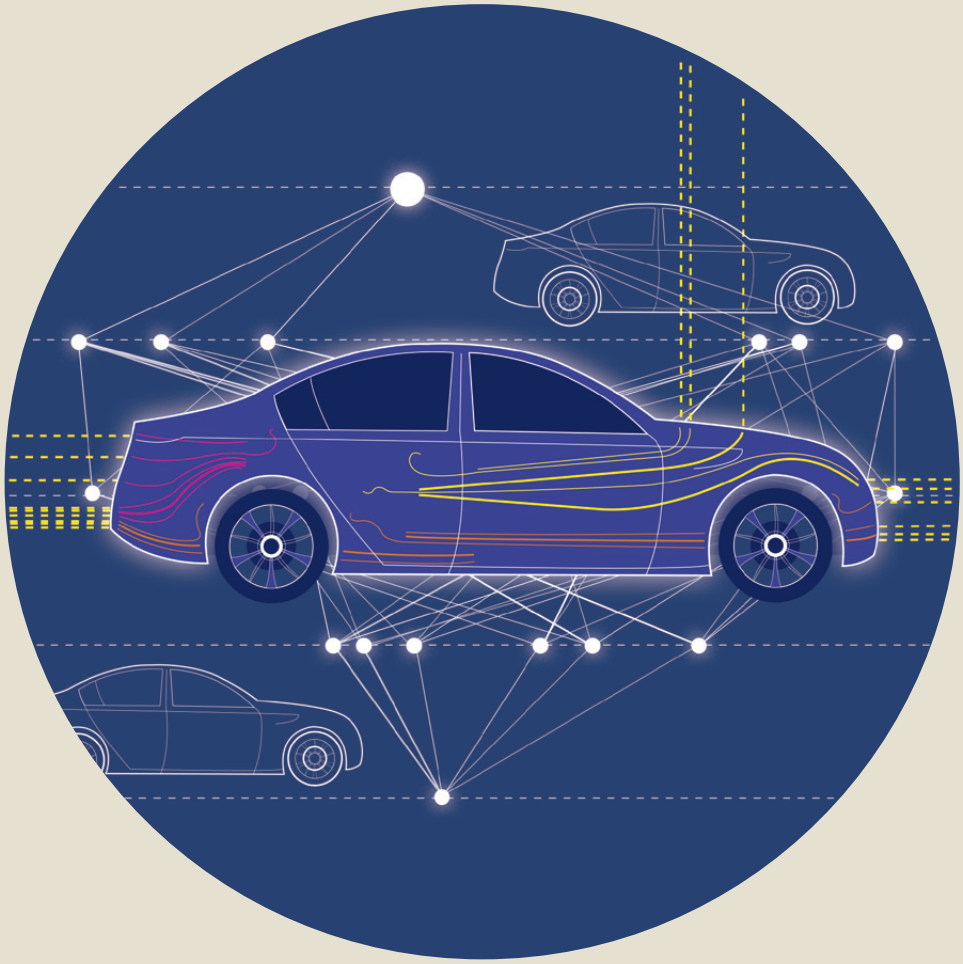
THESES



Civil, Geo and
Environmental
Engineering

Lailong Song

Commonality Design of Vehicle Architectures Concerning Crashworthiness Using Solution Spaces

TUM.University Press

Professur für Computational Mechanics
Ingenieurfakultät Bau Geo Umwelt
Technische Universität München

Lailong Song

Commonality Design of Vehicle Architectures Concerning Crashworthiness Using Solution Spaces

Vollständiger Abdruck der von der Ingenieurfakultät Bau Geo Umwelt der Technischen Universität München zur Erlangung des akademischen Grades eines Doktor-Ingenieurs (Dr.-Ing) genehmigten Dissertation.

Vorsitzender:

Prof. Dr.-Ing. Johannes Kai-Uwe Bletzinger

Prüfer der Dissertation:

1. Prof. Dr.-Ing. habil. Fabian Duddeck

2. Prof. Dr.-Ing. Thomas Vietor

3. Prof. Dr. Helmut Harbrecht

Die Dissertation wurde am 03.12.2018 bei der Technischen Universität München eingereicht und durch die Ingenieurfakultät Bau Geo Umwelt am 16.05.2019 angenommen.

The German National Library has registered this publication in the German National Bibliography. Detailed bibliographic data are available on the Internet at https://portal.dnb.de.

Imprint

Layout design and typesetting: Lailong Song
Layout guidelines for cover design: Designbuero Josef Grillmeier, Munich
Cover design: Caroline Ennemoser
Cover illustration: Caroline Ennemoser

ISBN: 978-395884-031-7
DOI: 10.14459/2019md1464340

TUM.University Press
Technical University of Munich
Arcisstrasse 21
80333 Munich

www.tum.de

Commonality Design of Vehicle Architectures Concerning Crashworthiness Using Solution Spaces

Abstract

The decision on commonality configurations of a vehicle architecture has to be made in early phase of vehicular crashworthiness design to achieve the functional feasibility and economic efficiency. In this early design phase, neither the hardware is available, nor the classical virtual methods can be applied. This is due to (i) non-availability of data and especially incompleteness of component development in this stage (lack-of-knowledge uncertainties), (ii) necessity to evaluate a large number of different designs together with non-affordable computational times for detailed models, and (iii) the fact that the number of possible commonality configurations scales up exponentially with the size of the architecture, resulting in NP-hardness to identify the optimal commonality configuration. To solve these problems, the following aspects are addressed here.

In order to evaluate the design feasibility efficiently, the Solution Space approach has been further developed, utilizing available information in the early phase to deal with more complex crash load cases also involving deformable crash barriers. The size of the identified Solution Space under nonlinear constraints derived from full-vehicle structural requirements determines the feasibility and flexibility of the structural design. Moreover, the Solution Space approach is extended to evaluate the entire vehicle architecture considering the common components shared by different vehicles.

The feasible architecture configurations with high commonality are thus identified by a predictive elimination procedure -- the hierarchical relationships between the architecture configurations together with the Solution Space evaluation of the configurations are exploited to heuristically reduce the complexity of the commonality optimization problem.

Therefore, in this dissertation, an approach to identify the vehicle architecture configurations with feasible crashworthiness design and satisfactory commonality is developed and validated. This approach involves the analysis of crash load paths, the construction of computationally efficient low-fidelity crash models, the identification of Solution Space under non-linear constraints for the entire vehicle architecture, and a heuristic elimination procedure ensuring optimal commonality configuration.

Kommunalitätsgestaltung von Fahrzeugarchitekturen bezüglich Strukturauslegung für Crashlastfälle durch das Lösungsraum-Verfahren

Abstract

Die Entscheidung hinsichtlich der Kommunalitätskonfiguration einer Fahrzeugarchitektur muss in der frühen Phase der strukturellen Auslegung getroffen werden, um die funktionale Zulässigkeit und Wirtschaftlichkeit zu erreichen. In dieser frühen Entwicklungsphase ist weder die Hardware verfügbar, noch können die etablierten virtuellen Verfahren angewendet werden. Dies liegt an (i) unzureichenden Daten und insbesondere an der Unvollständigkeit der Komponentenentwicklung in dieser Phase (Lack-of-Knowledge Ungewissheit), (ii) der Notwendigkeit, mehrere alternative Auslegungsentwürfe mit zu hohen Rechenzeiten im Detail zu bewerten, und (iii) an der Tatsache, dass die Anzahl der Kommunalitätskonfigurationen exponentiell mit der Anzahl der Fahrzeuge in der Architektur skaliert, was zu einer NP-Hardness führt, um die optimale Kommunalitätskonfiguration zu identifizieren. Um diese Probleme aufzulösen, werden die folgenden Aspekte thematisiert.

Um die Zulässigkeit der Strukturauslegung effizient zu bewerten, wird das Lösungsraum-Verfahren weiterentwickelt - die in der frühen Phase verfügbaren Informationen werden ausgenutzt, um die komplexeren Crash-Lastfälle mit deformierbarer Barriere zu behandeln. Die Größe des identifizierten Lösungsraums unter nichtlinearen Nebenbedingungen, die aus den strukturellen Anforderungen des Gesamtfahrzeugs abgeleitet werden, bestimmt die Zulässigkeit und Flexibilität der Strukturgestaltung. Darüber hinaus wird das Lösungsraum-Verfahren

erweitert, um die gesamte Fahrzeugarchitektur mit kommunalen Bauteile zwischen mehrere Fahrzeuge evaluieren zu können.

Die zulässigen Architekturkonfigurationen bezüglich der Lösungsraum-Bewertung mit hoher Kommunalität werden somit durch einen prädiktiven Eliminierungsprozess identifiziert – Die hierarchischen Beziehungen zwischen den Architekturkonfigurationen zusammen mit den dazugehörigen Lösungsraum-Bewertungen werden eingesetzt, um die Komplexität des Kommunalitätsoptimierung-Problems der Architektur heuristisch zu reduzieren.

In dieser Dissertation wird daher ein Verfahren zur Identifizierung der Architekturkonfigurationen mit zulässiger Strukturauslegung und zielführender Kommunalität entwickelt und validiert. Dieses Verfahren fasst die Analyse von Lastpfaden der Crashstruktur, den Aufbau von einem recheneffizienten Low-Fidelity-Model, die Identifizierung von Lösungsräume unter nichtlinearen Randbedingungen für die gesamte Fahrzeugarchitektur und einen heuristischen Eliminierungsprozess, um die optimale(n) Kommunalitätskonfiguration(en) zu entdecken, in einem neuen Workflow zusammen.

Contents

Listing of Figures

This is the dedication.

Acknowledgments

There are many people that earn my gratitude for their contribution to my dissertation. More specifically, I wish to express sincerely my appreciation:

To Prof. Dr.-Ing. habil. Duddeck, my thesis advisor, for the guidance and encouragement since my study of the master program *Computational Mechanics* and throughout my time as his student. He continually and convincingly conveyed a spirit of critical thinking, open discussion, and meticulous attitude in documentation during the scientific research.

To Dr.-Ing. Fender, my research project supervisor in BMW Group, who did not only organize the industrial funding to make this project possible but also gave inspiring and creative ideas in plentiful productive discussions. Moreover, he taught me how to build up the bridge between academic research and real world application, from which I am always benefiting in my current daily work.

To the rest of my dissertation committee members, Prof. Dr.-Ing. Vietor and Prof. Harbrecht for their great support and invaluable advices.

To the colleagues in my team in BMW Group, Volker Lange, Dr. Thomas Amler, and Johannes Stadler for their constructive suggestions and critical reflections on the application of the established approaches on industrial problems.

To my family who self-givingly supported me on a daily basis by giving me the time and the space to write this dissertation. Especially, to my dear wife Sijia. I thank her for the understanding when the dissertation took me away from spending time with her and from taking care of our son Caspian. It is also grateful that Caspian gave me many peaceful evenings to work on the dissertation although he cannot yet understand the mathematics and mechanics I wrote about.

Last but not least, to the contributors to the open-source communities, which essentially facilitate the scientific research nowadays.

"Engineering is the application of scientific and mathematical principles to practical ends such as the design, manufacture, and operation of efficient and economical structures, machines, processes, and systems."

Anonymous

1

Introduction

1.1 Product Development of Vehicle Architectures in the Concept Phase

1.1.1 Strategical criteria in product design: modularity and platform

Evalueserve estimates that by 2020, the 10 largest OEMs (in alphabetical order: Daimler, Fiat, Ford, General Motors, Honda, Nissan, PSA Peugeot Citroen, Renault, Toyota, and Volkswagen) will reduce their platforms by about a third from over 175 platforms in 2010, and will concentrate mass production across a few key core platforms (Seghal & Gorai 2012).

This strategy is not only a leading idea for the automakers with high number of units in production across various sub-brands but also adopted by the premium automakers.

For example: In 2016, the BMW Group steps into the second phase of the wide-ranging strategy referred to as Strategy Number One. In this stage, the UKL platform architecture (Untere Klasse, lower class in German) is developed to underpin up to 15 front-wheel-drive models. In addition, another platform architecture CLAR (Cluster Architecture), which is envisaged to underpin about 30 rear-wheel-drive models, is also under development. To enable a high model variety by using a reduced set of underlying designs and to reduce therefore the complexity is a key pillar in the company's future strategies.

The term **platform** in automobile sector exists since 1936 (The A-platform by General Motors) (N.N. 2018*c*,*d*). It is mainly described as a common foundation for various individual products. This foundation only needs to be designed once during the product development. This foundation can be divided into four categories: components, processes, knowledge, people and relationship. Essentially, a "product family" with a range of products based on this foundation (platform) is developed (Hou et al. 2017). The prime advantage of applying this common foundation lies thus in the economics perspective. Cost reduction has always been one of the dominating driving forces in industry with mass production.

The research presented in this thesis focuses on the perspective of engineering design of the components for a platform. The term **architecture** is thus adopted to emphasize the modularized component design with respect to their mechanical properties, i.e. the end products which belong to an architecture have the same modularization. The differentiation or the commonality of each module/component is the design parameter of this architecture. The **commonality** here is a property of the architecture, describing the sharing state of each module among the end products.

If the number of platforms is further reduced and a strong commonality is required among the different vehicle size classes (subcompact, compact, mid-size, full-size etc.), the realizability may reach a limit. To overcome this, a common

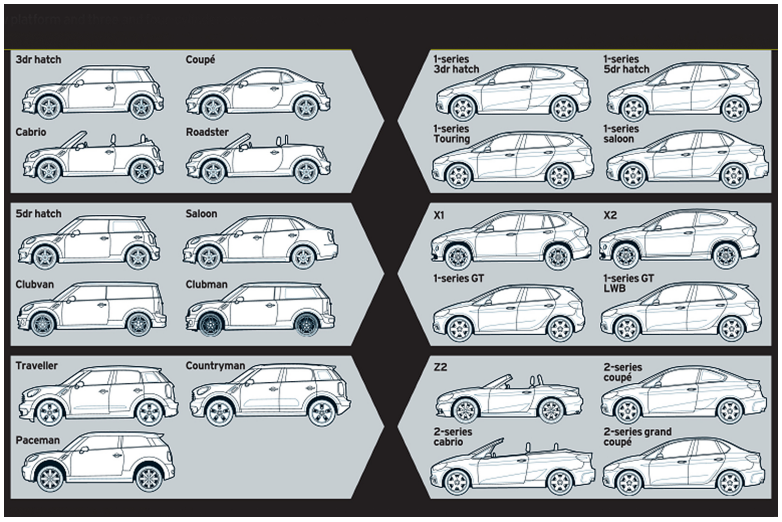


Figure 1.1: UKL platform developed by BMW Group (Hilton Holloway 2013).

platform must fulfill all constraints and expectations from each individual product.

Among all the structural functional requirements that should be considered for the architecture design, the passive safety is one of the leading characteristics due to the following reasons: 1) the "Vision Zero" strategy[i] to eliminate all traffic fatalities and severe injuries drives the government agencies, the independent customer test institutes, insurance companies, and auto manufacturers to tighten their requirements all over the time, thus introducing new crashworthiness design criteria for each architecture generation. 2) achieving the overall crashworthiness property is an iterative process by continuously improving the functional properties of the sub-system. This cross-functional process must be started as early as possible to avoid the multidisciplinary conflicts in the late development phase

[i] https://visionzeronetwork.org/about/what-is-vision-zero/

(Bois et al. 2004).

In order to fulfill significant design criteria regarding passive safety, it requires to identify the best compromise between a high number of functionalities, e.g. the total vehicle weight and mass distribution, the location and performance of the critical components and the restraint system, to absorb the crash energy effectively to protect passengers from getting injured. As a matter of fact, these configurations vary strongly between the vehicle size classes. Therefore, an adaptive architecture which not only maximizes the commonality but also compromises the distinctiveness is required.

A modular platform is discussed by (Jose & Tollenaere 2005) to ensure the distinctiveness of the end product; it allows a scalable and re-sizable design. Some modules/components can be easily plugged into the platform even if they are designed differently due to adaptivity requirements. During the implementation of such a modular platform, an interesting topic arises – ***how to make the optimal decision between commonality and distinctiveness for the architecture with respect to economic and design feasibilities?*** Here, feasibility means that a shared common component can be designed to fulfill all constraints and objectives from the vehicles in which it is integrated.

In order to bring the open question stated above from strategic level onto operational level, this work proposes methods to identify the optimal commonality

configurations of the architecture under the criteria claimed by crashworthiness design.

1.1.2 Vehicle Architecture

First, the mathematical description related to the problem must be derived to formulate this optimization problem. The terms used to quantify the properties of an architecture are established in following:

Quantification of Commonality

The commonality is a qualitative feature description of the components which are shared by vehicles in a single architecture. In order to illustrate the commonality of an architecture, the following example is investigated. As shown in Figure 1.2, the vehicle structure is modularized into five elements/components ($C1$ to $C5$). All vehicles which are derived based on this architecture, must contain a complete set of these components. As discussed in Section 1.1.1, each component, if possible, is expected to be common among all vehicles for economic efficiency and functional feasibility. However, this is not always possible due to individual constraints or particular boundary conditions for the components coming from each vehicle. The reason will be further discussed in Section 1.3. As a result, the necessary and realistic number of variants for the components in a vehicle architecture has

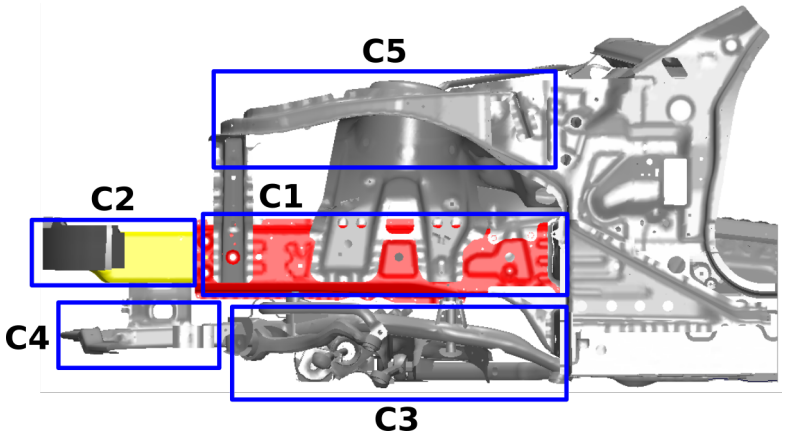


Figure 1.2: A front structure platform with five parts. Denotation of the components: $C1$–front rail; $C2$–crash box of the front rail; $C3$–front axle; $C4$–crash box of the front axle; $C5$–wheel housing frame.

to be decided. In addition, even with the same number of variants, the different combination set-ups of the common components influence the economic and design feasibility as well.

Assessment of Design Feasibility

The goal of reaching a higher commonality is always limited by the necessity of distinctiveness due to the design feasibility. In crashworthiness design, the structure assembly must fulfill certain design criteria; for instance, the homologations, the regulations, the star ratings defined by New Car Assessment Programs (NCAP), and structural performance requirements defined by insurance institutes. These criteria are broken down onto the component level and form constraints for the functional property of each component (Zimmermann & von Hoessle 2013, Zimmermann et al. 2017, Fender et al. 2017). These constraints are derived from

the vehicle configurations in which the component is integrated. In the design space, the constraints regulate the feasible domain for the design parameters of the components. When one component underpins several vehicles, its feasible design domain is normally bounded by more constraints than in the single vehicle case. This may result in an over-constrained feasible domain leading to overall infeasibility, which shows that the decision on the component sharing strategy is too ambitious. More variants of the components have to be created to resolve this infeasibility. In addition, a small feasible domain may also end up in an impractical commonality. The reasons are as follows:

In practice, the commonality configuration in the product family must be defined in the early design phase. ***The lack of knowledge in the early phase, concerning design details decided later in the process, leads to uncertainties on the system parameters.*** E.g. the deformable lengths of the crash components and mass distribution of the structure assembly are normally only estimated in the early phase. This may result in later changes of the constraints for component design. To describe this mathematically, the constraints of the components' functional properties are here denoted as inequality $g_i(x) \leq 0$. The feasible domain of the component functional properties can be thus noted as:

$$\mathbb{D} = \{\mathbf{x} \mid g_i(\mathbf{x}) \leq 0\} \ , i = 1, \ldots, n. \tag{1.1}$$

in which $\mathbf{x}$ represents the functional properties of the components in the i-th vehicle in the product family. The $\mathbb{D}$ is used to represent the feasible design space of the component, $\|\mathbb{D}\|$ denotes thus the size of the space. Apart from this, the components have multi-disciplinary design constraints; multi-disciplinarity in automotive design refers to the fact that different crash load cases have to be considered together with other functionalities like structural stiffness, vibrations, interior acoustics or aerodynamics. If the feasible design domain of the component is already strongly constrained by crashworthiness requirements, any further constraint from other disciplines may exclude all the possible designs. ***As a result, a larger feasible domain for the component considering all criteria in the early phase is attractive and can guarantee sufficient design flexibility to account for the aspects mentioned above.***

As a short summary, the optimal decision on the platform configuration in the early phase is to reach a high system commonality as well as to ensure feasible design space for all components to be larger than a critical threshold, i.e.

$$\|\mathbb{D}\| \geq \|\mathbb{D}\|_{crt} \quad . \tag{1.2}$$

For this purpose, an approach of evaluating the size of design spaces under commonality configuration must be established.

1.1.3 Scenario Analysis & Complexity

Keeping the commonality configuration problem for later, we first review an ancient Japanese parlor game – Genji-ko, in which guests receive five packages with incense to smell. The guests have to identify which ones are identical and which are different. The enumeration of the solutions is shown in Figure 1.3, which indicates the possible groupings of five elements. The elements which are under a common horizontal bar and which have the same color are grouped together. This ends up with 52 possibilities. This is the antecedent of applications of an important mathematical concept.

Getting back to the commonality decision for one component in the architecture, it is actually the same ***Set Partitioning*** process – the products which are grouped into the same subset share a common component.

The decision scenarios of an architecture need to be analyzed to reveal the complexity of the problem.

In this well-established mathematical problem of Set Partitions, the complexity to identify the optimal commonality decision is exponential to the number of vehicle derivatives and components types. Therefore, the combinatorial optimization problem described above cannot be tackled by ”brute-force” approaches.

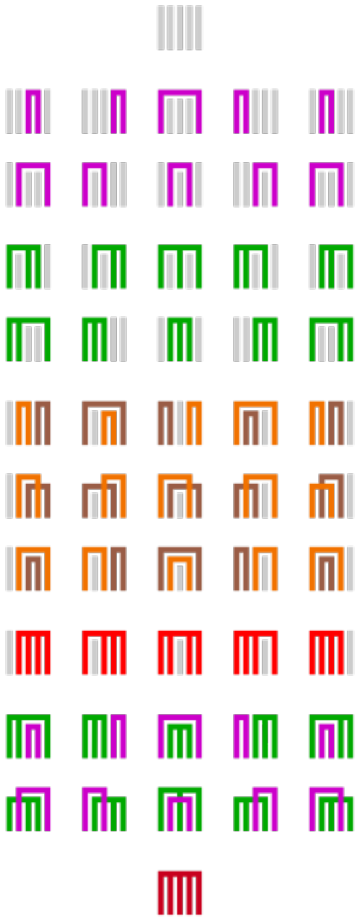


Figure 1.3: The "Genji-mon" symbols to show all the configurations to partition the set[ii].

1.2 Structural Crashworthiness Design

1.2.1 Importance of Rear Crash Test Load Case

One of the most important criteria for planning the commonality and distinctiveness of the vehicle body structure platform is crashworthiness. In the vehicle crashworthiness design process, the structure is optimized under various load cases,

[ii] https://commons.wikimedia.org/wiki/Category:Genji_chapter_symbols

including, but not limited to front crash, frontal small overlap and oblique crash, side crash, and rear crash. However, the research project introduced in this dissertation cannot cover and resolve all the load cases due to the complexity. This work focuses on one of the latest introduced load case with increasing criticality in the road traffic – rear crash for the following reasons:

According to the traffic accident statistics reported for 2014 by the National Automotive Sampling System (NASS) in the USA, among all 2,896 investigated accidents, the rear crash has a proportion of 17.1%; this is not so dominant as front crash (69.8%). However, as the electric vehicles are prompted to reduce the concentrated air population in the city, more and more vehicles equipped with large volumes of high voltage batteries and electric motors participate in the road traffic. In order to improve the driving range compared to cars with internal combustion engines, some car producers decide to increase the volume of high voltage battery, which leads to a higher fire hazard in an accident. Figure 1.4 shows an example of a regular design concept and the position of the high voltage battery and electric motor. We can see that these components are mounted relatively close to the deformation zone of a rear crash. The high voltage batteries may suffer high crushing loads during a rear crash. Additionally, micro city cars gained popularity with increasing fuel price and traffic density in cities. The crashworthiness of these vehicles, specially the integrity of the fuel tank in crash, must be approved.

Figure 1.4: Positioning of the high voltage battery and electric motor in Battery Electric Vehicle (BEV).

Both of these facts emphasize the significance of vehicle crashworthiness in rear crash load cases in the upcoming future. Therefore, we consider it as one of the main criteria on determining the commonality of the vehicle structure platform and it is hence investigated in an exemplary manner in this work. The methods can be transferred and further developed to other crash load cases.

1.3 Component Design and Commonality Decision

In the literature, the usage of specific terms needed for this thesis is often not precise enough. Hence, two naming conventions are clarified here:

- **Crash Component**, or Component in short, denotes a deformable component

which can absorb energy mainly by plastic deformations with specific force-deformation characteristics. These characteristics are the functional design parameters for the crash components' development.

- **Structure Assembly** consists of a set of Crash Components. The kinematics, i.e. acceleration, velocity, and intrusion, of this Structure Assembly are the direct result of the crash components' functional characteristics defining this assembly.

Normally, the crashworthiness criteria apply on the structure assembly level while the design parameters are associated with the crash components.

In the late development phases, crashworthiness of the structure assembly is assessed via explicit Finite Element (FE) simulations. An advanced and highly detailed model of a single design of a total vehicle is used with precise geometry and material data. Corresponding simulations are computationally expensive. The components are normally derived from different departments and commercial partners.

Furthermore, when making the commonality decision for crash components, the common crash components should be integrated into different vehicles (structure assemblies). This common component is thus subject to different boundary conditions from the vehicles, since its "neighborhood components" are different; it should perform together with others to satisfy different criteria predefined for the

vehicles. This dependency of the component criteria through commonality is described in Figure 1.5: the common component is built into n different vehicles, which differ in available deformation lengths, weights, and so on. This results in different working environments for the common component. Additionally, this common component should work together with other components in the structure assemblies to fulfill the design objectives (e.g. acceleration, energy absorption) defined for the entire structure assemblies.

As a consequence, to evaluate the feasibility of one specific commonality configuration for one component, i.e. to examine if the vehicles, where the common component is installed, fulfill the design objectives, each vehicle in the architecture has to be simulated. Any change in the common component or the "neighborhood components" leads to additional expensive vehicle simulations. However, all this is not possible in early design phases, in which the requirements are:

- Fast assessment of a set of alternative designs by different stakeholders;
- Efficient evaluation of different commonality configurations for the crash component;
- Non-detailed and flexible (parametric) modeling because of lacking information and knowledge;
- Hierarchical approach to handle complexity and break down requirements from structure assembly level to sub-system and crash component level;

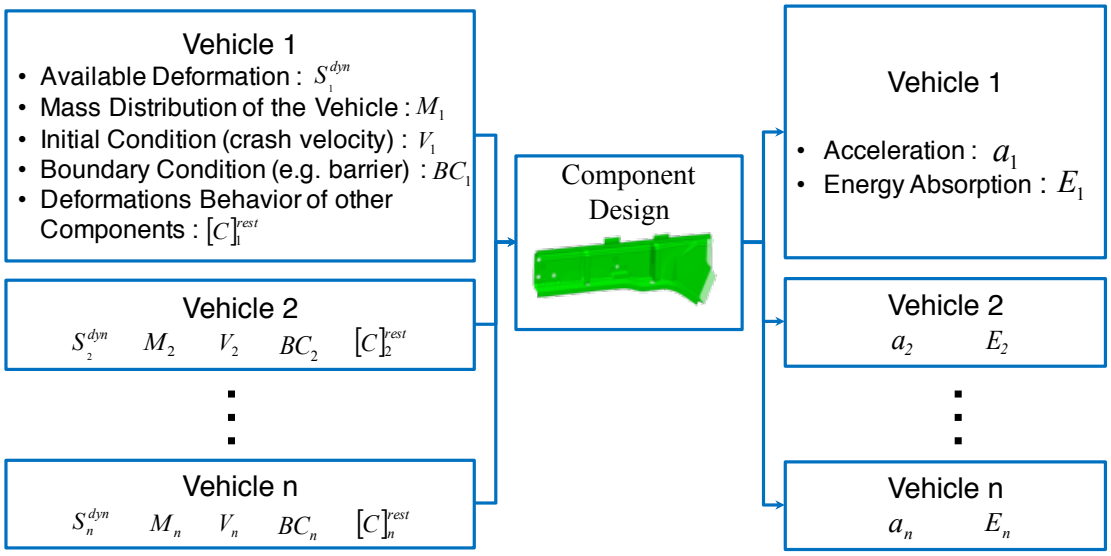


Figure 1.5: On the left hand side are the constraints and working environment in different vehicles for the common component while on the right hand side are the design objectives which the common component with its structure assembly must fulfill.

Considering all these aspects, an efficient simplified model which utilizes the limited information and takes into account uncertainties to assess the structure crashworthiness in rear-collision in the early development phase need to be developed. This simplified model should also enable the efficient evaluation of various commonality configurations.

From the perspective of system engineering, the V-Model in Figure 1.6 is used to describe the relationship between early and late design phase: in the early phase, overall system objectives are broken down or cascaded to sub-system and further to components level, each component receives its own functional requirement; in the later design phase, the geometries, materials, and manufacturing/connection approaches of these components are designed to fulfill the corresponding func-

tional requirements. When these components are assembled together, the overall crashworthiness requirements of the entire vehicle can be satisfied.

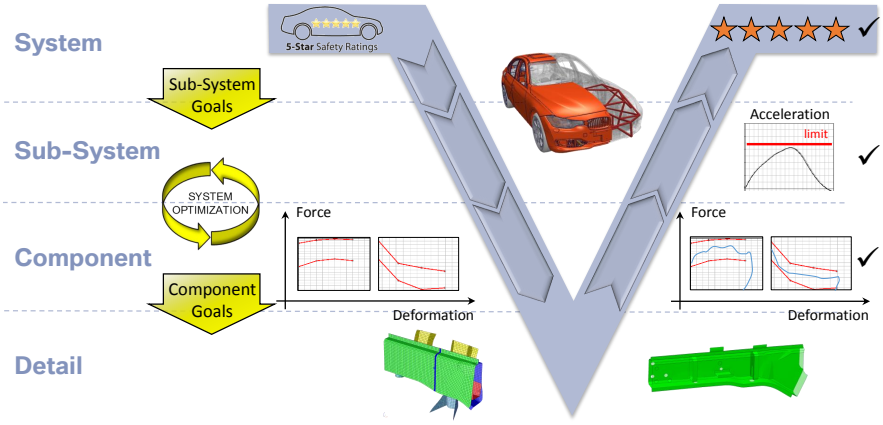


Figure 1.6: The V-Model for system engineering introduced in (Gausemeier & Moehringer 2002) is used to cascade the design process into different phases based on the availability of information by Fender et al. (Fender et al. 2014).

In order to develop approaches fulfilling the requirements stated above, more specifically, to identify the optimal commonality configuration with respect to crash criteria under the lack of knowledge uncertainties in the early design phase. The researchers have made the contributions in the following aspects.

"If I have been able to see further, it was only because I stood on the shoulders of giants."

Letter to Robert Hooke by Isaac Newton

2

State of the Art

To enable the precise definition of the Aims and Objectives of this thesis, the state of the art in product architecture/platform design, low-fidelity model for crashworthiness design, etc. are analyzed first.

2.1 Product Development Based on Architectures

Inspired by the first assembly lines of the Ford Motor Company in 1913, the *mass production* was popularized since 1926 to optimize the process of the product development. However, as the market preferences, parallel to the legislation and regulation, refined with time, it became more difficult that the mass product fulfill exactly the requirements. Consequently, Pine (Pine II 1993) extended the paradigm to *mass customization*, which aims to create higher variety of the product maintaining the low-costs of mass production. The necessity of higher product variety was also discussed by Ho and Tang (Ho & Tang 1998). One of the approaches to ensure product variety as well as design efficiency is a development based on the idea of a product architecture – a set of similar products derived from a common platform having specialized features to meet different functional requirements. Siddique and Rosen (Siddique & Rosen 2001) categorized the platform-based design problem into the following two problems: i) Platform Commonization (PC) Design Problem: Given a set of similar products, identify common modular platform architectures that satisfy multiple view-points. ii) Plat-

form Supported Product Variety (PSPV) Design Problem: Given a configuration of a common platform and product family information and constraints, identify the configuration of the different products that can be supported by the common platform. The PC Design Problem occurs during the configuration design stage to identify the possible architecture and commonality. However, only the possible combinations under the geometrical and topological constraints were investigated. The commonality of the architecture was not optimized with respect to the functional requirements of the design parameters. The PC Design Problem is also defined as the commonality optimization problem while the PSPV Design Problem is described as Product Positioning Problem. In this thesis, we focus on the former to optimize the commonality of the architecture in the early design phase. The necessary steps to optimize the commonality includes: i) defining product architecture and modularizing the sharable components; ii) establishing the functional requirements of the components and the objective(s) of the product architecture; iii) optimizing the commonality of the sharable components among the end products and sketching the product portfolio.

