	4387
Total compliance (8.59) in plasticity ($E_{\mathrm{iso}} = 712$ MPa, $\mathbb{H} = 105\mathbb{I}$MPa)	5092	4547

8.8 TO in Fracture Mechanics

This section presents a theory for optimizing structures which are not prone to fracture. The content of this section can be found in a recent publication (Desai et al. 2022). Structures subjected to a critical external load may undergo high stress, material damage, crack and an ultimate collapse. Shape and topology optimization to minimize damage can facilitate engineers to conceive robust structures, less susceptible to crack and hence with a longer life.

To model the fracture phenomena, one needs to choose an appropriate fracture model. In the framework of brittle fracture mechanics, we consider the so-called Francfort-Marigo model (Francfort and Marigo 1998), in the spirit of the pioneering energetic approach of Griffith (1921). This model relies on the quasi-static minimization of a total energy which is the sum of a linear elasticity energy and a fracture energy, under an irreversibility constraint. The original Francfort-Marigo model belongs to the class of free discontinuity problems (the crack is an unknown of the problem), which are notoriously difficult to solve numerically. Therefore, the same authors propose to approximate their original model with a regularized damage model (Bourdin et al. 2008), following a Gamma-convergence process, first suggested by Ambrosio and Tortorelli for the Mumford–Shah functional (Ambrosio and Tortorelli 1990). This Ambrosio-Tortorelli regularization features a small regularizing parameter $\ell > 0$.

In the regularization of this free discontinuity problem, the parameter ℓ can be viewed as a purely mathematical artifact, used to approach the exact minimization problem. However, this regularization has also a very clear physical interpretation as damage model with an internal length scale which is precisely the regularization parameter ℓ. Recall that a damage model features a damage variable α, measuring the extent of degradation of the material and varying continuously between 0 (no damage) and 1 (fully damage), and an elastic Hooke's tensor which is a decreasing function of α. Additional ingredients are, like in the fracture case, an irreversibility constraint and an energy criterion which determines the onset of damage. How the original fracture or free discontinuity problem is recast as a damage problem is precisely explained in Bourdin et al. (2008) and Pham et al. (2011). This approach is also called sometimes a smeared interface approach or a phase-field approach. Its main advantage is its simplicity of numerical implementation using the finite element method. Furthermore, it can easily detect initiation, branching and coalescence of cracks without the need of meshing the crack path. In the sequel we shall indifferently call this model a damage or a fracture model.

Combining the Francfort-Marigo fracture model with the level-set method, we propose a shape and topology optimization algorithm for preventing crack initiation and propagation in solid structures.

Topology optimization to minimize fracture has already been the topic of some works. In the level-set framework, the configuration of composite materials in a phase-field-based fracture model was optimized (Wu et al. 2020). There are more works on topology optimization using SIMP applied to various fracture models.

Topology optimization using SIMP was performed for a fracture model in Kang et al. (2017) or for a damage model to reinforce concrete in Amir (2013). Path dependency of the damage model was taken into account in SIMP (James and Waisman 2015a). Phase-field model of fracture was considered for topology optimization using extended BESO (Bi-directional Evolutionary Structural Optimization) (Da and Yvonnet 2020). Fracture governed by the phase-field model is considered for maximizing the fracture resistance of periodic composite (Da and Yvonnet 2020).

A common feature in all the previous works using SIMP is that several material properties must be approximated for the mixture of material and void, corresponding to the density variable. This is classical for the Young's modulus (Bendsoe and Sigmund 2013) but more delicate for other properties like, for example, the fracture toughness. Usually, these material properties are approximated by multiplying their values by the density raised to a certain exponent. This exponent is different for every property and ought to be chosen in an ad hoc manner, ensuring numerical stability. If the optimized shape has intermediate densities, the interpretation of damage is quite artificial and may cause numerical difficulties. On the contrary, in the level-set framework, since the material properties are never approximated, such artificial damaged zones are avoided.

Section 8.8.1 is devoted to the presentation of the Francfort-Marigo damage model. Although this model has nice properties, it features an irreversibility constraint (a damage region cannot heal and be again undamaged) which makes it a variational inequality, instead of a more standard variational equality. Unfortunately, the adjoint method for computing sensitivities or derivatives of an objective function is extremely involved and not practical for variational inequalities since it involves the notion of conical derivative (Mignot and Puel 1984; Sokolowski and Zolésio 1992). Therefore, we penalize the irreversibility constraint to transform the variational inequality into a more convenient variational equality which is amenable to the adjoint method.

Section 8.8.2 states the optimization problem and delivers its shape derivative (see Proposition 1), relying on the adjoint method and the well-known Hadamard and Céa's methods. Section 8.6.5 presents the discretization of the damage model and of the adjoint problem in space and time and the level-set algorithm. A key ingredient for the sequel is that, the damage model being non-convex, it is solved with a so-called backtracking algorithm, due to Bourdin et al. (2008), which is able to escape from local minima in the total energy minimization. Our finite element analyses are performed within the `FreeFEM` software (Hecht 2012). For the finer 3D meshes we use the parallel computational capacities of `FreeFEM` which rely on domain decomposition and the `PETSc` package (Balay et al. 2019). The level-set transport equation is solved with the `advect` library (Bui et al. 2012). Having a body-fitted mesh of the structure is crucial for an accurate evaluation of the damage. Indeed, there is no ersatz or weak material, so damage does not interact with this fictitious phase. Furthermore, since damage typically occurs in region of high stresses, those are more precisely computed with a body-fitted mesh. We also outline the details of our gradient-based algorithm for shape and topology optimization.

Finally, Sect. 8.7 is concerned with 2D and 3D numerical test cases. The objective function is the so-called total compliance (integrated in time), which is minimized under a volume constraint. These examples illustrate the efficiency of our proposed shape and topology optimization algorithm to obtain crack-free optimal structures. One specific difficulty in optimization of damage of fracture models is that these phenomena may be discontinuous: a small increase in the loadings, a small change in the structure's geometry may cause the sudden occurrence of a not-so-small crack and thus a large increase of the objective function (see the discussion in Remark 8). Nevertheless, our algorithm is able to sustain these large oscillations and, in the end, converge smoothly to optimal undamaged structures.

8.8.1 Governing Laws and Variational Formulation

Let $\Omega \subset \mathbb{R}^d$ be a smooth bounded open set representing the structure in Fig. 8.28 (with $d = 2, 3$ the space dimension). For $T > 0$, $[0, T]$ is the bounded time interval of interest. The imposed displacement is denoted by $\bar{\boldsymbol{u}} : \Omega \times [0, T] \mapsto \mathbb{R}^d$ and is assumed to be smooth, typically $\bar{\boldsymbol{u}} \in \mathcal{C}^0([0, T], H^2(\mathbb{R}^d)^d)$. Let $\partial\Omega = \Gamma_D \cup \Gamma \cup \Gamma_{\bar{\boldsymbol{u}}}$, be the union of disjoint boundaries and $\boldsymbol{n}$ denote the outward normal to $\partial\Omega$: Γ is the free boundary (no traction), Γ_D is the Dirichlet boundary and $\Gamma_{\bar{\boldsymbol{u}}}$ is the boundary where the displacement $\bar{\boldsymbol{u}}$ is imposed.

The elastic displacement vector field is $\boldsymbol{u} : \Omega \times [0, T] \mapsto \mathbb{R}^d$ and $\boldsymbol{e}(\boldsymbol{u}) = \frac{1}{2}\big(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T\big)$ denotes the second order strain tensor. The fourth-order Hooke's tensor $\mathbb{C}_0$ is assumed to be coercive on the set $\mathcal{M}_s^d$ of symmetric $d \times d$ matrices, i.e., $\exists\, c_{\min} > 0,\ c_{\max} > 0$ such that, $\forall \boldsymbol{\xi} \in \mathcal{M}_s^d$,

$$c_{\min}|\boldsymbol{\xi}|^2 \leq \mathbb{C}_0\boldsymbol{\xi} : \boldsymbol{\xi} \leq c_{\max}|\boldsymbol{\xi}|^2.$$

The rate of evolution of the imposed displacement $\bar{\boldsymbol{u}}$ is assumed to be small, resulting in a quasi-static evolution of the structure and hence a negligible acceleration $\ddot{\boldsymbol{u}} = \boldsymbol{0}$. As the loading increases with time t and the elastic energy in the structure exceeds a critical elastic energy density, the structure undergoes damage, which is measured with the damage variable $\alpha : \Omega \times [0, T] \mapsto [0, 1]$. The value $\alpha = 0$ corresponds to

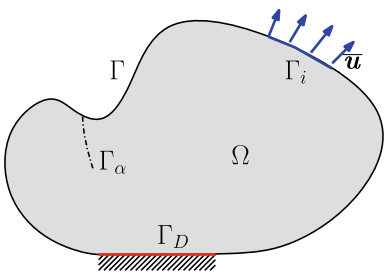


Fig. 8.28 Boundary conditions and the unknown crack Γ_α

no damage and $\alpha = 1$ to a complete damage. Damage is characterized by deterioration of the stiffness and is modeled by assuming the Hooke's tensor $\mathbb{C}(\alpha)$ to be a convex function of the damage variable α, such that

$$\mathbb{C}(0) = \mathbb{C}_0, \quad \mathbb{C}(1) = \mathbf{0} \text{ and } \mathbb{C}'(\alpha)\boldsymbol{\xi} : \boldsymbol{\xi} < 0 \; \forall \boldsymbol{\xi} \in \mathcal{M}_s^d. \tag{8.76}$$

The second expression in the above states that when the damage variable α attains unity, the stiffness becomes zero and structure undergoes a crack. The third expression in the above states that, when α increases, the stiffness must decrease.

The study of the damage model requires the introduction of some functional spaces. The space of admissible displacements is given by

$$V = \{\boldsymbol{u} \in H^1(\Omega)^d : \boldsymbol{u} = \mathbf{0} \text{ on } \Gamma_D\}. \tag{8.77}$$

For $\mathscr{v} = \dot{\bar{\boldsymbol{u}}}$, the affine space of admissible velocities is defined as

$$C_{\mathscr{v}} = \{\boldsymbol{v} \in V : \boldsymbol{v} = \mathscr{v} \text{ on } \Gamma_{\bar{\boldsymbol{u}}}\}. \tag{8.78}$$

The subspace of functions in $H^1(\Omega)$ which vanish on Γ_D and $\Gamma_{\bar{\boldsymbol{u}}}$, is denoted by

$$H_D^1(\Omega) = \{\beta \in H^1(\Omega) : \beta = 0 \text{ on } \Gamma_D \cup \Gamma_{\bar{\boldsymbol{u}}}\}. \tag{8.79}$$

The convex set of admissible damage is

$$\mathcal{D}_1 = \{\alpha \in H_D^1(\Omega) : 0 \leq \alpha(x) \leq 1 \; a.e.\, x \in \Omega\} \tag{8.80}$$

and the convex set of admissible damage evolution rate ($\gamma = \dot{\alpha}$) is

$$\mathcal{D} = \{\gamma \in H_D^1(\Omega) : \gamma(x) \geq 0 \; a.e.\, x \in \bar{\Omega}\}, \tag{8.81}$$

To simplify notations, we define the product space

$$Z = V \times H_D^1(\Omega).$$

The initial condition of the model is

$$(\boldsymbol{u}(0), \alpha(0)) = (\mathbf{0}, \alpha_0) \in V \times \mathcal{D}_1. \tag{8.82}$$

Remark 3 Typically, the set $\mathcal{D}_1$ of admissible damage variable and the set $\mathcal{D}$ of admissible damage evolution rate are defined for functions in $H^1(\Omega)$. In (8.80) and (8.81) we rather choose $H_D^1(\Omega)$ because we wish to forbid crack formation on Γ_D and $\Gamma_{\bar{\boldsymbol{u}}}$. Without this, the creation of cracks on these boundaries would require only half the energy needed for cracks occurring inside the structure Ω and thus would be artificial.

Following Ambati et al. (2015) and Pham et al. (2011) we introduce the elastic energy and the damage energy, respectively, as

$$\mathcal{E}(\boldsymbol{u},\alpha) = \frac{1}{2}\int_{\Omega} \mathbb{C}(\alpha)\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{u})\,\mathrm{d}x, \tag{8.83}$$

$$\mathcal{H}(\alpha) = \frac{G_c}{4c_w}\left(\int_{\Omega} \ell|\nabla\alpha|^2\mathrm{d}x + \int_{\Omega}\frac{w(\alpha)}{\ell}\mathrm{d}x\right), \quad \text{where} \quad c_w = \int_0^1 \sqrt{w(\xi)}\mathrm{d}\xi, \tag{8.84}$$

where G_c is the fracture toughness of the material, $\ell > 0$ is the characteristic length, measuring the thickness of the damaged zone around the fracture, and $w(\alpha)$ is the fracture energy density (Comi and Perego 2001), assumed to satisfy

$$w(0) = 0,\ w(1) = 1, \text{ and } w'(\alpha) \geq 0 \text{ for } 0 \leq \alpha \leq 1.$$

The sum of the integrands in the above two energies defines the energy density W_ℓ

$$W_\ell(\boldsymbol{u},\alpha) = \frac{1}{2}\mathbb{C}(\alpha)\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{u}) + \frac{G_c}{4c_w}\left(\ell|\nabla\alpha|^2 + \frac{w(\alpha)}{\ell}\right). \tag{8.85}$$

The total energy of the structure is then given by

$$\mathcal{P}(\boldsymbol{u},\alpha) = \int_{\Omega} W_\ell(\boldsymbol{u},\alpha)\,\mathrm{d}x. \tag{8.86}$$

The Francfort-Marigo regularized fracture model amounts to minimize the total energy $\mathcal{P}(\boldsymbol{u},\alpha)$ among all fields $(\boldsymbol{u}(t),\alpha(t)) \in V \times \mathcal{D}_1$, for $t \in (0,T]$, such that $(\dot{\boldsymbol{u}}(t),\dot{\alpha}(t)) \in C_{\nu} \times \mathcal{D}$ and with the initial condition (8.82). In particular, the definition of the space $\mathcal{D}$ contains the irreversibility condition $\dot{\alpha} \geq 0$ and that of the space C_{ν} implies that the boundary condition $\boldsymbol{u}(t) = \bar{\boldsymbol{u}}(t)$ on $\Gamma_{\bar{\boldsymbol{u}}}$ is satisfied at all times.

No body or surface forces are applied. Indeed, it is well-known (Francfort and Marigo 1998) that, in the case of complete damage ($\mathbb{C}(1) = \mathbf{0}$), applying a force leads to the breakdown of the structure, which is not physical.

The solution $(\boldsymbol{u}(t),\alpha(t))$ satisfies the energy balance

$$\int_{\Omega} \dot{W}_\ell(\boldsymbol{u},\alpha)\,\mathrm{d}x = \int_{\Gamma_{\bar{\boldsymbol{u}}}} (\mathbb{C}(\alpha)\boldsymbol{\varepsilon}(\boldsymbol{u}) \cdot \boldsymbol{n}) \cdot \dot{\bar{\boldsymbol{u}}}\,\mathrm{d}s, \tag{8.87}$$

where overdot represents derivative with respect to time and ds denotes the surface measure over $\Gamma_{\bar{\boldsymbol{u}}}$. The optimality conditions for the minimization of (8.86), written in a compact form, are Marigo (1989) and Nguyen (1987): for all $t \in (0, T]$, find $(\boldsymbol{u}(t), \alpha(t)) \in V \times \mathcal{D}_1$, satisfying the initial condition (8.82), such that $(\dot{\boldsymbol{u}}(t), \dot{\alpha}(t)) \in C_{\nu} \times \mathcal{D}$, $\boldsymbol{u}(t) = \bar{\boldsymbol{u}}(t)$ on $\Gamma_{\bar{\boldsymbol{u}}}$ and

$$d\mathcal{P}(\boldsymbol{u}, \alpha)(\boldsymbol{v} - \dot{\boldsymbol{u}}, \beta - \dot{\alpha}) \geq 0 \qquad \forall (\boldsymbol{v}, \beta) \in C_{\nu} \times \mathcal{D}. \tag{8.88}$$

From (8.88), one can derive the strong form of the quasi-static damage model: find $(\boldsymbol{u}(t), \alpha(t))$, satisfying the initial condition (8.82), such that

$$\dot{\alpha} \geq 0 \quad \text{in } \Omega \times (0, T], \tag{8.89a}$$

$$\frac{1}{2}\mathbb{C}'(\alpha)\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{u}) - \frac{G_c}{2c_w}\ell\Delta\alpha + \frac{G_c}{4c_w}\frac{w'(\alpha)}{\ell} \geq 0 \quad \text{in } \Omega \times (0, T], \tag{8.89b}$$

$$\dot{\alpha}\left(\frac{1}{2}\mathbb{C}'(\alpha)\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{u}) - \frac{G_c}{2c_w}\ell\Delta\alpha + \frac{G_c}{4c_w}\frac{w'(\alpha)}{\ell}\right) = 0 \quad \text{in } \Omega \times (0, T], \tag{8.89c}$$

$$\nabla\alpha \cdot \boldsymbol{n} \geq 0, \quad \dot{\alpha}\nabla\alpha \cdot \boldsymbol{n} = 0 \quad \text{on } \partial\Omega \times (0, T], \tag{8.89d}$$

$$\alpha = 0 \quad \text{on } (\Gamma_D \cup \Gamma_{\bar{\boldsymbol{u}}}) \times (0, T], \tag{8.89e}$$

$$\operatorname{div}(\mathbb{C}(\alpha)\boldsymbol{\varepsilon}(\boldsymbol{u})) = \boldsymbol{0} \quad \text{in } \Omega \times (0, T], \tag{8.89f}$$

$$\boldsymbol{u} = \bar{\boldsymbol{u}} \quad \text{on } \Gamma_{\bar{\boldsymbol{u}}} \times (0, T], \tag{8.89g}$$

$$\boldsymbol{u} = \boldsymbol{0} \quad \text{on } \Gamma_D \times (0, T]. \tag{8.89h}$$

Inequality (8.89b) is known as the damage criterion. Equation (8.89c) is the complementary relation which essentially states that the damage criterion is an equality only if the damage evolution rate is positive. The variational formulation (or weak form) of the system of equations (8.89) reads: for all $t \in (0, T]$ find $(\boldsymbol{u}(t), \alpha(t)) \in V \times \mathcal{D}_1$ such that $\boldsymbol{u}(t) = \bar{\boldsymbol{u}}(t)$ on $\Gamma_{\bar{\boldsymbol{u}}}$, $\dot{\alpha} \in \mathcal{D}$

$$\int_{\Omega} \mathbb{C}(\alpha)\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v})\,\mathrm{d}x = 0 \quad \forall \boldsymbol{v} \in V, \tag{8.90a}$$

$$\int_{\Omega} \frac{G_c}{2c_w}\left(\ell\nabla\alpha \cdot \nabla\beta + \frac{w'(\alpha)}{2\ell}\beta\right)\mathrm{d}x + \int_{\Omega} \frac{1}{2}\mathbb{C}'(\alpha)\beta\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{u})\,\mathrm{d}x \geq 0 \qquad \forall \beta \in \mathcal{D}. \tag{8.90b}$$

When the characteristic length ℓ is small enough, and since $\mathbb{C}(1) = \mathbf{0}$, the above gradient damage model (8.54) is known to approximate brittle fracture. In Fig. 8.28, Γ_α represents a crack which is a priori unknown. The model (8.54) is able to capture a crack Γ_α which is typically a line in 2D and a surface in 3D, where α takes a unit value, which is surrounded by a band of thickness 2ℓ where α is strictly positive. Thus, the length ℓ characterizes the thickness of the region that shall undergo damage.