2.1.1 Modularity of the product in the architecture

Product architecture can be defined as the way in which the functional elements of a product are arranged into physical units and the way in which these units interact (Ulrich & Eppinger 2011). The interactions are categorized into three types:

(1) functional modularity, (2) technical modularity, and (3) physical modularity (Jiao et al. 2007). The concepts of modularity are fundamental in structural architectures (Ulrich 1995). A module in the system refers to a subsystem that has the identical characteristics and is exchangeable. Modularity refers to the possibility to break a system down into more or less independent parts/modules (Newcomb et al. 1998). To improve this modularity, the interaction of design activities between the modules must be minimized (Ulrich 1995). This means that the degree of modularity shows the complexity of the functional dependencies of the components in the architecture. For this, Ulrich defined the function-to-component ratio for each product as one modularity metric (Ulrich 1995). Hölttä and Salonen (Holtta & Salonen 2003) proposed a measure of modularity based on singular value decomposition of the binary Design Structure Matrix (DSM), which is introduced by Steward (Steward 1981) to indicate the interactions between the sub-systems. The maximized degree of modularity leads to a minimal dependency of the components, which enables independent development of each modular/component and, thus, saves the efforts of unnecessary communication between different teams and/or processes.

One possible way to achieve this modularity was introduced recently; the so-called Solution Space approach allows to modularize and decouple the functional requirements of components/subsystems within the entire system in a structured

manner (Zimmermann & von Hoessle 2013, Song et al. 2015, 2017, Fender et al. 2017). Initially, the approach was applied on the structural crashworthiness design in a single load case, a predefined frontal collision of one vehicle against a rigid wall with full overlap; this represents a typical issue in system engineering, which is explained in the following. The crashworthiness of the structure assembly is determined by the collaboration of all involved components. These components, however, are developed by different departments and subjected to multi-disciplinary design criteria, including but not limited to crash, acoustics, driving dynamics and design. The Solution Space method is aiming to decouple the unnecessary synchronization and dependencies of the components' design by orthogonalizing the functional requirements of the components in the design space. A rough scheme of the proposed process to modularize the crashworthiness structural design is shown in Figure 1.6.

Alternatives to the Solution Space approach can be found in the literature, e.g. Hou et al. proposed a graph-based decomposition algorithm to modularize a typical body-in-white (BIW) structure. The BIW is described by a topological graph including components and their connections. The way of decoupling the BIW and connecting the sub-structure defines thus the design variables in the proposed method. These design variables are optimized so that the stiffness evaluated by Finite Element Method, the manufacturability evaluated by the number of the

molds used for stamping the sub-components, and the assembling ability evaluated by the number of welding spots reach the optimum (Hou et al. 2017).
The commonality of those decoupled modules is to be optimized for the architecture design.

2.1.2 Commonality metric of the architecture

In order to formulate the commonality as objective of an optimization problem, the commonality of the product architecture must be quantified. Therefore, in the literature, various metrics have been defined to evaluate the product architecture. Thevenot and Simpson compared various definitions of the commonality indices for assessing the component sharing (Thevenot & Simpson 2006), including the Degree of Commonality Index (DCI) (Collier 1981), the Total Constant Commonality Index (TCCI) (Wacker & Treleven 1986), the Product Line Commonality Index (PCI) (Kota et al. 2000), the Percent Commonality Index (%C) (Siddique et al. 1998), the Commonality Index (CI) (Martin & Ishii 1996, 1997), and the Component Part Commonality Index (Jiao & Tseng 2000). These commonality indices can measure the commonality of the whole product family from zero commonality to complete commonality. Here, we use $\mathbf{\Gamma}$ to denote the Commonality Description Matrix (CDM). The Commonality Index (CI) can be calculated based on the CDM, which leads to the general convention used in this work $CI(\mathbf{\Gamma})$. In the optimization problem, the metrics commonality ($CI(\mathbf{\Gamma})$) is to maximized.

Additionally, the distinctiveness, the cost, and the profit are utilized as metrics of the architecture as well (Jiao et al. 2007). However, these metrics can be derived from the commonality indices mentioned above.

2.1.3 Commonality optimization

The common components in different end products are subjected to various local functional requirements (FRs). Here, the term "local" indicates that the component is considered locally in one specific end product while the term "global" indicates the components are subjected to the information from all the end products in which they are integrated. The product architecture design solutions are generated in the physical domain by mapping FRs to design parameters (DPs) based on the commonality (Jiao et al. 2007). The design of a common component must fulfill the global requirements collected from all the end products. In the early phase of structural crashworthiness design, the functional requirements of each crash component are investigated via a simplified model. Those simplified models can handle the lack-of-knowledge situation in early phase as well as evaluate the functional characteristics efficiently. The application of the simplified model is to be reviewed in Section 2.2.

Algorithms for Commonality Optimization

With the framework of the modularized architecture, the product portfolio is optimized to maximize the commonality while respecting the functional requirements. The algorithms proposed in the literature can be sorted into three categories:

Heuristic selection: Green and Krieger established the preference-based conjoint analysis to identify and evaluate the new product concept in a heuristic procedure (Green & Krieger 1985). The conjoint-based analysis aims to identify the most influential attributes of the product on decision making by showing the combinations of different attributes to the survey respondents. This process results in combinatorial optimization problems because typically discrete attributes are used (Kaul & Rao 1995). This approach is utilized to investigate the influence of product attributes/commonalities on the customer decisions, which implies that the method is only applicable when the hidden rules in customer decisions are involved in commonality design. Tarasewich and Nair pointed out that not only the customer preference but also the engineer concerns have to be considered for applying the conjoint analysis (Tarasewich & Nair 2001). This approach involves normally the heuristic analysis of the customer psychology, which is too ambitious to be quantified and integrated into a mathematical optimization problem.

MAXIMIZE COMMONALITY W.R.T FUNCTIONAL REQUIREMENTS: The product design or commonality is optimized with respect to the performance requirements by means of genetic algorithms (Balakrishnan & Jacob 1996). Nayak et al. showed the conflict between commonality and individual performance requirements and formulated a compromised Decision Support Problem (DSP) to solve the trade-off between maximizing the commonality and satisfying the variety requirement (Nayak et al. 2002). The performance requirements are formulated as constraints. If there is a clearly defined leader-follower relationship between the product performance and the commonality, e.g. the platform designers act as leaders while the customization designers act as followers, Miao et al. established a bilevel optimization process to identify the optimal production family (Miao et al. 2016).

MULTI-OBJECTIVE OPTIMIZATION: Simpson et al. categorized this problem as a certain type of multi-objective optimization to determine the best design variable settings for the product platform and individual products within the family (Simpson & D'Souza 2004, Simpson et al. 2006). The multiple criteria decision-making are based on cost, revenue, and performance. A genetic algorithm (GA) is used to automatically vary the amount of commonality and identify the Pareto front. They used the Non-dominated Sorting Genetic Algorithm (NSGA), which was proposed by Goldberg (Goldberg 1989). A set of non-dominated design points on the Pareto front are kept in each generation and assigned with the same fitness

values. By doing this, multiple optimal points with respect to different objectives can co-exist in the population (Srinivas & Deb 1994). The fitness function of the GA optimization is consisted of two parts: the function evaluating the normal product performances and the Product Family Penalty Function (PFPF). The PFPF developed by Messac et al. (Messac et al. 2002) can assess the commonality level of the architecture by calculating the variations of the design variables within the product family. The PFPF was adopted by Simpson as a separate fitness function in GA to increase the commonality. The complexity of the optimization increases when the design space of the commonality becomes discrete and combinatorial.

COMMONALITY AS CONSTRAINTS: Fellini et al. considered commonality decision as a constraint in product family design (Fellini et al. 2002). This means, a pre-defined level of commonality was required while certain individual performance losses of each product were tolerated. This type of problem formulation can be resolved by infeasibility analysis to identify the necessary relaxation of the performance to ensure the prescribed commonality (Chinneck 2008).

To conclude, the discussion of the proposed algorithms for commonality optimization, it should be noted that the optimal configuration is normally neither unique nor decisive due to the subjective preference-based process and the stochastic characteristics of the non-deterministic algorithms (in the cases regarded here, genetic

algorithms were used).

2.2 Crashworthiness Design in Early Phases

2.2.1 Low-fidelity modeling

In the automotive industry, the commonality of vehicle architectures must be investigated and decided already in the early design phase, in which the detailed design of the single component is yet to be conducted. One of the most significant functional requirements (FRs) for commonality configurations is crashworthiness. This FR is normally investigated with the help of simplified modeling considering the robustness challenge as well as the lack-of-knowledge situation in the early design phase.

One way to realize simplified modeling (low-fidelity models) for crash is based on lumped mass–spring (LMS) models, which was introduced probably for the first time by Kamal in the early 1970s (Kamal 1970) for the full-overlap front crash. The structure of the vehicle is represented by concentrated masses connected by nonlinear springs. Deformations are here only possible in the driving direction, i.e. rotations and multi-dimensional kinematics are not represented in this first study. The deformable barriers have not been included in similar approaches, except in (Trella et al. 1991). In general, these simplified models can be used for the following two application types:

1. ***LMS models with forward kinematic behavior prediction***

   The parameters of the nonlinear springs (or more advanced, the nonlinear response force characteristics of the deformation elements) are derived from physical experiments or FE simulations based on the complex models, e.g. (Kamal 1970, Lust 1992, Ni & Song 1986). This is called here ***forward prediction*** because it is obtained directly from material and geometry assumptions. In early development phases, neither experiments nor high-fidelity models are available. Hence, using predecessor vehicles or estimates to identify parameters iteratively is questionable; the quality of this iterative design process depends strongly on the accuracy of the initial determination of the parameters of the springs;

2. ***LMS models with backward parameter identification***

   The parameters of all deformation elements are varied, e.g. based on shape functions as proposed by (Kim et al. 2001), and then determined as force-displacement curves for all components such that the overall kinematic requirements (acceleration, total deformation behavior, etc.) are fulfilled (backward identification). In the original paper (Kim et al. 2001), the backwards identification led to a single force-deformation curve as the design guideline for each of the deformation elements. In the frame of Solution Spaces (Zimmermann & von Hoessle 2013), this is replaced by an interval-

based approach where upper/ lower limits (corridors) for this curve are derived. The advantages are discussed in Section 2.3

2.2.2 Crashworthiness structural design for rear impact

In the rear crash load case shown in Section 4.1.1, the deformation front representing the geometrical interaction between vehicle and barrier is more complex than in a rigid wall impact because of additional shear effects and irregular deformation patterns (fork-effect). The deformation front as well as the energy distribution are dependent on the crushing strengths of the load paths in the vehicle structure, which themselves are design parameters. The stronger the load path is, the more the barrier deforms thus the more energy the barrier absorbs. An LMS model has been already proposed in (Trella et al. 1991) to characterize the overall energy absorption and reaction force of the barrier. However, this model cannot reproduce the deformation front and fork-effect between vehicle and barrier.

2.3 Solution Spaces

Crashworthiness performance of the complete vehicle structure is determined by the physical interaction of all components. However, in the product development process, these components are developed in parallel by different working groups. Thus, the design criteria have to be derived for the modularized sub-system and further on the component level from the full-vehicle level. Validation is then real-

ized in the opposite manner, often iteratively. This procedure can be illustrated by the V-model (Figure 1.6). As proposed by Fender et al. (Fender et al. 2014, 2017), simplified mechanical models (lumped mass-spring models) for the full vehicle can be used to derive the appropriate force distribution, or more precisely upper and lower force-deformation curves (corridors) for each component, such that the overall requirements (e.g. energy absorption, low acceleration and intrusion) are fulfilled. These corridors are then optimized for example with respect to size representing design ***feasibility***, ***flexibility***, and ***robustness*** in the early design phase. They are then used as component requirements allowing decoupled development on component level. All corridors together represent a decoupled subset of the feasible design space, which is called here Solution Space.

This Solution Space approach was derived for the full-overlapping rigid barrier frontal crash test, which is the crash load case with relative simple mechanics and clear structural boundary conditions. The design criteria of the functional requirements of the components were formed into linear constraints in the design space. As a consequence, the Solution Space identification can be achieved with linear programming methods. For the load case with deformable barrier, the Solution Space is subjected to non-linear constraints and thus the method must be developed further and extended for the more complicated load case.

"We cannot solve our problems with the same level of thinking that created them."

Albert Einstein

3

Aims and Objectives

3.1 Aims and Objectives

As shown by the discussion of the previous chapter, further research is needed. Hence, to especially address computational methods and modeling for concept development, this thesis proposes methods addressing following aims and objectives.

The main aim of this thesis is: To propose a new approach to develop vehicle architectures considering in particular commonality constraints for early phase crashworthiness design. This aim is addressed by considering the following objectives:

Objective 1: To establish a method for load path identification. For all crash load cases, the load paths through the structure are the essential design features in the early phase. The analysis of the load flow has to be complemented by a method to identify relevant load paths for different crash load cases, which consist of a set of structural components. This should be achieved from predecessor vehicle structural models or from basic geometrical considerations – the package requirements of aggregates and structural components are normally defined very early in vehicle design. The load paths should reflect the force transfer between the vehicle and barrier (deformable and undeformable) with a sufficient accuracy. The corresponding functional requirements of these load paths must be defined in

the early design phase.

Objective 2: To develop a low-fidelity modeling approach for crashworthiness. During the crash event, the main energy absorption and the protection of the safety cell for the passengers are assured by the force levels and deformation potentials along those identified load paths. The crash relevant components, which form these load paths, should be condensed in one model and used for functional design. In the early stage, due to unavailability of sufficient data, low-fidelity models representing the significant kinematics and mechanical behaviors are required here. These models should be utilized to explore the design space and identify the Solution Space of the design variables.

Objective 3: Generalize low-fidelity modeling such that load cases with deformable barriers can be considered. The existing low-fidelity approaches (e.g. Solution Space Fender et al. (2017), Zimmermann & von Hoessle (2013)) do not capture crash load cases sufficiently well where deformable barriers are involved (e.g. the FMVSS301 rear crash test, the IIHS side impact test, the Euro NCAP offset frontal impact test). Hence a corresponding approach to generalize the low-fidelity model dealing with various test boundary conditions has to be developed.

Objective 4: To derive a method for uncertainty consideration with respect to the lack-of-knowledge situation in early development phases. This is necessary because in early design stages, the knowledge on the exact geometries and package weight distribution of the vehicle to be developed later is limited. Hence, an approach is required, using the extended low-fidelity model addressed in Objective 3, which allows to capture necessary flexibility for later decisions and offers sufficient accuracy and efficiency in the early stage. This should be based on results from the first three objectives.

Objective 5: To extend the Solution Space approach enabling consideration of commonality conditions in vehicle architecture. The Solution Space approach has been established to allow decoupling of structural components and to break down full vehicle requirements to component objectives. For this, a modular representation of the vehicle structure is needed, which has now to be transferred to address a complete vehicle architecture. Hence, a method to modularize the functional requirements of the components should be developed.

Objective 6: To derive an evaluation method for relevant commonality configurations. Based on the developments addressed by the objectives above, a new approach has to be proposed, which allows to evaluate various commonality configurations of an architecture. For this, a comparison criterion has to

be established, which identifies the adequate or even optimal commonality configuration w.r.t. the functional requirements and design flexibility. This includes a method to study the complexity of the problem itself together with a complexity handling approach.

The methods and approaches in the following chapters will address the objectives stated above: Chapter 4 introduces a work-flow to analyze the load paths and extract the components relevant for crashworthiness. Based on this, a low-fidelity model considering a deformable barrier impact with the available premises and data in the early design phase is established. The Solution Space of the design variables in the low-fidelity mode is, thusly, identified by an optimization process. In Chapter 5, the approach is extended to deal with the Solution Space of a vehicle architecture. The optimal commonality configuration of the architecture w.r.t. design flexibility and efficiency is discussed as well. Chapter 6 demonstrates the results of several exemplary examples. The critical discussion together with outlook, and the conclusions of this approach are given in Chapter 7 and Chapter 8 respectively.

In one word, an approach to identify the feasible/optimal commonality configurations of vehicle architecture is provided and discussed in this thesis.

"The most practical solution is a good theory."

Albert Einstein

4

Early Phase Design for Single Vehicles

The overall approach proposed in this chapter involves the processes described in Figure 4.1.[i] The design targets/criteria that the entire vehicle structure assembly must fulfill are broken down onto the subsystem or component level. The essential steps are discussed and developed in the following section. In principle, the approach is proposed for all crash load cases. However, because of the high number of relevant crash tests, it is discussed here exemplarily for the rear impact case. In this case, most of the relevant aspects occur, in particular how to embed deformable barriers into the procedure.

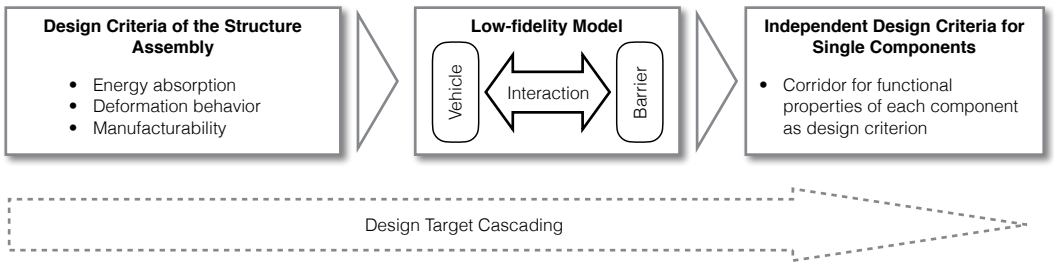


Figure 4.1: Main process proposed in this work to cascade the design targets in the early design phase from the full vehicle structure level to component level.

4.1 Configuration and Requirements of the Load Case

The crashworthiness of the vehicle structure must be assessed in various predefined load cases, involving different impacting directions, loading positions, and crash velocities. Additionally, the deformability of the barrier, when involved, has significance on the vehicle deformation. As a consequence, it is hard to generally

[i] A part of the developments summarized in this chapter has already been published by the author in (Song et al. 2017).

establish a standard design process, which properly works for all load cases. Hence, the approach is illustrated by regarding the load case defined for rear impacts. It is the high-speed crash test with the highest impacting velocity and of sufficient complexity because of the necessity to consider the role of a deformable barrier. Therefore it represents most of the relevant mechanical aspects of the established crash tests. For the crashworthiness design of vehicles under rear impacts, the requirements described in the next sections must be fulfilled.

4.1.1 Requirement I: Fuel System Integrity in high-speed Impacts

In order to reduce the fatalities and injuries caused by post-crash fires, the U.S. National Highway Traffic Safety Administration (NHTSA) issued the Federal Motor Vehicle Safety Standard 301 (FMVSS301). This standard prescribes requirements concerning the fuel system integrity after a rear-end collision (Parsons 1990). The FMVSS305 specifies further requirements for limitation of the electrolyte spillage from electric energy storage devices and prevention of electric shock during and after a crash in which electric-powered vehicles are involved (N.N. 2008). In the test, the rear end of the subject vehicle is struck by a 1,368 kg moving deformable barrier (MDB) at 80 km/h, with 70 percent overlap (Pai 2014). The scenario is depicted in Figure 4.2. The standard regulates post-crash behavior of the fuel tank. In crashworthiness design, this full vehicle requirement is interpreted into structural design criteria - the critical components (tank, high-voltage

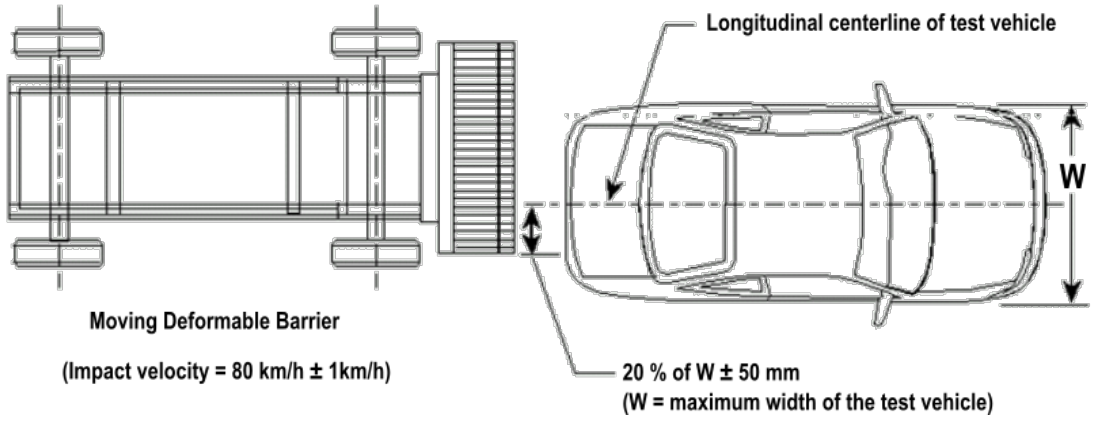
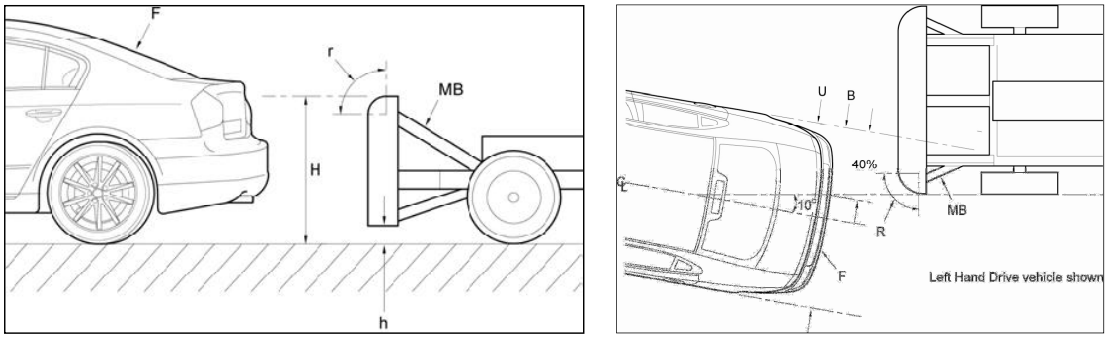


Figure 4.2: The FMVSS301 Moving Deformable Barrier Rear Test - 70% Overlap (Pai 2014).

battery, etc.) must neither be intruded nor overloaded during the crash, i.e. force and intrusion constraints are defined.

4.1.2 Requirement II: RCAR (Insurance) for reparability

The Research Council for Automobile Repairs (RCAR) has established a series of tests to assess a vehicle's damageability and reparability. One of these test scenarios is a bumper test with a relatively low speed (Figure 4.3) - a rigid-faced mobile barrier impacts the rear of the stationary vehicle at 15 km/h with an angle of 10° (N.N. 2011). After the crash, the plastic deformation zone of the structure must be limited to certain components for the low-cost reparability.



Key : MB = Mobile Barrier, H = Barrier height (700mm±10mm), h = Ground clearance (200mm±10mm), F=Test vehicle, R=Constant radius (150mm), r=Constant radius (50mm), U=40% Overlap, B=Vehicle width

Figure 4.3: The RCAR bumper rear test: rigid barrier with impact velocity of 15 km/h (N.N. 2011).

4.1.3 Requirement III: Reduction of whiplash injury

The high acceleration during a crash leads frequently to occupant injuries. In rear crash, the danger of whiplash injury is relatively high. Whiplash describes specific injuries of the neck caused by a sudden sequence of retraction and extension of the neck due to acceleration/deceleration. One of the most widely used criteria to evaluate the whiplash injury is the Neck Injury Criterion (NIC) proposed by Boström et al. (Boström et al. 1996). The NIC is defined to be of the form:

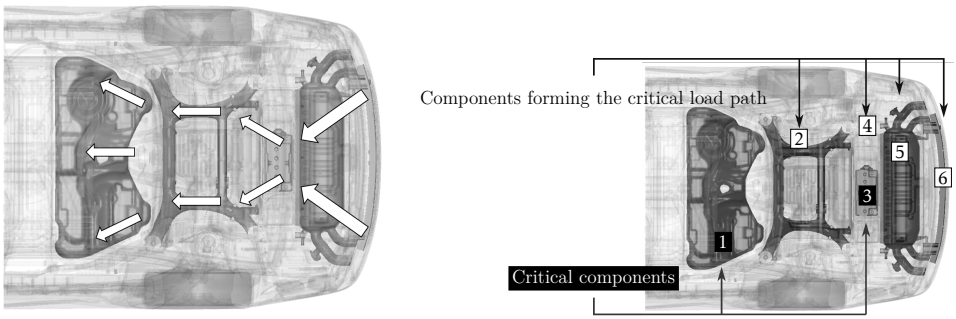
$$\mathbf{NIC} = a_{rel} * 0.2 + v_{rel}^2 \quad , \tag{4.1}$$

where a_{rel} is the relative acceleration and v_{rel} is the relative velocity between $C1$ (top of the cervical spine) and $T1$ (bottom of the cervical spine) in the horizontal direction.

Eichberger et al. confirm, without deeper analysis, the correlation between NIC and aspects related to the human body and seat (head restraint position, head and torso accelerations, head angulation, head angular acceleration, and neck torque) (Eichberger et al. 1998). Then they stated as well a relationship to the vehicle response in rear-end collisions, including vehicle velocity change (ΔV) and crash pulse. Krafft et al. investigated the influence of crash severity (change of velocity and change of acceleration) on the neck injuries (Krafft et al. 2000).

In the earlier stages of the development, no detailed information on dummy or human position or seat design is available. Therefore, the kinematics and especially the acceleration of the vehicle should be considered in the structural design.

Apart from that, the seat which pushes the occupant forward may break suddenly if the inertia force exceeds the maximum permissible loading of the seat. Without the constraint from the backrest of the seat, the occupants move randomly in the compartment and may undergo penetrating and crushing injuries. This is a second reason why the acceleration during the crash must be restricted. To enable structural development, the overall vehicle crashworthiness requirements derived above are transferred to structural criteria in the next section.



(a) Critical load paths. (b) Components in critical load paths.

Figure 4.4: Components layout in the critical load path: (1) - fuel tank, (2) - subframe, (3) - battery, (4) - battery tray and floor panel, (5) - exhaust system, (6) - bumper

4.2 Design Criteria on Structure Performance

In the rear–end of the vehicle, several components are designed to absorb the kinetic energy of the barrier mainly by plastic deformation and thus fulfill the requirements stated in Section 4.1. These components form several load paths aligned with the driving direction. The load paths containing the critical components, e.g. fuel tank or battery, are denoted as critical load paths, as shown in the left image of Figure 4.4. The kinetic energy of the impacting barrier must be absorbed sufficiently by plastic deformation and eventually fracture without leading to critical intrusions into the area of the critical components. This general requirement is detailed in the following:

(1) **Maximum whiplash impulse:** In the prescribed crash test, the stationary vehicle is accelerated by the moving barrier until they reach the same velocity. During this process, the interactions between the occupants and the seats are critical – if

the seat backrest is overloaded by the inertia force of the occupant, the rotation mechanism breaks down unanticipatedly. As a consequence, the occupant suffers from the risk of whiplash injury as introduced in 4.1.3. In the early development phase, a structural criterion is required representing the seat-occupant interaction, which is the specific accident capacity ("spezifische Unfallleistung, SPUL") (Lechner et al. 2017). Since the start velocity of the vehicle is zero, the SPUL of the vehicle in a rear crash is calculated as following:

$$\text{SPUL} = \frac{V_{Vehicle}(t)^2}{t} \quad , \tag{4.2}$$

in which $V_{Vehicle}(t)$ is the vehicle velocity during crash. This value is correlated to the rotation of the seat backrest caused by the dummy device in the rear crash. The rotation is restricted to prevent sudden seat failure. Therefore, the SPUL value of the vehicle should not exceed a predefined critical value in the early phase structural design.

(2) **Maximum force level:** During the crash, the maximal reaction force in the critical load paths must not exceed the permissible load on the critical components (tank (1) and battery (3) in the right part of Figure 4.4). In early phase design, permissible loads for these non-structural components are derived by physical experiments (quasi-static or dynamic depending on the component) and related to impact forces on critical areas. Furthermore, the maximal reachable force level

of each component is limited by manufacture, weight, and material requirements. The interpretation of these aspects with respect to force levels is based on empirical estimates proposed for early phase development.

(3) **No Intrusion into critical components:** When the critical load path with the shortest free deformation length is completely deformed, the vehicle must be already accelerated to the barrier's remaining velocity. If this is not the case, intrusion into critical components will occur. The time at which vehicle velocity equals that of the barrier is defined here as "end of crash", see Figure 4.5; because of elastic rebound, it is the time where the maximal deformation occurs. This means that the vehicle structure and barrier deform both to dissipate a certain amount of kinetic energy in the impact. Neglecting rotatory terms, which is allowed here because of the large overlap of the impacting barrier, the change in kinetic energy can be estimated analytically using a completely inelastic impact model:

$$E_{total} := E^v + E^b = \frac{1}{2}\frac{M^v M^b}{M^v + M^b} v_0^2 \ . \tag{4.3}$$

Here, E^v and E^b denote the deformation energy in vehicle and barrier while M^v and M^b denote the masses respectively. v_0 is the initial velocity of the barrier. E_{total} gives hence the kinetic energy, which was transformed into deformation energy during the impact. The intrusion into the critical components can be avoided when this amount of energy is absorbed by the available deformation

components at the end of crash.

(4) **Limitation of damage and reparability:** The reparability of the structure is assessed after the crash. E.g. the RCAR (N.N. 2011) requires that the structure deforms sequentially from rear to front. This means that force levels in the outer structure have to be smaller than for the inner parts. This is called here "order-of-deformation" criterion.

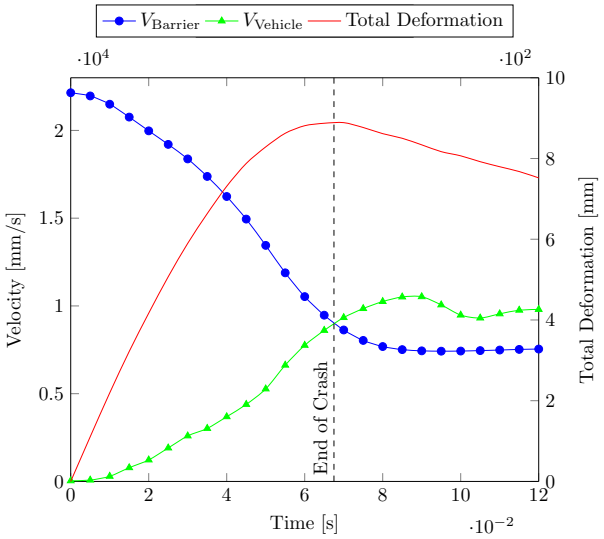


Figure 4.5: Time history of vehicle and barrier velocities showing the termination of the crash and the related maximal total deformation (sum of barrier and vehicle deformation).

4.3 Low-fidelity Model for Early Phase Design

In order to investigate all the criteria stated in Section 4.2 in the early design phase, a mechanical model should be established to relate the functional characteristics of the load paths to the major responses of the structural assembly. Load paths describe the principle load transfer behavior of the structure. They

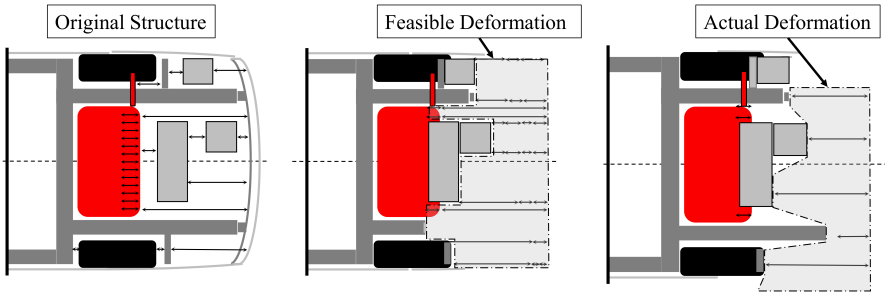


Figure 4.6: The schematic deformation behavior of the structure in rear crash (Song et al. 2017).

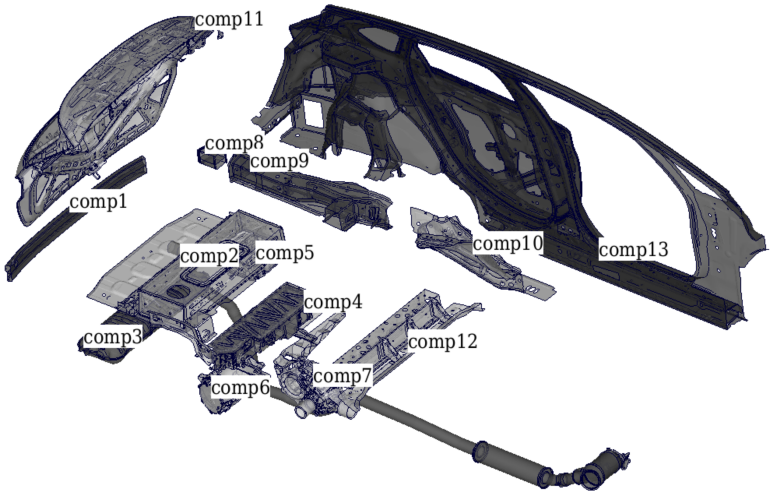
indicate the direction of the load transfer and represent the connectivity of the load carrying components. The topology and crushing strengths of components along these load paths play a significant role in structural crashworthiness. To the author's knowledge this was first discussed in (Volz 2011).

In a rear crash scenario, the final deformation state of the vehicle structure cannot be predicted alone by the available deformation lengths of the structure; the barrier and vehicle structure intrude into each other, as shown in Figure 4.6. The deformation front between the barrier and vehicle structure depends on the crushing strength of the components and contact surface against the barrier. This figure illustrates that the designed feasible deformation of some components may not be activated to absorb energy in the crash due to their higher crushing strengths against the deformable barrier. In order to model this appropriately and to extract the dominant interactions, a model reduction is proposed below to determine the crash relevant components. The reaction force of the barrier against different set-ups of the structure must be investigated as well.