Two choices of dissipation function $w(\alpha)$ are usually considered and the second one is chosen for the numerical test cases:

- `DQ` model [proposed in a different context by Ambrosio and Tortorelli (1992)]

$$w(\alpha) = \alpha^2, \qquad c_w = \frac{1}{2}. \tag{8.91}$$

  In this case, the damage onsets as soon as the external loading increments from zero.
- `DL` model [proposed in Pham et al. (2011)]

$$w(\alpha) = \alpha, \qquad c_w = \frac{2}{3}. \tag{8.92}$$

  In this case, the damage onsets only when a critical elastic energy density is exceeded.

The two choices of $w(\alpha)$, (8.92) and (8.91), make $w(\alpha)$ convex. Therefore, the functional $\alpha \mapsto \mathcal{W}_\ell(\boldsymbol{u}, \alpha)$ is convex. Similarly, for a fixed α, $\boldsymbol{u} \mapsto \mathcal{W}_\ell(\boldsymbol{u}, \alpha)$ is convex. This implies that fixing one variable at a time, one can alternately solve the minimization problem (8.54) (Bourdin 2007). However, $\mathcal{W}_\ell(\boldsymbol{u}, \alpha)$ is not convex with respect to the two variables $(\boldsymbol{u}, \alpha)$ together.

8.8.1.1 Penalization

The damage criterion (8.90b) is a variational inequality and is not shape-differentiable in the classical sense. For a class of inequalities, called of the first kind and of the second kind, one can determine the so-called conical derivative (Mignot and Puel 1984; Sokolowski and Zolésio 1992). Well-known examples of inequality of the first kind include the obstacle-problem and the frictionless contact mechanics problem. The damage inequality (8.90b) without the irreversibility constraint (only the box constraint $\alpha \in [0, 1]$) and for a convex $\mathbb{C}(\alpha)$ classifies as an inequality of the second kind, hence easy to analyze. But with the irreversibility constraint, inequality (8.90b) classifies neither as the first kind nor as the second kind. The analytical treatment of (8.90b) is thus complex and out of the scope of this chapter. Instead, we prefer to convert the inequality (8.90b) into an equation, using penalization. Let ϵ be a penalization factor such that $0 < \epsilon \ll 1$ and let $\mathcal{M}$ be the max function, defined as

$$\mathcal{M}(\beta) = \max(\beta, 0).$$

From now on, we replace the original model (8.54) by the following penalized problem: for all $t \in (0, T]$, find $(\boldsymbol{u}_\epsilon(t), \alpha_\epsilon(t)) \in Z$ such that $\alpha_\epsilon(0) = \alpha_0$, $\boldsymbol{u}_\epsilon(t) = \bar{\boldsymbol{u}}(t)$ on $\Gamma_{\bar{\boldsymbol{u}}}$, and

$$\int_\Omega \frac{G_c}{2c_w}\left(\ell \nabla\alpha_\epsilon \cdot \nabla\beta + \frac{w'(\alpha_\epsilon)\beta}{2\ell}\right) \mathrm{d}x + \int_\Omega \frac{1}{2}\mathbb{C}'(\alpha_\epsilon)\beta\boldsymbol{\varepsilon}(\boldsymbol{u}_\epsilon) : \boldsymbol{\varepsilon}(\boldsymbol{u}_\epsilon)\, \mathrm{d}x$$
$$+ \int_\Omega \frac{G_c}{\epsilon}\left(\mathcal{M}(\alpha_\epsilon - 1) - \mathcal{M}(-\dot{\alpha}_\epsilon)\right)\beta\, \mathrm{d}x = 0 \quad \forall \beta \in H^1_D(\Omega), \tag{8.93a}$$

$$\int_\Omega \mathbb{C}(\alpha_\epsilon)\boldsymbol{\varepsilon}(\boldsymbol{u}_\epsilon) : \boldsymbol{\varepsilon}(\boldsymbol{v})\, \mathrm{d}x = 0 \qquad \forall \boldsymbol{v} \in V. \tag{8.93b}$$

We call (8.93), the state problem, and the solution to it $(\boldsymbol{u}_\epsilon, \alpha_\epsilon)$, the state solution. Note that we implicitly assume in writing (8.93a) that α_ϵ admits a time derivative. In the sequel it is assumed that there exists a unique state solution. As is clear in (8.93a), only the bound constraints $\dot{\alpha}_\epsilon \geq 0$ and $\alpha_\epsilon \leq 1$ are penalized. The constraint $\alpha_\epsilon \geq 0$ is not penalized explicitly, rather implicitly by penalizing $\dot{\alpha}_\epsilon \geq 0$ and defining an initial condition $\alpha_0 \geq 0$. A similar penalization approach was studied numerically (Miehe et al. 2010), where authors penalize only the irreversibility criterion. Our work defers as we penalize the upper bound $\alpha_\epsilon \leq 1$ as well.

In the sequel, to simplify notations, we shall drop all ϵ indices and simply denote by $(\boldsymbol{u}, \alpha)$ the solution $(\boldsymbol{u}_\epsilon, \alpha_\epsilon)$ of problem (8.93).

8.8.2 *Optimization Problem*

We minimize an objective functional $J(\Omega)$ given by

$$J(\Omega) = \int_0^T \int_\Omega m(\boldsymbol{u}(\Omega), \alpha(\Omega))\, \mathrm{d}x\, \mathrm{d}t, \tag{8.94}$$

where $(\boldsymbol{u}(\Omega), \alpha(\Omega))$ is the solution of (8.93) and the function $m(\cdot, \cdot)$ is assumed to be $\mathcal{C}^1$ smooth with quadratic growth and linear growth for its derivative, so as to ensure that the objective function (8.94) is well-defined and the adjoint equation is well-posed. This objective functional represents a mechanical property such as total power, total elastic energy or total fracture energy. In Sect. 8.7, we shall maximize the total elastic energy and thus choose

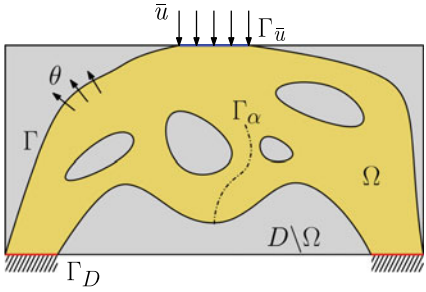


Fig. 8.29 Design domain D and shape Ω

$$m(\boldsymbol{u}, \alpha) = -\mathbb{C}(\alpha)\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{u}).$$

The justification for the above choice shall be given at the beginning of Sect. 8.7.

In practice, the shape Ω must be found inside a pre-fixed design space $D \subset \mathbb{R}^d$. Figure 8.29 shows the shape Ω (in gold) and the design space D (in gold and gray). The crack Γ_α is an unknown of the problem (8.93) that shall be determined for every shape Ω. This crack Γ_α might appear anywhere in the shape Ω and there is no postulated initial crack. The space of admissible shapes $\mathcal{U}_{\text{ad}}$ is given by Def.8.4. The minimization problem then reads

$$\min_{\Omega \in \mathcal{U}_{\text{ad}}} J(\Omega). \tag{8.95}$$

As is well known, very often there exists an optimal shape only if additional uniform smoothness conditions are imposed to the admissible shapes, that we shall not consider in the sequel. As usual, we content ourselves with computing numerical minimizers, using a gradient-descent method.

8.8.3 Shape Derivative Computation

To define the adjoint problem for the shape derivative, we introduce the subspace V_0 of V

$$V_0 = \{\boldsymbol{u} \in H^1(\Omega)^d : \boldsymbol{u} = \boldsymbol{0} \text{ on } \Gamma_D \cup \Gamma_{\bar{\boldsymbol{u}}}\}. \tag{8.96}$$

The adjoint variational formulation is defined as: find $(\boldsymbol{v}, \beta) \in H^1([0, T], Z)$, satisfying the final condition $\beta(T) = 0$, such that, for all $t \in [0, T)$,

$$\int_\Omega \left(\partial_{\boldsymbol{u}} m(\boldsymbol{u}, \alpha)\boldsymbol{\psi} + \mathbb{C}(\alpha)\boldsymbol{\varepsilon}(\boldsymbol{v}) : \boldsymbol{\varepsilon}(\boldsymbol{\psi}) + \mathbb{C}'(\alpha)\beta\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{\psi})\right) \mathrm{d}x = 0 \qquad \forall \boldsymbol{\psi} \in V_0, \tag{8.97a}$$

$$\int_\Omega \Big(\partial_\alpha m(\boldsymbol{u},\alpha)\varphi + \mathbb{C}'(\alpha)\varphi\, \boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v}) + \frac{G_c \ell}{2c_w} \nabla\beta \cdot \nabla\varphi + \left(\frac{1}{2}\mathbb{C}''(\alpha)\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{u}) + \frac{G_c}{4c_w}\frac{w''(\alpha)}{\ell} + \frac{G_c}{\epsilon}\mathcal{M}'(\alpha-1) \right) \beta\varphi \Big)\, \mathrm{d}x - \int_\Omega \frac{G_c}{\epsilon} \left(\frac{\mathrm{d}}{\mathrm{d}t}(\mathcal{M}'(-\dot{\alpha})\beta) \right) \varphi \, \mathrm{d}x = 0 \qquad \forall \varphi \in H^1_D(\Omega). \tag{8.97b}$$

In the sequel, we assume that there exists a unique solution of the adjoint equation. Note that (8.97b) features the time derivative of the derivative $\mathcal{M}'$ of the maximum function $\mathcal{M}(\beta) = \max(\beta, 0)$. Since $\mathcal{M}'$ is the Heaviside function, its time derivative is a Dirac mass, and the precise meaning of the last integral in (8.97b) is unclear. However, if $\mathcal{M}(\beta)$ was a regularization of $\max(\beta, 0)$, then everything makes sense, including the following proposition. In other words, our computation of the shape derivative below is fine for a smooth function $\mathcal{M}(\beta)$ and merely formal for the maximum function. Further time discretization will make clear in which sense the time derivative of the Heaviside function is computed (see Sect. 8.8.4).

Proposition 1 *Let Ω be a smooth bounded open set. Assume that there exists a unique solution $(\boldsymbol{u}, \alpha)$ to (8.93), which belongs to $H^1([0,T], Z)$, and that there exists a unique solution $(\boldsymbol{v}, \beta) \in H^1([0,T], Z)$ of the adjoint equation (8.61). If the state solution $(\boldsymbol{u}, \alpha) \equiv (\boldsymbol{u}(\Omega), \alpha(\Omega))$ is shape-differentiable, then the objective function (8.94) admits a shape derivative, given, for any $\boldsymbol{\theta} \in W_0^{1,\infty}(D, \mathbb{R}^d)$, by*

$$J'(\Omega)(\boldsymbol{\theta}) = \int_0^T \int_\Gamma \boldsymbol{\theta}\cdot\boldsymbol{n}\Big(m(\boldsymbol{u},\alpha) + \mathbb{C}(\alpha)\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v}) + \frac{1}{2}\mathbb{C}'(\alpha)\beta\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{u}) + \frac{G_c}{2c_w}\big(\ell\nabla\alpha\cdot\nabla\beta + \frac{w'(\alpha)\beta}{2\ell}\big) + \frac{G_c}{\epsilon}\Big(\mathcal{M}(\alpha-1) - \mathcal{M}(-\dot{\alpha})\Big)\beta\Big) ds\, dt. \tag{8.98}$$

Remark 4 The uniqueness of the solution of the damage model (8.93) is far from being obvious since this model is the optimality condition for the minimization of a non-convex energy, which thus may have multiple minima. For the same reason, the existence of a solution for the adjoint equation (8.61) is not obvious either because the corresponding operator is not coercive (nevertheless, see Sect. 8.8.4 for a positive result in this direction). Note that we also assume that the solutions are smooth with respect to time since they belong to $H^1([0,T], Z)$ and, in particular, are continuous with respect to time. We use this assumption in the variational formulations (8.93a) and (8.97b), which involve the time derivative of the damage variable α. Unfortunately, as discussed further in Remark 8, it is likely that, in some cases, the solution $(\boldsymbol{u}(\Omega), \alpha(\Omega))$ is discontinuous in time and thus the shape derivative (8.98) is not rigorously justified.

Proof The idea of the proof is well-known, based on Céa's method (J. Céa 1986). Introduce a few spaces on the full space $\mathbb{R}^d$ and thus independent of Ω:

$$\tilde{V} = \{\boldsymbol{v} \in H^1(\mathbb{R}^d)^d : \boldsymbol{v} = \boldsymbol{0} \text{ on } \Gamma_D\}, \quad \tilde{\mathcal{C}}_t = \{\boldsymbol{v} \in \tilde{V} : \boldsymbol{v} = \bar{\boldsymbol{u}}(t) \text{ on } \Gamma_{\bar{\boldsymbol{u}}}\},$$
$$\tilde{H}^1_D(\mathbb{R}^d) = \{\beta \in H^1(\mathbb{R}^d) : \beta = 0 \text{ on } \Gamma_D \cup \Gamma_{\bar{\boldsymbol{u}}}\}, \quad \tilde{Z}_t = \tilde{\mathcal{C}}_t \times \tilde{H}^1_D(\mathbb{R}^d),$$
$$\tilde{V}_0 = \{\boldsymbol{v} \in \tilde{V} : \boldsymbol{v} = \boldsymbol{0} \text{ on } \Gamma_{\bar{\boldsymbol{u}}}\}, \quad \tilde{Z}_0 = \tilde{V}_0 \times \tilde{H}^1_D(\mathbb{R}^d).$$

For independent variables $\tilde{\boldsymbol{u}}(t)$, $\tilde{\boldsymbol{v}}(t)$, $\tilde{\alpha}(t)$, $\tilde{\beta}(t)$ and $\tilde{\lambda}$, belonging to the spaces

- $(\tilde{\boldsymbol{u}}, \tilde{\alpha}) \in H^1([0,T], \tilde{Z}_t)$,
- $(\tilde{\boldsymbol{v}}, \tilde{\beta}) \in H^1([0,T], \tilde{Z}_0)$ (the Lagrange multiplier for the state equation (8.93)),
- $\tilde{\lambda} \in L^2(\mathbb{R}^d)$ (the Lagrange multiplier for the initial condition $\tilde{\alpha}(0) = \alpha_0$),

define a Lagrangian as

$$\begin{aligned}\mathcal{L}(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}, \tilde{\alpha}, \tilde{\beta}, \tilde{\lambda}, \Omega) = &\int_0^T \int_\Omega m(\tilde{\boldsymbol{u}}, \tilde{\alpha}) \mathrm{d}x \mathrm{d}t \\ &+ \int_0^T \Big(\int_\Omega \mathbb{C}(\tilde{\alpha}) \boldsymbol{\varepsilon}(\tilde{\boldsymbol{u}}) : \boldsymbol{\varepsilon}(\tilde{\boldsymbol{v}}) \, \mathrm{d}x \Big) \mathrm{d}t + \int_0^T \int_\Omega \Big(\frac{1}{2} \mathbb{C}'(\tilde{\alpha}) \tilde{\beta} \boldsymbol{\varepsilon}(\tilde{\boldsymbol{u}}) : \boldsymbol{\varepsilon}(\tilde{\boldsymbol{u}}) \\ &+ \frac{G_c}{2c_w} \Big(\ell \nabla \tilde{\alpha} \cdot \nabla \tilde{\beta} + \frac{w'(\tilde{\alpha}) \tilde{\beta}}{2\ell} \Big) + \frac{G_c}{\epsilon} \Big(\mathcal{M}(\tilde{\alpha} - 1) \\ &- \mathcal{M}(-\dot{\tilde{\alpha}}) \Big) \tilde{\beta} \Big) \mathrm{d}x \, \mathrm{d}t + \int_\Omega \tilde{\lambda}(\tilde{\alpha}(0) - \alpha_0) \, \mathrm{d}x. \end{aligned} \tag{8.99}$$

Since the boundaries $\Gamma_D \cup \Gamma_{\bar{\boldsymbol{u}}}$ are non-optimizable, the variables $\tilde{\boldsymbol{u}}(t)$, $\tilde{\boldsymbol{v}}(t)$, $\tilde{\alpha}(t)$, $\tilde{\beta}(t)$, and $\tilde{\lambda}$ are independent of Ω. When the optimality condition are applied to the Lagrangian (8.99) (that is, its partial derivatives with respect to its independent variables are set to zero), we obtain the state equation (8.93), the adjoint equation (8.61) and the shape derivative (8.98).

At first, differentiating the Lagrangian (8.99) with respect to the adjoint variable $(\tilde{\boldsymbol{v}}, \tilde{\beta})$ in the direction $(\boldsymbol{\psi}, \varphi) \in H^1([0,T], \tilde{Z}_0)$ and equating it to zero at $(\tilde{\boldsymbol{u}}, \tilde{\alpha}) = (\boldsymbol{u}, \alpha)$, we obtain

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \tilde{\beta}}(\varphi) = &\int_0^T \Big(\int_\Omega \frac{G_c}{2c_w} \Big(\ell \nabla \alpha \cdot \nabla \varphi + \frac{w'(\alpha)\varphi}{2\ell} \Big) \mathrm{d}x + \int_\Omega \frac{1}{2} \mathbb{C}'(\alpha) \varphi \boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{u}) \, \mathrm{d}x \\ &+ \int_\Omega \frac{G_c}{\epsilon} \big(\mathcal{M}(\alpha - 1) - \mathcal{M}(-\dot{\alpha}) \big) \varphi \, \mathrm{d}x \Big) \mathrm{d}t = 0 \quad \forall \varphi \in H^1([0,T], \tilde{H}^1_D(\mathbb{R}^d))), \end{aligned}$$

$$\frac{\partial \mathcal{L}}{\partial \tilde{\boldsymbol{v}}}(\boldsymbol{\psi}) = \int_0^T \left(\int_\Omega \mathbb{C}(\alpha) \boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{\psi}) \, \mathrm{d}x \right) \mathrm{d}t = 0 \qquad \forall \boldsymbol{\psi} \in H^1([0,T], \tilde{V}_0).$$

To obtain the initial condition $\alpha(0) = \alpha_0$, it suffices to differentiate (8.99) with respect to $\tilde{\lambda}$ at $\tilde{\alpha} = \alpha$. We thus recover the state equation (8.93).

Second, we differentiate the Lagrangian (8.99) with respect to $(\tilde{\boldsymbol{u}}, \tilde{\alpha})$ to recover the adjoint equation. By definition, $\tilde{\boldsymbol{u}} \in \tilde{\mathcal{C}}_t$, which is an affine space. The admissible perturbations $\boldsymbol{\psi}$ with respect to $\tilde{\boldsymbol{u}}$, must be such that $\tilde{\boldsymbol{u}} + \boldsymbol{\psi} \in \tilde{\mathcal{C}}_t$, hence $\boldsymbol{\psi} \in \tilde{V}_0$. Equating to zero the partial derivative in the direction $(\boldsymbol{\psi}, \varphi) \in H^1([0,T], \tilde{Z}_0)$, and denoting by $(\boldsymbol{v}, \beta)$ its solution for $(\tilde{\boldsymbol{u}}, \tilde{\alpha}) = (\boldsymbol{u}, \alpha)$ and $\tilde{\lambda} = \lambda$, we arrive at

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}}(\boldsymbol{\psi}) = \int_0^T \Big(\int_\Omega \big(\partial_{\boldsymbol{u}} m(\boldsymbol{u}, \alpha)\boldsymbol{\psi} + \mathbb{C}(\alpha)\boldsymbol{\varepsilon}(\boldsymbol{v}) : \boldsymbol{\varepsilon}(\boldsymbol{\psi}) \\ + \mathbb{C}'(\alpha)\beta\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{\psi})\big) \mathrm{d}x \Big) \mathrm{d}t = 0 \qquad \forall \boldsymbol{\psi} \in H^1([0,T], \tilde{V}_0), \end{aligned} \tag{8.101a}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \alpha}(\varphi) = \int_0^T \Big(\int_\Omega \big(\partial_\alpha m(\boldsymbol{u}, \alpha)\varphi + \mathbb{C}'(\alpha)\varphi\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v}) + \frac{G_c \ell}{2c_w} \nabla\beta \cdot \nabla\varphi \\ + \big(\frac{1}{2}\mathbb{C}''(\alpha)\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{u}) + \frac{G_c}{4c_w}\frac{w''(\alpha)}{\ell} \\ + \frac{G_c}{\epsilon}\mathcal{M}'(\alpha - 1)\big)\beta\varphi + \frac{G_c}{\epsilon}\mathcal{M}'(-\dot{\alpha})\dot{\varphi}\beta\big) \mathrm{d}x\Big) \mathrm{d}t + \int_\Omega \lambda\varphi(0) \mathrm{d}x = 0 \\ \forall \varphi \in H^1([0,T], \tilde{H}^1_D(\mathbb{R}^d)). \end{aligned} \tag{8.101b}$$

Varying the test function $\boldsymbol{\psi}$ in (8.101a), we get the boundary condition $\boldsymbol{v}(t) = \mathbf{0}$ on $\Gamma_{\bar{\boldsymbol{u}}}$ and for all $t \in [0, T)$

$$\int_\Omega \big(\partial_{\boldsymbol{u}} m(\boldsymbol{u}, \alpha)\boldsymbol{\psi} + \mathbb{C}(\alpha)\boldsymbol{\varepsilon}(\boldsymbol{v}) : \boldsymbol{\varepsilon}(\boldsymbol{\psi}) + \mathbb{C}'(\alpha)\beta\boldsymbol{\varepsilon}(\boldsymbol{v}) : \boldsymbol{\varepsilon}(\boldsymbol{\psi}) \big) \, \mathrm{d}x = 0 \qquad \forall \boldsymbol{\psi} \in V_0,$$

where we used definition (8.96) of V_0. We have thus derived the adjoint equation (8.97a). Now, to get rid of the time derivative $\dot{\varphi}$ in Eq. (8.101b), we integrate the term $\mathcal{M}'(-\dot{\alpha})\dot{\varphi}\beta$ by parts with respect to t and obtain

$$\begin{aligned}&\int_0^T \Big(\int_\Omega \big(\partial_\alpha m(\boldsymbol{u}, \alpha)\varphi + \mathbb{C}'(\alpha)\varphi\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v}) + \frac{G_c \ell}{2c_w}\nabla\beta \cdot \nabla\varphi \\ &+ \big(\frac{1}{2}\mathbb{C}''(\alpha)\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{u}) + \frac{G_c}{4c_w}\frac{w''(\alpha)}{\ell} + \frac{G_c}{\epsilon}\mathcal{M}'(\alpha - 1)\big)\beta\varphi\big) \mathrm{d}x \\ &- \int_\Omega \frac{G_c}{\epsilon}\Big(\frac{\mathrm{d}}{\mathrm{d}t}\big(\mathcal{M}'(-\dot{\alpha})\beta\big)\Big)\varphi \, \mathrm{d}x\Big) \mathrm{d}t \\ &+ \int_\Omega \lambda \, \varphi|_{t=0} \, \mathrm{d}x + \int_\Omega \frac{G_c}{\epsilon}\big(\mathcal{M}'(-\dot{\alpha})\beta\varphi\big|_{t=T} - \mathcal{M}'(-\dot{\alpha})\beta\varphi\big|_{t=0}\big) \mathrm{d}x = 0 \\ &\quad \forall \varphi \in H^1([0,T], \tilde{H}^1_D(\mathbb{R}^d)). \end{aligned} \tag{8.102}$$

This integration by part is legitimate if $\mathcal{M}$ is a smooth function but is purely formal if $\mathcal{M}$ is the maximum function since in such a case $\mathcal{M}'$ is a Heaviside function and its time derivative involves a Dirac function. Varying φ in (8.102), we find that the Lagrange multiplier λ is given by

$$\lambda = \frac{G_c}{\epsilon}\, \mathcal{M}'(-\dot{\alpha})\beta\big|_{t=0},$$

and that the adjoint problem (8.97b) for β holds true. Finally, since $J(\Omega) = \mathcal{L}(\boldsymbol{u}, \alpha, \tilde{\boldsymbol{v}}, \tilde{\beta}, \tilde{\lambda}, \tilde{\boldsymbol{\mu}}, \Omega)$, the shape derivative $J'(\Omega)(\boldsymbol{\theta})$ satisfies, for any $\boldsymbol{\theta} \in W_0^{1,\infty}(D, \mathbb{R}^d)$,

$$J'(\Omega)(\boldsymbol{\theta}) = \frac{\partial \mathcal{L}}{\partial \Omega}(\boldsymbol{\theta}) + \frac{\partial \mathcal{L}}{\partial \boldsymbol{u}}\left(\frac{\partial \boldsymbol{u}}{\partial \Omega}(\boldsymbol{\theta})\right) + \frac{\partial \mathcal{L}}{\partial \alpha}\left(\frac{\partial \alpha}{\partial \Omega}(\boldsymbol{\theta})\right)$$

as the variables $\tilde{\boldsymbol{v}}$, $\tilde{\beta}$, $\tilde{\lambda}$ and $\tilde{\boldsymbol{\mu}}$ are independent of Ω. Substituting these variables by the optimal ones $\boldsymbol{v}$, β, λ, $\boldsymbol{\mu}$ and using the adjoint equation (8.61), the two last terms in the above formula vanish, resulting in

$$J'(\Omega)(\boldsymbol{\theta}) = \frac{\partial \mathcal{L}}{\partial \Omega}(\boldsymbol{\theta}).$$

Consequently, formula (8.98) is deduced by a straightforward application of Lemma 1. □

8.8.4 *Time Discretized State and Adjoint Equations*

The adjoint equation (8.61) is a linear backward parabolic equation with a final condition at $t = T$. This equation was assumed to be well-posed in the statement of Theorem 1. A time discretized version of the adjoint equation can be shown to be well-posed (Desai et al. 2022).

To construct a time discretized version of the adjoint problem (8.61), we first time discretize the state equation (8.93) along with the objective function (8.94). The time interval $[0, T]$ is split in N sub-intervals of length $\delta t = T/N$. Let $(\boldsymbol{u}_n, \alpha_n)$ and $(\boldsymbol{v}_n, \beta_n)$ denote the discrete state and adjoint solutions, respectively, at the end of every n-th time interval. The discrete state is determined using an implicit scheme: initialize $(\boldsymbol{u}_0, \alpha_0) = (\boldsymbol{0}, \alpha_0)$ and, for $0 < n \leq N$, find $(\boldsymbol{u}_n, \alpha_n) \in Z$ such that $\boldsymbol{u}_n = \bar{\boldsymbol{u}}(t_n)$ on $\Gamma_{\bar{u}}$ and

$$\int_\Omega \mathbb{C}(\alpha_n)\boldsymbol{\varepsilon}(\boldsymbol{u}_n) : \boldsymbol{\varepsilon}(\boldsymbol{v})\, \mathrm{d}x = 0\, \mathrm{d}s \qquad \forall \boldsymbol{v} \in V, \tag{8.103a}$$

$$\int_\Omega \frac{G_c}{2c_w}\left(\ell\nabla\alpha_n\cdot\nabla\beta+\frac{w'(\alpha_n)}{2\ell}\right)\,\mathrm{d}x+\int_\Omega\frac{1}{2}\mathbb{C}'(\alpha_n)\boldsymbol{\varepsilon}(\boldsymbol{u}_n):\boldsymbol{\varepsilon}(\boldsymbol{u}_n)\,\mathrm{d}x$$
$$+\int_\Omega\frac{G_c}{\epsilon}\left(\mathcal{M}(\alpha_n-1)\beta\,\mathrm{d}x-\int_\Omega\mathcal{M}\left(\frac{\alpha_{n-1}-\alpha_n}{\delta t}\right)\beta\right)\mathrm{d}x=0\qquad\forall\beta\in H^1_D(\Omega).\tag{8.103b}$$

The above problem can be shown to admit at least one solution (Desai 2021), while uniqueness of the solution is not guaranteed [like its continuous counterpart (8.93)]. Nevertheless, we assume that the solution $(\boldsymbol{u}_n,\alpha_n)\in Z$ is unique. The discretization of the objective function (8.94) reads:

$$J_N(\Omega)=\sum_{n=0}^{N}\delta t\int_\Omega m(\boldsymbol{u}_n,\alpha_n)\,\mathrm{d}x.\tag{8.104}$$

Introducing a Lagrangian, as in the proof of Proposition 1, adapted to the above discretization, we obtain the following discrete adjoint problem: initialize $(\boldsymbol{v}_N,\beta_N)=(\boldsymbol{0},0)$ and, for $N-1\geq n\geq 0$, find $(\boldsymbol{v}_n,\beta_n)\in Z$ such that

$$\int_\Omega\left(\partial_{\boldsymbol{u}_n}m(\boldsymbol{u}_n,\alpha_n)\boldsymbol{\psi}+\mathbb{C}(\alpha_n)\boldsymbol{\varepsilon}(\boldsymbol{v}_n):\boldsymbol{\varepsilon}(\boldsymbol{\psi})+\mathbb{C}'(\alpha_n)\beta_n\boldsymbol{\varepsilon}(\boldsymbol{u}_n):\boldsymbol{\varepsilon}(\boldsymbol{\psi})\right)\,\mathrm{d}x=0$$
$$\forall\boldsymbol{\psi}\in V_0,\tag{8.105a}$$

$$\int_\Omega\frac{G_c\ell}{2c_w}\nabla\varphi\cdot\nabla\beta_n\,\mathrm{d}x$$
$$+\int_\Omega\left(\mathbb{C}'(\alpha_n)\varphi\boldsymbol{\varepsilon}(\boldsymbol{u}_n):\boldsymbol{\varepsilon}(\boldsymbol{v}_n)+\frac{1}{2}\mathbb{C}''(\alpha_n)\boldsymbol{\varepsilon}(\boldsymbol{u}_n):\boldsymbol{\varepsilon}(\boldsymbol{u}_n)+\frac{G_c}{4c_w}\frac{w''(\alpha_n)}{\ell}\right)$$
$$\varphi\beta_n\,\mathrm{d}x+\int_\Omega\frac{G_c}{\epsilon}\left(\mathcal{M}'(\alpha_n-1)+\frac{1}{\delta t}\mathcal{M}'(\alpha_{n-1}-\alpha_n)\right)\varphi\beta_n\,\mathrm{d}x$$
$$+\int_\Omega\partial_{\alpha_n}m(\boldsymbol{u}_n,\alpha_n)\varphi\,\mathrm{d}x=\frac{G_c}{\epsilon\delta t}\int_\Omega\mathcal{M}'(\alpha_n-\alpha_{n+1})\beta_{n+1}\varphi\,\mathrm{d}x\quad\forall\varphi\in H^1_D(\Omega).\tag{8.105b}$$

Remark 5 When $\mathcal{M}$ is the maximum function, its derivative is the Heaviside function, $\mathcal{M}'=\mathcal{H}$. The value $\mathcal{H}(0)$ is not precisely defined since the Heaviside function is discontinuous at zero. Numerically, we tested the adjoint equation (8.105b) (along with the corresponding shape derivative) for values $\mathcal{H}(0)=0$ and $\mathcal{H}(0)=1$, and both choices yield the same optimized shape (at least, for the 2D cantilever in Sect. 8.7).

8.8.5 Numerical Implementation

In this section, we expound the numerical aspects of the resolution of the state equation (8.93) and adjoint equation (8.61) and the proposed shape optimization algorithm. For all the numerical test cases presented in Sect. 8.7, the material is chosen to be concrete having the following properties (Bažant and Pijaudier-Cabot 1989): Young modulus $E = 29\,\text{GPa}$, Poisson ratio $\nu = 0.3$, ultimate tensile strength $\sigma_M = 4.5\text{MPa}$ and fracture toughness $G_c = 70\ \text{J/m}^2$. For the damage model, the characteristic length ℓ is calculated using the formula (Tanné et al. 2018)

$$\ell = \frac{3G_c E}{8\sigma_M^2}. \tag{8.106}$$

The domain Ω is discretized by a simplicial unstructured mesh Ω^h. The mesh is produced by the `MMG` software (Dapogny et al. 2014) which features two important input parameters: the minimal and maximal mesh size, denoted by $h_{\min}$ and $h_{\max}$, respectively. The mesh Ω^h is assumed to be uniform in the sense that $h_{\max}$ and $h_{\min}$ are of the same order of magnitude. The mesh size ought to be chosen such that the details smaller than the characteristic length ℓ are captured. Due to numerical experiments in Miehe et al. (2010), it was found that a uniform mesh size h chosen such that $2h < \ell$ suffices. Typically, in the mesh produced by `MMG`, the actual minimal mesh size is lesser than $h_{\min}$. Hence $h_{\min}$ in all our numerical experiments is not a precise lower bound on the mesh size, rather a parameter given to `MMG`. We performed numerical experiments and found that it suffices to choose

$$2h_{\min} < \ell. \tag{8.107}$$

We have to make one exception with the rule (8.107) in the test case of Sect. 8.9.3 (a realistic column of height 4 m) where we just enforce $h_{\min} < \ell$ in order to have a not too fine mesh which can be treated without resorting to high performance computing. The penalization parameter is chosen to be small, $\epsilon = \mathcal{O}(h_{\max}^2)$ (its precise value is given in the beginning of Sect. 8.7).

Remark 6 Despite the fact that the damage model (8.93) is non-local, the crack initiation is mesh-dependent. For instance, mesh refinement at corners of the shape Ω makes the crack initiation easier at these corners. For this reason we rely on uniform meshes ($h_{\max}$ and $h_{\min}$ of the same order), so that the crack initiation is unbiased.

The spaces V, C_{ν}, $H_D^1(\Omega)$, Z, defined by (8.77), (8.78), and (8.79) are discretized by piecewise affine continuous (linear) finite elements and their discrete counterparts are denoted by V^h, C_{ν}^h, $H_D^1(\Omega^h)$, Z^h, respectively.

The time interval $[0, T]$ is discretized in N intervals of length $\delta t = T/N$. The time at the end of the n-th time interval is denoted t_n, $n = 1, 2, \ldots, N$.

8.8.5.1 Solving the Fracture Model

We still denote by $(\boldsymbol{u}_n, \alpha_n) \in Z^h$ the time-space discretized solution at time interval t_n (we do not write its dependence to h). The space discretized version of (8.103) is simply the same variational formulation with the finite dimensional spaces V^h, $H_D^1(\Omega^h)$, Z^h replacing their continuous counterparts V, $H_D^1(\Omega)$, Z. Following (Bourdin et al. 2008), the nonlinear variational formulation (8.103) is solved by a sequentially alternate algorithm: fixing α_n, solve (8.103a) for $\boldsymbol{u}_n$; fixing $\boldsymbol{u}_n$, solve (8.103b) for α_n. This algorithm exploits the fact that the total energy (8.86) is separately convex in $\boldsymbol{u}$ and α, but not with respect to the couple $(\boldsymbol{u}, \alpha)$, which may hinder the convergence of a standard Newton algorithm. This algorithm of Bourdin et al. (2008) is precisely recalled in Algorithm 3.

Algorithm 3 Numerical resolution of (8.103)

Initialization: $(\boldsymbol{u}_n^0, \alpha_n^0) = (\boldsymbol{0}, 0)$ for $n = 0$ and $(\boldsymbol{u}_n^0, \alpha_n^0) = (\boldsymbol{u}_{n-1}^0, \alpha_{n-1}^0)$ for $n > 0$.
For $i = 0, 1, \ldots,$ do

1. Substitute $\alpha_n = \alpha_n^i$ in (8.103a), and solve it for $\boldsymbol{u}_n = \boldsymbol{u}_n^{i+1}$
2. Substitute $\boldsymbol{u}_n = \boldsymbol{u}_n^{i+1}$ in (8.103b), and solve it for $\alpha_n = \alpha_n^{i+1}$ using a Newton–Raphson or a fixed point algorithm
3. If $\left\|\alpha_n^{i+1} - \alpha_n^i\right\|_{L^2(\Omega^h)} < tol \left\|1\right\|_{L^2(\Omega^h)}$, then exit the loop, else repeat

The tolerance tol in Algorithm 3 is chosen to be 10^{-6}. The resolution for α_n (step 2 in the Algorithm 3) using Newton algorithm is easy given that there are no constraints on α_n, because of penalization. Without penalization, one needs to apply a constrained Newton algorithm, the numerical implementation of which is not straightforward. Algorithm 3 builds a sequence $(\boldsymbol{u}_n^i, \alpha_n^i) \longrightarrow (\boldsymbol{u}_n, \alpha_n)$ as $i \longrightarrow \infty$. Typically, its convergence is very fast when α_n is close to zero everywhere in Ω^h. Whereas the convergence is very slow when α_n approaches unity and there is crack formation in Ω^h.

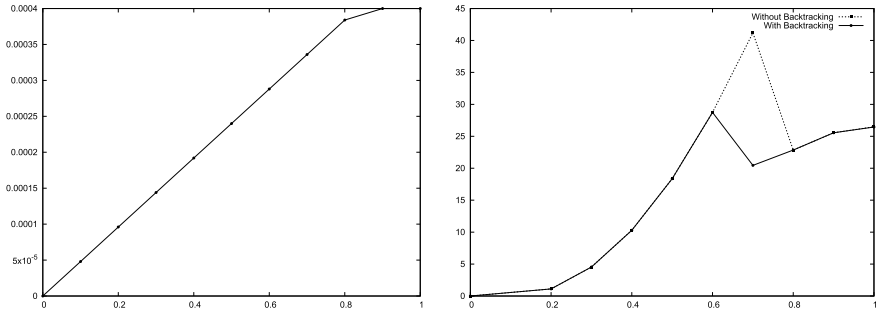
Of course, the solution $(\boldsymbol{u}_n, \alpha_n)$ depends on the mesh size $h_{\max}$ and on the time step δt. Furthermore, each solution at time t_n depends on the whole time history before t_n. In practice, the solution depends on the initialization $(\boldsymbol{u}_n^0, \alpha_n^0)$ which is usually taken as the solution at the previous time step $(\boldsymbol{u}_{n-1}, \alpha_{n-1})$. Therefore, it is not clear that Algorithm 3 delivers an approximation of the global minimizer of the total energy (8.86) (Chambolle 2004). Rather, we may end up in local minima. Following again (Bourdin et al. 2008), we rather use the backtracking Algorithm 4 which escapes from local minima in practice. The idea of Algorithm 4 is that, if the solution found at time step n using Algorithm 3 features a strong increase of damage, the solution at the previous time step $n - 1$ is re-computed using the solution at time step n as initial guess. In doing so, we expect to find a new solution $(\boldsymbol{u}_{n-1}, \alpha_{n-1})$ which is a better minimizer of the total energy (8.86). For example, consider the shape in Fig. 8.32a. This shape is subjected to boundary conditions shown in Fig. 8.31 and to

Algorithm 4 Numerical resolution of (8.103) with backtracking

Set $n = 1$ and $(\tilde{\boldsymbol{u}}, \tilde{\alpha}) = (\mathbf{0}, 0)$
While $n \leq N$ (time steps), do

1. Solve (8.103) for $(\boldsymbol{u}_n, \alpha_n)$ using Algorithm 3 using an initial guess $(\boldsymbol{u}_n^0, \alpha_n^0) = (\tilde{\boldsymbol{u}}, \tilde{\alpha})$
2. If $\|\alpha_n - \tilde{\alpha}\|_{L^\infty(\Omega^h)} > 0.5$ and $n > 1$

   then $(\tilde{\boldsymbol{u}}, \tilde{\alpha}) = (\boldsymbol{u}_n, \alpha_n)$, and set $n = n - 1$,
   else $(\tilde{\boldsymbol{u}}, \tilde{\alpha}) = (\boldsymbol{u}_n, \alpha_n)$, and set $n = n + 1$.