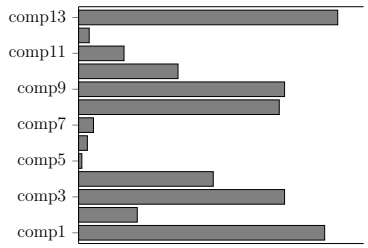
4.3.1 Low-fidelity Modeling of Vehicle Structures

Energy evaluation of the structure

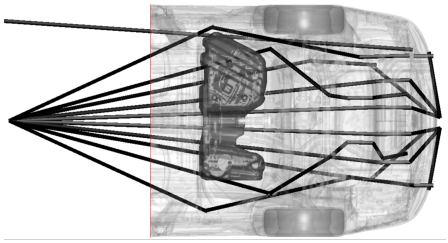
During the rear impact, the kinetic energy of the moving barrier is partially absorbed by the plastic deformations of the subjected vehicle and barrier, the rest remains as kinetic energy of the vehicle and the barrier in post-crash. In a principle simulation using a complex FE model of a prior design, the internal energy of each typical component can be evaluated and ranked. The components with relatively high energy rank are then defined as crash relevant components. Figure 4.7(b) shows an example of such an energy evaluation. A reduced FE model considering only these relevant components can be extracted and configured to represent the full scale simulation model. This reduced model is further simplified to represent only the load path topology and inertia of the load bearing structure, as shown in Figure 4.7(c). Note that the loads paths in the low-fidelity model only exist in the rear vehicle and that they are then connected to the center of gravity of the vehicle. This simplified model obtained from a generic model is then used for the new vehicle derivate imposing the same load path concept. The available deformation lengths of the rear structure along these load paths are a primary factor for energy absorption. They can be derived from the block lengths and the free space. This information is available and utilized for the crashworthiness design in the initial design phase.



(a) Relevant components.



(b) Energy distribution.



(c) Load path topology in rear crash. The long bar on the side represents the rocker and the side panel (comp13), which is also connected to the mass on the center of gravity.

Figure 4.7: The crash energy absorbed by components in the major load paths.

Load path analysis

The energy evaluation is helpful to identify the load bearing components. However, if the geometry and alignment of the component are not in the impact direction, the force flow and the interaction between the components are hard to be captured. Taking the underfloor panel structure between the longitudinal beams as

an example, it is challenging to define how the force is transferred through this component into the neighborhood components. This question cannot be answered solely by the energy evaluations.

The concept of load flow is widely used by design engineers to describe the way a structure carries applied loads from the point of application to the point of reaction in the structure (Kelly et al. 2001). Kelly et al., for example, visualized **the load flow in a given global direction** with the streamline technique of a static loading case. Based on the load path information, the load bearing structure is extracted automatically through a topology optimization process, which is derived from the method "Solid Isotropic Material with Penalisation" (SIMP) (Kelly et al. 2001, Kelly, Reidsema, Bassandeh, Pearce & Lee 2011, Kelly, Reidsema & Lee 2011, Kelly, Pearce, Ip & Bassandeh 2011). This process has been applied on structures with relative small deformations and unchanged contact situation. For crash applications with large deformations and complex contacts happening during explicit time integration, the approach established by these authors must be adapted and extended.

First, the original process is described preliminarily here to facilitate further discussion: Given a shell structure with the boundary conditions described in Figure 4.8, the loading can be decomposed into x-, y-direction under the global coordinate system. Suppose x is the impact direction, the load bearing structure for the

load coming in x-direction is defined as the load path relevant part(s) in impact direction. Kelly et al. adopted the streamlines derived normally via the Eulerian approach, used for computational fluid dynamics, into the Lagrangian domain, which is normally applied for solid mechanics.

The following operations can be conducted at each time step, when stress and deformation state of the whole structure field are available. The process is starting by choosing an arbitrary seed point P_1 near the loading boundary to start the streamline; then, the stress tensor at this point P_1 at the given time step is considered, denoted as

$$\boldsymbol{\sigma}_1 = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{bmatrix} . \tag{4.4}$$

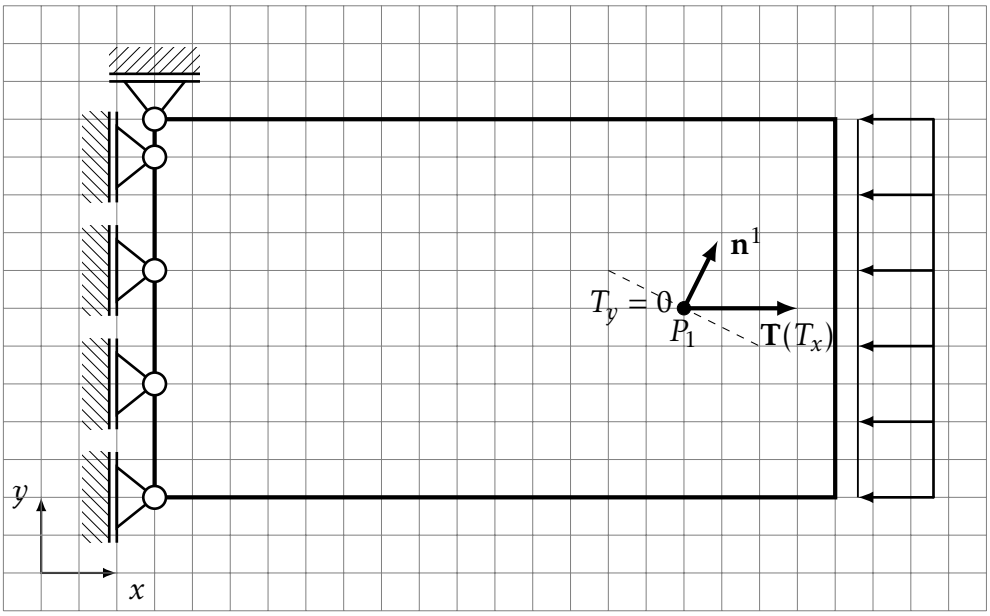
In the next step, a cutting surface is defined through P_1, on which the surface normal and the corresponding tangent vector are denoted as:

$$\mathbf{n}^1 = \begin{pmatrix} n_x^1 & n_y^1 \end{pmatrix}^T , \tag{4.5}$$

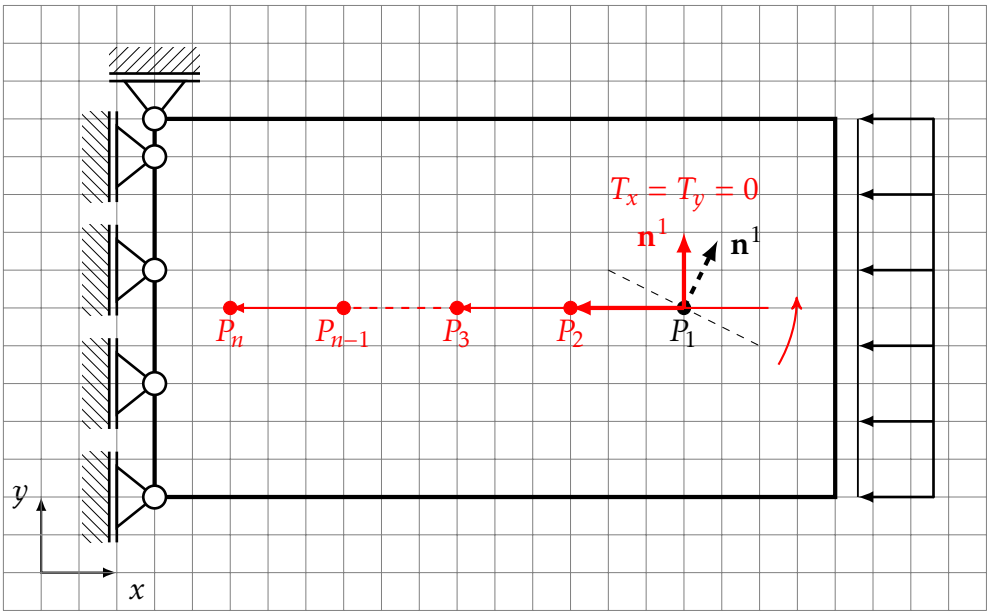
$$\mathbf{t}^1 = \begin{pmatrix} -n_y^1 & n_x^1 \end{pmatrix}^T . \tag{4.6}$$

The traction force at point P_1 on the given cutting surface can be calculated as:

$$\mathbf{T}_1 = \boldsymbol{\sigma}_1 \cdot \mathbf{n}^1 . \tag{4.7}$$



(a) Start point and load flow direction.



(b) Development of the load flow.

Figure 4.8: Direction of the load flow in global direction x.

The component of this traction force on P_1 in global x-direction is written as

$$\begin{aligned} T_{1,x} &= \sigma_{xx} \cdot n_x^1 + \sigma_{xy} \cdot n_y^1 \\ &= \mathbf{V} \cdot \mathbf{n}^1 \\ &= \Phi(x,y)|_{P_1} \cdot \mathbf{n}^1 \quad , \end{aligned}$$

in which the $\Phi(x,y) = \mathbf{V} = [\sigma_{xx}, \sigma_{xy}]$. Through this step, a vector field $\Phi(x,y)$ on the structure domain is defined.

As next step, the cutting surface is rotated so that the traction has zero component in the x-direction, i.e. $T_{1,x} = 0$. At this position, if the structure is split along the cutting surface, the two parts have no interaction in global x-direction (here the predefined analysis direction) on P_1. In addition, the vector $\mathbf{V}$ is perpendicular to the surface normal $\mathbf{n}^1$:

$$\mathbf{V} \cdot \mathbf{n}^1 = 0 \quad \Leftrightarrow \quad \frac{-n_y^1}{\sigma_{xx}} = \frac{n_x^1}{\sigma_{xy}} \tag{4.8}$$

As given by Definition 4.1, the unit direction vector of the streamline $\mathbf{x}_S$ at P_1 can be written as

$$d\mathbf{x}_S|_{P_1} = \begin{pmatrix} -n_y^1 & n_x^1 \end{pmatrix}^T \quad , \tag{4.9}$$

which aligns to the cutting surface.

Definition 4.1: Equation of streamline (White 2015)

If the components of the velocity are written $\mathbf{u} = [u, v]^T$, and those of the streamline as $\mathbf{x}(s) = [x(s), y(s)]$, the equation of streamline can be deduced:

$$\frac{dx(s)}{u} = \frac{dy(s)}{v} \quad , \tag{4.10}$$

which shows that the curves are parallel to the velocity vector.

The established process can be summarized to visualize the "force flow" in the structure in the **predefined analysis direction**, $\mathbf{n}^1$. In the situation stated above, the analysis direction is aligned with the global coordinate direction. Nevertheless, this is not necessary.

The further discussion of the difference between the load path vector field in the predefined analysis direction and the principal stress field is conducted here. The advantages of this approach for the purpose of load path analysis are the three following aspects:

- The first aspect can be illustrated by considering a structure (shown in Figure 4.9) under shear loading in x-direction. If the x-direction is the predefined analysis direction, the load flow is visualized as shown in Figure 4.9(a): the shear loading is transferred by the structure between the loading and fixed boundaries. If the y-direction is the predefined analysis direction, the

load flow can be visualized as shown in Figure 4.9(b): there are no streamlines connecting the loading boundary conditions. It is important to note that the principal stress shown in Figure 4.9(c) cannot provide the same information as the load flow does. The vector field shows only the mixed information without specification.

- The second aspect can be discussed by making the structure more complex with a cutout in the middle. The load flow indicates that the force transfers around the cutout (shown in Figure 4.9(d) and Figure 4.9(e)) while the principal stresses discontinue on the boundary of the cutout, shown in Figure 4.9(f).

- Finally, the streamline can only be generated for a vector field, e.g. the pointing vector $\mathbf{V}$; however, it is not feasible for the principal stress. Since the principal stress has no sense of the direction (i.e. the principal directions or the eigenvectors of the stress tensor only provide the orientations) and thus no unique integration direction for streamlines.

Nevertheless, this approach has two main shortcomings which must be overcome in the practice. Firstly, the established approach indicates both, the load flow in the predefined analysis direction (the real load path) and the irrelevant load flow vortices (local bending, torsion effect). In the real application, the structure always has regions, in which bending effects dominate. The curl of the vector field

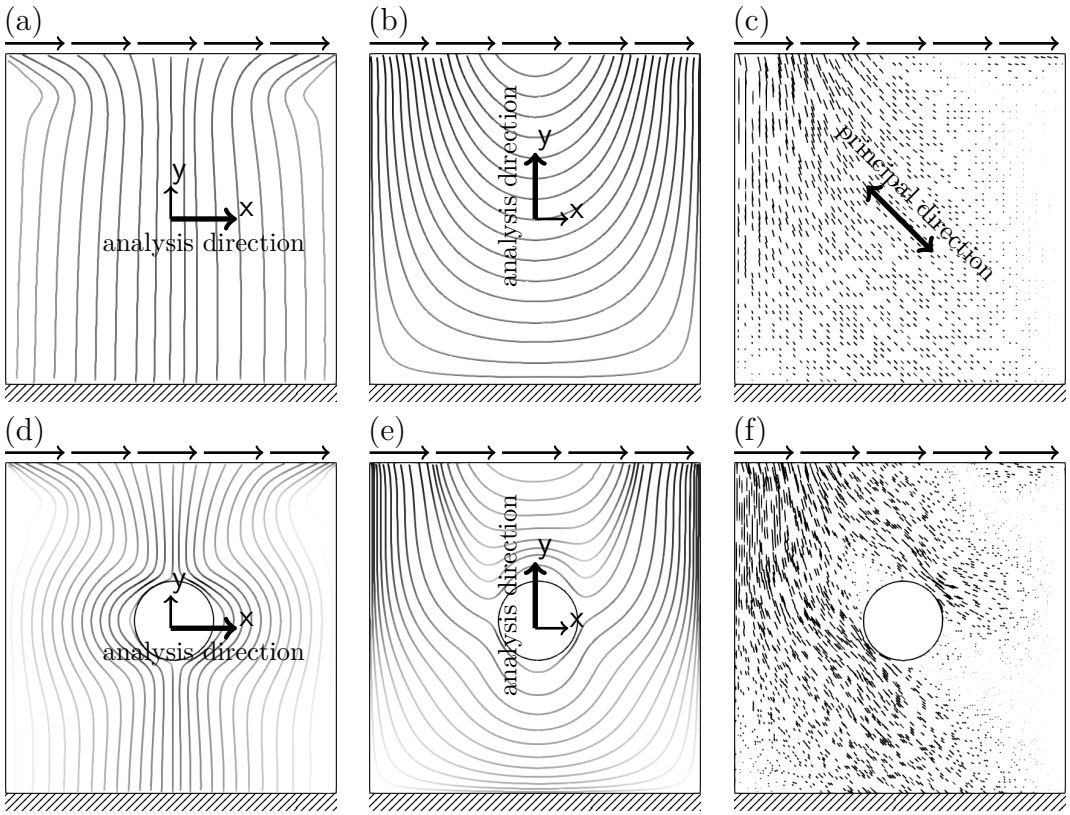
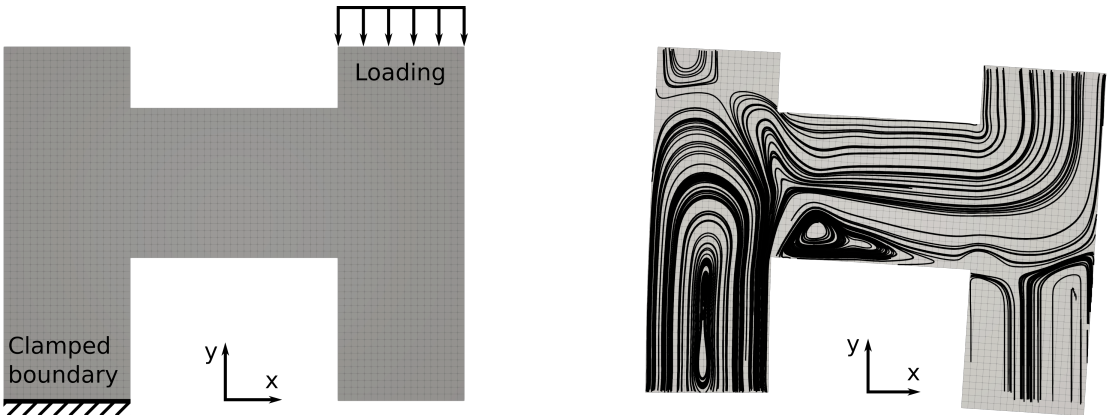


Figure 4.9: Comparison of two different approaches for the definition of load paths for two plates, with (d–f) and without (a–c) a central hole, under shear load on upper edge. Plates (a) and (d) represent the load paths with respect to the x-direction as predefined analysis direction, plates (b) and (e) represent the load paths with respect to the y-direction as predefined analysis direction, and plates (c) and (f) represent the visualization of the principal stresses.

$\nabla \times \Phi(x, y)$ is thus nonzero, which leads to the vortices of the streamlines in these structure domains. The swirling motion of the load flow distracts the load path analysis.

Secondly, the choice of seed points of the streamlines is crucial to capture the load path. If the seed points can be properly populated on the loading boundary, the load flow can be captured by the streamlines. However, for components involved in crash, their boundary conditions are changing with the unpredictable contact



(a) The description of the boundary condition.

(b) Streamlines to visualize load path visualization.

Figure 4.10: The load path visualization with y-axis as predefined analysis direction.

situations. It is hardly possible to cover all the locations with seed points efficiently. The improperly distributed seed points make the load path identification counterintuitive due to the quality of the streamlines, shown in Figure 4.10(b). For application in more complex geometries and load situations, the seed points must i) be of sufficient number to capture the complete force flow, especially for a load flow through contact areas created during the crash; ii) avoid generating too many vortices as stated above; iii) have a reasonable distribution (not too locally crowded) to restrict the computational efforts in the post-processing.

To overcome these two shortcomings for applications, the visualization technique can be enhanced in the following ways:

To meet these requirements, the dual streamline seeding technique introduced by Rosanwo et al. is adopted (Rosanwo et al. 2009). The authors made use of an auxiliary vector field (dual field) that is perpendicular to both the surface normal

and the vector field to be visualized (primal field). The roles of the dual and primal fields are alternately interchanged in an iterative refinement process: At each step, the intersections of all the field lines that have been calculated thus far are considered, and the seed point for the next primal (dual) line is placed right in the middle of the longest uninterrupted dual (primal) line segment. The iterative process is terminated when the lengths of the longest uninterrupted segments fall below a given threshold. This threshold controls the density of the streamlines in the visualization. To avoid clustering of the streamlines, Rosanwo et al. terminate the streamline integration if its distance to neighboring lines at an intersection is smaller than a threshold (Rosanwo et al. 2009).

This approach is adopted with modification here to improve the seeding algorithm: the algorithm is initialized by using a single dual line seed point for simple geometries and multiple evenly spaced seed points for more complex geometries. Multiple initial seed points reduce the risk of the algorithm "being stuck" in a region where stresses are ill-defined (with small magnitude), and it ensures that all structural parts are covered by the field lines. In complex case, the distribution of the multiple initial seed points may need to be examined with several attempts to explore the load flow in the whole structure domain. Additionally, the integration to generate the field line is not terminated when this line is getting close to the neighbors so that the complete generated field line can indicate the start and end

of the load path.

The generated streamlines are thus well stretched, more evenly distributed and almost independent of the start seed points, shown in Figure 4.11.

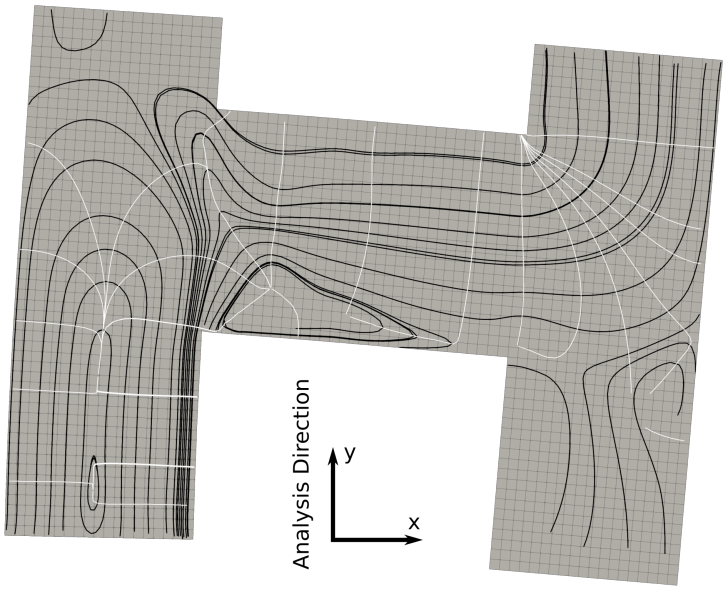


Figure 4.11: The load path visualization with y-axis as predefined analysis direction. The white lines are the dual streamlines used to optimize the distribution of the black streamlines for load flow.

In spite of the evenly distributed long streamlines, the vortices caused by the moment are still distracting. In order to get rid of this ambiguity, an approach is established here to categorize the streamlines.

For a specific streamline in the vector field, the pointing vector $\mathbf{V}$ and the predefined analysis direction $\mathbf{n}$ are evaluated in a given formula:

$$\varphi = \frac{1}{L}\int_0^L \frac{\sqrt{\mathbf{V}\cdot\mathbf{V} - (\mathbf{V}\cdot\mathbf{n})^2}}{|\mathbf{V}|} \quad \in [0,1] \ , \tag{4.11}$$

where L represents the total length of the streamline. For $\varphi = 0$, the streamline flows always in the analysis direction, which is the case shown in Figure 4.8. Otherwise if $\varphi = 1$, the streamline goes orthogonally to the analysis direction, which means only shear stress w.r.t. the analysis direction exists. Here, the shear effect is not of importance for the load path analysis, since it does not indicate the transition of the force in the analysis direction inside the structural domain between the boundaries.

For the streamline with a shear fraction, $\varphi < 1$, the sense of direction of the streamline is analyzed in the following. We define the **load path type index** for one single streamline as:

$$\lambda = \frac{L^+ - L^-}{L^+ + L^-} \quad \in [-1, 1] \ , \tag{4.12}$$

with

$$L^+ = \int_0^L (\mathbf{V} \cdot \mathbf{n})^+ ds, \ L^- = \int_0^L (\mathbf{V} \cdot \mathbf{n})^- ds \ , \tag{4.13}$$

where the $()^+$ and $()^-$ denote the positive and negative values respectively:

$$(X)^+ = \begin{cases} X \ \ \forall X > 0 \\ \\ 0 \ \ \text{otherwise} \end{cases}, \ (X)^- = \begin{cases} -X \ \ \forall X < 0 \\ \\ 0 \ \ \text{otherwise} \end{cases} . \tag{4.14}$$

With this definition, the streamlines which form closed vortices (bending load)

result in $\lambda = 0$, since $L^+ = L^-$. The streamlines which flow along the analysis direction (tension), result in $\lambda = 1$, since $L^- = 0$. Analogously, the streamlines, which flow against the analysis direction (compression), result in $\lambda = -1$, since $L^+ = 0$. All the streamlines with $\varphi < 1$ are combinations of the three fundamental cases. In the visualization, the streamlines are color-coded according to the λ value, and thus categorize the load path type within their flowing domain.

Figure 4.12 shows the improvement of the load flow representation with the help of dual prime seeding and load path type index λ.

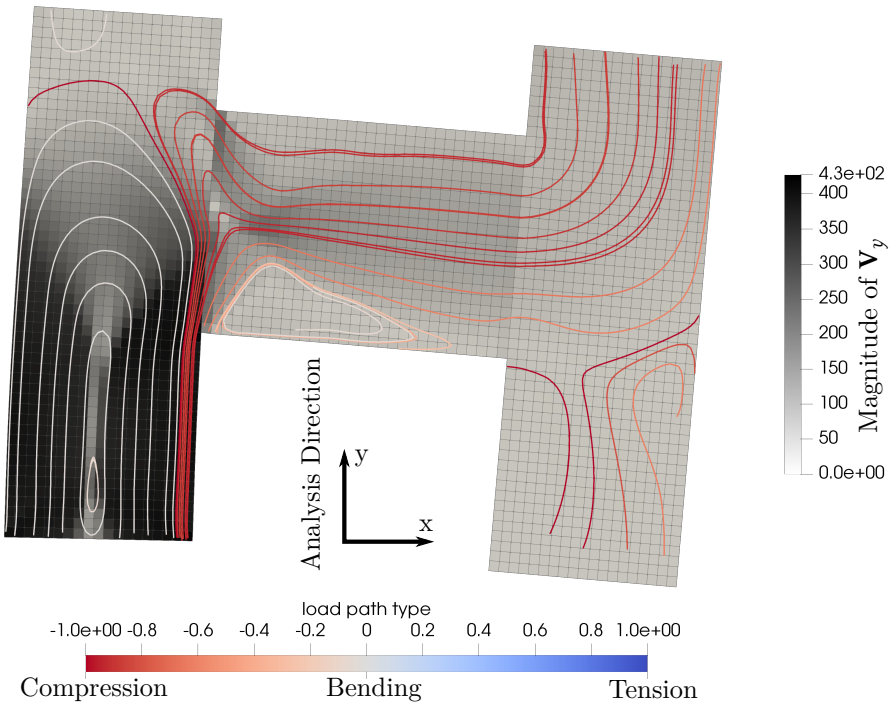
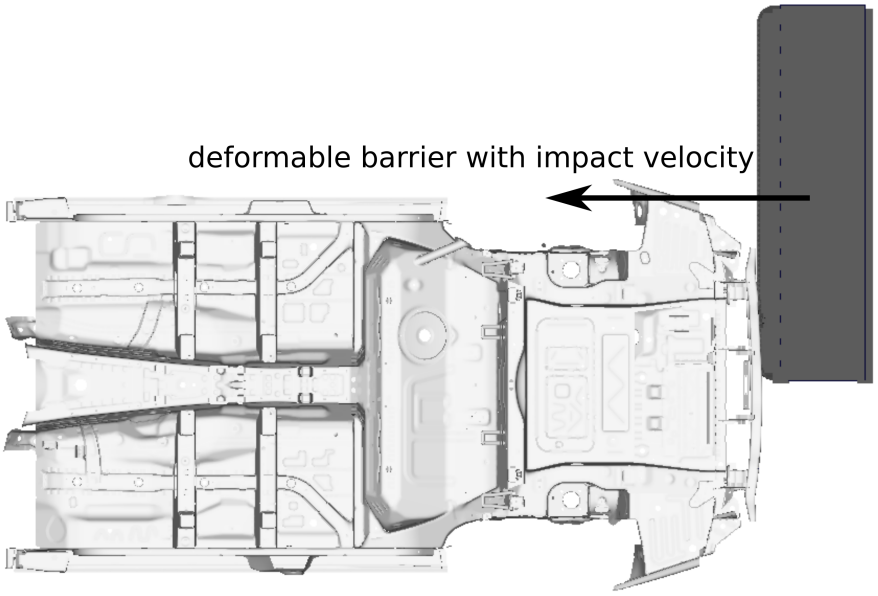


Figure 4.12: The load path visualization with y-axis as predefined analysis direction. The blue-red color scalar depicts the load path type index $\in [-1, 1]$. The red lines indicate how the compression load is transferred in y-direction between the boundaries. The white field lines indicate the bending moment.

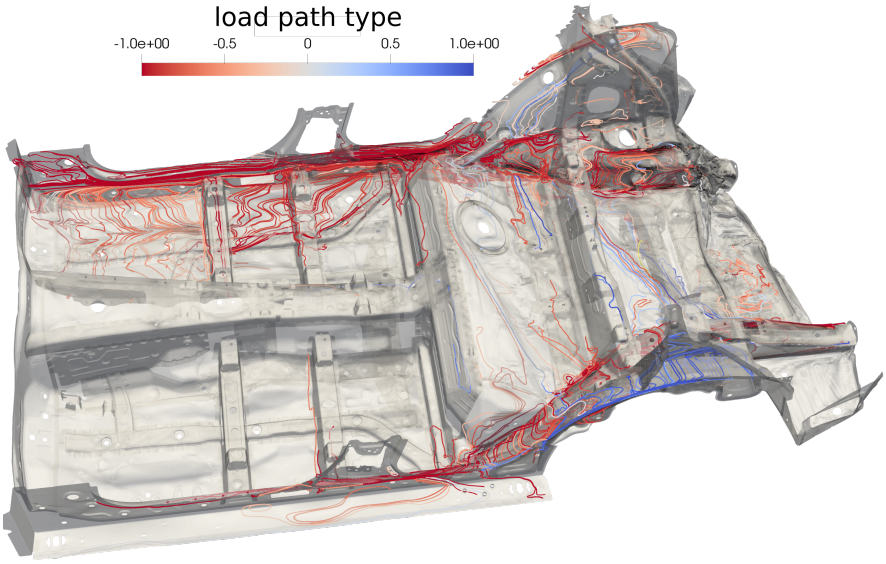
For the explicit crash simulation, this visualization can be performed at each time

step if the stress information of the entire structure is available. To clarify the proposed approach, a small example is regarded here. Figure 4.13 shows the load flow of a floor assembly in rear crash at a certain deformation state. Based on optical information in the visualization – the flowing direction of the streamlines after filtering based on a threshold intensity – it can tell that most of the load is transferred through the longitudinal beam on the impacting side into rocker and slightly into the cover of the oil tank, the second dominant load path is through the other longitudinal beam. Therefore, these components out of the chassis floor assembly should be taken for the low-fidelity modeling. A further detailed analysis of the result is demonstrated in Chapter 6.

After identification of relevant components on the defined load paths and their available deformation lengths, the functional requirements for these load paths have to be set up. This is only possible if a simplified barrier model is available, which is described in the next section.



(a) The description loading condition.



(b) Streamlines to visualize load paths.

Figure 4.13: The exemplary load path visualization of the rear impact.

4.3.2 Simplified modeling of deformable barriers [ii]

Assumptions for simplification

In the crash test involving a deformable barrier, the intrusion into the vehicle structure is decided by the deformation front between vehicle and barrier. For an undeformable barrier, this deformation front is just determined by the geometry of the barrier, e.g. in U.S. NCAP full frontal crash test, the vehicle structure is compressed by the rigid barrier homogeneously, the deformation front is thus simply the barrier plane. For deformable barriers, however, the deformation front is determined by the interrelationship of regional crushing forces between the vehicle structure and deformable barrier. Therefore, in order to obtain the barrier crushing forces corresponding to the impacting region into the structure, the following approach is proposed.

A moving deformable barrier (MDB) is regarded in an exemplary manner as impactor; the approach can be transferred to stationary deformable barriers. A detailed research on the crushing force and energy capacity of the barrier under specified impact conditions is therefore given in the following. The resistance force-deformation behavior of the barrier is used later as constraints for the design of the vehicle structure. These constraints can be determined prior to the vehicle structural design under the following assumptions:

[ii] This section is partially published in (Song et al. 2017)

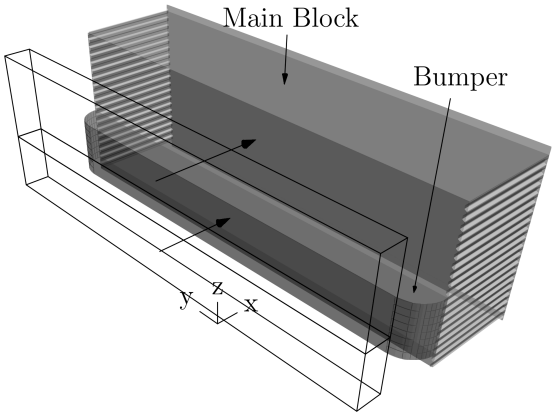
(1) Position and dimensions of the contact surfaces between vehicle and barrier can be estimated sufficiently well. When the packages of the vehicle are defined in the initial design phase, the positions of the load bearing structure/components towards the barrier are normally predictable. Together with the cross-sections of these components located in the rear-end, the contact surfaces of the load paths on the barrier can be estimated. The detailed shape information of these cross-sections is not required.

(2) Different contact surfaces deform the barrier independently and their interactions can be neglected. In case that several impactors crash simultaneously into the same barrier, the deformation caused by one impactor spreads out and may overlap with the deformation region of a second impactor. Although this can influence the interaction force between the impacting structures and the barrier, this effect is neglected here.

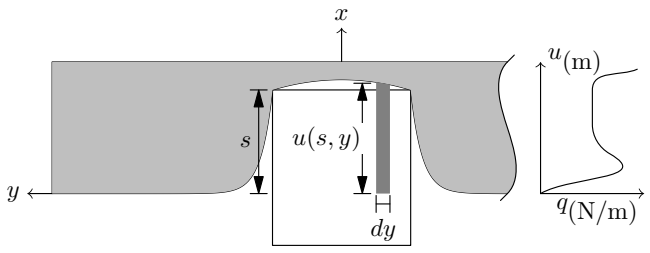
The second aspect may be in particular relevant for the stiffer barrier structure representing the bumper, because interaction may be stronger. Nevertheless, in the rear crash case regarded here, the barrier bumper is normally impacted by only a single load path such that the interaction of independently intruding vehicle structures becomes less important.

In addition, for the main block of the barrier, the shear resistance of the honeycomb structure mainly responsible for this interaction is of minor importance.

Regarding this more in detail, in the reality the load paths always have certain offsets in impact direction due to component positioning and components' crushing strengths. These offsets result in an impacting sequence of the load paths on the barrier, i.e. a simultaneous impact of multiple load paths is therefore not occurring and the second assumption is justified to the opinion and experience of the author. With these two assumptions, a method used to calibrate the force-deformation curve of the barrier under specific impact situations is proposed in the following.



(a) Fully overlapping impactor.



(b) Approximation of the barrier reaction force (in $-x$ direction) based on deformation front estimation. (top view of the barrier)

Figure 4.14: Impactor test of the barrier with fully and partially overlapping impactors.

Resistance forces of barriers

The crushing strengths of the barrier bumper and of the main block are obtained by conducting a quasi-static simulation with a fully overlapping rigid impactor against the corresponding parts, see Figure 4.14(a). The resistance forces $F(u)$ as function of the barrier deformation u are measured on the contact area until the barrier deformations reach the final state. The resultant resistance per unit length over barrier width is thus calculated by:

$$q(u) = \frac{F(u)}{L} \ , \tag{4.15}$$

in which L is the dimension of the impactor in y-direction. If the impactor intrudes the barrier by s, the shape of the deformation front over y is represented by the function $u(s,y)$; note that the aluminum honeycomb structure is transversely isotropic. The overall resistance force of the barrier against the impactor is estimated by integrating the unit barrier resistance force over the deformation front:

$$F(s) = \int_{y_r}^{y_l} q\,(u(s,y))\,dy \ . \tag{4.16}$$

y_r and y_l denote the right and left boundaries of the barrier respectively. The deformation of the barrier, $u(s,y)$, can be approximated by the following piecewise function (Song et al. 2017):

$$u(s,y) = \begin{cases} s \cdot \exp(\frac{y-y_-}{\alpha}) & \text{if } y < y_- \\ -\gamma y^2 + \gamma(y_- + y_+)y - \gamma y_- y_+ + s & \text{if } y_- \leq y \leq y_+ \\ s \cdot \exp(\frac{-y+y_+}{\beta}) & \text{if } y > y_+ \end{cases} \tag{4.17}$$

The coefficients α, β, γ, which are to be identified, describe the spreading of the deformation in the neighborhood of the direct contact surface: the exponential shape function used outside to describe the spreading of the intrusion while a quadratic function used to describe the deformation front. These coefficients are determined by the contact position and barrier properties. For example, the contact surface near the boundary leads to a wider propagation zone, as shown in Figure 4.15(d). The y_- and y_+ denote the right and left corner position of the contact surface. To model the coefficients, a response surface model, $\mathcal{R}$, is thus built to explore the relationship between the coefficients and the contact geometry:

$$[\alpha, \beta, \gamma] := \mathcal{R}(A, P) \ . \tag{4.18}$$

Here, A contains the area of the estimated rectangular impacting surface and the edge lengths of the rectangle while P is the center position of the impactor (given in coordinates). Combined with Equation (4.16), the reaction force-deformation curve of the barrier can be predicted from the contact geometry and position (A

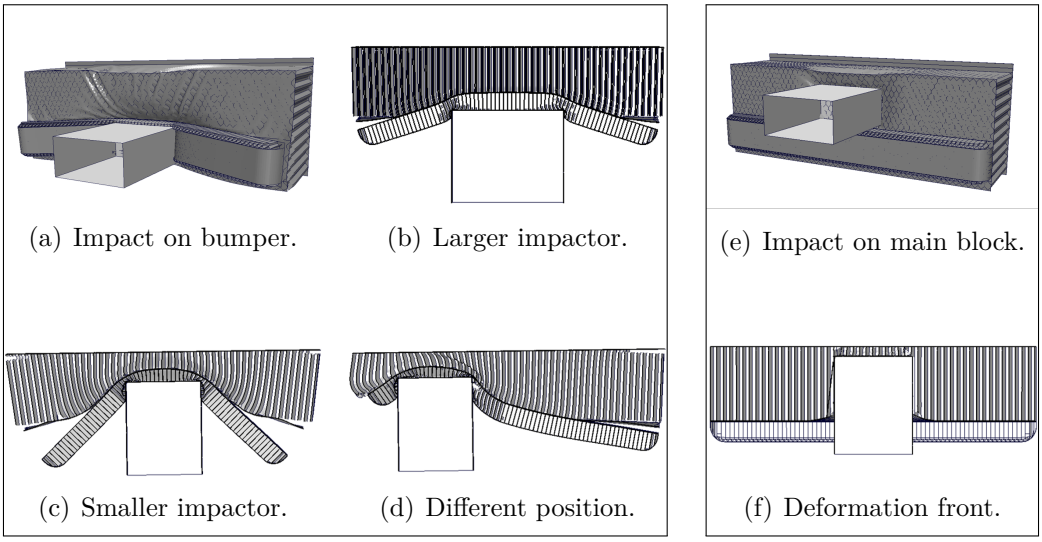


Figure 4.15: The variation of contact size and position (parameters A and P in Equation (4.18)) illustrating their influence on the deformation front.

and P):

$$[A,P] \overset{\mathcal{R}(.)}{\Longrightarrow} [\alpha,\beta,\gamma] \overset{Equation\ (4.17)}{\Longrightarrow} u(s,y) \overset{Equation\ (4.16)}{\Longrightarrow} F(s) \ . \tag{4.19}$$

4.3.3 The interaction between vehicle and barrier

Macroscopic kinematics

In the crash event, the contact force between vehicle and barrier accelerates the vehicle, decelerates the barrier in x-direction and causes the deformations in both. This contact force lasts until the two crash participants reach the same velocity. We consider this time point as the end of the crash event. The crash participants' motions are depicted in Figure 4.5. As defined by the FMVSS 301, the impact is

slightly eccentric, however, until the end of the crash event the rotations in x-y plane ($\leq 2°$) are negligible, because of the large overlap of 70%, i.e. the small offset of the barrier.

Deformation in load paths

The change of the distance between the vehicle's center of gravity and the rigid back plate of the barrier is defined here as the total crash deformation, denoted as D. The deformation of the j-th load path in the vehicle is d_j, its corresponding deformation in the barrier is d_j^b and δ_j, if any, is the gap in between. j is the index of the load path and N_{LP} is the total number of load paths considered. Since the rotations are neglected, the local deformations between load paths should fulfill the following condition:

$$d_j + d_j^b + \delta_j = D, \quad \forall j \in \{1, 2, \ldots, N_{LP}\} \ . \tag{4.20}$$

This condition is explained in Figure 4.16.

Additionally, when the contact geometry is determined, the force over intrusion curve of the barrier under this contact surface is determined by the approach proposed in Equation (4.19). Based on this curve, the intrusion into the barrier depends on the crushing force of the components in the corresponding load path on the vehicle side. The maximum force, which the components in j-th load

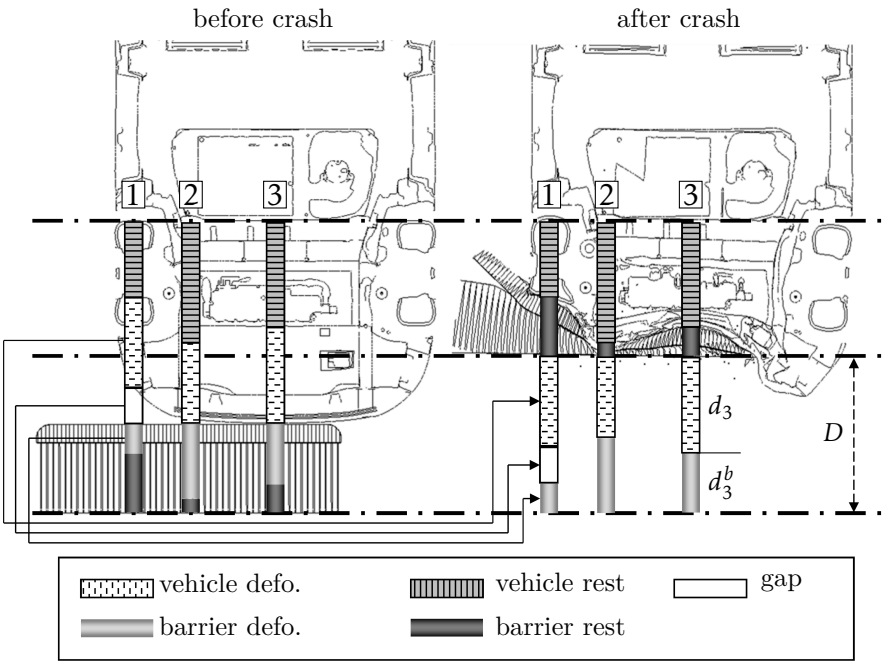


Figure 4.16: The relationship of intrusions among different load paths.

path can bear, is denoted as $\max F_j$. The intrusion into the barrier caused by the j-th load path can be calculated using the inverse of the force-deformation relationship defined in Equation (4.16) – the intrusion into the barrier can be estimated through the peak value of a given impact force:

$$d_j^b = s = F^{-1}(\max F_j) \ . \tag{4.21}$$

Combined with Equation (4.16), we can derive now that the energy absorbed by the barrier area with respect to the j-th load path (E_j^b) is related to the maximum

impact force in this load path.

$$E_j^b(\max F_j) = \int_0^{d_j^b} F(s)ds = \int_0^{F^{-1}(\max F_j)} \int_{y_r}^{y_l} q\,(u(s,y))\,dyds \ . \tag{4.22}$$

When the contact geometry is determined, the kinetic energy absorbed by the barrier under this contact surface can be predicted by providing the maximum contact force level, which is determined by the vehicle structure.

In the entire approach, the simplified model of barrier takes the crushing forces of the vehicle structure as input and predicts the barrier intrusion depth and the absorbed energy. The rest of the energy defined in Equation (4.3) must be absorbed by the deformable vehicle structure. This rest energy, which is dependent on the vehicle crushing force – the design parameter, delivers a nonlinear boundary condition of the structural design. The structural design process is discussed in the following.

4.3.4 Solution Space for early phase design of the structure

The deformation space model(DSM) of the structure

During the rear crash, the dominant momentum change happens in the driving direction (x-direction). As a consequence, the crushing strengths of the load paths in x-direction have significant influence on the crash performance. Therefore, we

define the functional requirements for the components only in x-direction and neglect the minor contributions in the other directions. This means we focus on the requirements for force-deformation curves of the components in x-direction. Since only the deformable parts of the structure contribute to energy absorption, a so-called deformation space model(DSM) is built based on the simplified model to summarize all the available information in the initial design phase (Lange et al. 2018):

- The lengths of the bars represent the available deformation lengths of the components;
- The topology of the bars indicates the load flow in the structure;
- The bars can be connected transversely to indicate that the load flow can be transferred from one path to the next;
- The contact surfaces between vehicle structure and barrier are approximated as rectangles, which do not change during the crash;
- The mass is distributed discretely onto concentrated points;
- The bars are discretized into sections; within one section, the reaction force of the bar in x-direction is assumed to be constant, neglecting the elastic reactions and considering only plastic deformations without hardening and failure.

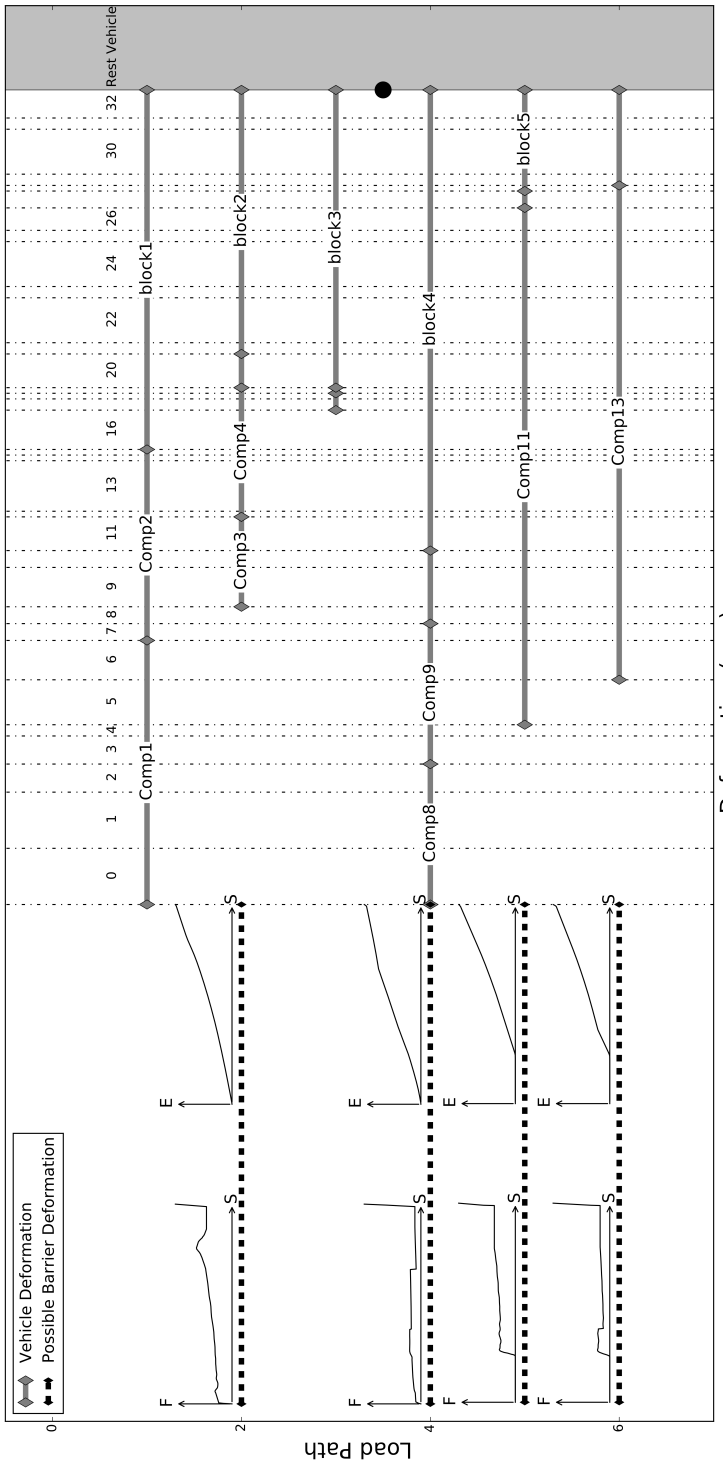


Figure 4.17: Deformation space model: in this example, the curves on the left hand side are the predicted force/energy over deformation of the barrier under the assumed contact surfaces; the bars on the right hand side illustrate the topology and components of the load paths. The vehicle load path 1, 2 and 3 (the three load paths on the top) impact onto the same contact surface.

Constraints of the Solution Space boundaries

The main goal for the structural crashworthiness design in the initial phase is to set up the target force-deformation characteristics for components located in load paths. However this goal is ambitious due to insufficient information. As introduced in Chapter 2, the Solution Space method proposed in (Fender et al. 2017)

- establishes a robust, compatible and directive guideline for component design for the U.S. NCAP front crash, in which the constraints are mainly linear.

- decouples the interdependency of the component designs, such that the components can be designed/optimized independently. When each component fulfills its local functional goal defined by the Solution Space, the objectives for the overall vehicle regarding crashworthiness design are consequently fulfilled.

In the following section, an approach is set up to identify the Solution Space for the vehicle structure in crash against deformable barrier. The established approach is to find the largest possible Solution Space (axis parallel hypercube) under linear constraints. This increases the flexibility and enables to react later in the development process such that changes in one single component can be realized without affecting the design of other components in the structure assembly. Here, due to the nonlinear influence of the barrier, the approach must be extended to identify

the Solution Space under the nonlinear constraints formulated in Section 4.2. The intervals of the force-deformation curves are the unknowns to be identified in the solution process. $F_{u,ij}$ and $F_{l,ij}$ are used to denote the upper and lower boundaries of the feasible interval for the force level in section i, load path j. The $\mathbf{F}_u$ and $\mathbf{F}_l$ are the corresponding vector forms, i.e.

$$\mathbf{F} = [\mathbf{F}_l, \mathbf{F}_u] \quad \text{and} \quad F_{l,ij} \leq F_{u,ij} \ , \tag{4.23}$$

where $F_{l,ij} \in \mathbf{F}_l$ and $F_{u,ij} \in \mathbf{F}_u$. These boundaries of the intervals have to fulfill certain constraints so that the structure assembly conforms with the criteria discussed in Section 4.2. We assume here that the deformation space model(DSM) has overall N sections and M load paths (in the example shown in Figure 4.17 we have $N = 32$ and $M = 6$). The criteria for structural design stated in Section 4.2 can be now quantified as follows:

(1) **Maximum load on critical components:** In each load path, the force should not exceed the feasible load of the critical components (e.g. fuel tank and high-voltage battery). Additionally, the designed force level must be realistic regarding manufacturing restrictions. Suppose the structure in the j-th path is discretized in N sections, the upper boundaries of the force in the i-th section of the component

and the j-th load path is then constrained by:

$$F_{u,ij} \leq \min\left(F_{\mathrm{crt},j}, F_{\mathrm{max},ij}\right) \ . \tag{4.24}$$

Here, $F_{\mathrm{crt},j}$ is the maximum permissible load on the critical component in the j-th load path and $F_{\mathrm{max},ij}$ denotes the maximum manufacturable crushing strength in the corresponding component of i-th section and j-th load path.

(2) **Limited intrusion into critical components:** In order to avoid the intrusion into critical components, the velocity difference between vehicle and barrier should be reduced to zero when the free crash length(s) along the critical load path(s) is/are totally consumed. Lower boundaries of the force-deformation curves together with barrier deformation should guarantee the energy absorption defined in Equation (4.3). The energy absorbed by the vehicle structure assembly can be calculated from the force-deformation characteristic in each load path:

$$E^v = \sum_{j}^{M} E_j^v = \sum_{j}^{M} \sum_{i}^{N_j^*} F_{l,ij} d_{ij} \ . \tag{4.25}$$

In this discrete form, d_{ij} is the deformable length of section i in load path j and N_j^* denotes the index of the last deformed section in load path j, i.e. $N_j^* \leq N$.

The energy absorbed by the barrier can be estimated using Equation (4.22):

$$E^b = \sum_j^M E_j^b\Big(\max_i F_{l,ij}\Big) \ . \tag{4.26}$$

The total deformation energy, E_{total}, is predefined by the load case, i.e. it equals the initial kinetic energy. When the lower boundaries of the force levels in the structure can absorb this energy within the given available deformation, the critical components are not intruded. Hence, this constraint for the lower boundaries of the force-deformation curves can be written as:

$$\sum_j^M \left[\sum_i^{N_j^*} F_{l,ij} d_{ij} + E_j^b\Big(\max_i F_{ij}\Big) \right] \geq E_{total} \ . \tag{4.27}$$

(3) **The order of deformation:** In the crash, the order of deformation of some components in a load path should be controlled. This has two reasons:

- If the impact velocity is relatively low, it is not necessary to dissipate energy by collapsing all components. For example the crash box must deform before the rear longitudinal beam starts to deform plastically. This reduces repair cost.

- The critical components, though sometimes located in the middle of a load path, are the last to be deformed plastically.

On these accounts, the order of deformation criterion requires that:

$$\max F^{C_1}(u) \leq F^{C_2}(u = 0) \ . \tag{4.28}$$

In this configuration, component C_1 deforms before component C_2 starts to deform. $F^{C_1}(u)$ and $F^{C_2}(u)$ are the plastic force (F) - deformation (u) characteristics for each component. The criterion requires that the crushing force of component C_1 is always lower than the yielding force of the component C_2, $u = 0$ is the start of the plastic deformation.

This order of deformation constraint can be reformulated considering corridor boundaries as follows: If the upper boundary of the force-deformation curve of component C_1 stays below the starting level of the lower boundary of the force-deformation curve of component C_2. The component C_2 always deforms plastically after the component C_1. This can be written as:

$$F^{C_1}_{u,i^*j} \leq F^{C_2}_{l,i_0j} \ . \tag{4.29}$$

Here i^* is any section in component C_1 while i_0 is the first section in component C_2.

Up to here, the constraints on the force-deformation characteristics of each vehicle component are defined with the available information in the deformation space

model(DSM).

Identification of Solution Spaces under non-linear constraints with sequential linear programming

The shape of the Solution Space should be adapted to the properties of the design, i.e. components with strongly oscillating force-deformation characteristics receive relatively wide corridors, while components tending to have stable force-deformation characteristics can fulfill narrower corridors. This relationship is controlled by predefined weighting factors (ω_{ij}):

$$F_{u,ij} - F_{l,ij} - \omega_{ij}\Delta F = 0 \ . \tag{4.30}$$

In which ΔF is the scalar measurement of the overall size of the Solution Space. Larger ω ensures a wider corridor. The constraints defined in Equations (4.24), (4.29), and (4.30) form a set of linear constraints, while the constraint in Equation (4.27) is strongly non-linear with respect to the design parameters. This is explained in the following paragraph. Under the constraints stated above, the optimal Solution Space for the vehicle structure is identified through an optimization process. In general, it is advantageous to provide to the components wider intervals/corridors, namely larger ΔF, for the force-deformation characteristics design. A larger ΔF increases the design flexibility. As a consequence, the objective

of the optimization is to maximize the size of the Solution Space, which is here defined as a hypercube. There are different size measures possible, see (Fender 2014); here we propose to consider an objective, which maximizes the minimal edge length of the hypercube :

$$\max \left(\min_{i \in N_j^*, j \in M} \frac{F_{u,ij} - F_{l,ij}}{\omega_{ij}} \right) . \tag{4.31}$$

The optimization problem stated above is a non-linear problem for rear crash. The non-linearity of the constraint Equation (4.27) is the result of the following:

Left-hand-side (LHS): the actual deformation lengths of the components can be determined only when the corresponding barrier deformation is given. The barrier deformation depends on the force level of the components, which are the design parameters. Thus, the vehicle deformation in load path j (d_j) is dependent on the design variables.

Right-hand-side (RHS): the actual energy absorption in the barrier depends non-linearly on the strengths or force levels of the vehicle components. Since the gradient of this constraint is not available, the Karush-Kuhn-Tucker (KKT) condition cannot be evaluated for this constraint. Hence, a sequential linear programming solution is developed to linearize this constraint in each step.

The iterative solution process is: At step $k+1$, the energy constraint is determined by assuming the vehicle structure carrying over the force level obtained in

step k. The coefficients $d_{ij} \in \mathbf{d}$ on the LHS, which define the minimum required deformation of the vehicle structure, are determined successively based on the deformation condition specified in Equation (4.20): Assuming that the available deformation length (d_{crt}) in the critical load path is used up, its maximum force level during the deformation determines the corresponding barrier deformation (d^b_{crt}). The total deformation in the critical load path (D_{crt}) is thus fixed in step $k+1$, shown in Figure 4.18 (the upper plot is the critical load path). The required total deformations in other parallel load paths are thus fixed by D_{crt} due to Equation (4.20). The bottom plot in Figure 4.18 shows the process to distribute the required deformation between vehicle and barrier in a non-critical load path. Meanwhile the RHS is updated by interpolating the two successive states of design parameters to accelerate convergence. The coefficients at the LHS and the RHS of the constraint Equation (4.27) are constant in step $k+1$ and a standard linear optimization problem for this step is formulated:

$$\sum_{j}^{M} \sum_{i}^{{}^{k}N} {}^{k+1}F_{l,ij} d_{ij} \geq \underbrace{\alpha \left(E_{total} - \sum_{j}^{M} E^b_j (\max_i {}^{k}F_{l,ij}) \right) + {}^{k}\mathrm{RHS}(1-\alpha)}_{{}^{k+1}\mathrm{RHS},\ \text{Constant in step } k+1} \ . \tag{4.32}$$

As a consequence, in each step a linear optimization problem with the linearized constraint Equation (4.32) and other linear equality/inequality constraints is solved using the Interior Point Method (Nocedal & Wright 2006). Figure 4.19 shows

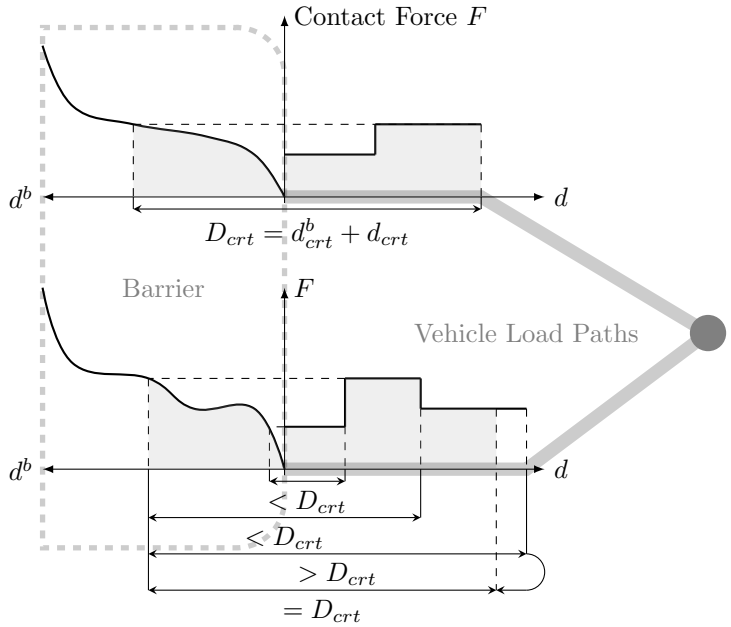
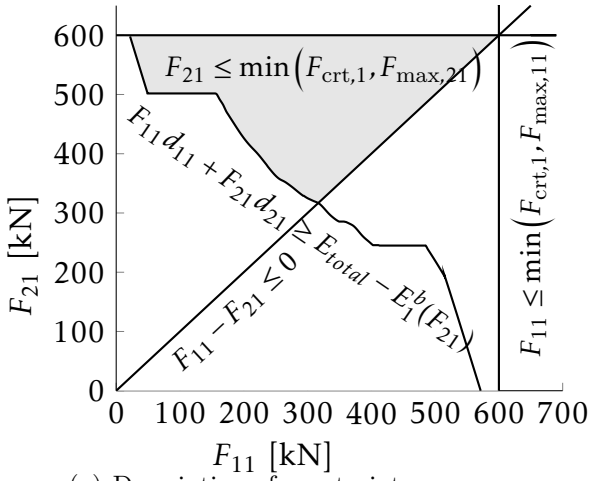
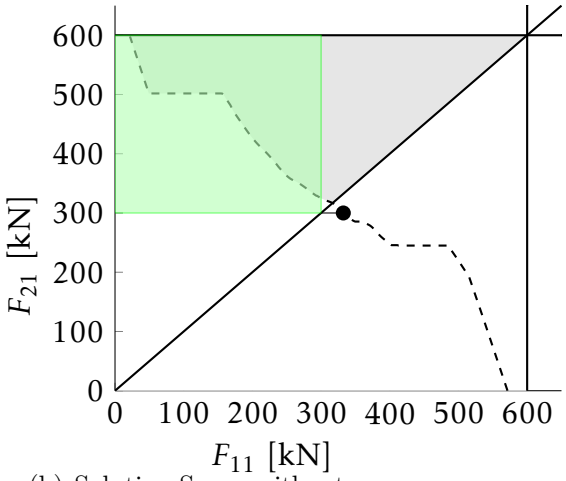


Figure 4.18: The deformations of vehicle and barrier in all load paths are increasing interactively until the total deformation is equal to that in the critical load path. The x-axis with d^b denotes the deformation length of barrier while the x-axis with d denotes that in vehicle load paths. Two load paths are shown in this figure and the upper one is the critical load path, of which deformation is limited to D_{crt}.

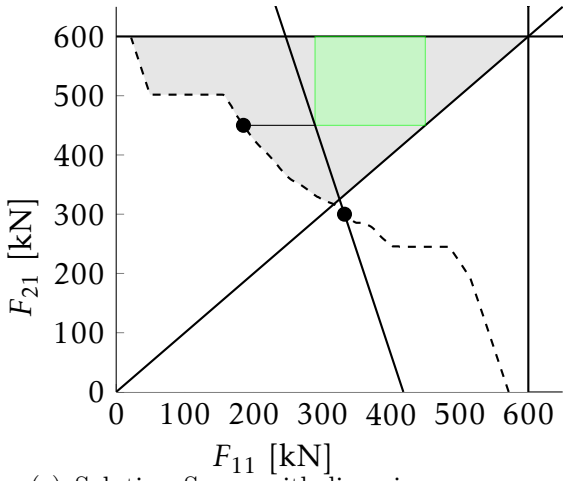
schematically the linearization steps of the energy constraint for a two-dimensional Solution Space. The only changing constraint due to linearization influences the size of the Solution Space, which is represented with the green square.



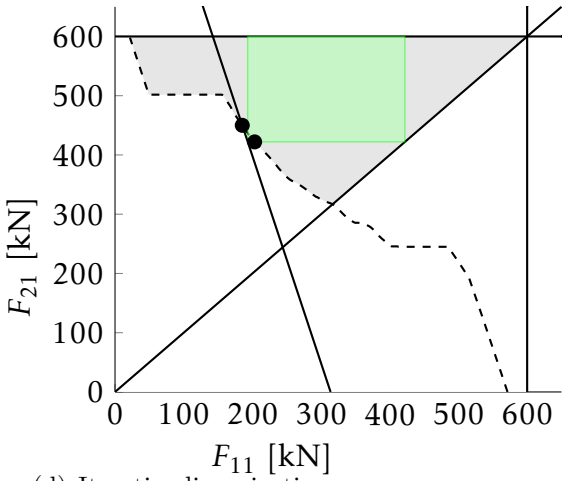
(a) Description of constraints.



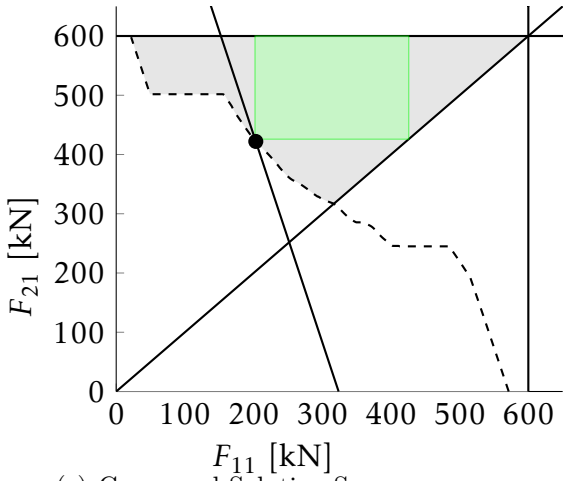
(b) Solution Space without energy constraint at step $k = 1$.



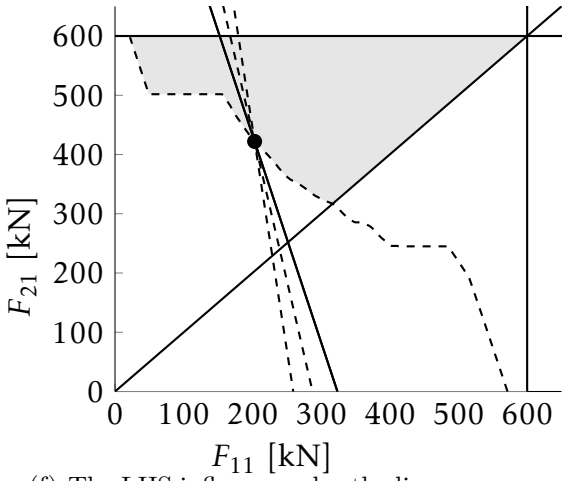
(c) Solution Space with linearization of RHS energy constraint.



(d) Iterative linearization process.



(e) Converged Solution Space.



(f) The LHS influences also the linearization.

Figure 4.19: Sequential linearization of the nonlinear constraint to identify the optimal Solution Space.

We obtain the largest Solution Space ${}^{k+1}\Delta F$ with the boundaries for the force level in each component section ${}^{k+1}\mathbf{F} = \left[{}^{k+1}\mathbf{F}_l, {}^{k+1}\mathbf{F}_u\right]$. The complete work-flow of the optimizer is described in Figure 4.20. In the end of the optimization process, the maximized Solution Space for the structural design is obtained. Each dimension of the Solution Space defines an interval for one piece of force-deformation curve in one component.

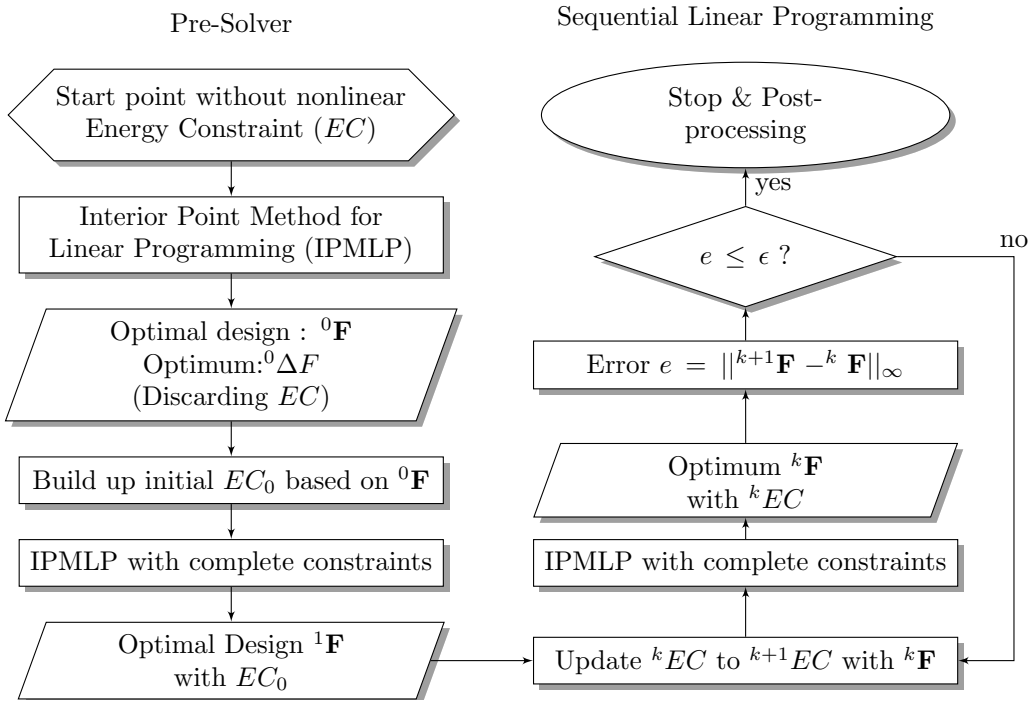


Figure 4.20: Iterative linearization solving process to identify the Solution Space.

4.4 Summary of the Proposed Method

With the method described above, an approach to set up the requirements concerning the force-deformation curves for components relevant for rear crashes is

established. The steps are the following:

(1) Extract the load path model out of the total vehicle model. The energy evaluation as well as load path visualization can be conducted when the FE model is already available. Otherwise, the load paths can be derived from the structural packages. The available deformation lengths, mass distribution and load path topology are determined in this step.

(2) Estimate the contact surface of each load path against barrier. The resistance force levels from the barrier impacting the vehicle structure are thereby determined. The simplified barrier model has to be calibrated only when a new barrier model is adopted.

(3) Pre-define the requirements on the vehicle structure. The critical components and their admissible loads, the order of the deformation between components and the maximum possible strengths of the components have to be provided as constraints for the structural design.

(4) Identify the Solution Space. The information collected in step (1)-(3) is used to formulate a sequential linear programming problem. The optimum is the largest Solution Space, which generates the feasible intervals for force-deformation curves of each component.

(5) Use the optimized Solution Space to realize a decoupled development of the

structural components assuring fulfillment of the overall crash objectives for rear impact.