(a) $\bar{\boldsymbol{u}}(t)$, defined by (1.114), versus time t (b) $\mathcal{E}(\boldsymbol{u}, \alpha)$, defined by (1.83), versus time t

Fig. 8.30 Evolution of elastic energy for the shape in Fig. 8.32a with and without the backtracking Algorithm 4

an applied displacement given in (8.114). The equation (8.103) is resolved with and without Algorithm 4 and the evolution of the elastic energy $\mathcal{E}(\boldsymbol{u}, \alpha)$ (8.83) is plotted in Fig. 8.30. The spike in elastic energy indicates the time instant after which cracks are formed. As seen in Fig. 8.30, the solution obtained via backtracking exhibits crack earlier than the solution obtained without backtracking. The elastic energy after $t = 0.8$ is the same with and without backtracking algorithm and the final crack configuration is the same for the two cases. From a physical standpoint, the better minimizer of the total energy (8.86) (obtained via backtracking) consumes lesser amount of energy to exhibit cracks (as can be seen in Fig. 8.30). The cracks which consume lesser amount of energy are more likely to appear physically. Considering such minimizers (obtained via backtracking) in a gradient-based shape optimization ensures the minimization of the higher likelihood of cracks.

8.8.5.2 Solving the Adjoint Problem

We still denote by $(\boldsymbol{v}_n, \beta_n) \in Z^h$ the time-space discretized adjoint solution at time interval t_n. As for the state solution in the previous subsection, the space discretized version of (8.105) is simply the same variational formulation with the

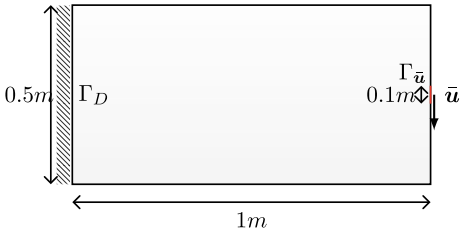


Fig. 8.31 2D Cantilever boundary conditions

finite dimensional spaces V^h, $H_D^1(\Omega^h)$, Z^h replacing their continuous counterparts V, $H_D^1(\Omega)$, Z. As usual the adjoint problem is solved backward in time, i.e., for decreasing indices $n = N-1, \ldots, 1, 0$. One ought to solve the state equation (8.103) (using Algorithm 4) until the last time step, store the solutions $(\boldsymbol{u}_n, \alpha_n)$ for every time step and retrieve the solutions starting from the last time step. As explained in Sect. 8.8.4, the bilinear form in the variational formulation (8.105) is not coercive. Hence, for numerical implementations, one ought to use a direct solver or an iterative technique like `GMRES`, that is capable of resolving indefinite matrices.

Remark 7 The numerical resolution of (8.105) using a direct solver or `GMRES` is slow. If the damage variable is small $\|\alpha_n\|_{L^\infty(\Omega^h)} \ll 1$, the adjoint problem is close to a simple linear elasticity problem and one can rather use an iterative solver meant for positive definite matrices, for instance, `CG`, to save computational effort.

Finally, the space-time discretized version of the shape derivative of (8.104) is

$$\begin{aligned} J_N'(\Omega^h)(\boldsymbol{\theta}) = \int_\Gamma \boldsymbol{\theta}\cdot\boldsymbol{n} \sum_{n=0}^{N} \delta t \big(& m(\boldsymbol{u}_n, \alpha_n) + \mathbb{C}'(\alpha_n)\beta_n \boldsymbol{\varepsilon}(\boldsymbol{u}_n) : \boldsymbol{\varepsilon}(\boldsymbol{u}_n) \\ & + \frac{G_c \ell}{2c_w} \nabla\alpha_n \cdot \nabla\beta_n + \frac{G_c}{4c_w} \frac{w'(\alpha_n)}{\ell} \beta_n + \mathbb{C}(\alpha_n)\boldsymbol{\varepsilon}(\boldsymbol{u}_n) : \boldsymbol{\varepsilon}(\boldsymbol{v}_n) \\ & + \frac{G_c}{\epsilon} \left(\mathcal{M}(\alpha_n - 1) - \mathcal{M}\Big(\frac{\alpha_{n-1} - \alpha_n}{\delta t}\Big) \right) \beta_n \big) \mathrm{d}s. \end{aligned} \tag{8.108}$$

8.8.5.3 Regularization and Extension of the Shape Derivative

The shape derivative (8.108) is defined only on the boundary Γ, while it is needed in the full computational domain D for solving the transport equation (8.19). Furthermore, both the derivative and the shape may be not smooth enough, which may result in poor numerical efficiency. Therefore, it is convenient to regularize the shape derivative (8.108) (Allaire et al. 2021; Burger 2003), still ensuring that it is a descent direction. A classical possibility is to consider the $H^1(D)$ scalar product (instead of the $L^2(\Gamma)$ scalar product) for identifying the gradient. In other words, introducing a mesh D^h of D and denoting by $H^1(D^h)$ the linear finite element subspace of $H^1(D)$, we seek a function $dj(\Omega^h) \in H^1(D^h)$ such that

$$\int_D \left(4h_{\min}^2 \nabla \mathrm{d}j(\Omega^h) \cdot \nabla\varphi + \mathrm{d}j(\Omega^h)\varphi\right) \mathrm{d}x = \int_\Gamma j'(\Omega^h)\,\varphi\,\mathrm{d}x \qquad \forall \varphi \in H^1(D^h), \tag{8.109}$$

where $h_{\min}$ is the fixed minimal mesh size and $j'(\Omega^h)$ is the function defined by formula (8.108) with

$$J_N'(\Omega^h)(\boldsymbol{\theta}) = \int_\Gamma \boldsymbol{\theta} \cdot \boldsymbol{n}\, j'(\Omega^h)\,\mathrm{d}s.$$

For the implementation details of the above, an interested reader can refer to Desai (2021).

8.8.5.4 Shape Optimization Algorithm

We consider the shape optimization problem (8.95) where the admissible shapes must satisfy a constraint on the target volume V_f. Hence, we introduce a Lagrangian in the same manner as before

$$\mathcal{L}(\boldsymbol{u}, \alpha, \boldsymbol{v}, \beta, \Omega^h, \lambda) = J(\Omega^h) + \frac{\lambda}{C_V}\left(\int_{\Omega^h} \mathrm{d}x - V_f\right), \tag{8.110}$$

where λ is the Lagrange multiplier for the volume constraint and C_V is a normalization constant. Denoting by Ω_0^h the initial shape, the constant C_V is defined by

$$C_V = \left|\int_{\Omega_0^h} \mathrm{d}x - V_f\right|. \tag{8.111}$$

The optimization strategy is implemented in the same as before, resulting in Algorithm 2.

8.8.5.5 Parallel Implementation

All our numerical experiments are performed with the open-source software `FreeFEM 4.8` (Hecht 2012), installed on a workstation featuring an Intel(R) Xeon(R) Gold 6230 CPU and 40 processors. In the next section we shall present a large scale topology optimization test case, consisting of approximately 1.3 million tetrahedra. It would be impossible to perform this test case on a single processor, because its CPU time would be at least a few weeks. Indeed, 90% of the total computation time is spent on the first three steps of Algorithm 5, namely on the finite element analyses.

Algorithm 5 Shape optimization for the damage model

Initialize with a shape Ω_0^h and repeat over $i = 1, \ldots, \mathcal{N}$

1. Solve for the state $(\boldsymbol{u}, \alpha)$ in Ω_i^h marching in time from t_1 until t_N using Algorithm 4
2. Solve for the adjoint $(\boldsymbol{v}, \beta)$ in Ω_i^h backward in time from t_N up to t_1
3. Compute the shape derivative using (8.108) and regularize it with (8.109) to deduce $dj(\Omega_i^h)$
4. Update the Lagrange multiplier λ_{i+1} with (8.31)
5. Solve the transport equation (8.19) with the velocity given by (8.32) for the pseudo-time step τ given by (8.30) to obtain the new level-set function $\tilde{\phi}_{i+1}$
6. Re-initialize $\tilde{\phi}_{i+1}$ to the signed distance function ϕ_{i+1} (defining a new shape Ω_{i+1}^h)
7. Compute the volume V_{i+1}. If it is close to the volume target, apply the projection algorithm (8.33) to satisfy exactly the volume constraint.
8. Remesh the box D using `MMG` (Dapogny et al. 2014) to obtain the body-fitted mesh of the new shape Ω_{i+1}^h

Therefore, in order to minimize the total computation time, we must perform parallel computation for the finite element analysis. Fortunately, `FreeFEM` comes with built-in OpenMPI and the open-source package `PETSc` (Balay et al. 2019). The resolution of the state equation (8.103) (using Algorithm 4), the adjoint equation (8.105) and the regularization of the shape derivative (8.109) are performed using the parallel-solver of `PETSc`. The mesh is partitioned in sub-domains using the open-source package `METIS` (Karypis and Kumar 1997). The finite element rigidity matrix is thus partitioned accordingly, and the linear systems are solved in parallel with the `GAMG` (geometric algebraic multigrid) preconditioner. The state equation and the regularization problem are solved by the conjugated gradient algorithm, while the adjoint equation is solved by `GMRES`. For details of implementation we refer to the tutorial on the parallel version of `FreeFEM` (Jolivet 2020). Some examples of total computational times are given in Remark 9.

8.9 Numerical Results: Fracture

We now present 2D and 3D shape optimization results for the damage model (8.93), which prove the effectiveness of our algorithm to produce crack-free structures.

We choose to maximize rigidity, namely to minimize an objective function which is the total compliance. As already explained in Sect. 8.8.1, no body or surface forces can be considered in the Francfort-Marigo damage model and, rather, one has to impose given displacements to the structure. In such a case, there is a subtle definition of compliance, see, e.g., Barbarosie and Lopes (2011), which takes into account the reaction force on the surface where the displacement is imposed. Since the work done by this reaction force is equal to the elastic energy, the total compliance is defined as

$$J(\Omega) = -\int_0^T \int_\Omega \mathbb{C}(\alpha)\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{u}) \mathrm{d}x \, \mathrm{d}t. \tag{8.112}$$

The objective function (8.112) is minimized in all test cases, except otherwise mentioned.

The final time is $T = 1s$ and the time step is $\delta t = 0.15s$. The solution $(\boldsymbol{u}, \alpha)$ is computed by solving the penalized formulation (8.103) using Algorithm 4. The penalization parameter is chosen equal to $\epsilon = 10^{-5}$.

An isotropic degradation function, with a small residual stiffness $\mathbb{C}_{\text{res}}$, is considered

$$\mathbb{C}(\alpha) = (1-\alpha)^2 \mathbb{C}_0 + \mathbb{C}_{\text{res}}. \tag{8.113}$$

In practice the residual stiffness is $\mathbb{C}_{\text{res}} = 10^{-6}\mathbb{C}_0$ but it could even be taken to be zero if the finite element solver is robust enough (which is the case with `FreeFEM`). For all test cases, only the dissipation function (8.92) of the `DL` model is considered.

Since we are in a quasi-static evolution framework, the rate of increment of the imposed displacement has no effect on the solution $(\boldsymbol{u}, \alpha)$ at the final time $T = 1s$. But the rate does have a strong influence on the objective function (8.112) (since it contains a time-integral). In order to see a greater influence of the damage variable α on the optimized shape, we consider an imposed displacement $\bar{\boldsymbol{u}}(t)$ that grows from zero to a certain value and then remains constant for some period.

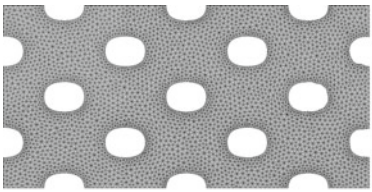
8.9.1 2D Cantilever

We study a 2D cantilever beam represented by a rectangle of dimensions $1m \times 0.5\,\text{m}$ as shown in Fig. 8.31. The cantilever beam is clamped all along its leftmost edge and subjected to an applied vertical displacement,

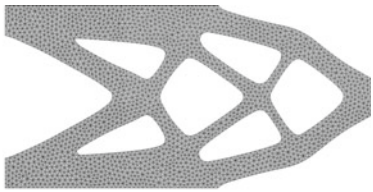
$$\bar{\boldsymbol{u}}(t) = (0, 4\min(1.2t, 1)) \times 10^{-4} m, \quad t \in [0, 1], \tag{8.114}$$

on a centered part of its rightmost edge $\Gamma_{\bar{\boldsymbol{u}}}$ of length 0.1 m. The above displacement (8.114) is chosen in such a way that the initial shape of Fig. 8.32a suffers from a crack as seen in Fig. 8.34a. This displacement is kept constant near the end of the loading process so that the weight of the largest loading is increased in the objective function and the optimization algorithm works harder to get rid of the possible cracks. A target volume $V_f = 0.25\,\text{m}^2$ is imposed for all the test cases in this subsection. The parameters of the remeshing tool `MMG` are: $h_{\min} = 0.0064m$, $h_{\max} = 0.0128\,\text{m}$.

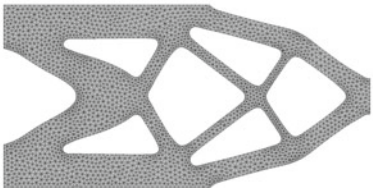
The initial shape is displayed in Fig. 8.32a. The shape obtained by minimizing (8.112) for linear elasticity (without any damage) is plotted in Fig. 8.32b. The shape obtained by minimizing (8.112) for the damage model is plotted in Fig. 8.32c. The convergence history is plotted in Fig. 8.33.



(a) Initialization

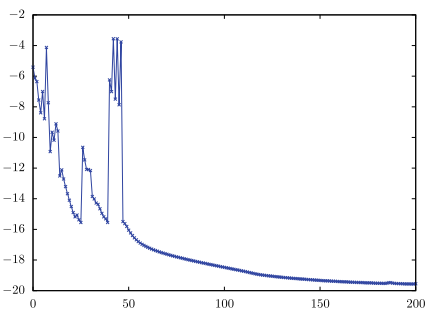


(b) Final shape obtained for linear elasticity

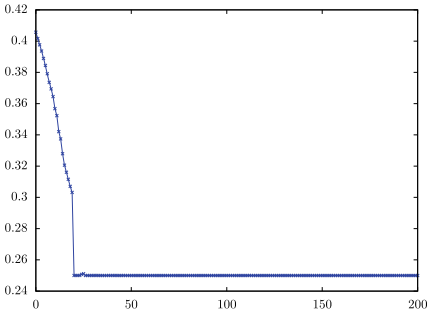


(c) Final shape for the damage model

Fig. 8.32 Initial and final meshes for the cantilever shapes of Sect. 8.9.1



(a) Total elastic energy (1.112) v/s iterations



(b) Volume v/s iterations

Fig. 8.33 Convergence history for the optimized shape of Fig. 8.32c

As can be seen in Fig. 8.32, the optimized shapes, with or without damage in the mechanical model, are slightly different, but share the same topology. The shape in Fig. 8.32c do not undergo a crack, unlike the shape in Fig. 8.32b that does undergo a crack (see Fig. 8.35a). For the shape in Fig. 8.32c, the damaged region in the intermediate shapes is plotted in Fig. 8.34. The optimization algorithm indeed tries hard to remove every damaged or cracked region that appears. We observe that the crack appears in the cantilever at several locations, taking different configurations at each iteration and disappears finally after the 45-th iteration.

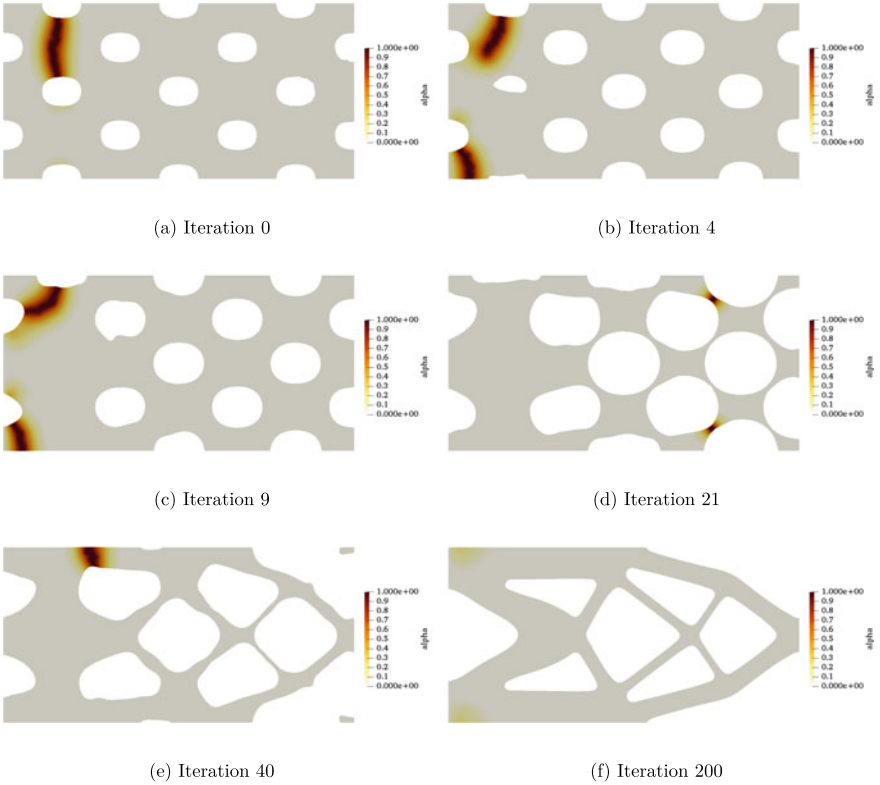
Remark 8 One can see in Fig. 8.33 that the objective function (8.112) features very strong oscillations during the first 50 iterations although the volume constraint is nicely satisfied after 20 iterations. These peaks in the convergence history occur whenever there is a transition from a shape without any crack to a shape with a crack, or when the crack changes from one position to a completely different one (see Fig. 8.34). In other words, small perturbations in the shape Ω can result in the appearance or disappearance of cracks, leading to abrupt changes in the objective function (8.112). Reducing the descent step τ would not help here because the onset of fracture is a discontinuous process with respect to load or geometry variations. Typically, the growth in time of a fracture can be discontinuous. Therefore, it is plausible that a small change in the geometry of the shape can induce a large change in the crack profile and thus in the value of the objective function (8.112). This non-smooth character of fracture or damage is well documented not only in the theoretical literature (Benallal and Marigo 2006; Chambolle et al. 2009, 2010, 2008; Pham et al. 2011) but also in the numerical literature (Bourdin et al. 2008). Note that our derivation of the shape derivative in Sect. 8.8.3 was performed under the assumption of a smooth solution of the damage model (8.93).

The regularization of the damage field α with the characteristic length ℓ or the penalization process of the damage irreversibility do not help at all on this matter. Our numerical experiments confirm this non-smoothness of the damage problem and the discontinuity of the objective function with respect to shape variations. Therefore, it is questionable to use a gradient-descent method to minimize the objective function (8.112). Nevertheless, the presented test cases show that, after some early oscillations, our gradient algorithm does converge to a crack-free optimal shape.

This is merely a numerical evidence for a finite family of examples and it may well be different for other geometries and sets of loading and material parameters.

There are two key ingredients for this relative success. First, although the descent step τ is adapted at each iteration by formula (8.30), we do not test if the objective function decreases at each iteration and we never step back with a smaller descent step. Second, we rely on the backtracking Algorithm 4 [following Bourdin et al. (2008)] which plays a pivotal role in ensuring a stable damage evolution. As a consequence, the shapes obtained in the final iterations are more stable in the sense that small geometric perturbations do not cause the appearance of a crack. Hence, oscillations in the objective function are avoided and a smooth convergence of the objective function is attained. This is confirmed by our attempt to replace the backtracking Algorithm 4 by the simpler Algorithm 3 (without backtracking) in our shape optimization Algorithm 5. We noticed that the fluctuations in the objective function were more violent than the ones obtained with backtracking, which was hindering convergence.

One could think that changing the initialization could improve the convergence of the shape optimization for the damage model. For example, instead of starting from the periodically perforated initial shape in Fig. 8.32a, it is possible to initialize the damage model optimization with the optimal shape for linear elasticity in Fig. 8.32b. We perform this new test case and the result in Fig. 8.35b is quite deceiving. Indeed,

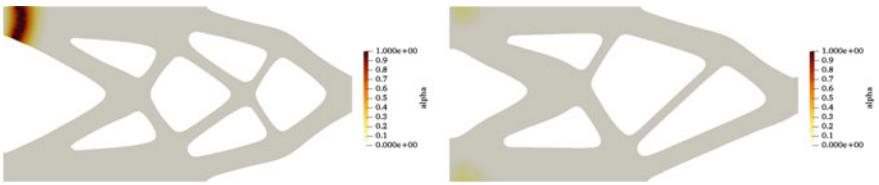


(a) Iteration 0 (b) Iteration 4

(c) Iteration 9 (d) Iteration 21

(e) Iteration 40 (f) Iteration 200

Fig. 8.34 Damage variable α at the final time T plotted for several optimization iterations for the shape in Fig. 8.32c (black regions indicate cracks)



(a) Initial shape as in Fig. 1.32b (b) Final shape

Fig. 8.35 Damage variable α at the final time T for the initial shape, as in Fig. 8.32b, and for the resulting optimal shape

it takes 600 iterations (3 times more, see Fig. 8.36) to converge to the shape of Fig. 8.35b, which is different from that previously obtained in Fig. 8.32c, slightly less optimal since the objective function for Fig. 8.32c is 0.7% better than for the shape of Fig. 8.35b. Note that both shapes of Fig. 8.35b and 8.32c feature no crack.

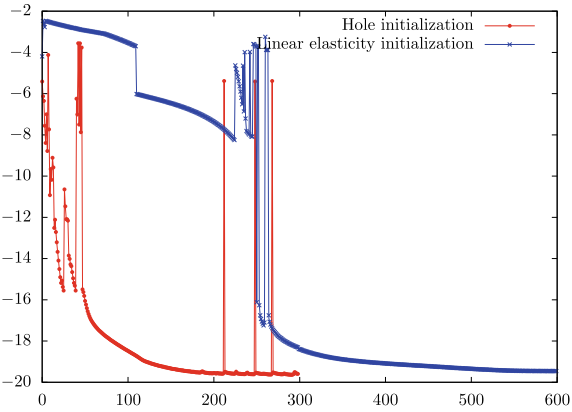


Fig. 8.36 Objective function (8.112) versus iterations for the shapes (8.32b) and (8.35b)

8.9.1.1 Traction-Only Degradation

Only for once, we replace the isotropic degradation function (8.113) by the following traction-only degradation function (Miehe et al. 2010)

$$\mathbb{C}(\boldsymbol{u},\alpha) = \Big(\mathcal{H}(\mathrm{d}iv\boldsymbol{u})(-1+(1-\alpha)^2)+1\Big)\mathbb{C}_0 + \mathbb{C}_{\mathrm{res}}, \tag{8.115}$$

where $\mathcal{H}$ denotes the Heaviside function. If $\mathrm{d}iv\boldsymbol{u} \geq 0$ the material is said to be in traction, otherwise it is in compression. The degradation function (8.115) is constructed in such a way that damage occurs only under tension. In other words, when $\mathrm{d}iv\boldsymbol{u} < 0$, whatever the value of α, one has $\mathbb{C}(\boldsymbol{u},\alpha) = \mathbb{C}_0$. Such a traction-only degradation function is more realistic since it can make a difference between an opening and a closing crack (this idea was introduced in Amor et al. (2009) with a slightly different degradation function). The Hooke's tensor $\mathbb{C}(\boldsymbol{u},\alpha)$ obviously depends on $\boldsymbol{u}$, and furthermore is not even differentiable with respect to $\boldsymbol{u}$. Nevertheless, for the numerical test here, we ignore this dependence and do not take it into account in the adjoint equation.

The same 2D cantilever beam, as in Sect. 8.9.1, is considered for the new degradation function (8.115). Starting from the initialization in Fig. 8.32a, the algorithm converges to the shape of Fig. 8.37. The convergence history, in Fig. 8.38, is slightly smoother than in Fig. 8.33. The optimized shape in Fig. 8.37 is somehow intermediate between those in Fig. 8.32 for linear elasticity and the original isotropic degradation function (8.113).

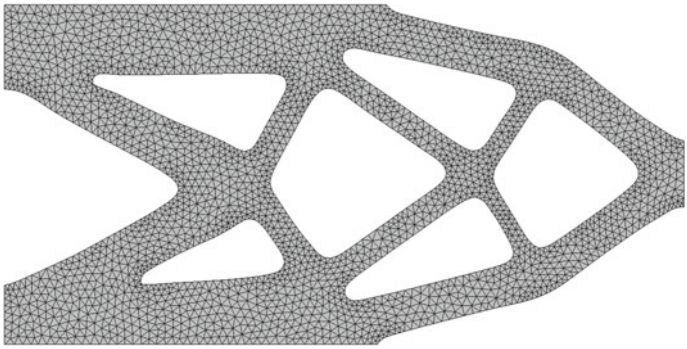
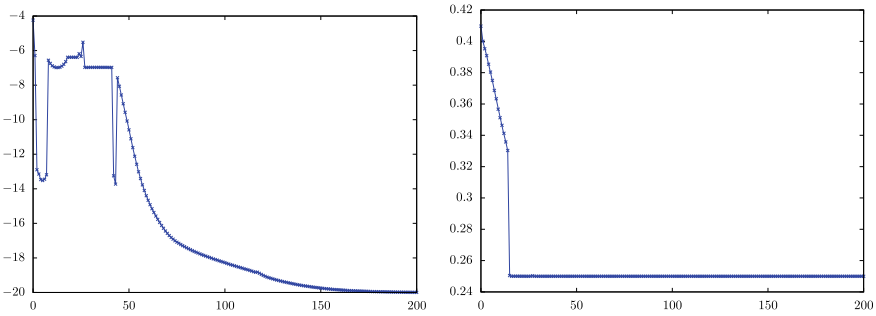


Fig. 8.37 Final mesh for the anisotropic degradation function (8.115)

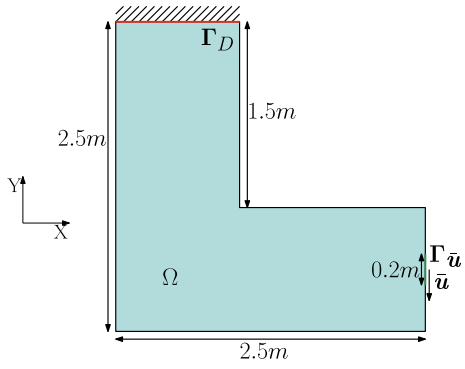


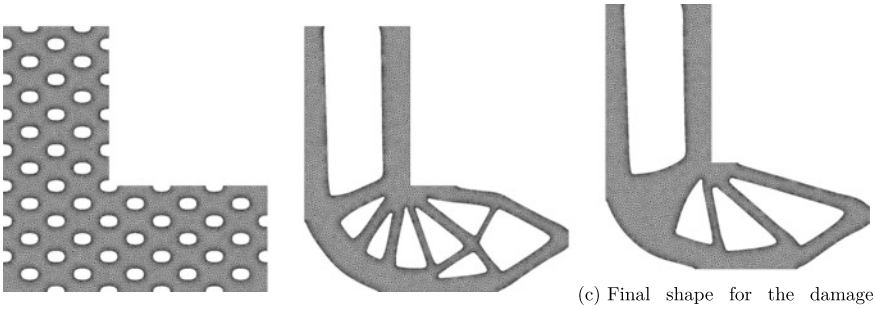
(a) Objective function (1.112) v/s iterations

(b) Volume v/s iterations

Fig. 8.38 Convergence history for the shape of Fig. 8.37

Fig. 8.39 2D L-beam boundary conditions





(a) Initialization mesh (b) Final shape for linear elasticity (c) Final shape for the damage model

Fig. 8.40 Initial and final meshes for the 2D L-beam

8.9.2 2D L-Beam

The last 2D example is an L-beam which is a meaningful test case because of its re-entrant right-angled corner, which is prone to crack initiation. As shown in Fig. 8.39, the L-beam is fixed on its topmost edge, subjected to an imposed displacement,

$$\bar{\boldsymbol{u}}(t) = (0, 1\,\min(1.2t, 1)) \times 10^{-3}\,\mathrm{m}, \quad t \in [0, 1], \tag{8.116}$$

on a small part of its rightmost edge with a vertical force as shown in Fig. 8.39. A target volume $V_f = 2\,\mathrm{m}^2$ is considered. The input parameters for `MMG` are $h_{\min} = 0.01$ m and $h_{\max} = 0.02$ m. From the initial shape in Fig. 8.40a, the minimization of the objective function (8.112) for linear elasticity leads to the final shape in Fig. 8.40b. Contrary to the previous test cases, we now initialize the optimization for the damage model with this shape of Fig. 8.40b, instead of the periodically perforated initialization of Fig. 8.40a. It yields the final shape of Fig. 8.40c. In order to compare the shapes in Fig. 8.40b and in Fig. 8.40c, we perform a damage computation for both shapes with $\bar{\boldsymbol{u}}(t)$, given by (8.116), and plot the damage variable α at the final time in Fig. 8.41: obviously, the shape in Fig. 8.40c does not undergo a crack, unlike the shape in Fig. 8.40b.

8.9.3 Coarse 3D Column

This new test case is a 3D column (see Fig. 8.42a) which is 4 m high, fixed at the bottom (in red) and subjected to an imposed displacement $\bar{\boldsymbol{u}}(t)$ on the top (in yellow). The precise geometrical definition of this column can be found in Gaudillière et al. (2019).

Here again, we investigate the impact of increasing the magnitude of the imposed displacement on the optimized shape and hence consider two functions

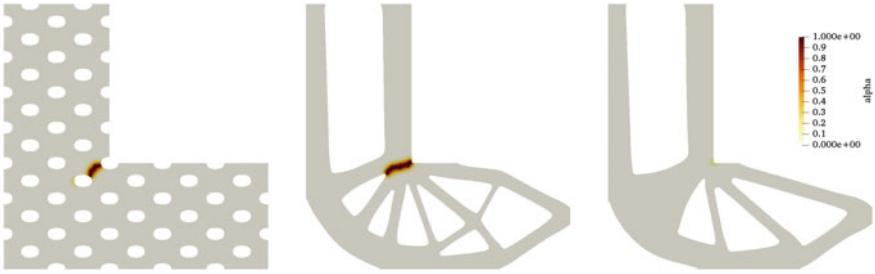


Fig. 8.41 Damage variable α at the final time T plotted for the shapes in Fig. 8.40 (black regions indicate cracks)

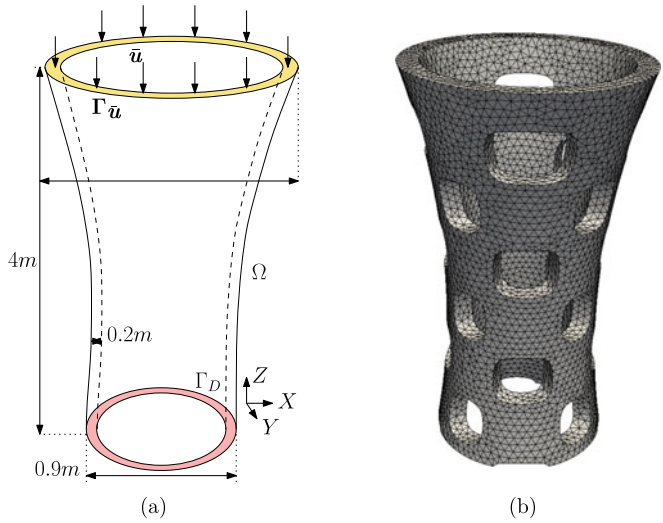


Fig. 8.42 3D column: boundary conditions (left), initial shape (right)

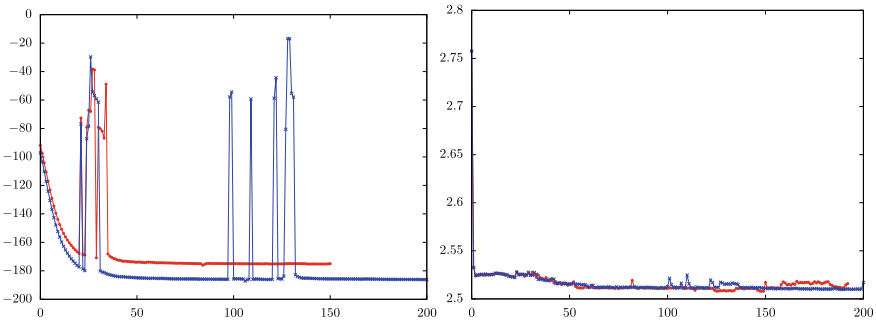
$$\bar{\boldsymbol{u}}(t) = (0, 0, 2.88 \min(1.2t, 1)) \times 10^{-4}\,\mathrm{m}, \quad t \in [0, 1]s, \quad \text{and} \tag{8.117}$$

$$\bar{\boldsymbol{u}}(t) = (0, 0, 2.97 \min(1.2t, 1)) \times 10^{-4}\,\mathrm{m}, \quad t \in [0, 1]s. \tag{8.118}$$

The input parameters of the remesher MMG are $h_{\min} = 0.02\,\mathrm{m}$ and $h_{\max} = 0.04\,\mathrm{m}$. Since the characteristic length (using the formula (8.106)) is $\ell = 0.0375\,\mathrm{m}$, the condition (8.107) is violated. Instead of choosing a smaller $h_{\max}$, we increase the characteristic length to $\ell = 0.075\,\mathrm{m}$ to ensure that the condition (8.107) is satisfied. The target volume is $2.5\,\mathrm{m}^3$. The shape is initialized as shown in Fig. 8.42. The final shape obtained for linear elasticity is plotted in Fig. 8.43a. For the damage model, it is plotted in Fig. 8.43b, c for the imposed displacement (8.117) and (8.118), respectively. Clearly again, we see that a slight increase in the imposed displacement results in a very different optimized shape for the damage model (Figs. 8.44 and 8.45).

(a) Linear elasticity (b) Imposed displacement (1.117) (c) Imposed displacement (1.118)

Fig. 8.43 Final shapes obtained from the initialization in Fig. 8.42a for different imposed displacements

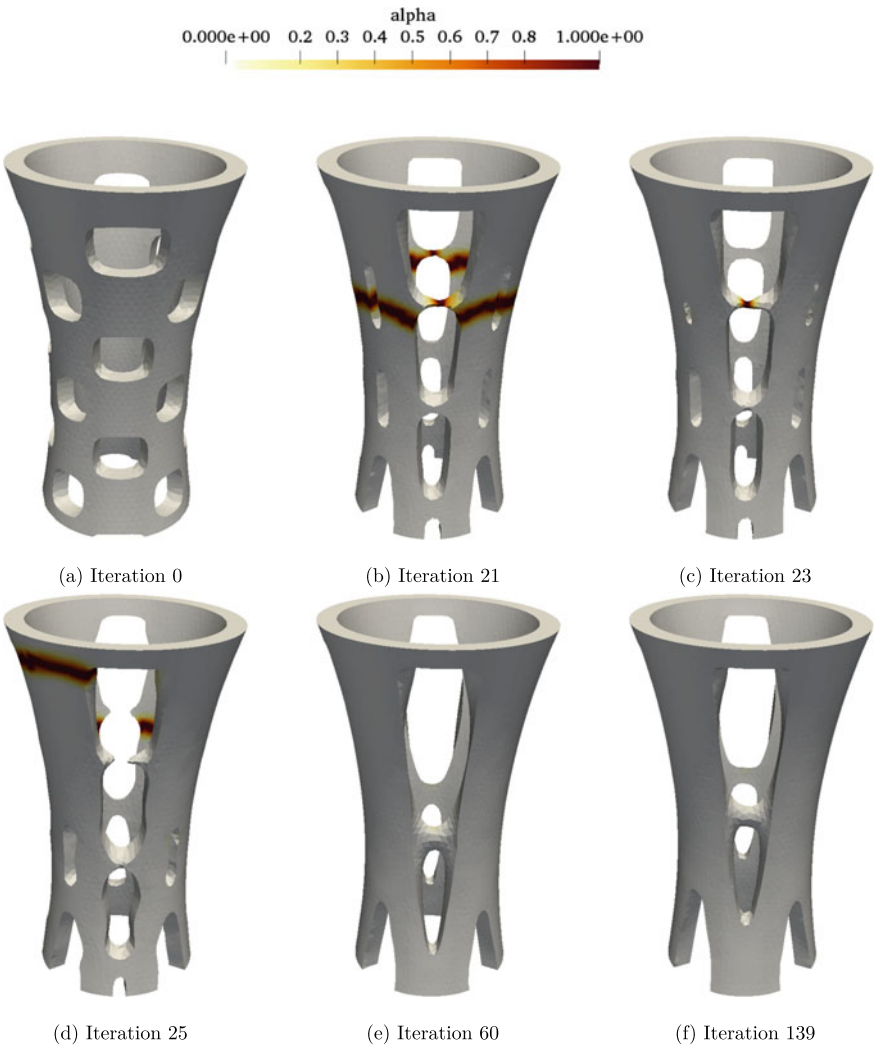


(a) Objective function (1.112) v/s iterations (b) Volume v/s iterations

Fig. 8.44 Convergence history for the shapes in Fig. 8.43b (in red) and Fig. 8.43c (in blue)

We compare the performance of the three shapes in Fig. 8.43 for the same damage model and for the same linear elasticity system with the imposed displacement (8.118) and plot the objective function (8.112) values in Table 8.5. As can be expected, the shapes optimized for damage have much better performances with the damage model. But, surprisingly, the performance of the three shapes for linearized elasticity is very similar (the optimal shape for linear elasticity is less than a fraction of percent better than the two other ones). The damage variable α for those three shapes is plotted in Fig. 8.46. As one can check in Fig. 8.46, the shapes in Fig. 8.43b, c do not undergo a crack whereas the shape in Fig. 8.43a does.