"The ability to simplify means to eliminate the unnecessary so that the necessary may speak." [a]

Hans Hofmann

[a] quoted in Zen and the Art of Stand-up Comedy (1998) by Jay Sankey

5

Solution Spaces for Vehicle Architectures

5.1 Commonality of Vehicle Architecture

The Solution Space approach for single vehicle cases has been discussed in Section 4. There, the size of the feasible Solution Space was optimized to achieve the robustness and flexibility of the structural design. This has to be modified when a complete vehicle architecture is considered, i.e. several vehicles with comparable proportions and load path topologies, which share as many as possible components. The design criteria of the corresponding structural components become now coupled across the vehicle architecture. Figure 5.1 shows an example of a vehicle architecture. In general, there are two types of coupling relationships:

Direct Coupling: When several vehicles share certain components, the shared components may have different deformation behavior due to the blocking packages in their neighborhoods, i.e. the deformations may be blocked by some stiff devices, the buckling is thus triggered differently or the impact positions into the barrier are different. For these cases, only the components with similar deformation behavior, represented the resultant force-deformation curve, are considered as shared components with respect to crashworthiness functions. The constraints for these components are directly coupled in the vehicle architecture. In the example shown in Figure 5.1, this applies to the front rail (marked as red) shared by Veh. 1 and Veh. 5 such that the constraints of this component are directly coupled.

Indirect Coupling: The indirect coupling describes the situation that several vehicles do not directly share components. However, they have at least one vehicle, which has directly coupled components with these vehicles. In Figure 5.1 none of the components in Veh. 1 and Veh. 4 is directly coupled. Nonetheless, since vehicle 1 shares components with vehicle 5 as the vehicle 4 also does, the components in Veh. 1 and 4 are indirectly coupled.

The consequence of the direct coupling among components in vehicle architecture

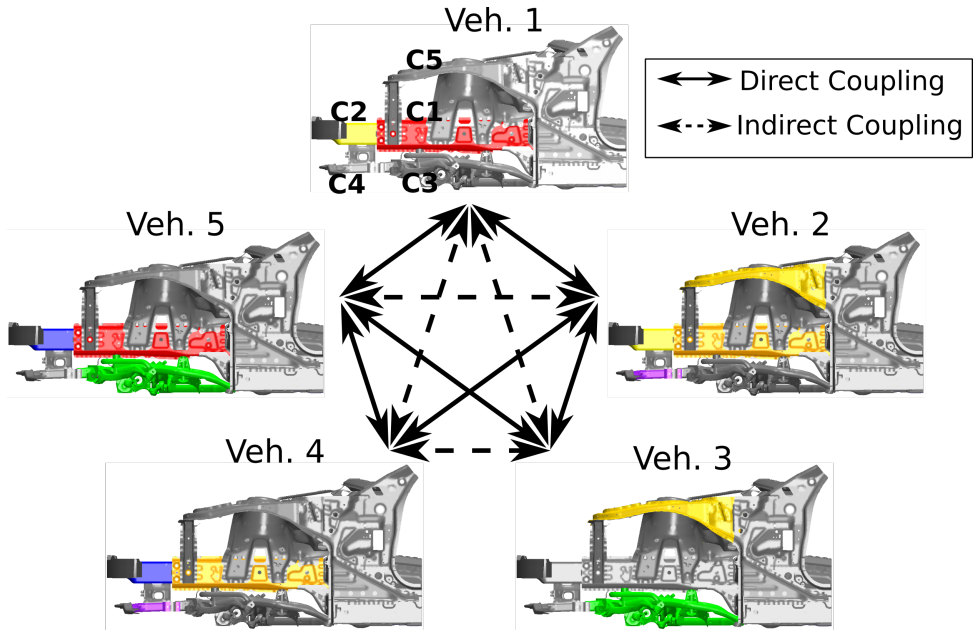


Figure 5.1: Example for a coupling situation of components within an architecture for five vehicles: identical colors signify common components (Song et al. 2015). Denotation of the components: $C1$–front rail; $C2$–crash box of the front rail; $C3$–front axle; $C4$–crash box of the front axle; $C5$–wheel housing frame.

is straight forward: the coupled components must fulfill all the constraints coming from each vehicle in which they are integrated. Here, we define the Commonality Description Matrix (CDM) to denote this relationship. If an architecture involves

P_e vehicles and each of these vehicles can be modularized into E types of components. The CDM, $\mathbf{\Gamma} \in \mathbb{N}^{P_e \times E}$, which describes the example stated in Figure 5.1 is:

$$\mathbf{\Gamma} = \begin{array}{c} \begin{matrix} \text{C1} & \text{C2} & \text{C3} & \text{C4} & \text{C5} \end{matrix} \\ \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 2 & 2 \\ 3 & 2 & 2 & 1 & 2 \\ 2 & 3 & 1 & 2 & 1 \\ 1 & 3 & 2 & 1 & 1 \end{pmatrix} \end{array} \begin{matrix} \text{Veh.1} \\ \text{Veh.2} \\ \text{Veh.3} \\ \text{Veh.4} \\ \text{Veh.5} \end{matrix} \tag{5.1}$$

Each column represents the commonality configuration of a component. The same integers indicate the shared components. E.g. for component C1 – the front rail, Veh. 1 and Veh. 5 sharing the red variant and represented as 1 in the first column; Veh. 2 and Veh. 4 sharing the yellow variant and represented as 2 in the first column; Veh. 3 has its own gray variant and represented as 3 in the column.

One drawback of this notation is that the CDM describing one commonality configuration is not unique. Again taking the C1 as example, the first column used to describe the commonality of the $C1$ among vehicles has the following equivalent

representations:

$$\begin{pmatrix}1\\2\\3\\2\\1\end{pmatrix} \Longleftrightarrow \begin{pmatrix}2\\1\\3\\1\\2\end{pmatrix} \Longleftrightarrow \begin{pmatrix}2\\3\\1\\3\\2\end{pmatrix} \tag{5.2}$$

All the combinations above describe the same configuration. For convenience and clarity, the canonical form is introduced to ensure uniqueness of the CDM, describing one specific commonality configuration of the vehicle architecture.

Definition 5.1: Canonical representation

A canonical representation of a set partition $\pi = \pi_1\pi_2\ldots\pi_n$ of $[n]$ is a word π such that $\pi_1 = 1$, and the first occurrence of the letter $i \geq 1$ precedes that of j if $i < j$ (Mansour 2012).

With this definition, only the first string in Equation (5.2) is used to describe the commonality configuration of $C1$ among all vehicles: the notion separates for example a set 12345 into partitions 15/24/3. The canonical form is thus 12321, having $\pi_1 = \pi_5 = 1$, as both 1 and 5 are in the first block. Likewise $\pi_2 = \pi_4 = 2$ and $\pi_3 = 3$.

5.2 Identification of Solution Spaces for Multi-vehicles

5.2.1 Handling of the Common Components

The common components represented by the same integer in the CDM should have identical deformation length and lumped mass in the deformation space model(DSM) explained in Section 4.3.4. The force-deformation curve of the component is discretized by sections in the deformation space. As a consequence, the common components, which share the same Solution Space, must have the same section discretization. In other words, the section division of common components should have identical lengths and quantity. If the common component has different section discretization in the vehicles into which it is integrated, as shown in Figure 5.2, additional artificial sections should be inserted into the deformation spaces to synchronize the section division of the common components. For instance, three deformation spaces with common components are synchronized in the following way:

(1) Independent building of the deformation spaces: as shown in Figure 5.3, on step I, the three deformation space models are built independently for each structure assembly. The building process is the same as for a single structure – sections are inserted where a lumped mass point is located or endings of components meet.

(2) Consecutive synchronization of sections for common components: New artifi-

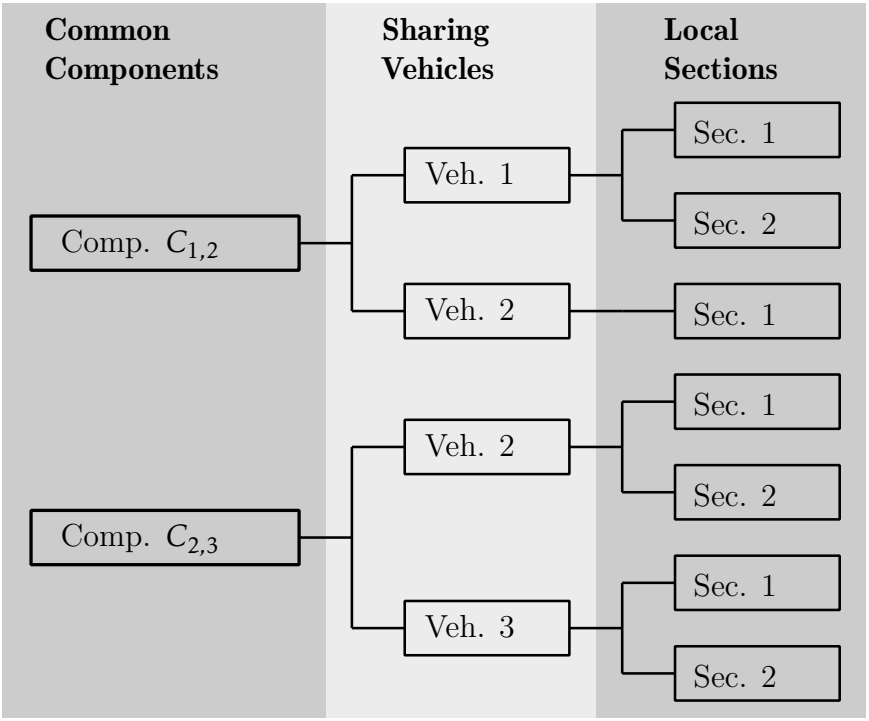


Figure 5.2: Examples for common components having individual section discretization in the vehicles in which they are integrated.

cial sections are inserted so that the common components have an identical section division, as shown in Figure 5.3 by step II and III. Since the sections are defined transversely through all parallel load paths, the parallel components are affected as well. This synchronization leads to a finer discretization of the deformation space. When the common components have comparable relative spatial positions among the structure assemblies, the section count for the common components converges. Section bounds are inserted eventually on the important positions (mass points and endings of components) within the spatial range of the common components.

(3) Section refinement for all common components: the last process is repeated for all common components.

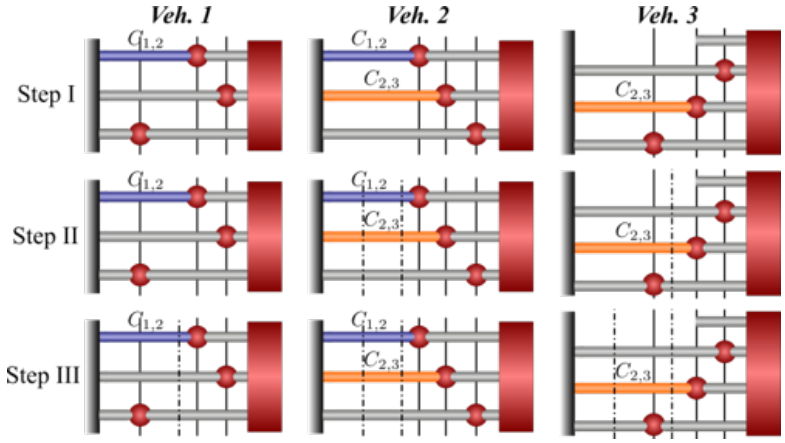


Figure 5.3: Insertion of artificial sections (dotted lines) consecutively to synchronize the section division of the common components in deformation space models.

5.2.2 COUPLING OF SYSTEM CONSTRAINTS

In Section 4.3.4, the sequential linear programming process is established to identify the Solution Space. Inequality constraints of Veh. i (i represents the numbering of the vehicle in the architecture) are denoted as $\mathbf{A}^i_{ineq}(\mathbf{F})$ while an equality constraint defined in Equation (4.30) is given as $\mathbf{A}^i_{eq}$. The optimization problem for vehicle i is now formulated as:

$$\begin{aligned} \max \ : \ & \mathbf{c}^T\mathbf{F} \ , \\ \text{s.t.} \ : \ & \mathbf{A}^i_{ineq}\mathbf{F} - \mathbf{b}^i_{ineq} \leq 0 \ , \\ & \mathbf{A}^i_{eq}\mathbf{F} - \mathbf{b}^i_{eq} = 0 \ . \end{aligned} \tag{5.3}$$

To couple the constraints for common components, which have a synchronized section discretization, the constraint matrices of each vehicle in the architecture are stacked diagonally. Additionally, the assembled matrix of equality constraints must be extended as follows:

$$
\begin{array}{l}
\\
\text{Comp. } C_{1,2} \\
\\
\text{Comp. } C_{2,3}
\end{array}
\left(\begin{array}{c|c|c}
\begin{bmatrix} & \mathbf{A}^1_{eq} & \end{bmatrix} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \begin{bmatrix} & \mathbf{A}^2_{\mathbf{eq}} & \end{bmatrix} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \begin{bmatrix} & \mathbf{A}^3_{\mathbf{eq}} & \end{bmatrix} \\
\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \end{bmatrix} & \begin{bmatrix} -1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & 0 \\ 0 & 0 & -1 & \cdots & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \\
\begin{bmatrix} 0 & \cdots & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} \cdots & 1 & 0 & 0 & 0 \\ \cdots & 0 & 1 & 0 & 0 \\ \cdots & 0 & 0 & 1 & 0 \\ \cdots & 0 & 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} \cdots & -1 & 0 & 0 & 0 \\ \cdots & 0 & -1 & 0 & 0 \\ \cdots & 0 & 0 & -1 & 0 \\ \cdots & 0 & 0 & 0 & -1 \end{bmatrix}
\end{array}\right)
\tag{5.4}
$$

The extended part of this matrix couples the common components, i.e. the upper and lower boundaries of the corridors for the sections of common components should be identical. The corridors of these common components are constrained by the equalities and inequalities from the total vehicle system in which these common components are integrated. As a result, the Solution Space of the vehicle architecture is identified with the same sequential linear programming procedure

as described in Section 4.3.4, but now under the extended set of constraints.

5.3 Evaluation of Commonality Configurations

In vehicle architecture design, the commonality configuration must be evaluated for appropriate design decisions. In this thesis, two norms are developed for this purpose: **Commonality Index** and **Size of Solution Space** respectively:

5.3.1 Commonality Index of a Given Vehicle Architecture Configuration

Definition 5.2: Commonality Index of Single Components

For one COMPONENT type, the difference between the total number of end products in the product family (denoted as P_e) and the number of the variants (V_e) of this component is defined as the *Commonality Index* of this component, denoted as:

$$CI_e = P_e - V_e \quad . \tag{5.5}$$

Based on this definition, the commonality index of a component reaches the minimum – zero, if the component is distinct in each end product (every vehicle in the architecture in this case). Conversely, a commonality index $(P_e - 1)$ means that this component is shared by all end products. The benefit of this definition over

those introduced in Chapter 2 is the consistency with the CDM representation, $\mathbf{\Gamma}$: the commonality index of component j can be calculated by the maximal value of the corresponding j-th column of the $\mathbf{\Gamma}$, i.e.

$$CI_{e,j} = P_e - \max_i \Gamma_{i,j} \ , \tag{5.6}$$

in which the i and j are the row and column indices of the CDM respectively. P_e is the number of the vehicles in the architecture, thus constant for a given architecture set-up. And in this chapter from now on, $i \in [1, P_e]$ always denotes the considered vehicle while j stands for the different types of components with maximum value E, as defined in Definition 5.3.

One end product contains normally several component types. Between two component types, the same commonality configuration though may not lead to equal benefits. To compare the commonality between component types, the weighting factor e_j is introduced. This weighting factor correlates to the relative importance of the commonality of the corresponding component. In practice, these factors are calibrated by the costs – the more cost the commonality saves by one component, the higher weighting factor is assigned to this component. With this factor, we can define the commonality index for the total architecture. In this thesis, the weighting factors are assumed to be one – the commonality for all the component types are equally important.

Definition 5.3: Commonality Index of Vehicle Architecture

For one ARCHITECTURE, the *Commonality Index* is the weighted sum of the Commonality Indices of all component types which are parts of this architecture; it is denoted as:

$$CI = \sum_{j=1}^{E} e_j CI_{e,j} \ , \tag{5.7}$$

in which, E is the total number of the component types in one end product of the architecture.

Based on this definition, the Commonality Index of a certain commonality configuration can be calculated when the CDM, $\mathbf{\Gamma}$, is given:

$$CI(\mathbf{\Gamma}) = E \cdot P_e - \sum_{j=1}^{E} e_j \max_i \Gamma_{i,j} \ . \tag{5.8}$$

5.3.2 Size of Solution Space of a Given Vehicle Architecture Configuration

The Solution Space can be identified using the approach presented in Chapter 4; the size of the Solution Space is subjected to the following logic:

Given the design parameters $\mathbf{F}_i$, which represent the crushing strengths of crash

components, and the simplified model M_i, the mechanical responses $\mathbf{y}_i$ of the vehicle i can thus be predicted:

$$\mathbf{y}_i = M_i(\mathbf{F}_i) \ . \tag{5.9}$$

The prescribed design criteria of the structure performance (energy absorption, acceleration, order of deformation, etc.) apply on the structural response and can be denoted as $\mathbf{y}_i^{crt}$. Accordingly, the boundaries of the design parameters are defined by taking the inverse of the design criteria on the total vehicle response. This process is comparable to the classical parameter identification problem. An exemplary problem is discussed by Kim et al. (Kim et al. 2001). The difference is that only the boundaries/constraints of the design parameters are identified:

$$\mathbf{F}_i^{bound} = M_i^{-1}(\mathbf{y}_i^{crt}) \ . \tag{5.10}$$

The feasible domain of the design parameters should thus fulfill the constraint:

$$\mathbf{F}_i^{fsb} =: \{\mathbf{F}_i \mid \mathbf{F}_i - \mathbf{F}_i^{bound} \leq 0\} \ . \tag{5.11}$$

The optimal Solution Space of the vehicle structure, which is the maximized hypercube in the feasible design domain, can be denoted as a sub-space:

$$\mathbb{D}_i \subseteq \mathbf{F}_i^{fsb} \quad . \tag{5.12}$$

If multiple vehicles in the architecture are considered, the feasible domain of the functional characteristics (e.g. crushing strength) of the common components is subjected to the constraints from all relevant vehicles; this implies, if the CDM, $\mathbf{\Gamma}$, is given, that the feasible domain of the design variables of the specific component type J can be written as:

$$\mathbf{F}^{fsb}(\mathbf{\Gamma}) = \Big(\bigcap_{\forall \Gamma_{i,J}=1} \mathbf{F_i}^{fsb} \Big) \cup \Big(\bigcap_{\forall \Gamma_{i,J}=2} \mathbf{F_i}^{fsb} \Big) \cup \cdots \cup \Big(\bigcap_{\forall \Gamma_{i,J}=\max_i \Gamma_{i,J}} \mathbf{F_i}^{fsb} \Big) \quad . \tag{5.13}$$

Here, the index i denotes the vehicles in the grouped architecture. Analog to Equation (5.12), the hypercube Solution Space of the entire vehicle architecture is $\mathbb{D}(\mathbf{\Gamma}) \subseteq \mathbf{F}^{fsb}$.

5.4 Conflicts of Commonality and Design Feasibility

5.4.1 Maximizing the Design Feasibility

In Chapter 4, the size of the Solution Space reflects the width of the corridors for the force-deformation design. In the practical applications, the goal is to design

the component so that the force-deformation curve is positioned within its corridor. Therefore, it is normally easier to achieve this goal with a wider corridor, i.e. the wider the corridor is, the more flexible the design is in the current development phase. This flexibility enables further adaptations in subsequent design stages.

For vehicle architecture design, if the goal is simply to maximize the size of the Solution Space and the design flexibility by configuring the CDM, $\mathbf{\Gamma}$, the commonality requirements must be discarded. Otherwise, they couple the constraints of the Solution Spaces from individual vehicles and confine the size of the overall Solution Space. The following academic example demonstrates the situation stated above: the Solution Spaces of Veh. 1, Veh. 2 and Veh. 3 are identified by maximizing the hypercubes under the given constraints, shown as grey boxes in Figure 5.4(b). When seeking for a common Solution Space which fulfills all the constraints of Veh. 1 and Veh. 2, the largest box 12 in the common feasible domain should be identified. In this process, the common feasible domain can be written as

$$\mathbf{F}^{fsb}(\mathbf{\Gamma}^{12/3}) = \left(\mathbf{F}_1^{fsb} \cap \mathbf{F}_2^{fsb}\right) \cup \mathbf{F}_3^{fsb} \quad , \tag{5.14}$$

as shown in Figure 5.4(c). The CDM here is:

$$\mathbf{\Gamma}^{12/3} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{pmatrix}. \tag{5.15}$$

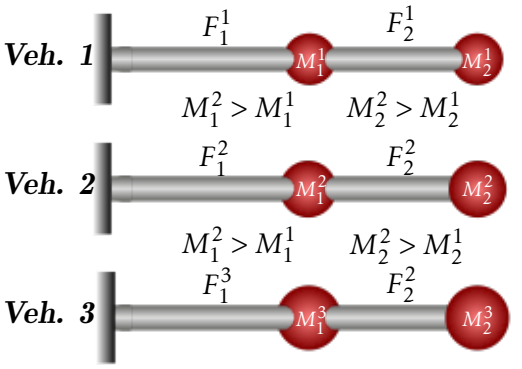
Analogously, Figure 5.4(d) shows the feasible design space of commonality configuration with CDM $\mathbf{\Gamma}^{1/23} = \left(\begin{smallmatrix} 1 & 1 \\ 2 & 2 \\ 2 & 2 \end{smallmatrix}\right)$. Obviously in this case, if all three vehicles are required to be common, the feasible domain is an empty set – there is no overlap between the feasible domains of all these three vehicles.

The common feasible domain cannot be larger than the original feasible domain without commonality. Therefore, the common Solution Space in the common feasible domain cannot be larger than any of the individual Solution Spaces, which means in this exemplary example: $\left\|\mathbb{D}(\mathbf{\Gamma}^{12/3})\right\| \leq \left\|\mathbb{D}(\mathbf{\Gamma}^{1/2/3})\right\|$, $\left\|\mathbb{D}(\mathbf{\Gamma}^{1/23})\right\| \leq \left\|\mathbb{D}(\mathbf{\Gamma}^{1/2/3})\right\|$, and certainly $\left\|\mathbb{D}(\mathbf{\Gamma}^{123})\right\| \leq \left\|\mathbb{D}(\mathbf{\Gamma}^{1/2/3})\right\|$.

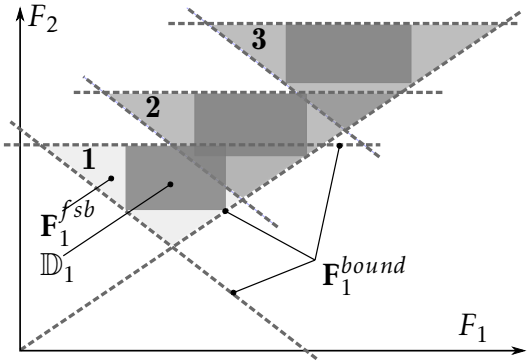
As a consequence, in order to ensure a large Solution Space and, hence, design feasibility, the commonality should be suppressed.

5.4.2 Maximizing the Commonality

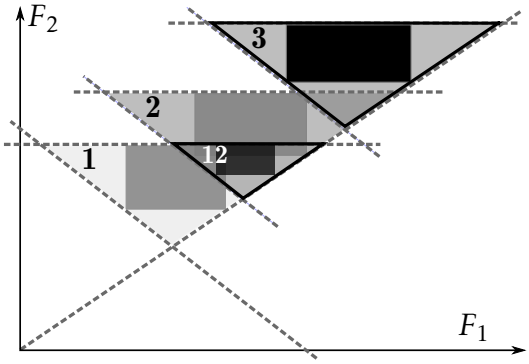
Vehicle architecture design aims to achieve a high commonality index. This leads to the coupling of the components. In the formulation of the constraints stated



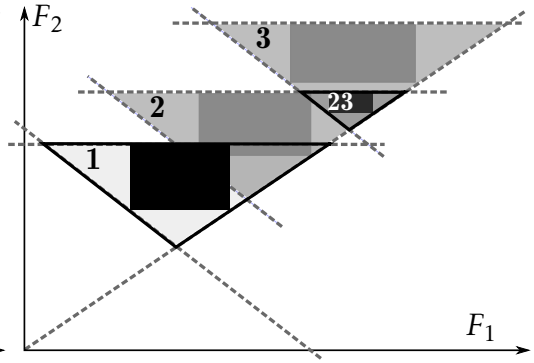
(a) Simplified Models.



(b) Solution Spaces for three independent vehicles.



(c) Solution Spaces for commonality configuration with CDM $\mathbf{\Gamma}^{12/3}$.



(d) Solution Spaces for commonality configuration with CDM $\mathbf{\Gamma}^{1/23}$.

Figure 5.4: The constraints introduced by commonality requirements restrict the common Solution Space of the architecture.

in Equation (5.4), depending on the number of synchronized sections, a certain number of equality constraints is inserted into the system.

Suppose a common component C_i with commonality index CI_e^i has s_i sections after the synchronization process described in Section 5.2.1; the number of equality

constraints inserted to enlarge the constraint matrix is calculated as

$$2 \cdot s_i \cdot \binom{CI_e^i + 1}{2} . \tag{5.16}$$

Here, (.) is the binomial coefficient.

Using E denoting the number of components in each vehicle, the total number of new equality constraints due to commonality can be estimated:

$$\#\text{ of additional constraints} = 2 \sum_{i=1}^{E} s_i \binom{CI_e^i + 1}{2} , \tag{5.17}$$

which indicates that increasing the component commonality index CI_e^i leads to more equality constraints in the system. These additional equality constraints couple the degrees of freedom of the optimization problem. As a consequence, the optimal Solution Space is shrinking; it can even turn out that the feasible set is empty.

To summarize the conflicts between the objectives mentioned above: the dilemma in the vehicle architecture design lies in the conflict between commonality and design flexibility – on one hand, if high commonality is prescribed, the design flexibility is decreased due to the narrowing Solution Space. Even worse, blindly pursuing a high commonality may lead to an empty feasible design space. On the other hand, the most flexible and robust design requires each vehicle having its

own individualized component configuration, i.e. no components are shared and coupled among the vehicles.

5.5 Multi-objective Optimization

In order to resolve this design dilemma, a multi-objective optimization must be formulated to balance the two objectives, i.e.

$$\mathbf{\Gamma}^{opt} = \arg\max_{\mathbf{\Gamma} \in \mathbb{Z}^{P_e \times E}} \Big(\big\| \mathbb{D}(\mathbf{\Gamma}) \big\|, CI(\mathbf{\Gamma}) \Big) \ . \tag{5.18}$$

The design parameter here is the Commonality Description Matrix (CDM), $\mathbf{\Gamma}$ – an all-ones $\mathbf{\Gamma}$ defines the highest commonality and worst Solutions Space size; while a $\mathbf{\Gamma}$, whose columns are filled with consecutive numbers, represents the none commonality design for the best design flexibility.

Here, a scenario analysis with an architecture of four vehicles is conducted to reveal the complexity and essence of the problem. For the rear longitudinal beam marked with colors, the commonality configuration is described in the Figure 5.5: from left to right, the Veh. 1 is equipped with a certain type of rear longitudinal beam. Considering Veh. 2, the component can be either common to that in the Veh. 1 or a different type (difference in material, geometry, etc.). Based on the decision made for Veh. 2, the beam in Veh. 3 could be common to Veh. 1, to Veh. 2 or different to all. The same logic applies to the Veh. 4.

In total, the architecture in this example can be set up with a single rear longitudinal beam for all, can have two different variants of beams with seven distinct allocations, can have three variants of beams in six configurations, and can get four completely different variants of beams.

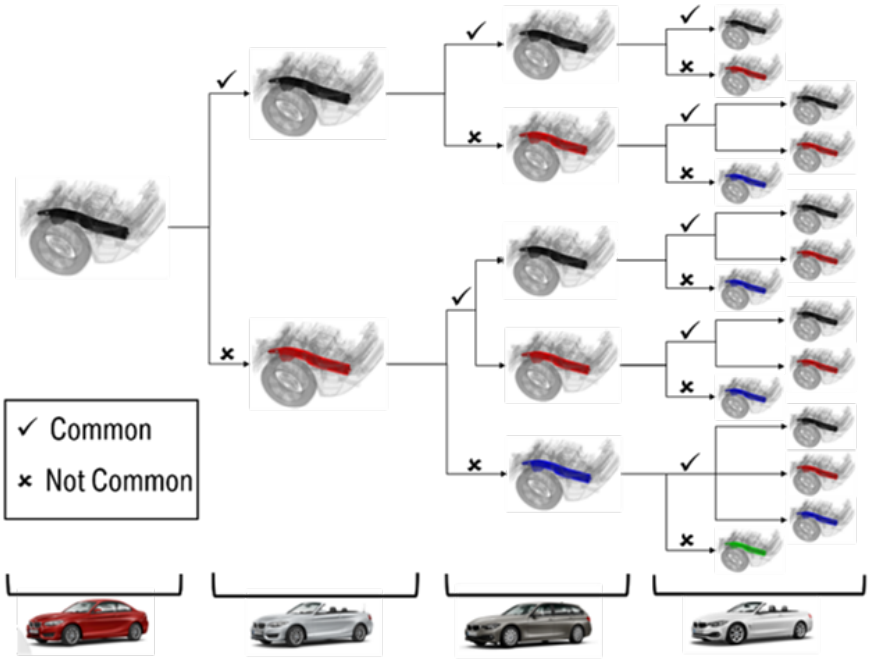


Figure 5.5: Scenario analysis of the possible commonality decisions.

Analogously, the commonality set-up of all other components, e.g. bumper, crashbox, can be enumerated in this process. The tree structure can be extended to describe all the scenarios in Figure 5.6.

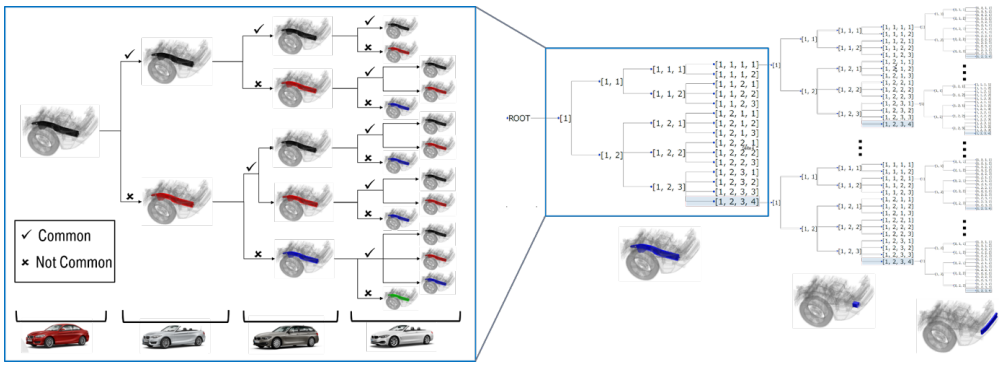


Figure 5.6: Scenario analysis of the commonality configurations of the architecture.

The allocation of the common components among vehicles in the architecture is mathematically categorized into a Set-Partition problem:

Definition 5.4: Partition of a set (Mansour 2012)

A *set partition* of a set S with n elements is a collection B_1, B_2, ..., B_k of non-empty disjoint subsets of S such that $\bigcup_{k}^{i=1} B_i = S$. The elements of a set partition are called blocks, and the size of a block B is given by $|B|$. The set of all set partitions of S is denoted by $\mathcal{P}(S)$.

5.5.1 Representation of Set Partitions

The partition of the component sets can be intuitively presented by different means (Mansour 2012).

Tree representation shown in Figure 5.5 lists all the partitions by following the branches from root to the leaf nodes. The cardinality of the leaf nodes

indicates the number of different commonality configurations. If the order of the vehicles is reorganized, the branches must be updated to maintain the partition information.

String representation, which uses the integers in canonical form adopted for CDM, generates all the possible partitions in lexical order. For instance, the partition of one component among four vehicles can be written as: 1111, 1112, 1121, 1122, 1123, 1211, 1212, 1213, 1221, 1222, 1223, 1231, 1232, 1233, and 1234.

Diagram representation introduces the Hasse Diagram, shown in Figure 5.7, to emphasize the generating relations between the partitions, i.e. the refinement and the coarsening of a given partition. These two concepts will be further discussed later in Section 5.5.5. Additionally, within each level/generation of Hasse Diagram, the partitions have the identical Commonality Index.

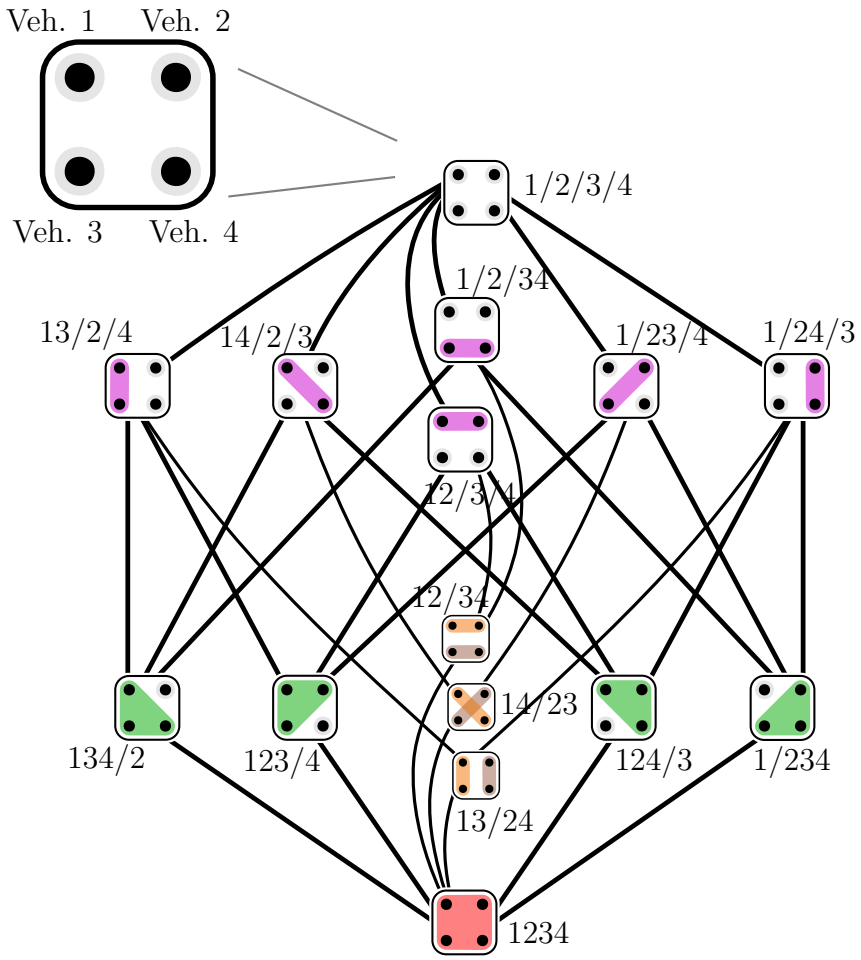


Figure 5.7: In graph theory, the Hasse Diagram is used to represent a finite partially ordered set. (Figure is modified based on that from wikipedia[i].)