Remark 9 The damage model-based optimization is significantly more expensive than linear elasticity-based optimization. We use parallel computing with 40 processors (see Sect. 8.8.5.5 for more details) to compare the two methods. The complete



(a) Iteration 0 (b) Iteration 21 (c) Iteration 23

(d) Iteration 25 (e) Iteration 60 (f) Iteration 139

Fig. 8.45 Damage variable α at the final time T plotted for several optimization iterations for the shape in Fig. 8.43b (black regions indicate cracks)

Table 8.5 Comparison between shapes in Fig. 8.43a–c for the imposed displacement (8.118)

	Fig. 8.43a	Fig. 8.43b	Fig. 8.43c
Objective (8.112) for linear elasticity	−183.47	−183.65	−186.83
Objective (8.112) for damage model	−179.75	−186.49	−186.13

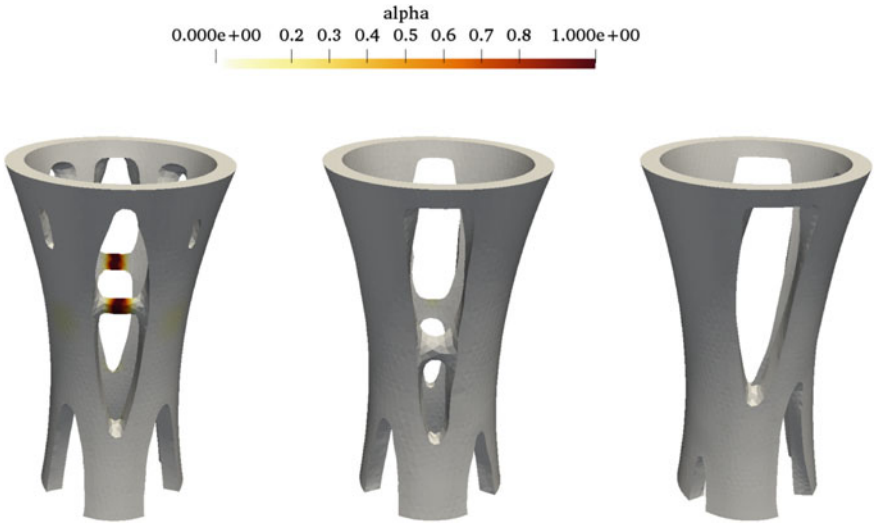


Fig. 8.46 Damage variable α at the final time T for the optimized shapes in Fig. 8.43 and the imposed displacement (8.118) (black regions indicate cracks)

optimization of the 2D cantilever beam (in Sect. 8.8.5.1) takes about 2 min for linear elasticity versus 90 min for the damage model. The complete optimization of the 3D coarse column example (in Sect. 5.8) takes 100 min for linear elasticity versus 6 h for the damage model.

8.9.4 Fine 3D Column

Eventually, we revisit the same column, studied above, but with a highly refined mesh. The goal of this example is to show that our optimization approach is amenable to high performance computing (HPC). The details on the parallel implementation are given in Sect. 8.8.5.5. The column is subjected to the same imposed displacement (8.118). The input parameters of the remesher `MMG` are $h_{\min} = 0.015$ m and $h_{\max} = 0.03$ m. It implies that the initial mesh has 243,641 vertices (1,359,805 tetrahedra) and the final mesh has 241,852 vertices (1,365,125 tetrahedra). The characteristic length, given by (8.106), is $\ell = 0.0375$ m, satisfying condition (8.107). The target volume is again chosen to be $2.5\,\text{m}^3$. An incremental residual strategy is used, where the residual stiffness is defined with $\mathbb{C}_{\text{res}} = \kappa_i \mathbb{C}_0$, and the residual stiffness parameter is defined as

$$\kappa = \frac{1}{2}\left(1 - \frac{i}{\mathcal{N}}\right)^8,$$

(a) Initialization

(b) Final shape obtained for linear elasticity

(c) Final shape obtained for damage model

Fig. 8.47 Column shapes for a fine mesh and the imposed displacement (8.118)

(a) Iteration 15 (b) Iteration 17 (c) Iteration 18

(d) Iteration 43 (e) Iteration 83 (f) Iteration 200

Fig. 8.48 Damage variable α at the final time T plotted for several optimization iterations for the shape in Fig. 8.47 (black regions indicate cracks)

where i is the iteration number and $\mathcal{N}$ is the total number of iterations. This incremental approach is preferred in order to ensure that the intermediate shapes do not undergo brittle fracture too often. The mesh is so fine that the time or descent step τ, given by (8.30), is very small and the convergence is too slow. Therefore, for this test case, we multiply it by a factor of 2, namely $\tau = h_{\min}/C_i$. The initial shape (with a rich topology) can be seen in Fig. 8.47a. The final shape obtained for linear elasticity is plotted in Fig. 8.47b and, for the damage model, is plotted in Fig. 8.47c. The damage variable α is plotted for some intermediate shapes in Fig. 8.48. Actually, there

were only 3 intermediate shapes, which were cracked. Our incremental approach of the residual stiffness was thus not able to eliminate completely the appearance of cracks during the optimization process. The final shape obtained is very similar to the one obtained for linear elasticity in Fig. 8.47b. The total computational time for this optimization was 7 d.

8.10 Conclusion

This chapter provides a brief literature review on the existing methods of shape optimization, while delineating on the level-set method with body-fitted remeshing. The Céa's method is used for computing the shape derivative for linear elasticity and numerical results are presented in 2D and 3D. Compared to the classical level-set method on a fixed mesh, the level-set method with body-fitted remeshing renders a simpler topology, which is more intuitive and easier to fabricate (see Sect. 8.5).

Section 8.6 presents topology optimization for plasticity with linear kinematic and linear isotropic hardening. The non-differentiable plasticity model is rendered differentiable via penalization and regularization and the shape derivative is computed using Céa's method. From the numerical results in 2D and 3D, one can observe that the shapes obtained for plasticity are indeed very similar to ones obtained for linear elasticity. When one minimizes the compliance of the structure, one naturally expects the optimal structure to undergo less plastic deformation. But in reality, the optimal shapes are the ones in which the plastic flow occurs everywhere (e.g., see Fig. 8.21). This helps the structure as plastic flow is followed by hardening, which causes an increase in the yield limit. This increase renders the structure more resistant to an increasing external loading.

Section 8.8 resorts to a gradient damage model to perform topology optimization of structures in order to prevent fractures. The important ingredients of the presented approach are a backtracking algorithm for solving the damage model, the use of body-fitted meshes and a parallel implementation to mitigate the high CPU cost in 3D. Indeed, the proposed optimization approach is more expensive by approximately one order of magnitude than a traditional compliance minimization in linear elasticity. Nevertheless, the methodology is able to deliver crack-free optimal structures (for reasonable choices of the loading conditions and material parameters). In some cases, the crack-free optimal structures resemble the ones obtained by optimizing for linear elasticity but the latter ones can undergo a crack in many instances (see, e.g., the 2D cantilever in Sect. 8.9.1, the 2D L-beam in Sect. 8.9.2, the 3D coarse column in Sect. 8.9.3).

References

Allaire G (2007) Conception optimale de structures. Math Appl 58:280
Allaire G (2012) Shape optimization by the homogenization method. Springer Science & Business Media, vol 146
Allaire G, Dapogny C, Frey P (2011) Topology and geometry optimization of elastic structures by exact deformation of simplicial mesh. Comptes Rendus Mathematique 349(17–18):999–1003
Allaire G, Dapogny C, Frey P (2014) Shape optimization with a level set based mesh evolution method. Comput Methods Appl Mechan Eng 282:22–53
Allaire G, Dapogny C, Jouve F (2020) Shape and topology optimization. In: Bonito A, Nochetto R (eds) Geometric partial differential equations, part II. Handbook of numerical analysis, vol 22, pp 1–132
Allaire G, Dapogny C, Jouve F (2021) Shape and topology optimization, volume 22 of Handbook of numerical analysis. In: Bonito A, Nochetto R (eds) Geometric partial differential equations, part II. Elsevier, pp 1–132
Allaire G, Delgado G (2016) Stacking sequence and shape optimization of laminated composite plates via a level-set method. J Mechan Phys Solids 97:168–196
Allaire G, Jouve F, Michailidis G (2016) Molding direction constraints in structural optimization via a level-set method. In: Variational analysis and aerospace engineering, pp 1–39
Allaire G, Jouve F, Toader AM (2002) A level-set method for shape optimization. Comptes Rendus Mathematique 334(12):1125–1130
Allaire G, Pantz O (2006) Structural optimization with Freefem++. Struct Multidiscipl Optim 32(3):173–181
Ambati M, Gerasimov T, De Lorenzis L (2015) A review on phase-field models of brittle fracture and a new fast hybrid formulation. Comput Mechan 55(2):383–405
Ambrosio L, Tortorelli VM (1990) Approximation of functional depending on jumps by elliptic functional via t-convergence. Commun Pure Appl Math 43(8):999–1036
Ambrosio L, Tortorelli VM (1992) On the approximation of free discontinuity problems. In: Bollettino dell'unione matematica italiana, pp 105–123
Amir O (2013) A topology optimization procedure for reinforced concrete structures. Comput Struct 114:46–58
Amor H, Marigo JJ, Maurini C (2009) Regularized formulation of the variational brittle fracture with unilateral contact: numerical experiments. J Mechan Phys Solids 57:1209–1229
Balay S, Abhyankar S, Adams M, Brown J, Brune P, Buschelman K, Dalcin L, Dener A, Eijkhout V, Gropp W, Karpeyev D, Kaushik D, Knepley M, May D, Curfman McInnes L, Mills R, Munson T, Rupp K, Sanan P, Smith B, Zampini S, Zhang H, Zhang H (2019) PETSc users manual. Technical report
Barbarosie C, Lopes S (2011) A generalized notion of compliance. Comptes Rendus Mécanique 339(10):641–648
Bažant P, Pijaudier-Cabot G (1989) Measurement of characteristic length of nonlocal continuum. J Eng Mechan 115(4):755–767
Benallal A, Marigo JJ (2006) Bifurcation and stability issues in gradient theories with softening. Model Simul Mater Sci Eng 15(1):S283
Bendsoe M, Sigmund O (2013) Topology optimization: theory, methods, and applications. Springer Science & Business Media
Bendsøe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. Comput Methods Appl Mechan Eng 71(2):197–224
Bendsøe MP, Sigmund O (1999) Material interpolation schemes in topology optimization. Archive Appl Mechan 69(9):635–654
Bendsoe MP, Sigmund O (2004) Topology optimization: theory, methods, and applications, 2nd edn. Springer, Berlin
Bogomolny M, Amir O (2012) Conceptual design of reinforced concrete structures using topology optimization with elastoplastic material modeling. Int J Numer Methods Eng 90(13):1578–1597

Bourdin B (2007) Numerical implementation of the variational formulation for quasi-static brittle fracture. Interf Free Bound 9(3):411–430
Bourdin B, Francfort GA, Marigo JJ (2008) The variational approach to fracture. J Elast 91(1–3):5–148
Buhl T, Pedersen C, Sigmund O (2000) Stiffness design of geometrically nonlinear structures using topology optimization. Struct Multidiscipl Optim 19(2):93–104
Bui C, Dapogny C, Frey P (2012) An accurate anisotropic adaptation method for solving the level set advection equation. Int J Numer Methods Fluids 70(7):899–922
Burger M (2003) A framework for the construction of level set methods for shape optimization and reconstruction. Interf Free Bound 5(3):301–329
Burman E, Claus S, Hansbo P, Larson MG, Massing A (2015) CutFEM: discretizing geometry and partial differential equations. Int J Numer Methods Eng 104(7):472–501
Chambolle A (2004) An approximation result for special functions with bounded deformation. J de mathématiques pures et appliquées 83(7):929–954
Chambolle A, Francfort GA, Marigo JJ (2009) When and how do cracks propagate? J Mechan Phys Solids 57(9):1614–1622
Chambolle A, Francfort GA, Marigo JJ (2010) Revisiting energy release rates in brittle fracture. J Nonlinear Sci 20(4):395–424
Chambolle A, Giacomini A, Ponsiglione M (2008) Crack initiation in brittle materials. Arch Ration Mech Anal 188(2):309–349
Chenais D (1975) On the existence of a solution in a domain identification problem. J Math Anal Appl 52(2):189–219
Coleman BD, Gurtin ME (1967) Thermodynamics with internal state variables. J Chem Phys 47(2):597–613
Comi C, Perego U (2001) Fracture energy based bi-dissipative damage model for concrete. Int J Solids Struct 38(36–37):6427–6454
Costa G, Montemurro M, Pailhès J (2019) NURBS hyper-surfaces for 3D topology optimization problems. In: Mechanics of advanced materials and structures, pp 1–20
Céa J (1986) Conception optimale ou identification de formes, calcul rapide de la dérivée directionnelle de la fonction coût. ESAIM: Math Model Numer Anal 20(3):371–402
Da D, Yvonnet J (2020) Topology optimization for maximizing the fracture resistance of periodic quasi-brittle composites structures. Materials 13(15):3279
Dambrine M, Kateb D (2010) On the ersatz material approximation in level-set methods. ESAIM: Control Optim Calculus Var 16(3):618–634
Dapogny C (2013) Shape optimization, level set methods on unstructured meshes and mesh evolution. PhD thesis, Paris, p 6
Dapogny C, Dobrzynski C, Frey P (2014) Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems. J Comput Phys 262:358–378
Dapogny C, Frey P (2012) Computation of the signed distance function to a discrete contour on adapted triangulation. Calcolo 49(3):193–219
De Gournay F (2006) Velocity extension for the level-set method and multiple eigenvalues in shape optimization. SIAM J Control Optim 45(1):343–367
De Gournay F, Allaire G, Jouve F (2008) Shape and topology optimization of the robust compliance via the level set method. ESAIM: Control Optim Calculus Var 14(1):43–70
Desai J (2021) Topology optimization in contact, plasticity, and fracture mechanics using a level-set method. Phd thesis, to appear, Université de Paris
Desai J, Allaire G, Jouve F (2022) Topology optimization of structures undergoing brittle fracture. J Comput Phys 458:111048
Desai J, Allaire G, Jouve F, Mang C (2021) Topology optimization in quasi-static plasticity with hardening using a levet-set method. preprint on https://hal.archives-ouvertes.fr/hal-03259408/
Feppon F, Allaire G, Bordeu F, Cortial J, Dapogny C (2019) Shape optimization of a coupled thermal fluid-structure problem in a level set mesh evolution framework. SeMA J 76(3):413–458

Feppon F, Allaire G, Dapogny C, Jolivet P (2020) Topology optimization of thermal fluid-structure systems using body-fitted meshes and parallel computing. J Comput Phys 417:109574
Francfort GA, Marigo JJ (1998) Revisiting brittle fracture as an energy minimization problem. J Mechan Phys Solids 46(8):1319–1342
Gaudillière N, Duballet R, Bouyssou C, Mallet A, Roux Ph, Zakeri M, Dirrenberger J (2019) Chapter 3–building applications using lost formworks obtained through large-scale additive manufacturing of ultra-high-performance concrete. In: Sanjayan JG, Nazari A, Nematollahi B (eds) 3D concrete printing technology. Butterworth-Heinemann, pp 37–58
Gomes HM (2011) Truss optimization with dynamic constraints using a particle swarm algorithm. Exp Syst Appl 38(1):957–968
Griffith AA (1921) VI: the phenomena of rupture and flow in solids. Philos Trans Roy Soc Lond 221(582–593):163–198
Han W, Reddy BD (2013) Plasticity: mathematical theory and numerical analysis. Number XVI in 9, 2 edn. Springer, New York
Han W, Reddy BD, Schroeder GC (1997) Qualitative and numerical analysis of quasi-static problems in elastoplasticity. SIAM J Numer Anal 34(1):143–177
Hassani B, Hinton E (1998) A review of homogenization and topology optimization iii-topology optimization using optimality criteria. Comput Struct 69(6):739–756
Hassani B, Hinton E (1998) A review of homogenization and topology optimization i-homogenization theory for media with periodic structure. Comput Struct 69(6):707–717
Hecht F (2012) New development in FreeFEM++. J Numer Math 20(3–4):251–266
Henrot A, Pierre M (2018) Shape variation and optimization. EMS Tracts Math 28
Huang X, Xie YM (2009) Bi-directional evolutionary topology optimization of continuum structures with one or multiple materials. Comput Mechan 43(3):393–401
James KA, Waisman H (2015) Topology optimization of structures under variable loading using a damage superposition approach. Int J Numer Methods Eng 101(5):375–406
James KA, Waisman H (2015) Topology optimization of viscoelastic structures using a time-dependent adjoint method. Comput Methods Appl Mechan Eng 285:166–187
Jolivet P (2020) FreeFEM tutorial. http://jolivet.perso.enseeiht.fr/FreeFem-tutorial/
Kang Z, Liu P, Li M (2017) Topology optimization considering fracture mechanics behaviors at specified locations. Struct Multidiscipl Optim 55(5):1847–1864
Karypis G, Kumar V (1997) METIS: a software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices
Kato J, Hoshiba H, Takase S, Terada K, Kyoya T (2015) Analytical sensitivity in topology optimization for elastoplastic composites. Struct Multidiscipl Optim 52(3):507–526
Klarbring A, Strömberg N (2013) Topology optimization of hyperelastic bodies including non-zero prescribed displacements. Struct Multidisciplin Optim 41(3):37–48
Lazarov BS, Sigmund O (2011) Filters in topology optimization based on Helmholtz-type differential equations. Int J Numer Methods Eng 86(6):765–781
Li L, Zhang G, Khandelwal K (2017) Design of energy dissipating elastoplastic structures under cyclic loads using topology optimization. Struct Multidiscipl Optim 56(2):391–412
Lindgaard E, Dahl J (2013) On compliance and buckling objective functions in topology optimization of snapthrough problems. Struct Multidiscipl Optim 47(3):409–421
Liu P, Luo Y, Kang Z (2016) Multi-material topology optimization considering interface behavior via XFEM and level set method. Comput Methods Appl Mechan Eng 308:113–133
Marigo JJ (1989) Constitutive relations in plasticity, damage and fracture mechanics based on a work property. Nucl Eng Design 114(3):249–272
Marigo JJ (2000). From Clausius-Duhem and Drucker-Ilyushin inequalities to standard materials. Continuum Thermomechan 289–300
Martínez-Frutos J, Herrero-Pérez D (2018) Evolutionary topology optimization of continuum structures under uncertainty using sensitivity analysis and smooth boundary representation. Comput Struct 205:15–27

Martínez-Frutos J, Herrero-Pérez D, Kessler M, Periago F (2016) Robust shape optimization of continuous structures via the level set method. Comput Methods Appl Mechan Eng 305(Suppl C):271–291

Maury A, Allaire G, Jouve F (2018) Elasto-plastic shape optimization using the level set method. SIAM J Control Optim 56(1):556–581

Maute K, Schwarz S, Ramm E (1998) Adaptive topology optimization of elastoplastic structures. Struct Optim 15(2):81–91

Miehe C, Welschinger F, Hofacker M (2010) Thermodynamically consistent phase-field models of fracture: variational principles and multi-field fe implementations. Int J Numer Methods Eng 83(10):1273–1311

Mignot F (1976) Contrôle dans les inéquations variationelles elliptiques. J Funct Anal 22(2):130–185

Mignot F, Puel JP (1984) Optimal control in some variational inequalities. SIAM J Control Optim 22(3):466–476

Mohammadi B, Pironneau O (2001) Applied shape optimization for fluids. Clarendon Press, Oxford

Nakshatrala P, Tortorelli DA (2016) Topology optimization of multiscale elastoviscoplastic structures. J Numer Methods Eng 106(6):430–453

Nguyen QS (1987) Bifurcation and post-bifurcation analysis in plasticity and brittle fracture. J Mechan Phys Solids 35(3):303–324

Osher S, Fedkiw R (2003) Signed distance functions. Level set methods and dynamic implicit surfaces. Springer, Berlin, pp 17–22

Osher S, Fedkiw R (2006) Level set methods and dynamic implicit surfaces. Springer Science & Business Media, vol 153

Osher S, Santosa F (2001) Level set methods for optimization problems involving geometry and constraints: I. Frequencies of a two-density inhomogeneous drum. J Comput Phys 171(1):272–288

Pedersen P (1991) On thickness and orientational design with orthotropic materials. Struct Optim 3(2):69–78

Pham K, Amor H, Marigo JJ, Maurini C (2011) Gradient damage models and their use to approximate brittle fracture. Int J Damage Mechan 20(4):618–652

Reddy BD, Martin JB (1994) Internal variable formulations of problems in elastoplasticity: constitutive and algorithmic aspects. Appl Mechan Rev 47(9):429–456

Seo YD, Kim HJ, Youn SK (2010) Isogeometric topology optimization using trimmed spline surfaces. Comput Methods Appl Mechan Eng 199(49–52):3270–3296

Sethian JA, Wiegmann A (2000) Structural boundary design via level set and immersed interface methods. J Comput Phys 163(2):489–528

Sigmund O (2007) Morphology-based black and white filters for topology optimization. Struct Multidiscipl Optim 33(4–5):401–424

Sigmund O, Maute K (2013) Topology optimization approaches. Struct Multidiscipl Optim 48(6):1031–1055

Simo JC, Hughes TJR (2006) Computational inelasticity. Springer Science and Business Media, vol 7

Simo JC, Taylor RL (1986) A return mapping algorithm for plane stress elastoplasticity. Int J Numer Methods Eng 22(3):649–670

Sokolowski J, Zochowski A (1999) On the topological derivative in shape optimization. SIAM J Control Optim 37(4):1251–1272

Sokolowski J (1992) Zolésio JP (1992) Introduction to shape optimization. Springer, Berlin

Tanné E, Li T, Bourdin B, Marigo JJ, Maurini C (2018) Crack nucleation in variational phase-field models of brittle fracture. J Mechan Phys Solids 110:80–99

Villanueva CH, Maute K (2014) Density and level set-XFEM schemes for topology optimization of 3-D structures. Comput Mechan 54(1):133–150

Wallin M, Ivarsson N, Ristinmaa M (2015) Large strain phase-field based multi-material topology optimization. Int J Numer Methods Eng 104(9):887–904

Wallin M, Jönsson V, Wingren E (2016) Topology optimization based on finite strain plasticity. Struct Multidiscipl Optim 54(4):783–793

Wang F, Lazarov BS, Sigmund O (2011) On projection methods, convergence and robust formulations in topology optimization. Struct Multidiscipl Optim 43(6):767–784

Wang M, Wang X, Guo D (2003) A level set method for structural topology optimization. Comput Methods Appl Mechan Eng 192(1–2):227–246

Wu C, Fang J, Zhou S, Zhang Z, Sun G, Steven GP, Li Q (2020) Level-set topology optimization for maximizing fracture resistance of brittle materials using phase-field fracture model. Int J Numer Methods Eng 121(13):2929–2945

Xie YM, Steven GP (1993) A simple evolutionary procedure for structural optimization. Comput Struct 49(5):885–896

Yamada T, Izui K, Nishiwaki S, Takezawa A (2010) A topology optimization method based on the level set method incorporating a fictitious interface energy. Comput Methods Appl Mechan Eng 199(45–48):2876–2891

Chapter 9
Design of Quarter Car Model for Active Suspension System and Control Optimization

Vyomkumar P. Patel, Vijaykumar S. Jatti, and Vinaykumar S. Jatti

Abstract Quarter car model can be used to approximate a response of the suspension systems to obtain a behavioral relationship between the suspension and the body. The major aim of the quarter is to obtain a stable working control system for achieving three major control states. The three states include the passenger's comfort design, the road handling design, and a balanced design. The objective is to obtain the three control systems by using the H-infinity synthesis and designing a controller based on the defined states and control inputs of the car. The designed controller can therefore be optimized for the account of uncertainty. The use of μ-synthesis for the optimization of the designed controller and the balanced design is considered for the optimization. For the analysis of the control system, initially the system is given a disturbance of 7 cm. For the optimization, the input was increased and a bump of 10 cm was considered for achieving a greater disturbance and gains in the measurements. The bode plots recorded for the calculation would verify the performance of the control system which would take the quarter car model as a state space and the controller for the feedback module for the control of the suspension system.

Keywords Quarter car · Active suspension system · Bode plots · MATLAB · Control optimization

V. P. Patel
D. Y. Patil College of Engineering, Akurdi, Pune, Maharashtra, India

V. S. Jatti (✉) · V. S. Jatti
Symbiosis Institute of Technology (SIT), Symbiosis International University (SIU), Lavale, Pune, Maharashtra 412115, India
e-mail: vijaykumar.jatti@sitpune.edu.in

V. S. Jatti
e-mail: vinay.jatti@sitpune.edu.in

I. R. Kale and A. Sadollah (eds.), *Optimization Methods for Structural Engineering*, Engineering Optimization: Methods and Applications,
https://doi.org/10.1007/978-981-99-2378-6_9

9.1 Introduction

Mathematical modeling deals with representing a system or a component of system mathematically. The key for representing a system mathematically is behavior study of the system and its components. The obtained mathematical model shall behave the same as the actual model under the usual circumstances. The behavior of the components in the system serves as the building blocks for the response of the system. The behavior of each components their connections with other system and their control of gains with respect to the system objective lead to a successful mathematical model. The mathematical model of a component is usually determined by the behavior of the component; i.e., if the components work with the displacement gain (spring), it generally works linearly. The dampers on the other hand works with the velocity gains, and thus the dampers work in a parabolic response. The components' performance may be different from the desired output since the practical circumstance and conditions play a vital role for the actual response. Mathematical models are used for representing systems and its corresponding components in mathematical format. The models are a great way in determining the system definition and obtain a more calculative approach. The system in the simulation system can be edited for a change in any of the component, and the simulation results can provide a similar result of the actual change of response in the system.

The mathematical modeling of a system usually starts with components and their respective behavior; usually, an equation or a state space is used for the definition of the components. The components involved in the suspension system are spring which responds linearly, the damper which responds parabolically, and the masses which vibrates with the road disturbance. As we solve the mathematical model of the system, all the corresponding components shall be tested for achieving the response of the suspension system.

The mathematical model of a **spring** can be determined as a **linear function**.

$$\boldsymbol{F = kx}$$

The mathematical model of a **damper** can be defined as **first-order differential**.

$$\boldsymbol{F = c\frac{\partial x}{\partial t}}$$

The mathematical model of the **masses** falls under the **second-order differential**.

$$\boldsymbol{F = m\frac{\partial^2 x}{\partial t^2}}$$

As we sum up all the terms we obtain an equation for the suspension system which can be used for the analysis.

$$m\frac{\partial^2 x}{\partial t^2} + c\frac{\partial x}{\partial t} + kx = F$$

The actuator can be represented in the cumulative force in the ***rhs*** of the equation.

Thite et al. (2012) developed a refined model for quarter car analysis. The research focused around the discomfort caused due to the vibration. Sharaf et al. (2013) presented quarter car analytical model and depicted the effect of vehicle speeds on ride comfort. Yu et al. (2019) illustrated experiments on series active variable geometry suspension system and modeled using H-infinity control scheme. Ahmed et al. (2021) employed linear quadratic regulator and fuzzy PID controllers to get optimized active suspension system of a quarter car. Hassaan et al. (2015) developed a passive suspension-based quarter model car and analyzed using MATLAB. Had et al. (1992) developed a semi-active suspension system based on 2 DoF with actual road conditions simulation. Paliwal et al. (2020) presented effect of different road profiles on nonlinear quarter car model. Allamraju et al. (2016) depicted the numerical modeling of quarter car model suspension system to get modal parameters. Ebrahimi-Nejad et al. (2020) utilized sports car to develop a modal equations using Lagrange's equations. Alvarez-Sánchez et al. (2013) used linear mathematical model to have a robust control scheme. Based on the literature review, it is observed that design of car with active suspension system and control optimization is at nascent stage. This study aims to obtain the three control systems by using the H-infinity synthesis and designing a controller based on the defined states and control inputs of the car.

9.2 Methodology

The simulation of the active suspension system in the simulation system requires defining all the components in the system and defining the design setup. A quarter car model is generally designed by defining the individual components and setting up gains from all the components according to their connection to the car and suspension system. The normally used suspensions usually have a spring and a damper (Zeng 2019). These are usually connected to the car body on one end and to the wheel on the other end. Under the components, the specific characteristics like the spring constant and the damping coefficient of the damper can be changed to achieve the desired results. Under the modern suspension system, the same configuration observes an increase in the installation of the actuator which is normally hydraulic and it is controlled by a feedback controller. The components need to be individually defined in the MATLAB for the generation of the state space of the quarter car model, as depicted in Fig. 9.1. The mass of the car body is defined in terms of **mb** (in kg). The mass of the wheel assembly is denoted by **mw** (in kg). The spring and damper are defined by **ks** and **bs**, respectively. The compressibility of the tire is accounted as **kt**. The linear first-order measurements and the disturbances are defined as **xb** (body travel), **xw** (wheel travel), and **r** (road disturbance). All the linear measurements are in meters. The hydraulic actuator discussed earlier is denoted as the force that is used

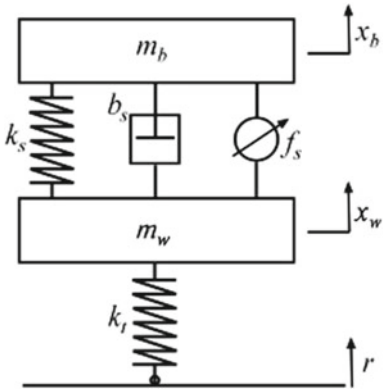


Fig. 9.1 Mathematical model of active suspension system

to control the system as **fs** (in kN). The actuator serves as the active component of the suspension system.

The values of the components can be referred as given below

$$(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) = (\boldsymbol{x}_b, \dot{\boldsymbol{x}}_b, \boldsymbol{x}_w, \dot{\boldsymbol{x}}_w)$$

The state-space equations for the quarter car model are defined as listed below

$$\dot{\boldsymbol{x}}_1 = \boldsymbol{x}_2$$

$$\dot{\boldsymbol{x}}_2 = -\left(\frac{1}{\boldsymbol{m_b}}\right)\left[\boldsymbol{k_s}(\boldsymbol{x}_1 - \boldsymbol{x}_3) + \boldsymbol{b_s}(\boldsymbol{x}_2 - \boldsymbol{x}_4) - 10^3 \boldsymbol{f_s}\right]$$

$$\dot{\boldsymbol{x}}_1 = \boldsymbol{x}_2$$

$$\dot{\boldsymbol{x}}_4 = -\left(\frac{1}{\boldsymbol{m_w}}\right)\left[\boldsymbol{k_s}(\boldsymbol{x}_1 - \boldsymbol{x}_3) + \boldsymbol{b_s}(\boldsymbol{x}_2 - \boldsymbol{x}_4) - \boldsymbol{k_t}(\boldsymbol{x}_3 - \boldsymbol{r}) - 10^3 \boldsymbol{f_s}\right]$$

Further, defining all the elements in a state matrices which would aid in providing with the control of the quarter car model. After defining the matrices of the state space, we would define the state space of the quarter car model and define the inputs and outputs of the system. The transfer function related to actuator and chassis travel makes zero with the imaginary axis with natural frequency, i.e., **tire-hop frequency**. Similarly, actuator and the suspension travel give a zero with the imaginary axis at the natural frequency known as the **rattle space frequency**. The road disturbance is one of the factors that affect the motion of a vehicle, the suspension primarily targets the road disturbance, and the required control is used for achieving the specific design target. The road disturbance usually provides an input to the car state space with the deflection in the suspension, and the control is achieved by the force applied by the actuator. There are zeros of the imaginary axes that affect the feedback control and

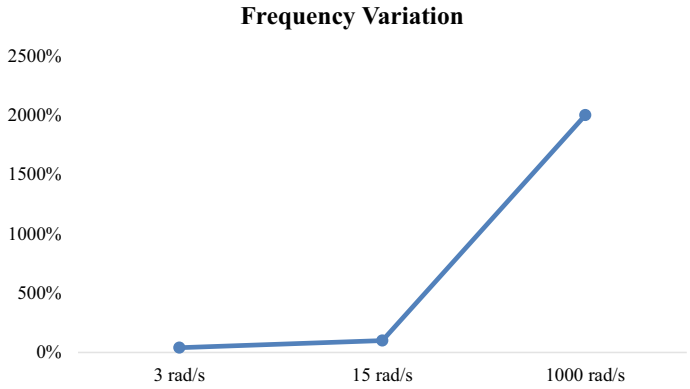


Fig. 9.2 Variations in the actuator with respect to the frequency

would cause not to improve the response from road disturbance to the acceleration of the body that is obtained at the tire-hop frequency. The similar would be the case of achieving the control of the suspension travel at the rattle space frequency. There is a relation that defines that the wheel position follows the road surface $x_w = x_b - \text{sd}$. Under the conditions of lower frequencies, i.e., less than 5 rad/s, there is observed a trade-off between passenger comfort and suspension travel. The actuator is used for the force application to the car body, and the wheel is usually a hydraulic actuator. The actuator is placed between car body $\boldsymbol{m_b}$ and the wheel $\boldsymbol{m_w}$. The hydraulic actuator dynamics can be best represented by the first-order transfer function $1/((1 + \boldsymbol{s})/60)$ with a maximum allowable displacement of **0.05 m**. The defined actuator model, but the actuator is named as a nominal model of a true actuator. This is due to the results that may be obtained would be only an approximation of the true actuator. This family of actuators would consist of a nominal model, the uncertainty would be frequency dependent. Figure 9.2 shows the result variations in the actuator with respect to the frequency. The weighting function $\boldsymbol{W}_{\mathbf{unc}}$ can be used to interpret the amount of uncertainty with frequency.

The two major goals of the suspension system are the passenger's comfort and the vehicle handling. The goals refer to the control of the body acceleration and the suspension travel. There are other factors that account for the control design which includes the actual surface of the road, data/measurements received from the sensors, noise of the sensor, and the actual working constraints of the actuator. The H-infinity synthesis algorithms can be used to express the objectives with the help of a single cost function and the target would be to minimize the cost function for achieving the control of the system. The $\boldsymbol{H}_\infty$ controller would use the measurements $\boldsymbol{y}_1$, $\boldsymbol{y}_2$ of the travel of suspension $\boldsymbol{sd}$ and chassis acceleration $\boldsymbol{ab}$ this will be used for obtaining the signal **u**, which will then be used to drive the actuator. There are three external sources of disturbance:

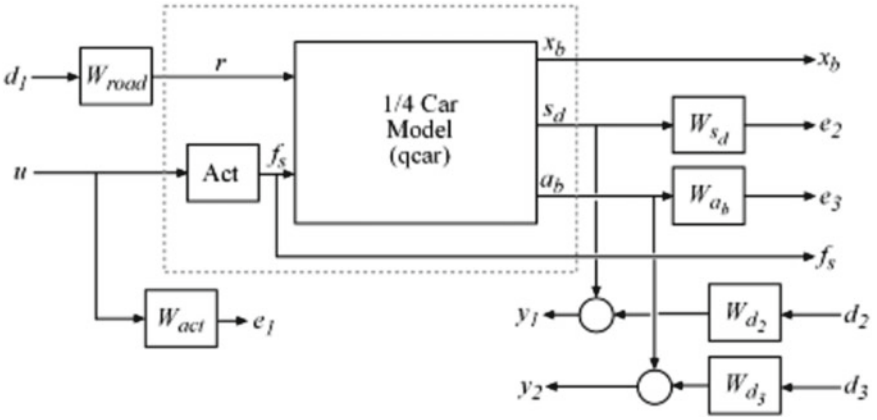


Fig. 9.3 Design setup for active suspension system

- The surface of the road $\boldsymbol{r}$ is returned as a normalized signal $\boldsymbol{d}_1$ which is further classified under the weighting function $\boldsymbol{W_{road}}$. To account for the road bump of height 7 cm, the weight is constant $\boldsymbol{W_{road}} = 0.07$
- The sensors used for the measurement of the current disturbance results with the noise, and the noise of the sensors is accounted as $\boldsymbol{W}_{d2}$ and $\boldsymbol{W}_{d3}$. The weighting functions are used, $\boldsymbol{W}_{d2} = 0.01$ and $\boldsymbol{W}_{d3} = 0.5$ to account for the noise of the sensor. The noise intensity is kept constant as **0.01** and **0.5**. The constant intensity locks the scope of the analysis and a result in a nominal result under the fixed intensity of noise. The analysis can be made realistic by adding a frequency-dependent sensor noise relationship and a weight function that corresponds to the noise dependency can be defined. Figure 9.3 depicts the quarter car model design setup with active suspension system.

After the definition, all the components and their respective signal models corresponding to the weight functions. The road surface signal is accounted for as $\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3$ on a combined weight function for the control signal $\boldsymbol{u}$, suspension travel $\boldsymbol{sd}$, and body acceleration $\boldsymbol{ab}$. The result obtained of the simulation would be in terms of H∞ norm, the greater the system the lesser the control authority. Thus, the H∞ norm can be represented as the impact of the road surface to the car. The result can be summarized as the design of the controller accounting the inputs $\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3$ to error signals $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$. To proceed further, we need to define the weight functions of the design and denote the input and output channels for accounting the interconnection of the different weights defined. We need to use a high-pass filter for the weight of the actuator $\boldsymbol{W_{act}}$ to account for the high-frequency content of the control signal. This would aid in limiting the range of operation of the controller. The need arises for specifically defining the closed-loop gain targets for the road disturbance r to the suspension deflection sd and the body acceleration ab (Gawad 2021; Shao 2020; Elattar et al. 2016; Liu 2019). Due to the actuator uncertainty, the system shall seek the control only below 10 rad/s. The weights W_{sd}, W_{ab} can be

defined as the reciprocals of the comfort and handling goals. To observe the variation between passenger comfort and road handling, we would develop three sets of weights (βW_{sd}, $(1-\beta)W_{ab}$) corresponding to three different variations: comfort ($\beta = 0.01$), balanced ($\beta = 0.5$), and handling ($\beta = 0.99$). The use of connect function for constructing a model `qcaric` of the block diagram. `qcaric` is an amalgamation of three models, one for each design point β. `qcaric` can be interpreted as an uncertain model. The reason for the uncertainty is the presence of the actuator which we earlier termed as an uncertain model and derived only the nominal values for the simulation. The use of `hinfsyn` for computing an H_∞ controller for every value of the blending factor β. Now we need to define closed-loop systems for observing the different results of the road disturbance impact to x_b, s_d, a_b for the passive and active type suspensions. The results obtained reveal that the controllers reduce the suspension deflection and the other control variable below the rattle space frequency, i.e., 21.52 rad/s.

9.3 Results

The frequency of the wheel assembly that is attained due to the road disturbance is called tire-hop frequency and the obtained value after the simulation is 56.27 rad/s. Similarly, the frequency of the car body that is transferred from the suspension to the body is called rattle space frequency. The frequency obtained after the simulation is 21.52 rad/s. Figure 9.4 shows the bode plots for suspension system for open-loop gain from disturbance and actuator force to the acceleration of body and travel of suspension.

Figure 9.5 depicts the comparison of actuator response values with nominal values. Response of the hydraulic actuator with total 20 samples for its comparison with the nominal value.

Figure 9.6 depicts the comparison of open-loop to closed-loop targets obtaining results through bode plots for achieving open-loop targets and closed-loop targets.

Figure 9.7 depicts closed-loop model tests according to design points. The closed-loop target represents the active suspension system, and the reaction of the suspension system to various parameters aids in understanding the working of the suspension system. Figure 9.8 depicts the results for response of suspension system. Figure 9.9 depicts the comparison of design objectives with suspension deflection and control force.