For further development, a comparison between these three illustrations of the set partition problem is listed in Table 5.1.

[i] https://commons.wikimedia.org/wiki/File:Set_partitions_4;_Hasse;_circles.svg

Table 5.1: Comparison between the different descriptions of the set partitions

	Benefits	Drawbacks
Tree Repr.	++ Easy generating + Intuitive dependencies among partitions	-- Generating order is vehicle sorting dependent
String Repr.	+++ Easiest generating by "flipping" + Direct indication of the cardinality	-- Predefined generating order - No relationship among partitions
Hasse Diag.	+++ Clear relationship among partitions +++ Direct connection between level and Commonality Index	- Generation complexity

Based on the comparison, the Hasse Diagram shows intuitively the refinement of the partitions with monotonically increasing Commonality Index. This feature later facilitates the commonality optimization. Thus, the Hasse Diagram is adopted for this thesis.

5.5.2 Complexity of the Problem

With the representation and convention defined, the next step is to analyze the complexity of the problem. A decision problem belongs to as least one of the following classes in the computational complexity theory: polynomial time decidable (P); non-deterministic polynomial time decidable (NP); hardest of all problems in NP (NP-hard); in NP and being NP-hard (NP-Complete) (Dean 2016).

Theoretically, to identify the optimal architecture design, all the possible set partitions must be evaluated. The complexity of this exhaustive search is dependent on the number of vehicles P_e and number of component types E. The cardinality of the combinatorial set partition of a component type among all vehicles is described by the Bell Number:

Definition 5.5: Bell Number (Bell 1934, N.N. 2018a)

The number of ways a set of n elements can be partitioned into non-empty subsets is called a *Bell Number* and is denoted as $B(n)$, which is calculated as:

$$B(n) = \sum_{k=1}^{n} \text{Stir}(n,k) \quad . \tag{5.19}$$

The values $\text{Stir}(n,k)$ are called *Stirling Numbers of the Second Kind*, which denote the number of set partitions of $[n]$ into k blocks and satisfy the recurrence

$$\text{Stir}(n+1,k) = k\text{Stir}(n,k) + \text{Stir}(n,k-1) \quad . \tag{5.20}$$

with $\text{Stir}(1,1) = 1$, $\text{Stir}(n,0) = 0$ for $n \geq 1$, and $\text{Stir}(n,k) = 0$ for $n < k$ (Mansour 2012). This number can be calculated by the explicit formula

$$\text{Stir}(n,k) = \frac{1}{k!} \sum_{i=0}^{k} (-1)^i \binom{k}{i} (k-i)^n \quad , \tag{5.21}$$

in which $\binom{k}{i}$ is a binomial coefficient. Commonality scenarios of one component type can be counted by this number. The growth rate of the Bell Number is proved to fulfill the inequality $2^n \leq B_n \leq n! \leq 2^{n \log n}$ with limited n (Berman & Idziak 2005). The complexity is further established by Berend and Tassa (Berend & Tassa 2010) to be $\mathcal{O}((\frac{cn}{\ln(n+1)^n})^n)$ (c is a constant) which means that the Bell Number grows exponentially. This is also shown by the generating functions of the Bell Number – one of them is *Dobiński's Formula*(Dobiński 1877):

$$B(n) = \frac{1}{\mathrm{e}} \sum_{i=0}^{\infty} \frac{i^n}{i!} \ , \tag{5.22}$$

in which i are non-negative integers and e is the Euler's number. By applying this formula, the total number of possible configurations of the CDM can be calculated: for each column which represents the partition of one component type among all vehicles, the number of the combinations is exactly counted by the Bell Number, while the total number of combinations of different component types is counted by the product of the column cardinality:

$$|\,\mathbf{\Gamma}\,| = \underbrace{B(P_e) \cdot B(P_e) \ldots B(P_e)}_{\text{In total, E times Bell Numbers}} = \mathrm{e}^{-E} \Big(\sum_{i=0}^{\infty} \frac{i^{P_e}}{i!} \Big)^E \ . \tag{5.23}$$

Theoretically, to identify the optimal commonality configuration, all the configurations stated above should be evaluated by the optimal sizes of corresponding

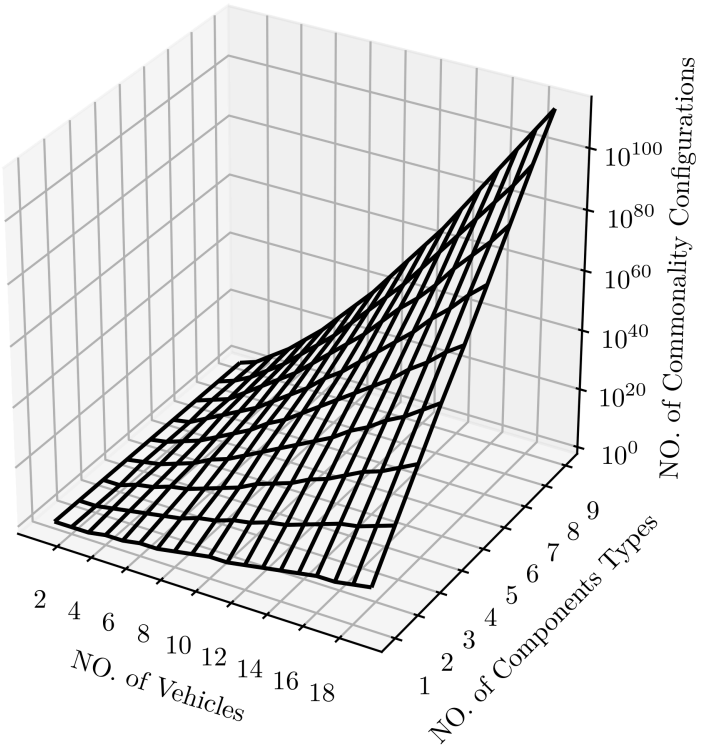


Figure 5.8: The increasing number of configurations with respect to number of vehicles and component types.

Solution Spaces. The complexity of this evaluation with respect to the number of vehicles and the number of component types is shown in Figure 5.8. In the work of Kovalyov et al. (Kovalyov & Pesch 2010), a generic approach to prove NP-hardness of partition type problems has been proposed. The architecture with one single component type for all vehicles is yet a partition problem, therefore NP-hard. The architecture with multiple component types is reducible. Consequently,

the commonality optimization problem is NP-hard.

In order to resolve the complexity, depending on the application use cases, three strategies are proposed to help making the decision to compromise the commonality and design flexibility in architecture design.

5.5.3 Pareto-Frontier

The trade-off between the design flexibility and commonality is evaluated by the size of Solution Space and the Commonality Index. To quantify the conflict between these two goals, the Pareto front should be identified between the size of Solution Space and the Commonality Index. Further, the optimal architecture design can be determined on the Pareto front according to the decision preferences. However, identification of the best Solution Space size under each given Commonality Index deterministically requires the exhaustive evaluation of all the configurations, as shown in Figure 5.9. A stochastic approach can be developed to avoid this expensive process. The development of this approach is not within the scope of this thesis and is left for future research projects. The intension here is to deterministically eliminate certain configurations a priori before they are evaluated by the Solution Space solver.

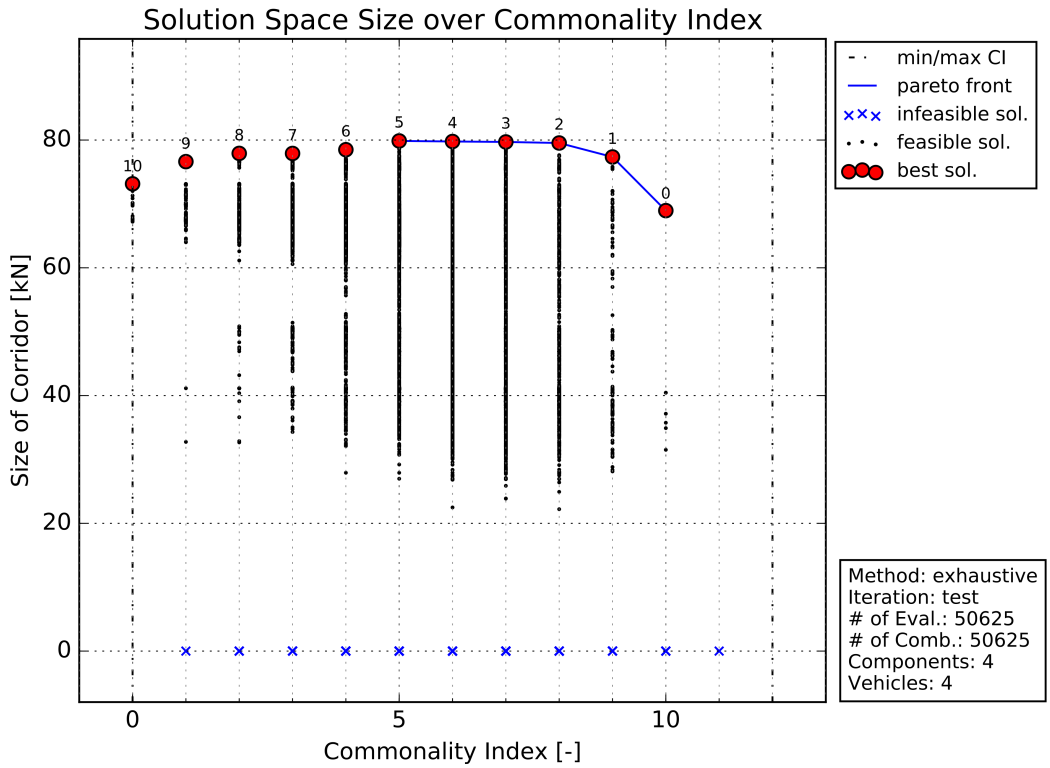


Figure 5.9: An exemplary Pareto front illustrating the trade-off between Solution Space Size (indicates design feasibility) and Commonality Index (assessment of architecture commonality)

.

5.5.4 Prescribed Commonality Configuration

In industrial applications, the commonality configurations are normally not completely free. The premises on the commonality configuration can be categorized as follows:

Prescribed Component: for example: "The rear axle must be identical for all vehicles", "The crash boxes must be identical among vehicle A, B, and C; while the crash boxes between vehicle D and F are identical but different from

those of vehicle A, B, C" etc. These are all statements to prescribe a specific commonality configuration of specific components. If the commonality configurations of one or several components are prescribed, the influence on the total number of combinations is directly calculated by Equation (5.23). The number of remaining combinations is given by:

$$\mathrm{e}^{-E+\delta}\Big(\sum_{i=0}^{\infty}\frac{i^{P_e}}{i!}\Big)^{E-\delta} \quad , \tag{5.24}$$

δ denotes the number of prescribed components. Note that the reduction is exponential.

Predefined Variants of a Component: for example: "The rear rail must have no more than two variants". This is a requirement to predefine the number of variants of specified components. In case the number of variants of one component is constrained between $\hat{\delta}$ and $\check{\delta}$ ($\hat{\delta} \leq \check{\delta}$), the rest of the total number of combinations can be calculated:

$$\mathrm{e}^{-E+1}\Big(\sum_{i=0}^{\infty}\frac{i^{P_e}}{i!}\Big)^{E-1} \cdot \sum_{j=\hat{\delta}}^{\check{\delta}} \mathrm{Stir}(P_e, j) \quad . \tag{5.25}$$

The reduction rate is dependent on the predefined $\hat{\delta}$ and $\check{\delta}$.

Connected Components: for example: "The rear rail and the crash box must have the same commonality configuration". This statement requires several

components to share the same commonality configuration. If δ components are set to have an identical commonality configuration, the remaining combinations can be counted with:

$$e^{-E+\delta+1}\Big(\sum_{i=0}^{\infty}\frac{i^{P_e}}{i!}\Big)^{E-\delta-1} \quad ,\delta \geq 2 \quad . \tag{5.26}$$

If these connected components are restricted by "predefined variants of component" as stated above, the Formula (5.26) can be combined with the Formula (5.25). This means that the reduction rates are also superposed.

Connected Vehicles: for example: "The rear axle in vehicle A and vehicle B must be the same". This exemplary statement requires that several vehicles are grouped together in the set partitioning process, which consequently reduces the Bell Numbers. If ρ vehicles are connected by one component, the number of the remaining combinations is calculated as follows:

$$e^{-E}\Big(\sum_{i=0}^{\infty}\frac{i^{P_e}}{i!}\Big)^{E-1}\cdot\sum_{i=0}^{\infty}\frac{i^{P_e-\rho+1}}{i!} \quad ,\rho \geq 2 \quad . \tag{5.27}$$

All the constraints stated above help to reduce the complexity of the problem – the stricter the constraints are, the more combinations are eliminated. Theoretically, the complexity of the problem after reduction lies between $\mathcal{O}(1)$ and $\mathcal{O}((\frac{cn}{\ln(n+1)^n})^{n\cdot E})$. Therefore, the solvability of the problem is strongly dependent

on the premises of the real world applications.

5.5.5 Prescribed Design Feasibility and Heuristic Elimination

Last, and most importantly, the size of the Solution Space sets up a straightforward approach to evaluate a given commonality configuration. If a critical size of Solution Space, $\left\|\mathbb{D}\right\|_{\text{crt}}$, is prescribed, any given commonality configuration can be verified in polynomial time and thus is a non-deterministic polynomial-time (NP) problem. Together with the NP-Hardness proof from Kovalyov (Kovalyov & Pesch 2010), the problem is now transformed into an NP-Complete problem; the solution strategy can be chosen from **exact algorithms**, **approximation algorithms**, **randomization**, **parallelism** or **heuristic approaches** (Reus 2016).

Except for the randomization, the other approaches stated above result in a deterministic solution, which are more suitable for a discrete commonality optimization problem. The non-deterministic randomized methods may converge to different commonality configurations in case of repeated analyses. Therefore, a deterministic approach is proposed in this thesis.

Elimination Approach based on a Survival Criterion

The $\left\|\mathbb{D}\right\|_{\text{crt}}$ sets up a "survival criterion" for the commonality configurations with respect to the size of generated Solution Space – The specific commonality configuration is eliminated when the corresponding Solution Space is smaller than the

critical size.

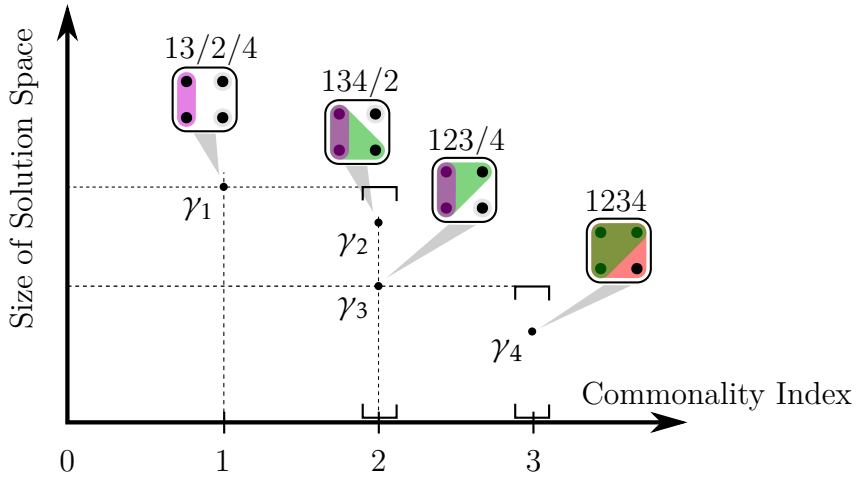


Figure 5.10: The relationship between size of Solution Space and Coarsening Partitions.

For linearly constrained Solution Spaces, increasing commonality of the architecture introduces more constraints, thus downsizes the Solution Spaces. This means, if the commonality index increases, the size of Solution Space cannot become larger. This logic can be utilized to speed up the elimination and is clarified by the following example: specific combinations (γ_1 to γ_4 shown in Figure 5.10) of one component grouped among four vehicles are selected out of the Hasse Diagram in Figure 5.7. By the configuration γ_1, Vehicle 1 and 3 share the same component. When Vehicle 4 is set up to share the same component with Vehicle 1 and 3, indicated by configuration γ_2, or Vehicle 3 shares the same component with Vehicle 1 and 2, shown by γ_3, the size of Solution Spaces of these two configurations γ_2 and γ_3 cannot be larger than that of γ_1; the reason has been thoroughly discussed

in Section 5.4.1. Therefore, the size of Solution Space of γ_2 and γ_3 can only be within the interval shown in Figure 5.10 at Commonality Index = 2. Further, the configuration γ_4 has all vehicles sharing the same component, the size of Solution Space is thus not larger than that of γ_3, which has at least the Vehicle 4 yet not coupled. As a consequence, the interval at Commonality Index = 3 continuously decreases. The deterministic decrement of the Solution Space size when the configurations satisfy these relationships can help to eliminate multiple configurations by only one evaluation; i.e. if $\left\|\mathbb{D}(\gamma_1)\right\| < \left\|\mathbb{D}\right\|_{\text{crt}}$, the configuration γ_1 is eliminated, all the other configurations which contain this "bad gene" (marked in purple) can also be eliminated without further Solution Space evaluations. In this case, the γ_2, γ_3, and γ_4 are to be eliminated consequently. The elimination is comparable to the "Domino Effect".

The relationships between the commonality configurations used by the Domino-like elimination are exactly illustrated in the Hasse Diagram, shown in Figure 5.11: starting from the combination 13/2/4 marked in red, three configurations are generated along the red dash lines. This operation is defined as **Stepwise Coarsening**: each subset of the "parent" configuration (denoted as $targetPartition$ in Algorithm 1, marked in red in Figure 5.11) is enlarged by one single element, and the union set of all the generated "offspring" configurations gives the resultant partitions of the stepwise coarsening operation.

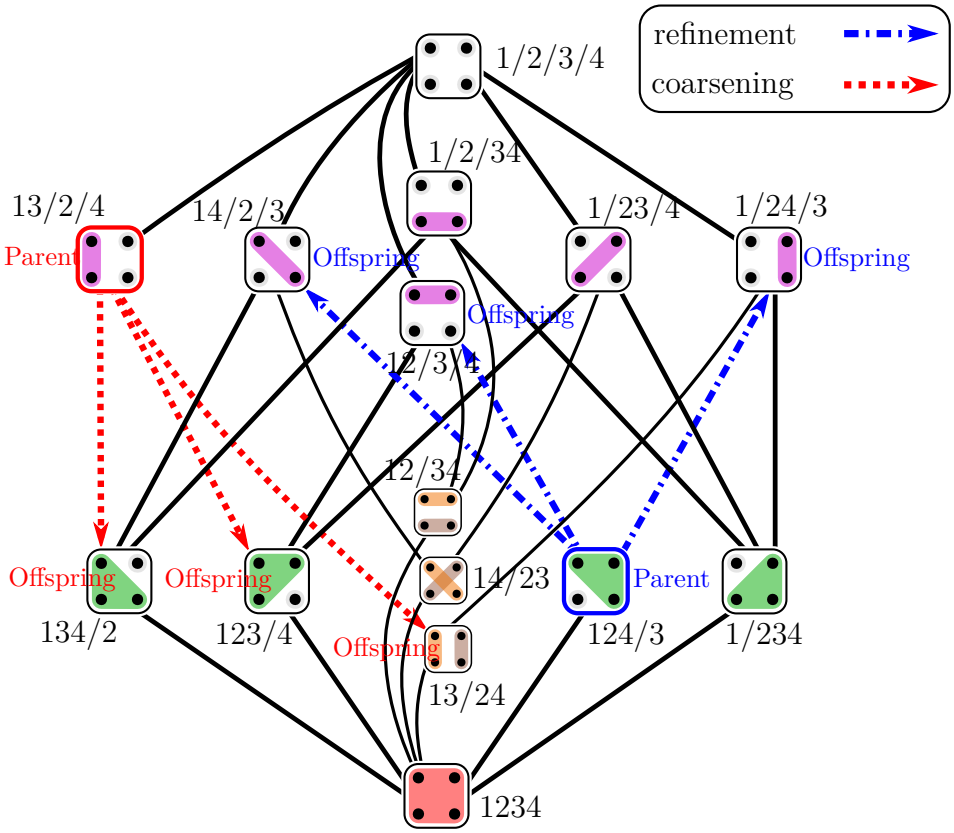


Figure 5.11: The exemplary stepwise coarsening and refinement operator of the set partition shown in the Hasse Diagram. (Figure is modified based on that from wikipedia[ii].)

The Stepwise Coarsening Operator is described in Algorithm 1. The generated configurations are the immediate offsprings of the target partition configuration.

[ii] https://commons.wikimedia.org/wiki/File:Set_partitions_4;_Hasse;_circles.svg

Algorithm 1 Stepwise coarsening operation of a given set partition

Input: *targetPartition* – one specific partition ▷ e.g. 12/3/4
Output: *resltPartitions* – partitions generated ▷ e.g. 123/4, 124/3, 12/34
procedure StepwiseCoarsening(*targetPartition*)
    initialization:
    convert target partition to canonical representation ▷ 12/3/4 →1123
    $partNum$: the number of parts in target partition ▷ partNum = 3
    $eleNum$: the number of elements in the partition ▷ eleNum = 4
    end initialization
    for $partID \leftarrow 1, partNum$ **do**
        for $i \leftarrow partID + 1, partNum + 1$ **do**
            initialize: *coarsenPartition*
            for $j \leftarrow 1, eleNum$ **do**
                if $targetPartition(j) = i$ **then**
                $coarsenPartition(j) \leftarrow partID$
                else if $targetPartition(j) < i$ **then**
                $coarsenPartition(j) \leftarrow targetPartition(j)$
                else
                $coarsenPartition(j) \leftarrow targetPartition(j) - 1$
                end if
            end for
            convert from canonical to standard form ▷ e.g. 1112 → 123/4
            append *coarsenPartition* in *resltPartitions*
        end for
    end for
    return *resltPartitions*
end procedure

Reversely, the **Stepwise Refinement Operator** has been defined with the help of the Hasse Diagram as well: Each subset (if it contains more than one element) in the "offspring" configuration (denoted as *targetPartition* in Algorithm 2 and 3, marked in blue in Figure 5.11) is reduced by one element in a traversal manner. All the generated "parent" configurations are the results of the stepwise refinement operations. This process identifies all configurations, which are the immediate

parents and can be coarsened to generate this "offspring" configuration. An exemplary operation is shown in Figure 5.11. The algorithm of stepwise refinement operation is described in Algorithm 2 and Algorithm 3.

Algorithm 2 Stepwise refinement operation of a given set partition (part 1)

Input: $targetPartition$ – one specific partition; $recursiveFlag$ ▷ e.g. 123/4
Output: $resltPartitions$ – partitions generated ▷ e.g. 12/3/4, 13/2/4, 1/23/4
procedure StepwiseRefinement($targetPartition$, $recursiveFlag$)
    initialization:
    convert target partition to canonical representation ▷ 12/3/4 →1123
    $partNum$: the number of parts in target partition ▷ partNum = 2
    $eleNum$: the number of elements in the partition ▷ eleNum = 4
    end initialization
    if $partNum = 1$ **then** ▷ initialization for recursion.
        for $i \leftarrow targetPartition(recursiveFlag - 1) - 1, recursiveFlag$ **do**
            $refinePartition(i) = i + 1$
            if $i > 1$ **then**
                break
            end if
            if $recursiveFlag \neq eleNum$ **then**
                return StepwiseRefinement($targetPartition$, $recursiveFlag + 1$) ▷ recursive function call
            else if $recursiveFlag = eleNum$ **then**
                if number of parts of $refinePartition = partNum + 1$ **then**
                    append $refinePartition$ in $resltPartitions$
                end if
            end if
        end for

Algorithm 3 Stepwise refinement operation of a given set partition (part 2)

```
23:     else if partNum ≠ 1 then                          ▷ for rest of the recursion.
24:         for partID ← 0, partNum do
25:             currentPart ← partID^th part of targetPartition
26:             tempPartitions = StepwiseRefinement(currentPart,
    recursiveFlag = 1)                                    ▷ recursive function call
27:             for partition in tempPartitions do
28:                 set count ← 0
29:                 for j ← 1, eleNum do
30:                     if targetPartition(j) = partID + 1 then
31:                         refinePartition(j) = partID + partition(count)
32:                         count = count + 1
33:                     else if targetPartition(j) < partID + 1 then
34:                         refinePartition(j) = targetPartition(j)
35:                     else
36:                         refinePartition(j) = targetPartition(j) + 1
37:                     end if
38:                 end for
39:                 append refinePartition in resltPartitions
40:             end for
41:         end for
42:     end if
43:     return resltPartitions
44: end procedure
```

With these two defined operators, an efficient elimination process can be performed along the Hasse Diagram.

Elimination Algorithms based on Hasse Diagram

The elimination process is illustrated intuitively by creating the Hasse Diagram generation-wise. In Figure 5.12, the process is decomposed into four steps:

Step I: Generating All configurations (No. 1 to 6) with $CI = 1$ are generated

by stepwise coarsening from configuration 0. These generated commonality configurations are evaluated by the Solution Space solver. In this case, the configurations 1, 4, and 5 are found out not fulfilling the critical Solution Space size and thus eliminated. The complexity of this process is limited to $\mathcal{O}(n^2)$, since even all the configurations in this generation must be exhaustively evaluated, the cardinality of the first generation is $\binom{P_e}{2} = P_e(P_e-1)/2$ – quadratic to the number of vehicles.

Step II: Selective Coarsening Only the configurations which survive the elimination with respect to Solution Space size in Step I are allowed to generate offsprings by stepwise coarsening. In this case, configuration 8 is not generated.

Step III: Backwards Refinement Before evaluating the newly generated configurations (No. 7, 9, 10, 11, 12, and 13), the stepwise refinements are conducted backwards. The relationships between the eliminated configurations in the last generation and the new generation are established.

Step IV: Predictive Elimination The configurations 7, 9, 10, 11, 12, and 13 are predictively eliminated without the necessity of Solution Space evaluation, since they are derived from the previously eliminated configurations (No. 1, 4, and 5). This predictive elimination logic is shown by the blue dash lines in the fourth sub-figure in Figure 5.12. In this case, all the newly

generated generations are eliminated before the Solution Space evaluation. It is not necessary to generate and evaluate further configurations. The best configurations exist among No. 2, 3, and 6, depending on their Solution Space size.

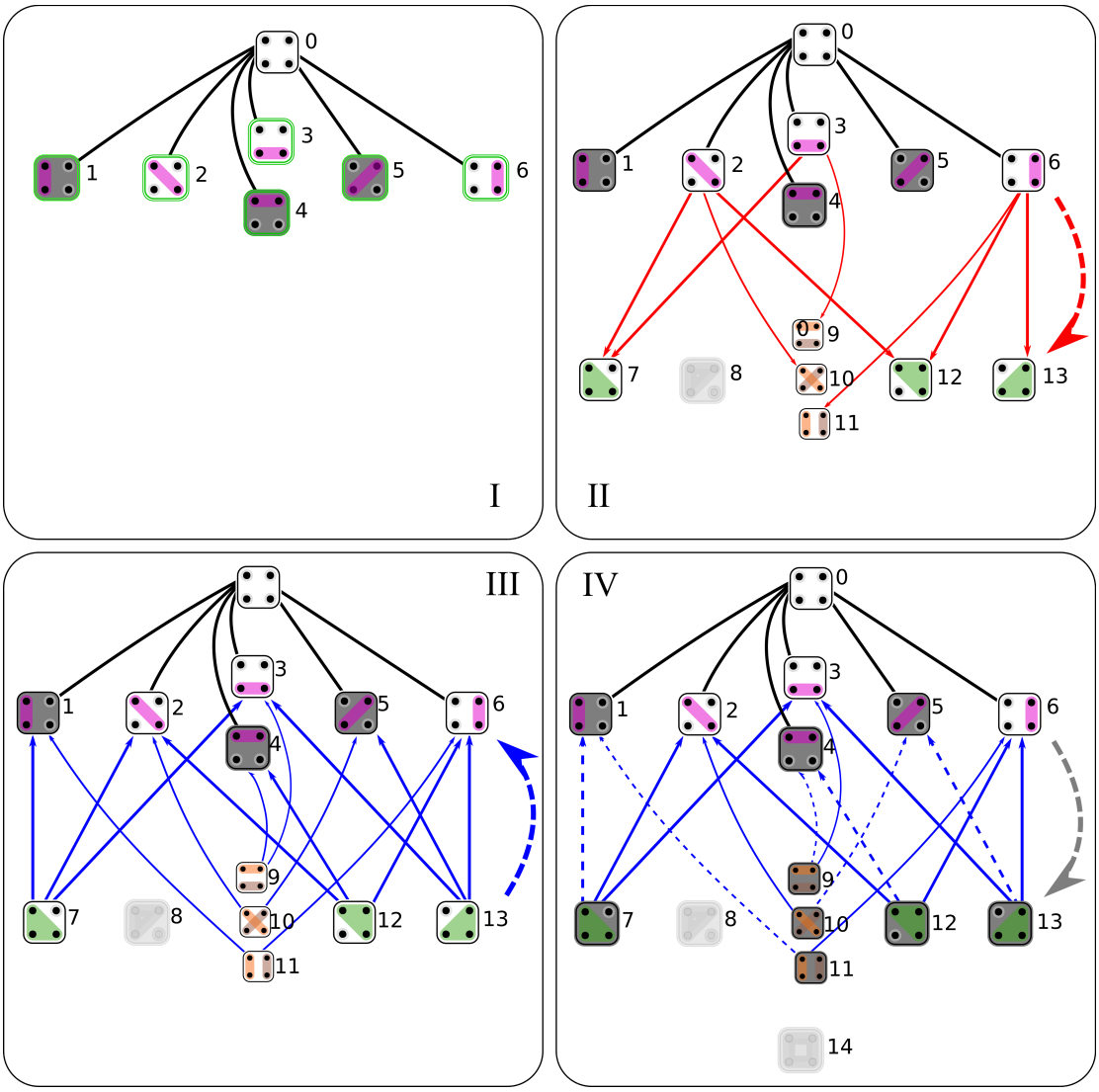


Figure 5.12: Eliminate the configurations which do not satisfy the Solution Space size criterion in four steps. (Figure is modified based on that from wikipedia[iii].)

For the example stated above, instead of performing the exhaustive search (in total 15 Solution Space evaluations), only 7 combinations in Step I are evaluated. The entire predictive elimination algorithm based on the generations in the Hasse Diagram is shown in Figure 5.13. Here, the process marked with blue has already been established in Figure 4.20.

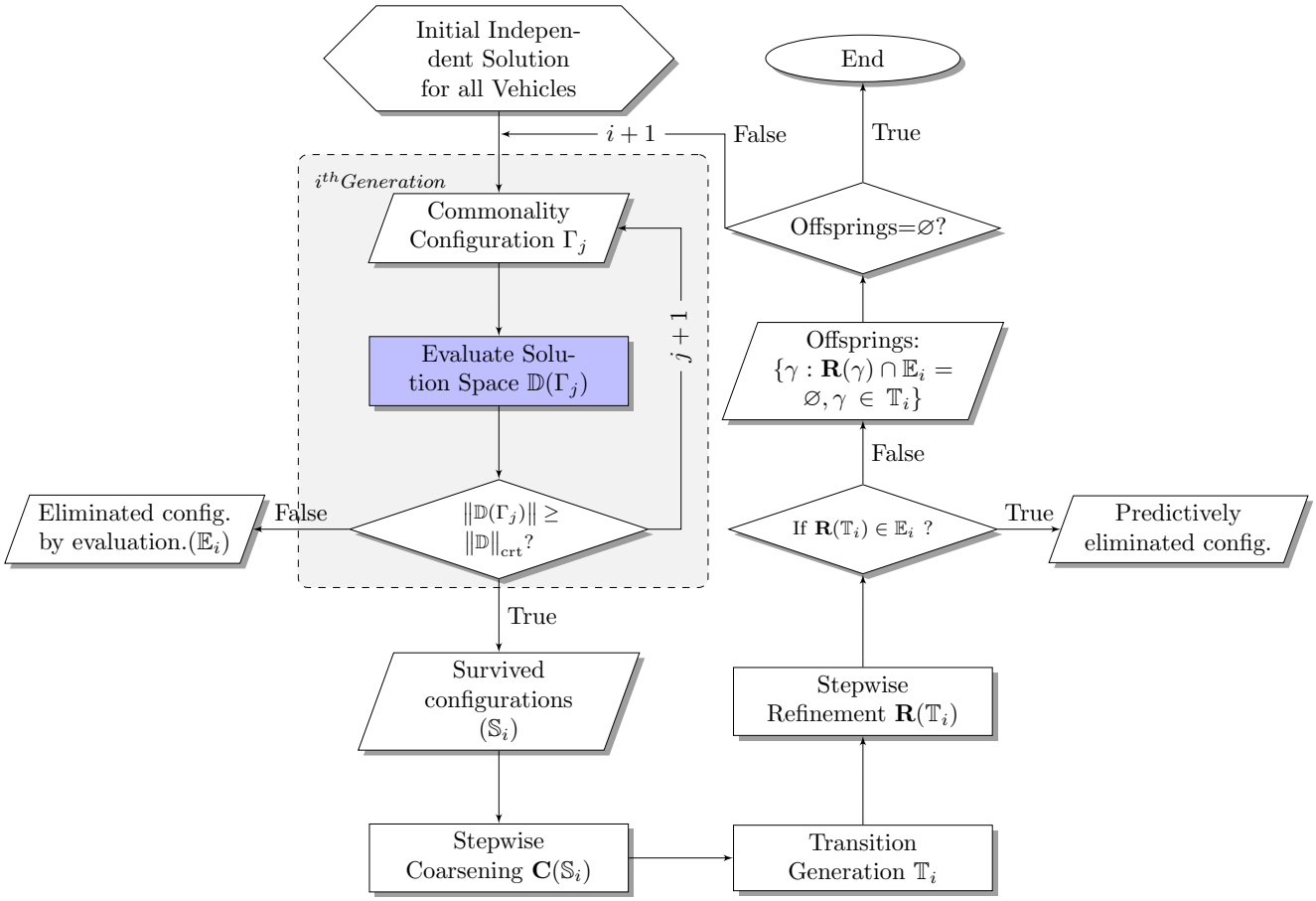


Figure 5.13: General work-flow of the predictive elimination algorithm.

With this established heuristic elimination algorithm, a considerable number of

[iii] https://commons.wikimedia.org/wiki/File:Set_partitions_4;_Hasse;_circles.svg

configurations can be predictively eliminated without Solution Space evaluation. Furthermore, all the procedures within one generation can be computed in parallel to improve the speed-up. The increased efficiency to solve the problem is discussed in the following section.

Reduction Rate of the Predictive Elimination Algorithm

Since the algorithm is heuristic, the reduction rate is first discussed here by specific examples. The discussion is then extended to the general case as far as possible. As an assumption, only the Solution Space evaluation operation is considered here to influence the complexity. The generation effort, in comparison to the Solution Space evaluation, is negligible.

The first step of the algorithm is to evaluate all the configurations in the first generation with $CI = 1$, which is only $\mathcal{O}(n^2)$ complex. If ONE configuration results in a critical Solution Space and thus has to be eliminated, the remaining configurations can be counted with:

$$B(P_e) - B(P_e - 1) = B(4) - B(3) = 10 \ , \tag{5.28}$$

which is the case shown in Figure 5.14:

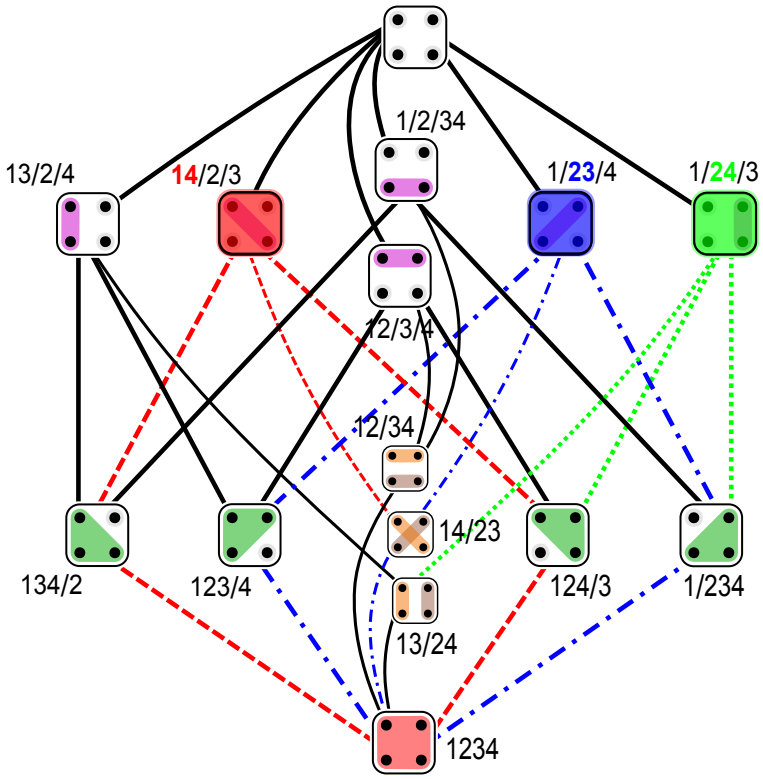


Figure 5.14: The scenarios of predictive elimination to analyze the reduction rate. (Figure is modified based on that from wikipedia[iv].)

if only the configuration 14/2/3 is eliminated, the combination "14" can be treated as "bad gene", all the configurations containing this bad gene "14" can be predictively eliminated. The configurations containing this bad gene are counted by $B(3)$, which explains Equation (5.28) – the number of the configurations containing bad gene is subtracted from the total number of configurations to result in the number of remaining configurations.

What happens if multiple bad genes are detected after the evaluation of the config-

[iv] https://commons.wikimedia.org/wiki/File:Set_partitions_4;_Hasse;_circles.svg

urations in first generation. An exemplary case is illustrated in Figure 5.14, where 14/2/3 (marked in red), 1/23/4 (marked in blue), and 1/24/3 (marked in green) result in Solution Spaces smaller than the prescribed criterion. The corresponding bad genes are 14 (g_1), 23 (g_2), and 24(g_3) respectively. The law for a single bad gene discussed above does not apply anymore. Otherwise it leads to double/triple counting of the eliminated configurations. This situation is described by a Venn Diagram (Venn 1880) in Figure 5.15: here, the sets of configurations which contain

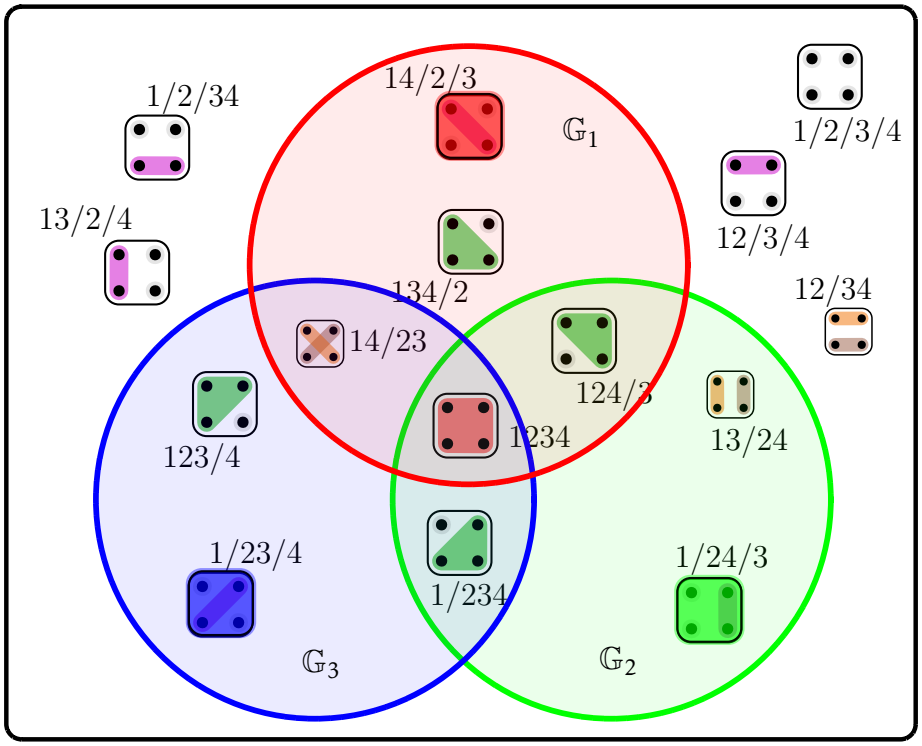


Figure 5.15: The scenarios of predictive elimination to analyze the reduction rate.

those bad genes are $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_3$. The number of the remaining configurations

after elimination, $|\mathbb{R}|$, can be calculated as:

$$\begin{aligned}
|\mathbb{R}| &= B(P_e) - |\mathbb{G}_1 \cup \mathbb{G}_2 \cup \mathbb{G}_3| \\
&= B(4) - |\mathbb{G}_1| - |\mathbb{G}_2| - |\mathbb{G}_3| + |\mathbb{G}_1 \cap \mathbb{G}_2| + |\mathbb{G}_1 \cap \mathbb{G}_3| + |\mathbb{G}_2 \cap \mathbb{G}_3| - |\mathbb{G}_1 \cap \mathbb{G}_2 \cap \mathbb{G}_3| \\
&= B(4) - 3B(3) + 3B(2) - B(1) \\
&= 15 - 3 \times 5 + 3 \times 2 - 1 = 5
\end{aligned} \tag{5.29}$$

Extending this logic to a general case with m bad genes (g_i, $i = 1, 2 \cdots, m$) identified in the first generation, the rest of the configurations after predictive elimination can be calculated as:

$$\begin{aligned}
|\mathbb{R}| &= B(P_e) - \sum_{i=1}^{m} |\mathbb{G}_i| + \sum_{1 \le i \le j \le m} |\mathbb{G}_i \cap \mathbb{G}_j| + \ldots + (-1)^m |\mathbb{G}_1 \cap \mathbb{G}_2 \ldots \cap \mathbb{G}_m| \\
&= \sum_{i=0}^{m} (-1)^i B(P_e - i) \cdot \binom{m}{i} \\
&= A(P_e, m) \ ,
\end{aligned} \tag{5.30}$$

in which $A(P_e, m)$ is proposed here to calculate the number of remaining configurations of a case with P_e end products and m bad genes being identified in the first generation by $CI = 1$.

In the next steps, the generality of $A(P_e, m)$ should be assessed: In the exemplary Case I, shown in Figure 5.16, the graph formed by the bad genes does not include

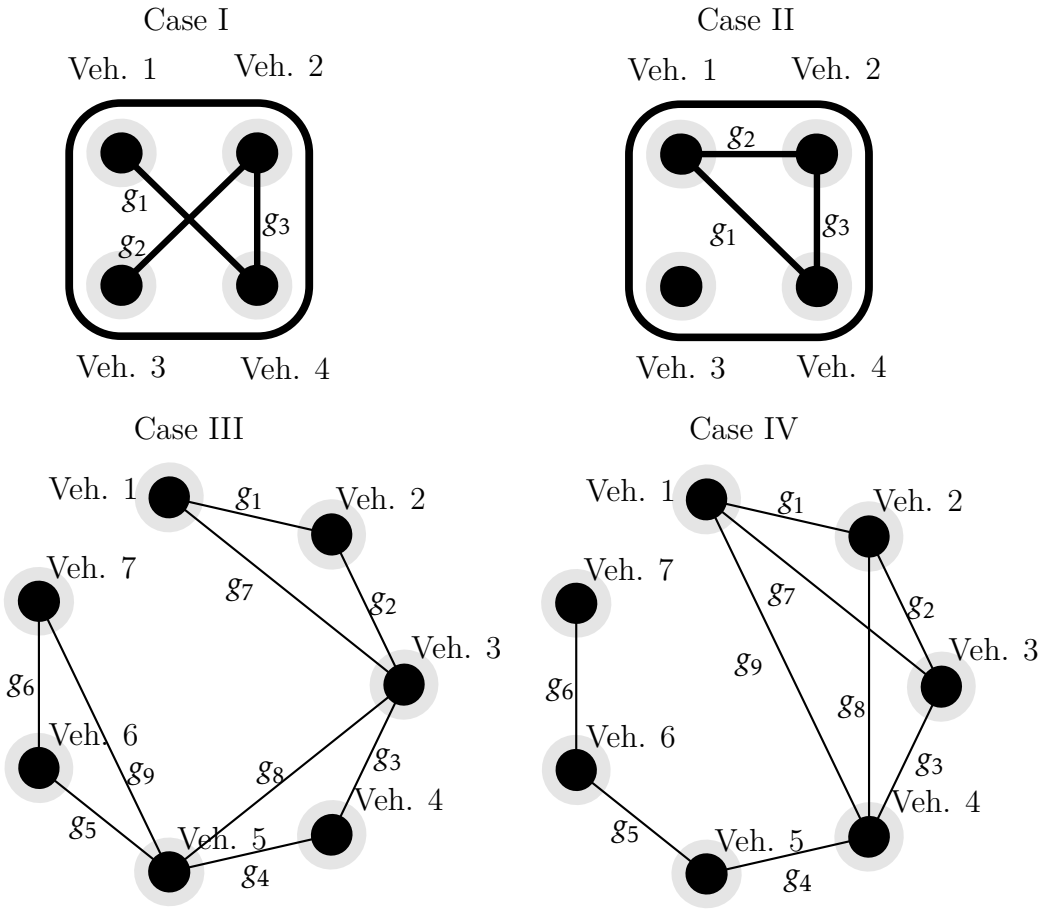


Figure 5.16: The general cases of the graphs formed by bad genes.

any cycle. The reduction rate of Case I can be resolved by the Equation (5.29). In the exemplary Case II in Figure 5.16, there is a cycle formed in the bad genes graph, the Formula (5.29) does not apply to this case (Formula (5.29) results in 5 configurations; however, only 4 are left after the predictive elimination).

To solve this problem, a "characteristic polynomial" is constructed in this work for the given number of vehicles P_e and known number of bad genes m. Based on the Equation (5.29), the definition x^p is introduced. p denotes the number of

cycles in the graph of bad genes. The characteristic polynomial is defined to fulfill the following properties:

Property 1. p represents the number of genes which form the cycle;

Property 2. $x^p \equiv A(P_e - p, m - p)$;

Property 3. When no cycles (i.e. $p = 0$), $x^0 = 1$ and is equivalent to $A(P_e - 0, m - 0) = A(P_e, m)$, which is consistent to Equation (5.30).

Based on this, the number of remaining configurations can be represented by the characteristic polynomial:

$$R(x) = 1 - (-x)^{p-1} \quad . \tag{5.31}$$

Applying this formula to Case II results in:

$$\begin{aligned} R(x) &= 1 - (-x)^{3-1} \\ &= 1 - x^2 \\ &\equiv A(4,3) - A(4-2, 3-2) \\ &= \sum_{i=0}^{3} (-1)^i B(4-i) \cdot \binom{3}{i} - \sum_{i=0}^{1} (-1)^i B(2-i) \cdot \binom{1}{i} \\ &= 4 \ , \end{aligned} \tag{5.32}$$

which is exactly the same compared to the result obtained by numerical valida-

tion.[v]

The Case III (Figure 5.16) generalizes the graph with multiple cycles formed by bad genes: 1^{st} cycle formed by bad genes g_1, g_2, and g_7; 2^{nd} cycle formed by g_3, g_4, and g_8; 3^{rd} cycle formed by g_5, g_6, and g_9. The characteristic polynomial can be formulated as:

$$\begin{aligned} R(x) &= \underbrace{[1-(-x)^{3-1}]}_{\text{1st cycle}} \cdot \underbrace{[1-(-x)^{3-1}]}_{\text{2nd cycle}} \cdot \underbrace{[1-(-x)^{3-1}]}_{\text{3rd cycle}} \\ &= 1 - 3x^2 + 3x^4 - x^6 \ . \end{aligned} \tag{5.33}$$

Substitute the $x^p = A(P_e - p, m - p)$ into Formula (5.33), the number of remaining configurations for Case III is 87, which is consistent to the result via numerical validation. The Formula is numerically confirmed to work on the "Cactus Graph" as exemplarily shown in Case III. [vi]

Case IV includes the cycles which intertwine together. The characteristic polynomial in the current given form cannot resolve this case. To find out the right form of characteristic polynomial for arbitrary intertwined cycles is by itself $\mathcal{O}(2^m)$ complex. Therefore, it is not worthwhile to formulate a reduction rate estimation problem as complicated as the original problem.

Here, a heuristic approach is suggested. For m bad genes generated by P_e vehi-

[v] The numerical validation is conducted by creating the complete Hasse Diagram and counting the remaining configurations exactly.

[vi] In graph theory, a cactus graph is a connected graph in which any two simple cycles have at most one vertex in common (N.N. 2018*b*).

cles, the minimal number of cycles formed by the bad genes is $\max\{0, m - P_e + 1\}$. Additionally, all the cycles are assumed to be formed by three bad genes (the smallest cycle in the graph of bad genes used in this work). These result in the characteristic polynomial to estimate the number of remaining configurations:

$$R(x)_{\mathbf{est}} = \max\{0, (1 - x^2)^{\max\{0, m - P_e + 1\}}\} \ . \tag{5.34}$$

With this estimation in the predictive elimination method, after the first generation, the number of the left configurations can be estimated. And the complexity of the problem is strongly reduced.

5.6 Summary of the Proposed Method

With the method described above, an approach to efficiently identify the commonality configurations, which fulfill the Solution Space size criterion is established. The steps are the following:

(1) The commonality configuration of the vehicle architecture is assigned by the Commonality Description Matrix (CDM), $\mathbf{\Gamma}$. The commonality extent of the architecture can be evaluated by the Commonality Index, which directly results from the $\mathbf{\Gamma}$.

(2) The size of Solution Space for the vehicle architecture prescribed by $\mathbf{\Gamma}$ can

be calculated by the extended approach, which is based on the Solution Space method established in Chapter 4.

(3) An efficient predictive elimination approach with the help of a Hasse Diagram is applied to eliminate the commonality configurations of the architecture based on the design prerequisites and prescribed design feasibility. The commonality configurations, which do not consist of the prescribed commonality or result in Solution Space smaller than the critical size, are eliminated.

(4) The survived commonality configuration with the highest Commonality Index gives the optimal commonality configuration(s) of the vehicle architecture.

"The most practical solution is a good theory."

Albert Einstein

6

Validation via an Industrial-sized Problem

In Chapter 6, the established approaches are validated and illustrated further via a series of more practical and complex problems. Therefore, it is assessed how far the questions put forward in Chapter 3 can be answered and explained via an industrial-sized application and where questions remain open or new issues arise. The seven methods (M1 to M7 labeled in Figure 6.1) proposed in the previous chapters are integrated into one work-flow shown in Figure 6.1 to tackle the practical problems.

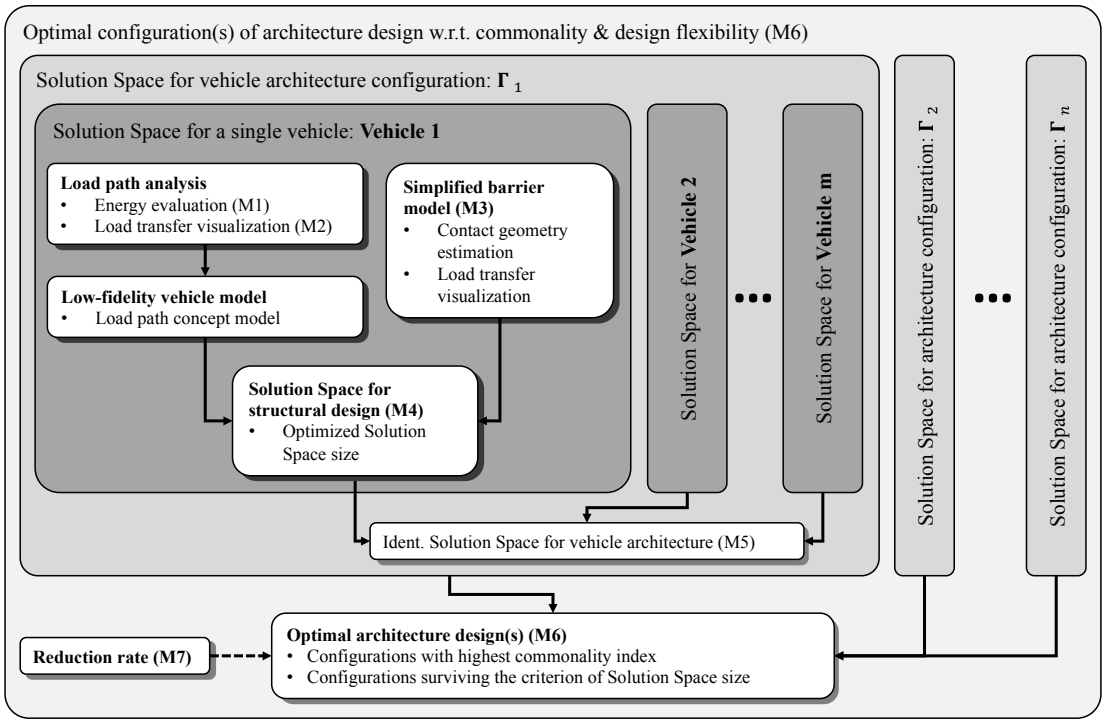
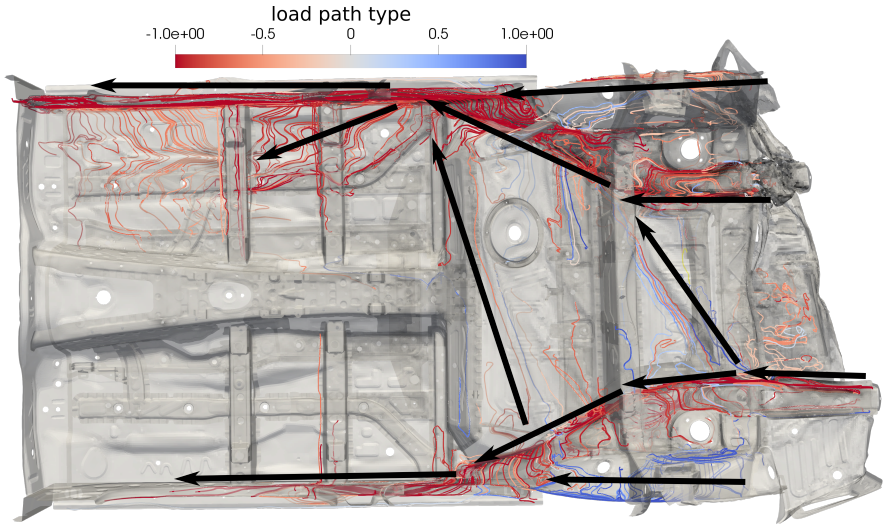


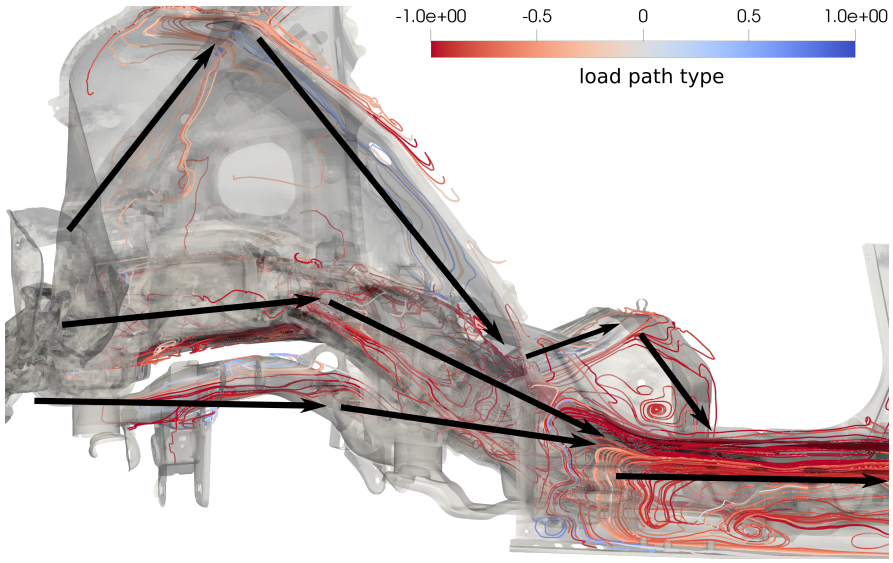
Figure 6.1: The proposed methods can solve a practical problem starting from establishing the low-fidelity models for each vehicle and its corresponding barrier response in the architecture (M1, M2, and M3), identifying the Solution Spaces for crashworthiness structural design (M4), extending the Solution Space evaluation for an arbitrary architecture configuration (M5), and optimizing the commonality configurations of the architecture with reduced complexity (M6 and M7).

6.1 Low-fidelity Model of Structure

For a vehicle subjected to the rear crash load, the components relevant for crashworthiness are identified by an energy evaluation of a predecessor vehicle, which is the **first method** of the overall approach proposed in this thesis. The basic result has already been shown in Chapter 4 and explicitly illustrated in Figure 4.7. This is complemented by the second method, which defines the load paths for the crash case regarded in the application. For axially oriented structures, this is rather trivial, while for more complex parts, this requires using the **second method** proposed in this thesis. The most non-axially oriented crash relevant "planar" structures are located in the floor group. For these components, the load path or load flow is ambiguous. Here, with the help of the load transfer visualization approach established in Section 4.3.1, the load path for the rear impact load case depicted in Figure 4.13(a) at any time step can be identified: this does not require a high precision, i.e. it is sufficient to define manually a rather coarse net of flow lines. Important is that they are sufficiently connected to enable the transfer of the forces. In the example regarded here, the black arrows in Figure 6.2 are drawn along the moderate flowing directions of the streamlines and thus indicate the selected load paths.



(a) The load path analysis (top view).



(b) The load path analysis (side view).

Figure 6.2: The load paths (shown as black arrows) are identified manually with the help of the load flow visualization for the rear impact.

For crashworthiness design, the goal is to determine the functional characteristics of the components within these load paths based on the overall design criteria and available information in the early phase. Hence, in a third step, the deformation

space model for the identified structures for the new vehicle with an analogous concept is constructed based on the expected structural deformation behavior. This deformation behavior is derived from the geometric information of the blocking packages, the location of critical components, the possible deformation ratio of these components, and a pre-defined deformation concept/mode. A more detailed discussion concerning the deformation space model is published with contributions of the author of the thesis at hand in (Lange et al. 2018).

Figure 4.17 shows a deformation space model of a real vehicle structure: on the right hand of the deformation space plot, six load paths are extracted and loaded by the impacting barrier through four contact surfaces[i]. These load paths are identified partially through the energy evaluation and partially manually with the help of the load flow visualization. The two-dimensional representation of the load paths in deformation space model is explained here in short: the original load paths depicted in Figure 6.2 are in the three-dimensional space, the analysis direction (discussed in Section 4.3.1) is predefined in global x-direction, which means that only the load transfer in x-direction is interesting. Consequently, the three-dimensional load path representation can be projected into x-direction, resulting the two-dimensional description in the deformation space model. The components in the deformation space model are those forming the load paths. The reaction

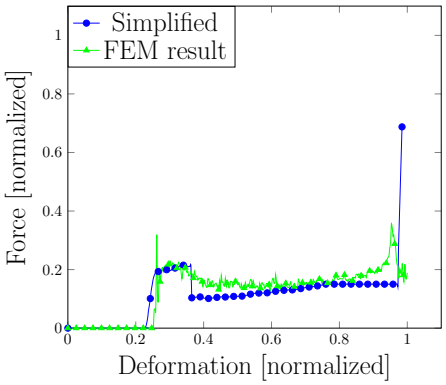
[i]Due to the confidentiality agreement, the data of the vehicle represented by this deformation space model cannot be published. Thus, another vehicle is used for the load path visualization in Figure 6.2.

forces of the barrier on the structure are then calculated using the **third method** proposed in this thesis, which is described in the next section.

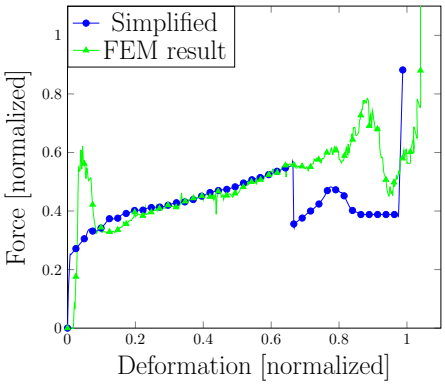
6.2 Response of Deformable Barrier

In a first step, the size of the contact surfaces between vehicle and barrier are estimated based on the package and geometric information of existing vehicles with comparable architecture. Then using the **third method** described in Section 4.3.2, the force-deformation curves of the barrier under each contact surface is obtained, as shown in the left part of Figure 4.17.

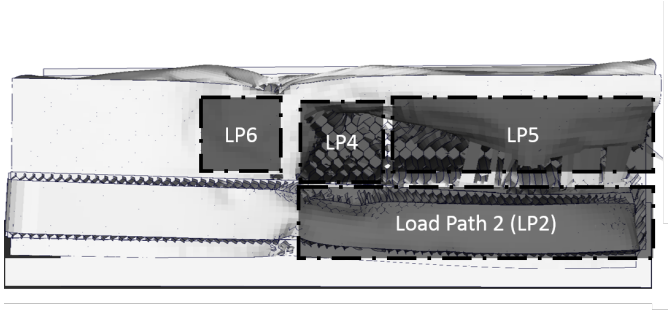
In addition, Figure 6.3 helps to indicate the validity and generation of these barrier curves. The curves of predicted force-deformation characteristics of the contact surfaces are validated against the finite element simulation. Figure 6.3(a) shows the curve of one contact surface intruding the main block, while Figure 6.3(b) shows one contact surface impacting the barrier bumper. Here, the approximation error is in the tolerance in early design phase, since the uncertainty of the estimation of contact geometry is normally higher. This validation of the barrier model only needs to be performed once if the property of the barrier does not change. The reaction force-deformation curves of the barrier for the given deformation space model are thus calculated based on the contact geometries shown in Figure 6.3(c).



(a) Contact force on main block of the barrier.



(b) Contact force on bumper of the barrier.



(c) Distribution of contact surfaces on the barrier and the corresponding load paths: Load paths 1, 2, 3 impact together onto contact surface LP2; load paths 4, 5, 6 have separate contact surfaces LP4, LP5, and LP6.

Figure 6.3: The comparison of the barrier reaction forces from FE simulation and simplified model. The simplified barrier model established in Section 4.3.2 generates the four force-deformation curves in deformation space model based on the illustrated contact geometries.

Up to here, the complete deformation space model of a real vehicle structure together with the predicted response of the deformable barrier is created. The Solution Space for functional characteristics of the components is identified and discussed in the following section.

6.3 Solution Spaces Identification for Single Vehicles

The proposed optimizer in the **fourth method** utilizes in the next step (here for a single vehicle) the information embedded in the deformation space model and the corresponding barrier boundary conditions to identify the maximal Solution Spaces for the force-deformation characteristics of the components, fulfilling all the constraints. Since each dimension of the Solution Space defines an interval of the force-deformation curve in one section, the intervals for each component are retrieved by connecting the upper and lower boundaries of the intervals separately based on the sequence of sections. This forms, for each component, a corridor in the force-deformation plot. These corridors are thus given to constrain the force-deformation characteristics in the crashworthiness component design, as shown in Figure 6.4.

The optimized Solution Spaces are utilized to guide the component design. To illustrate this, two sample cases are regarded here in an exemplary manner. This means, two detailed vehicle concepts using detailed finite element models for all components are taken. The first generates a set of force-deformation curves, which are shown in Figure 6.4 as "Bad Design". In this design, the intrusion in the vehicle violates the critical value, i.e. the tank is also deformed to absorb the crash energy. More explicitly, in Figure 6.4 it can be observed that the force-deformation curves of component 4 and component 10 show a relatively high violation of the limits

defined by the corridor values while component 9 does not deform at all as the corridor requires. This corridor violation information from the "Bad Design" is now used to modify specifically the design of component 4, 9, and 10 to generate an acceptable design. The proposed changes are the following (note that other modifications may also be successful)[ii]:

- The crushing strength of COMP4 is reduced by changing the wall thickness so that the component deforms with lower reaction force;
- the deformation of COMP9 is triggered by a set of beads/cutouts to fulfill the corridor requirement with respect to the deformation length;
- the deformation of COMP10 is limited by a set of ribbons while the crushing strength is reduced in the component design.

The new design shown in Figure 6.4 as "Good Design" has thus a minimized violation of the Solution Spaces. As a result, the modified design fulfills the crashworthiness criteria, here in particular the intrusion criterion concerning the tank in contrast to the previous "Bad Design". It is obvious that the wider the corridors are, the higher the design flexibility is.

[ii]The geometric designs of the components are not accessible due to the confidentiality agreement. The approaches of optimizing the structure are not the focus of this thesis.

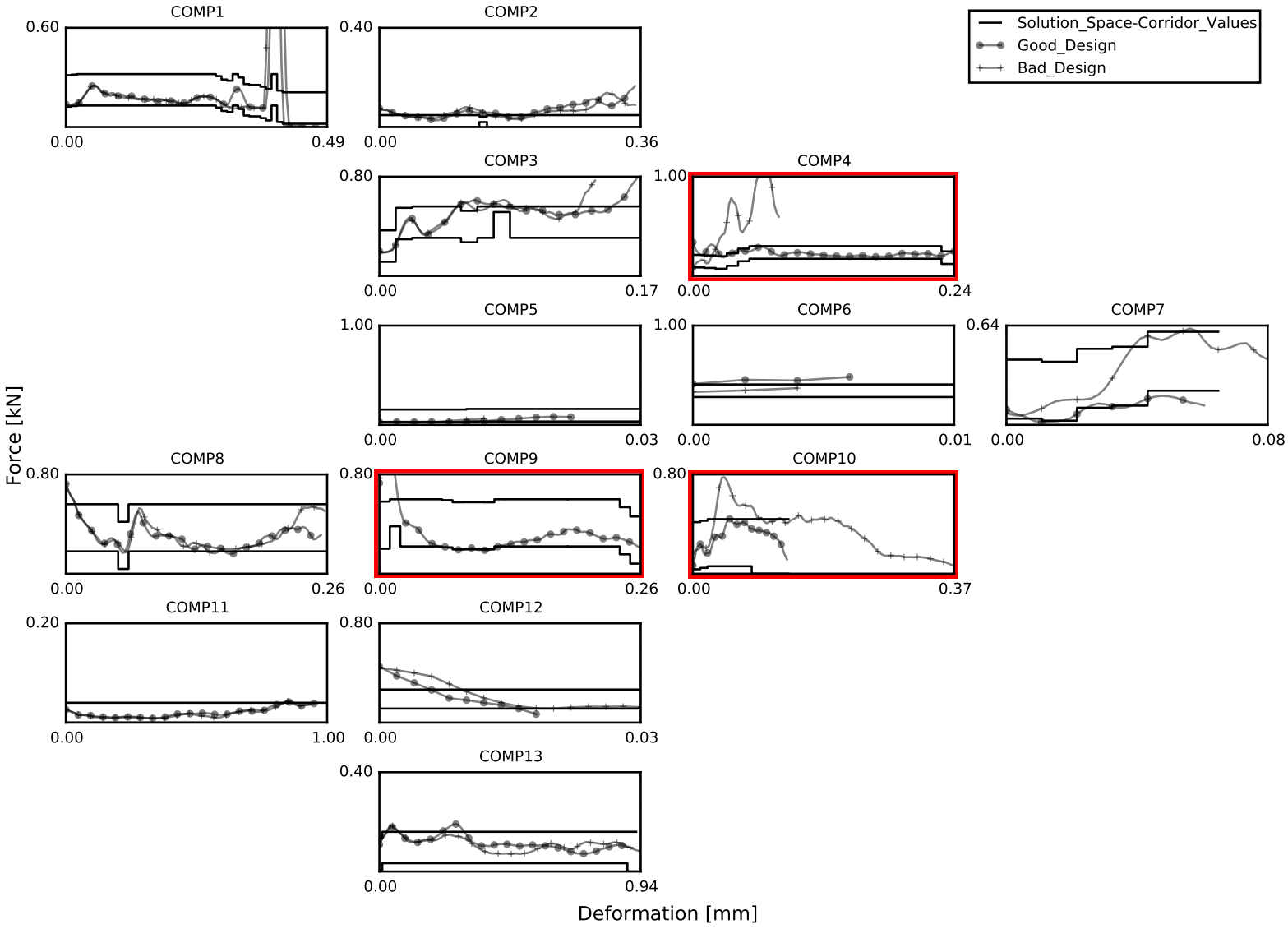


Figure 6.4: Solution spaces (Corridor values) of the force-deformation curves of each crashworthiness relevant component. The three components COMP4, COMP9, and COMP10 being modified regarding corridors are highlighted.