The controller that has been designed in the earlier stages has been working for only three different conditions according to the control required. The major three conditions included the control for comfort, handling, and balanced. Under the balanced condition, both the comfort and handling were observed to be compromised in a considerable manner. The error in this condition is observed due to the actuator model used for the analysis. The actuator model used for the quarter car analysis was only accounting the nominal values, and it served only as an approximation to the true actuator. The model didn't encounter the model errors and uncertainty that may

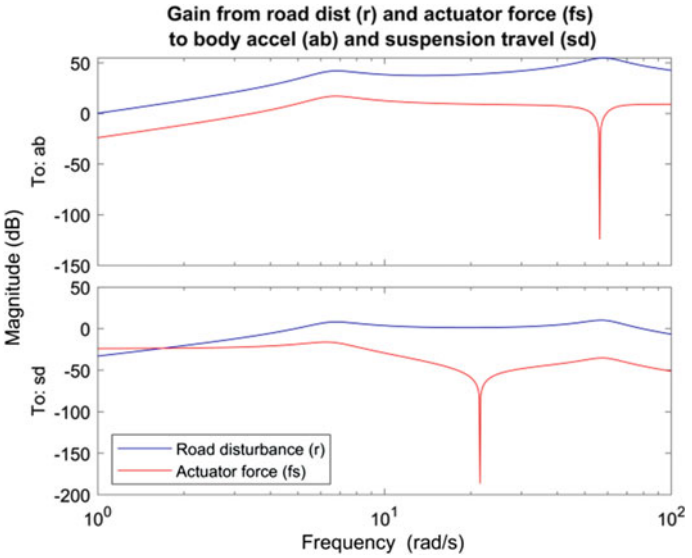


Fig. 9.4 Behavior of passive suspension system with actuator

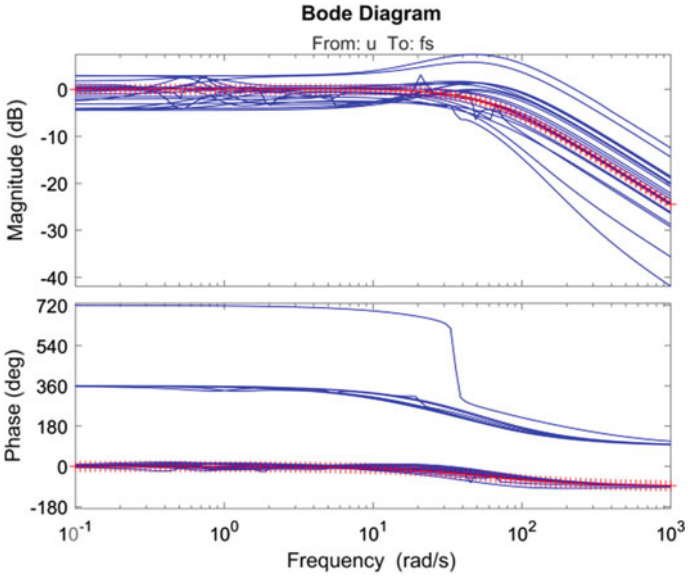


Fig. 9.5 Actuator model comparison with nominal actuator

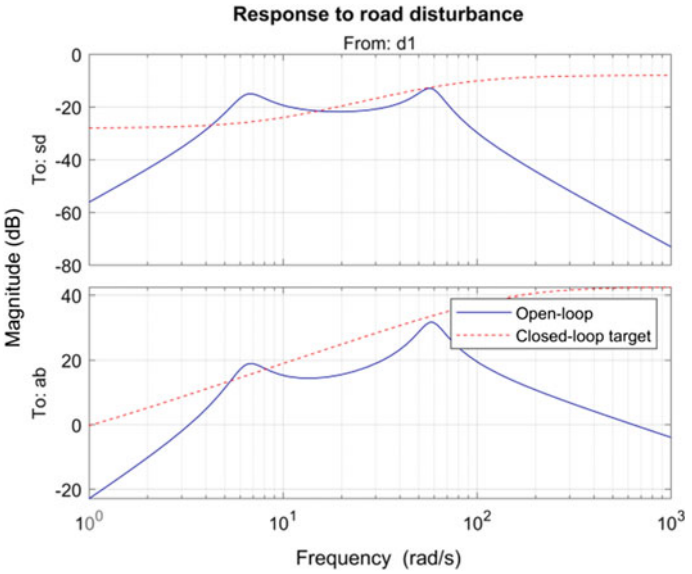


Fig. 9.6 Comparison of active suspension system with passive suspension system

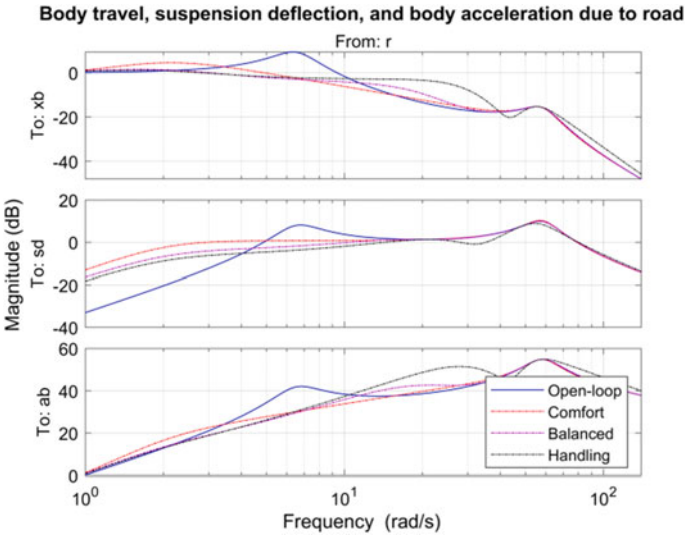


Fig. 9.7 Comparison of parameters with different design objectives

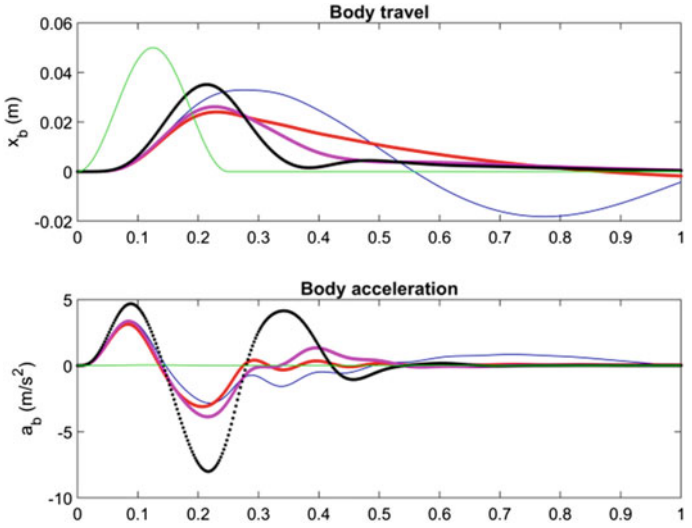


Fig. 9.8 Comparison of design objectives with body travel and body acceleration

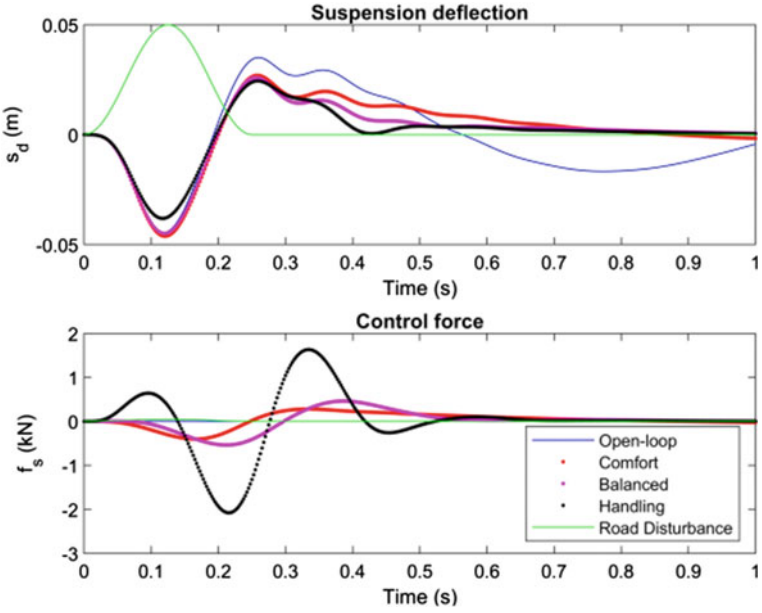


Fig. 9.9 Comparison of design objectives with suspension deflection and control

be observed in the true actuator. In the earlier stage of analysis, during the definition of the actuator model, we only used the nominal value of the actuator but have defined the actuator family including of total 20 sample actuators. The 20 samples generated for the actuator models would aid in accounting for the uncertainty. For obtaining the desired response of the suspension to a bump, we need to add extra feedback to the existing control system which would in turn act as the reduction in the uncertainty in the response and would provide a more robust response from the controller. The aim is to achieve a robust performance according to the road conditions, and accordingly, the control authority can be achieved for the desired system. The μ synthesis is used for obtaining the robust performance in reducing the uncertainty.

The term robust states "the ability of the system to resist change without adapting the initial stable configuration." In terms of control systems, robustness can be defined as "Approach for a Controller Design that explicitly deals with Uncertainty." μ-synthesis falls under the class of the sliding mode control criterion. The control criterion states that "it's a nonlinear control method that alters the dynamics of a nonlinear system by applying a discontinuous control signal that forces a system to 'slide' along a cross section of a system's normal behavior." The current suspension refers to the system as we have defined the state space of the suspension system. Under our analysis, the synthesis can change the feedback control from one continuous structure to another based on the current position of the suspension system. Thus, our control system can be termed as a hybrid dynamical system which contains a continuous state space and discrete control modes. The state space in the synthesis can be referred to the state space of the car, and the uncertain and discrete control modes can be referred to the family of actuator models which were generated earlier. The use of the `musyn` function of MATLAB is used for the analysis. The balanced performance is selected for the analysis, and a robust performance is to be obtained for the balanced design point. The output results in a *D-K* Iteration which provides us with the robust performance achievement.

The results show that the robust performance was achieved best as 1.09 for a bump of 10 cm. Further, we need to obtain the performance of the closed-loop control with the help of the **Krob** controller currently obtained. The result of the **`Krob`** controller performance is shown in Fig. 9.10. The controller performance of the H_∞ controller is to be calculated for the family of actuators that were generated earlier. The result of the H_∞ controller with the different discrete actuator models is shown in Fig. 9.11. To obtain robust performance, only some of the discrete actuator shall be used and the range of the currently obtained H_∞ controller needs to be reduced. The results with the robust **Krob** controller are shown in Fig. 9.12.

9.4 Discussion

The results we have obtained till now are the performance of two controllers, and the initially obtained results were from the nominal H_∞ controller. The results were not optimal since the actuator model used was possessing only the nominal values

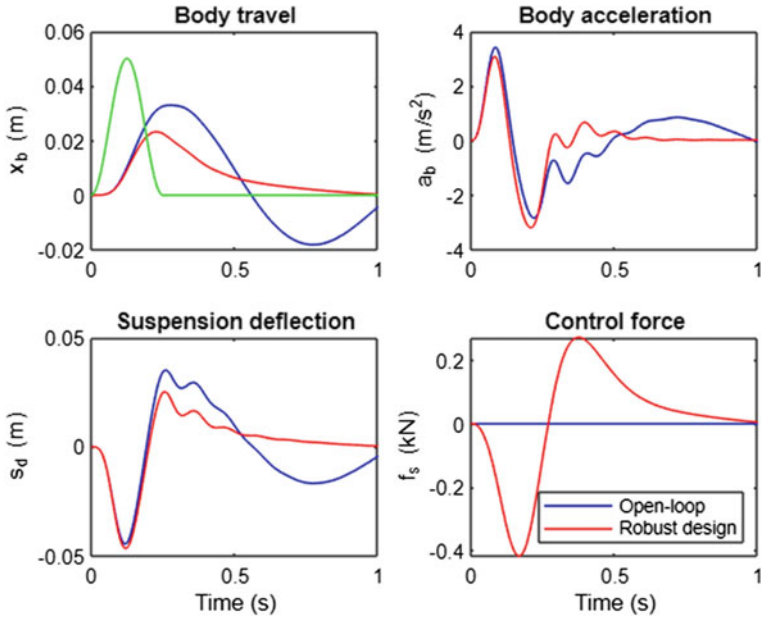


Fig. 9.10 Performance of robust controller

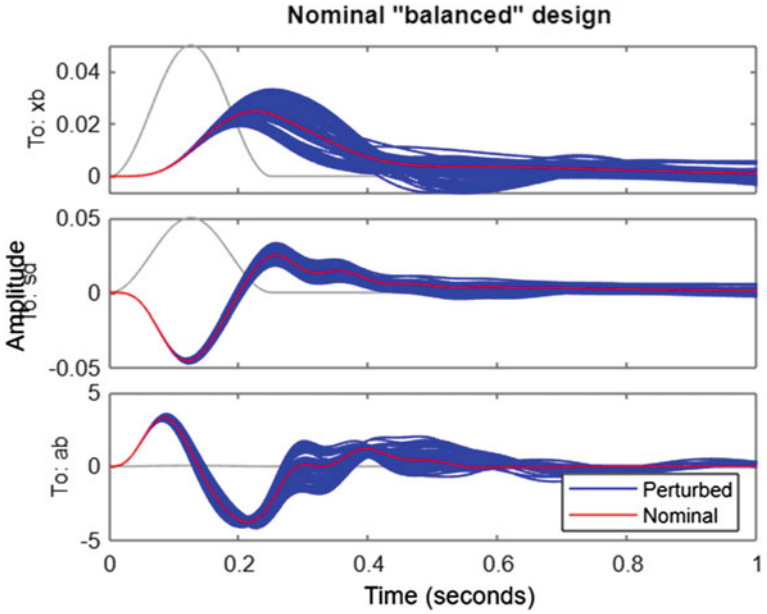


Fig. 9.11 Performance of initial controller with 20 actuator models

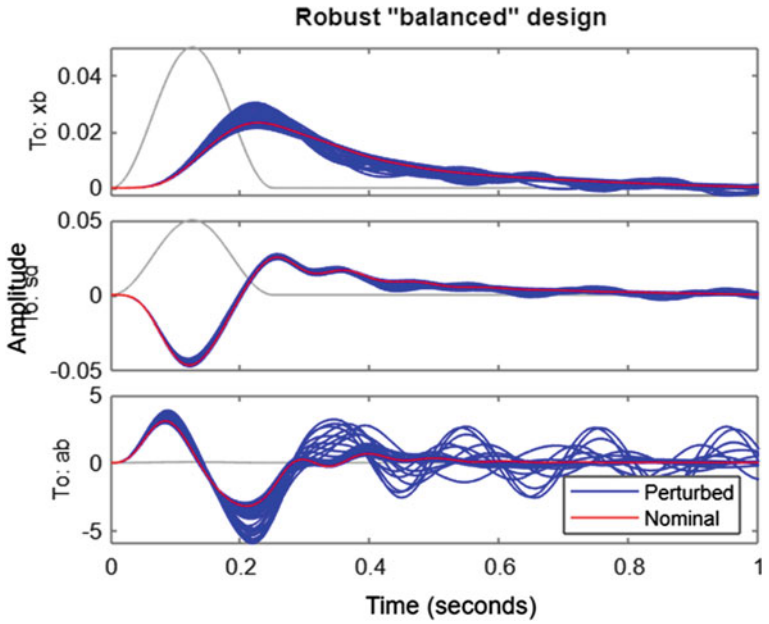


Fig. 9.12 Performance of robust controller amplitude versus time

of the true actuator. Thus, the results obtained were only an approximation of the true response of the actuator, if the controller were to be in actual practice. The second major result obtained was from the family of actuators (discrete generated actuators). The results obtained have been plotted with respect to the response of the suspension to body acceleration, suspension deflection, and body travel. The results show the range in which the results may variate according to the situation. The last results obtained were the performance of the robust controller which was designed for the optimizing the H_∞ controller feedback and obtaining a robust performance. The results obtained after the μ-synthesis works mainly for reducing the variation in the peak MU obtained, i.e., the peak gain in the μ-synthesis. As soon as the variation stops and a consistent performance is achieved, the said performance is called robust performance. Under the use of Krob controller, the results were considerably narrowed down and the desired robust control was achieved. Further, the results were compared to the passive suspension system. The results clearly depict that the robust performance of the controller aided in obtaining a balanced control between the passenger comfort and road handling. The same can be achieved for prioritizing the other design points, the design would vary accordingly, and the gains and their control would change accordingly.

9.5 Future Scope

The results obtained were obtained by the optimization of the initially used H_∞ controller by the μ-synthesis method. The process used for the optimization results in the controller with an auto-defined order controller. Thus, the order of the controller may be more comparatively and there is a chance that the same or similar performance may be achieved by a relatively lower-order controller. The obtained controller can further be reduced in terms of the operational order. The fine-tune of the controller order can be done to reduce the complexity of the operation and can be optimized in that domain. Alternatively, we can use a fixed order controller, obtain results for it, and compare it to the robust performance of results. As the desired results are obtained, the controller obtaining the nearby performance can be used for application. The currently used model was defined and analyzed for only a quarter of a car. The same can be done for the half car model which includes two independent quarter car models. Their respective dependencies and relations can be defined in a state space and the inputs need to change accordingly. The controllers we have been using are the nominal H∞ controller, and the optimization can be carried out with the μ-synthesis as we carried out in the quarter car model. The entire analysis can be done on a full car model, which would be a true analysis for a car since the quarter car only accounts for the response to the road disturbance. But under practical circumstances, the car's movements may result under some deflection to the suspension system as well and counting only the road disturbance for the analysis would be only an approximation. The use of dependency equations for the four wheels and their effects on one another need to be studied for the relative response to the road disturbance. The full car model analysis would result in a much complex result and achieving the desired design points would be more complex since there would be more variables to be optimized.

9.6 Conclusions

The simulation of the quarter car model has been carried out by defining the system and their entire models. The system was defined in terms of the state-space equations and further was converted into state-space matrices. The state-space matrices have been then used as the main systems. The suspension system is then considered for completing the entire system. The suspension system consists of various components like spring, damper, tire, the car body, and the actuator. All the components need to be individually defined and then all the components are to be connected to form a system of suspension. As we defined the system and its components, we now have the system ready for simulation and the next step was to define the control criteria and the disturbances to be given as input. All the values of the system are defined as different weight functions. The weight functions include the control inputs, road disturbance, error signals, and the measured output. While defining the weight

function, we have been keeping the error signal intensity and the road disturbance to be constant. Thus, the dependency of variables is reduced. The entire system control is defined as the active suspension system. The controller that provides the command for maintenance of the design goal is designed using the H- infinity synthesis. The controller designed works on the reduction on the H norm achieved at the end of every output measured from the system. Thus, a gain reduction principle is followed, and for testing the system, a bump was set to 7 cm. The design of the controller was initially designed by the H-infinity synthesis, the results obtained achieved the basic goals of achieving the control authority. The controller designed only achieved the performance with only the nominal actuator in the suspension system. The practical conditions would have an uncertain actuator model, and to model the practical/true actuator, the family of actuators are considered and 20 different actuator models were generated and simultaneously tested. The current controller was also optimized using the μ- synthesis approach. The same controller was tested for the 20 different actuator models and provided with a range of control with the aid of different actuators.

References

Ahmed AA (2021) Quarter car model optimization of active suspension system using fuzzy PID and linear quadratic regulator controllers

Allamraju VK (2016) Modal analysis of quarter car model suspension system

Alvarez-Sánchez E (2013) A quarter-car suspension system: car body mass estimator and sliding mode control

Ebrahimi-Nejad S (2020) Multi-objective optimization of a sports car suspension system using simplified quarter-car models

Elattar YM (2016) PDF versus PID controller for active vehicle suspension

Gawad MAA (2021) Mechatronic suspension systems: a survey and directions for future work

Had A (1992) Optimal semi-active suspension with preview based on a quarter car model

Hassaan GA (2015) Car dynamics using quarter model and passive suspension, part VI: sprung-mass step response

Liu C (2019) General theory of skyhook control and its application to semi-active suspension control strategy design

Paliwal V (2020) Effect of varying road profile amplitude on the behavior of a nonlinear quarter car model

Shao SJ (2020) Study on the stability and vibration reduction of nonlinear active suspension system with time-delayed feedback control

Sharaf AM (2013) Ride comfort analysis using quarter car model

Thite AN (2012) Development of a refined quarter car model for the analysis of discomfort due to vibration

Yu M (2019) Quarter-car experimental study for series active variable geometry suspension

Zeng Q (2019) Adaptive vehicle stability control of half-car