6.4 Solution Spaces for a Given Vehicle Architecture

In a next validation step, the Solution Space (corridors) calculation is extended to an architecture with several vehicles having common components. The coupling matrix introduced in Equation (5.4) requires that to each section of a common component the identical corridor is allocated. Figure 6.5 shows an example corridor of an architecture with three vehicles sharing three types of components calculated by the **fifth method**[iii]. The components, when shared between vehicles, have the same corridors. If the functional characteristics of one component fulfills one of these common corridors, this component design fits into all the other vehicles.

6.5 Commonality Optimization Based on Predictive Elimination

To summarize the Solution Space/ corridor properties discussed above: i) The corridor and the force-deformation characteristics of a specific design in Figure 6.4 indicate that, the wider the corridor is, the easier it is to adjust the force-deformation curves to fit into the corridor, thus fulfilling the overall design criteria. Consequently, the width of the corridor determines the design flexibility and the corresponding robustness with respect to uncertainties typical for early phases; ii) For

[iii]Due to the limitation from the real project, the single vehicle discussed in Section 6.3 was not extended to an architecture. Here, an architecture with vehicles having four load paths is considered instead in this section, which is comparable from the theory, method, and complexity point of view.

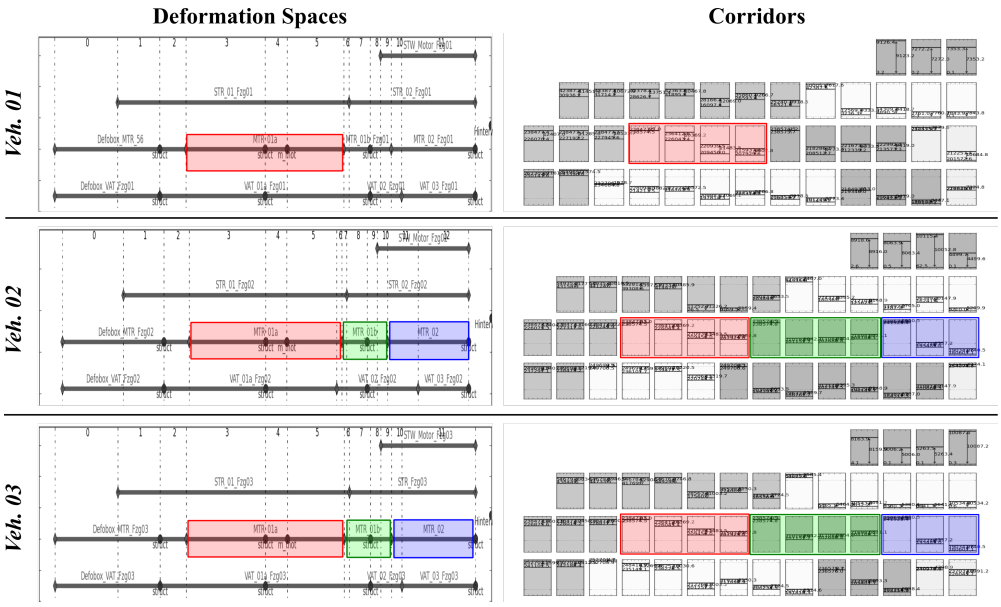


Figure 6.5: The common components are marked with identical colors in the deformation space models. The corresponding corridors of these common components are identical. (Note: The corridors of the components marked in red are identical. The optical discrepancy results from the different axes' scalings/ranges between load paths in the plot.)

the case of an architecture with multiple vehicles, the Solution Spaces of the common vehicles are coupled by the equality constraints introduced in Section 5.2.2. Any given commonality configuration can be evaluated by calculating the corridors.

If the corridors of a given commonality are too narrow, this cuts down the design flexibility and hence is not acceptable in the design criterion cascading. In order to find out the optimal commonality configurations which guarantee the design flexibility, the predictive elimination process established in Section 5.5 is applied on the following exemplary vehicle architecture: four vehicles with three

components types result in totally $B(4)^3 = 3,375$ configurations[iv]. In the visualization shown in Figure 6.6, each point represents one Solution Space evaluation of the corresponding commonality configuration. The color scale indicates the normalized corridor width (the darker the point, the wider the corridor). If the predictive elimination process is applied with the requirement that the size of the Solution Space should be larger than zero – there exists at least one feasible solution. Instead of evaluating all the configurations exhaustively as shown in Figure 6.6, only the configurations highlighted in Figure 6.7 are to be evaluated by the Solution Space calculation. Finally, the configurations which survive the elimination process with highest commonality index are marked in Figure 6.7 and given as suggested architecture design by applying the **sixth method**.

In this process, two phenomena are visible for the logic followed in the elimination process: Figure 6.8 indicates that if one configuration (the largest point in the figure) is infeasible, all the offspring generated by coarsening are not feasible. Therefore, these configurations can be predictively eliminated due to infeasibility. In addition, Figure 6.9 illustrates that if one configuration (the largest point in the figure) is generated by some feasible configuration but has at least one infeasible parent, this configuration is eliminated without evaluation.

[iv] For a readable visualization, the Hasse Diagrams here are demonstrated only for four vehicles and two components, resulting in $B(4)^2 = 225$ configurations.

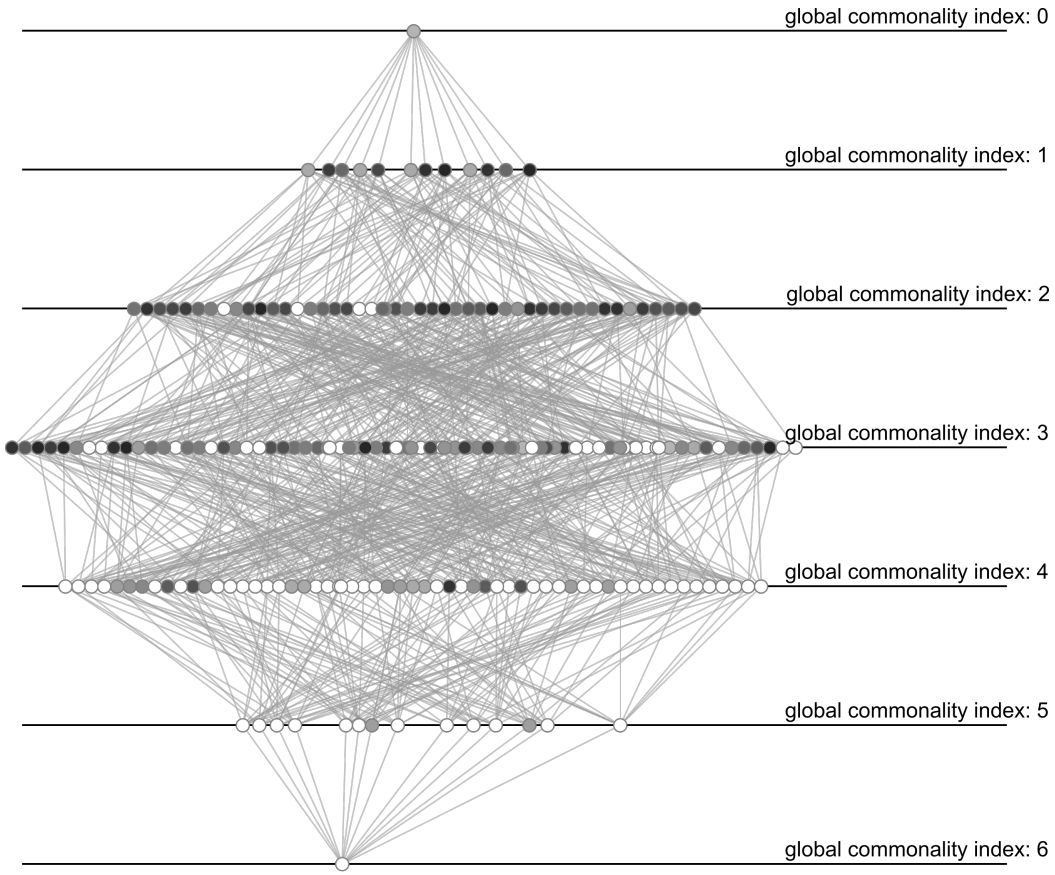


Figure 6.6: A full Hasse Diagram showing the combinatorics of set partitions for an architecture with four vehicles and two modules/components. There are in total 225 points in the diagram. Each point represents an architecture configuration. The color of the point indicates the size of the corresponding Solution Space – the darker the point, the larger the Solution Space of this configuration. The white points represent infeasible configurations. The lines connecting the points show the parent-offspring relations. All the points located on the same horizontal line have the same commonality index(CI).

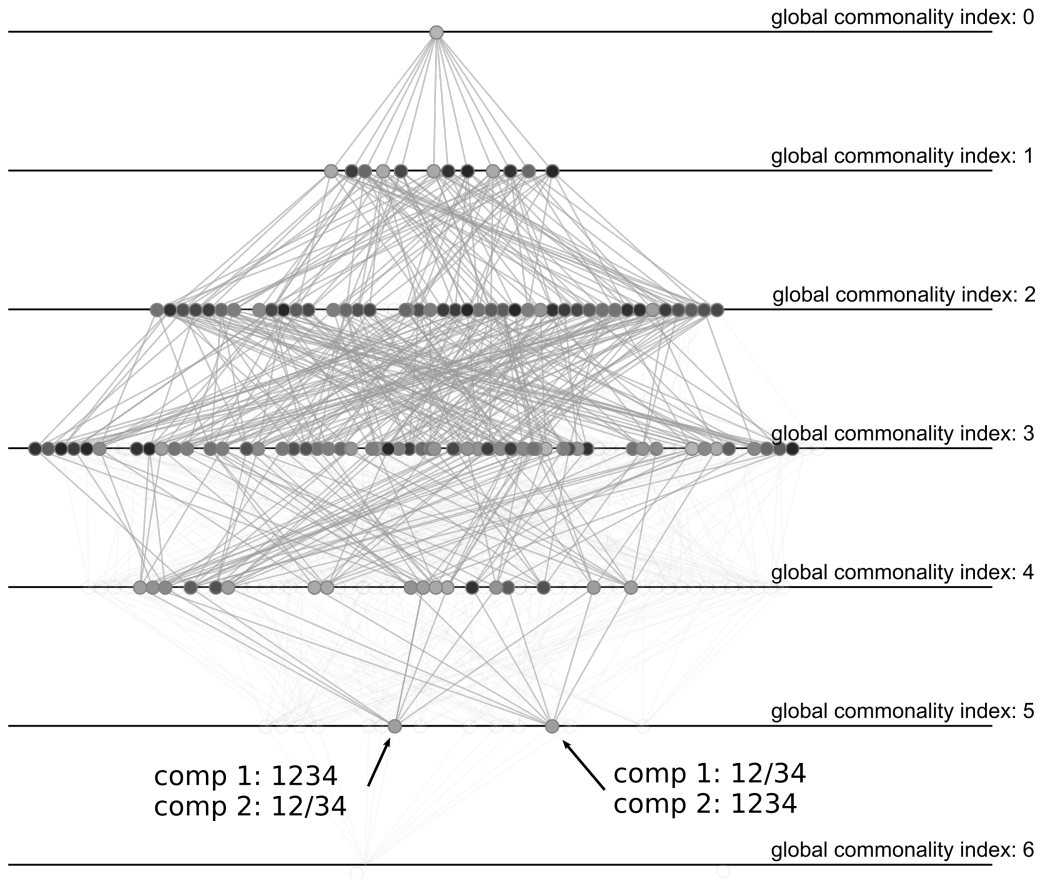


Figure 6.7: The evaluated configurations after the predictive elimination process. The reduced number of evaluated configurations reduces the complexity of the original problem. The two points with $CI = 5$, fulfilling the surviving criterion $\|\mathbb{D}\| > 0$, are given as optimal architecture designs.

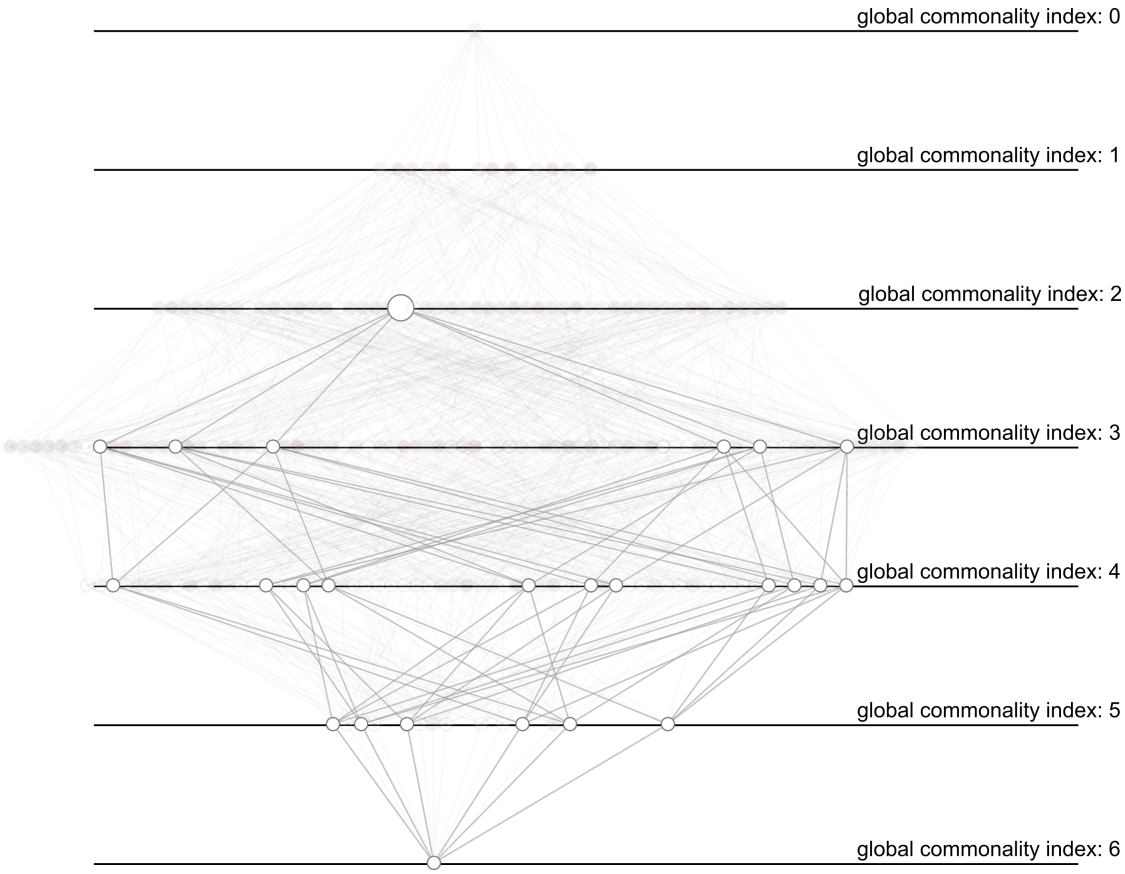


Figure 6.8: If one configuration (the largest highlighted white point) has infeasible Solution Space, all its off-springs are infeasible.

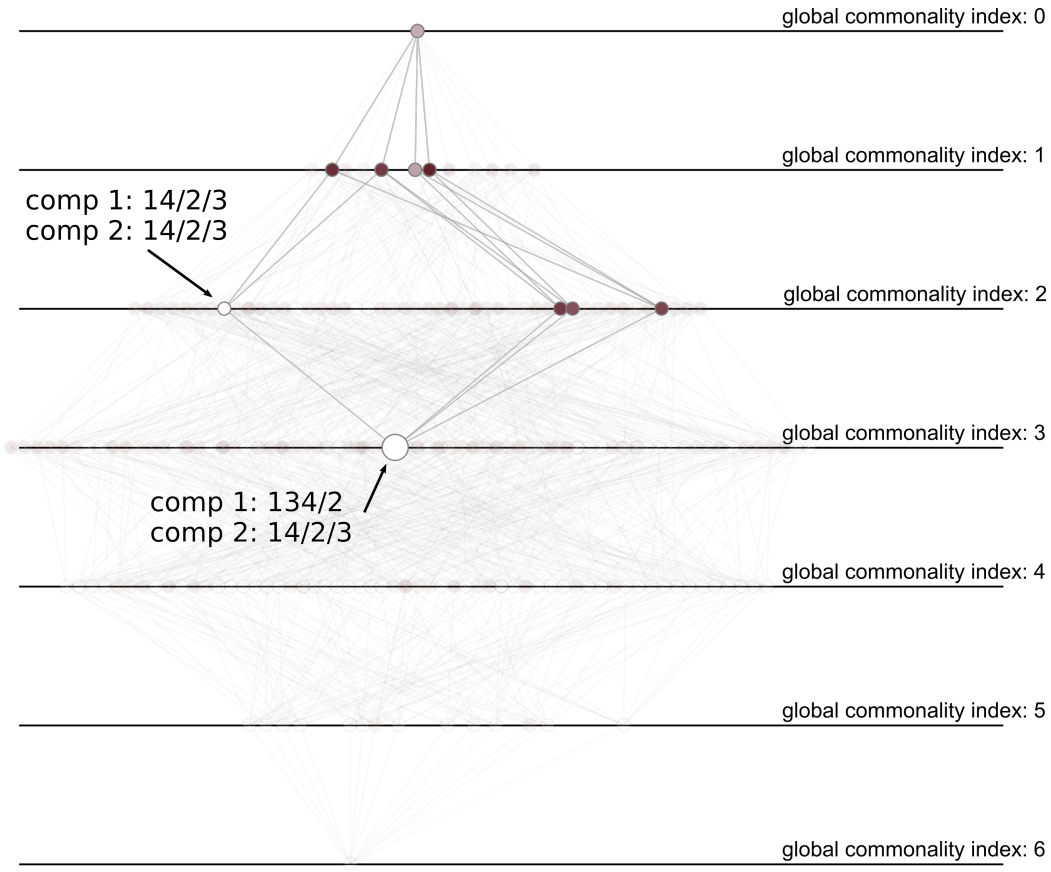


Figure 6.9: If one configuration (the largest highlighted point) has at least one infeasible parent (the smaller highlighted white point), this configuration is for certain infeasible and, thus, can be eliminated.

6.5.1 Reduction Rate

A heuristic rule based on the derived characteristic polynomial to predict the reduction rate has been proposed in Section 5.5.5. In the example stated in the last section, there are no infeasible configurations being eliminated in the first generation. The heuristic prediction rule does not apply in this case.

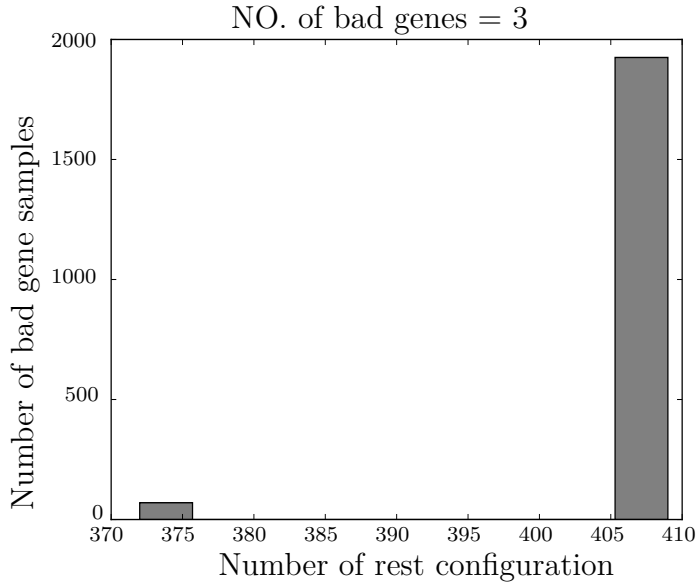
For the validation purpose in this chapter, in order to check the validity of this

prediction rule, an even more general numerical experiment is designed as following:

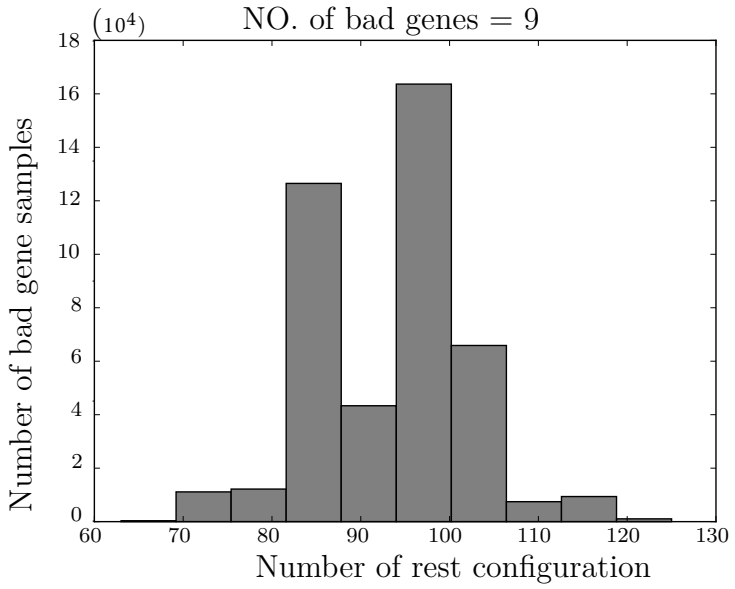
Assume an architecture with $P_e = 7$ vehicles, the commonality of one component for these vehicles is to be configured. This setup compromises between the numerical effort and generality. The population size of the first generation, where $CI = 1$, is calculated by $\binom{P_e}{2} = \binom{7}{2} = 21$. This means that the number of bad genes ranges from 0 to 21.

In the numerical experiment, the possible bad genes configurations under certain counts (from 1 to 21) are generated randomly to cover the various cycle topologies. The hence corresponding remaining configurations after the predictive elimination are counted exhaustively and summarized in histograms. For instance, if there are 9 bad genes in the first generation, the randomly generated combinations of 9 bad genes[v] result in 26 different numbers of remaining commonality configurations after the elimination. Most of the combinations of 9 bad genes leave 87 commonality configurations surviving after the elimination, as shown in Figure 6.10. Another example for 3 bad genes is given in the histogram as well. To summarize, the same number of bad genes with different combinations in the first generation result in scattered number of remaining configurations after the predictive elimination.

[v]For instance, "12, 23, 34, 45, 56, 67, 17, 26, 35" and "13, 26, 35, 47, 57, 67, 15, 24, 46" are two different combinations of 9 bad genes.



(a) In the numerical experiment, 1995 random combinations of 3 bad genes are generated, 1925 of them lead to 409 remaining configurations after elimination.



(b) In the numerical experiment, 440895 random configurations with 9 bad genes are generated. The rest configuration after elimination is in the range from 63 to 125.

Figure 6.10: Two examples showing that the same number of bad genes with different configurations result in scattered number of surviving configurations.

The Formula (5.34) (set up in seventh method) is applied to estimate the number of remaining configurations based on the number of bad genes. In Figure 6.11, the sizes of the dots indicate the possibility that how often can the corresponding remaining number of configurations after elimination happen. It is shown that the number of remaining configurations obtained most often (shown by the larger dots at each number of bad genes) is consistent between analytical prediction and numerical counting. The disagreement between analytical predication and numerical validation happens when the number of bad genes increases, which leads to more complicated topologies of the cycles formed by those bad genes. However, this prediction error is deemed to be acceptable in real world applications.

In this chapter, the proposed methods are applied and validated by industrial applications. Despite of the solved problems, some further questions and difficulties arise, which are discussed in the next chapter.

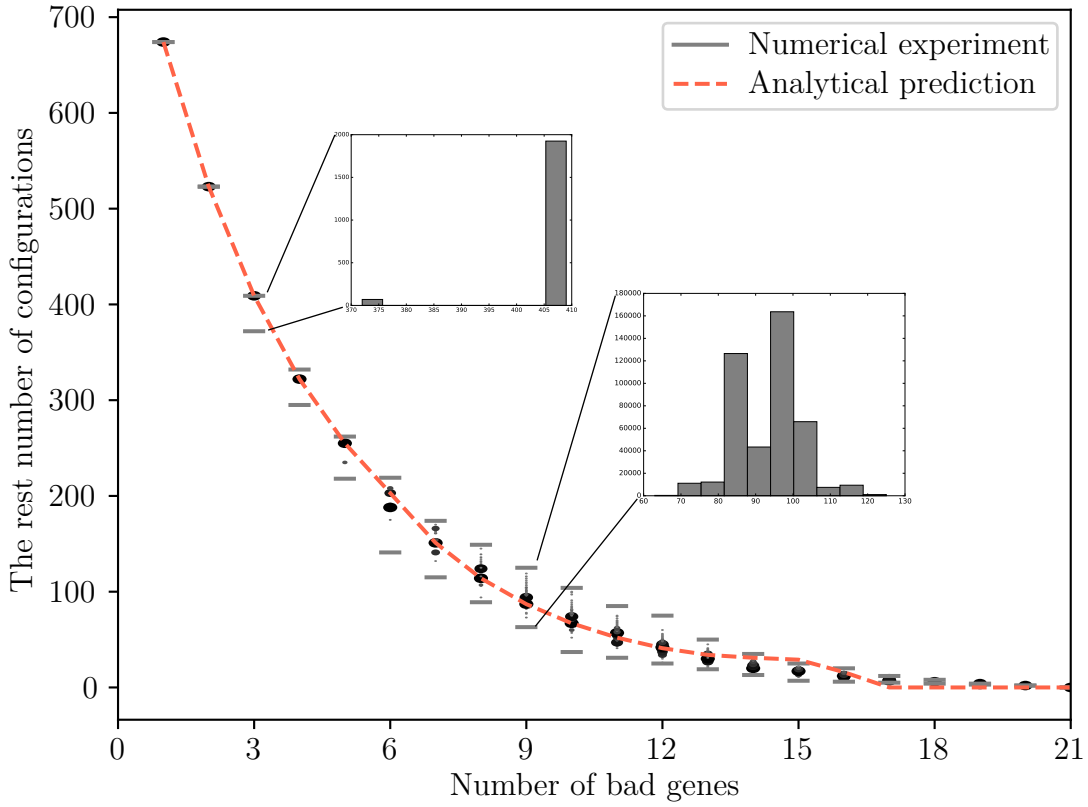


Figure 6.11: The number of bad genes is in the range from 1 to 21. The numerical experiment established in this section is conducted for every number of bad genes. It is evident that the same number of bad genes in first generation always results in scattered number of remaining configurations. The analytical prediction rule can almost predict the most often remaining configurations by each number of bad genes.

"The important thing is not to stop questioning."

Albert Einstein

7

Critical Reflections

In order to answer the research questions been raised in Chapter 3, a systematic approach is proposed in Chapter 4 and 5. However, several assumptions and simplifications are agreed to avoid the unnecessary complexity. These simplifications lead to loss of generality in the practical application, which is to be discussed in this chapter. Furthermore, potential solution strategies to overcome these limitations are provided.

7.1 About the Low-Fidelity Model

7.1.1 Simplified vehicle model

The proposed approach assumes that all relevant components of the vehicle structure can be approximated by simplified elements, which deform one-dimensionally (axially). Hence, the characteristics of these elements are described here by single axial force–deformation curves. However, in the real crash case, the transverse components, such as vehicle bumper, are subjected to lateral loading as well. Additionally, if other load cases with a deformable barrier, e.g. side crash, are to be resolved by the approach established in this research, the crash relevant components are mostly loaded laterally. This means that functionalities of these components should be considered by a multi-dimensional modeling, which can then account for the coupling of load paths via bending of components. The basic element of the simplified model can be enhanced by defining not only the

translational force-deformation characteristic curves but also moment-curvature characteristics. This is to be investigated and the proposed methodology will be extended correspondingly.

The established load path analysis approach in Section 4.3.1 to assist constructing this one-dimensional simplified model requires the stress information of the entire analysis structure domain. The generation of the load path field lines in each simulation time step demands stepwise integrals over the structure domain, which is computationally costly for large models. In order to overcome this limitation, the Line Integral Convolution (LIC), seen in Definition A.1, can be adopted. This visualization can be performed at each simulation time step. A texture is generated all over the surface of the structure, which is much more efficient compared to stream line generation. As a pragmatic solution, the LIC can be firstly used to identify roughly the interesting regions as well as the important simulation time steps. Based on which the load path field lines can be further generated. An LIC example is shown in Figure 7.1

7.1.2 Simplified barrier model

In the proposed barrier model, the reaction forces of the barrier are calculated based on the estimated contact surfaces. In the established approach, it is assumed that the interactions between the contact surfaces are negligible, which is a reasonable assumption for the main block but not for the bumper. These

Figure 7.1: The LIC visualization: the color scale indicates the intensity of the load flow (yellow color tone represents higher magnitude of the vector field) while the texture depicts the orientation of the vector field.

interactions need to be considered for in a load case in which multiple contact surfaces happen on the deformable barrier bumper, as partially shown in Figure 7.2. The suggestion to resolve this problem is to quantify the interactive influences by a shape-function-like weighting curve, shown in the image underneath the barrier drawing in Figure 7.2. This solution will be discussed more in detail in an additional research project, in which the interaction is significant.

7.1.3 Interaction between deformable barrier and vehicle

Despite the critical aspects of the simplified modeling for vehicle and barrier respectively, the overall acceleration of the vehicle during the crash, which is normally limited to protect the occupants from injuries (e.g. whiplash), needs to be

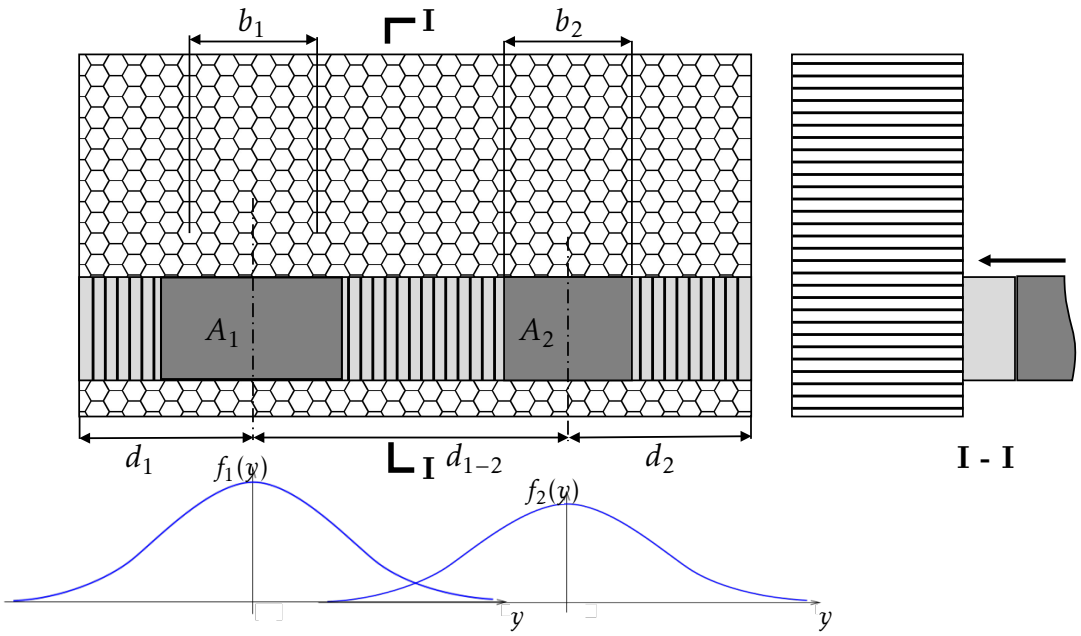


Figure 7.2: Two impactors intrude into the barrier bumper together. The interactions are shown schematically by the two curves in the lower part of this figure.

evaluated. In order to obtain the acceleration of the vehicle, the sum of the forces in the components from parallel load paths, which deform simultaneously, must be calculated. In the method described here, the acceleration history is not available since the order of deformation between the barrier and load paths is not uniquely prescribed. Extra constraints must be inserted to predefine an order of deformation between the barrier and vehicle for the evaluation of acceleration-relevant design criteria. This will be also addressed in the future.

7.2 About the Combinatorial Optimization

7.2.1 Complexity after predictive reduction

A heuristic predictive elimination process has been developed to reduce the complexity of the problem. The advantages and necessities of this approach have been stated in Chapter 5. In this section, some difficulties this approach faces are discussed.

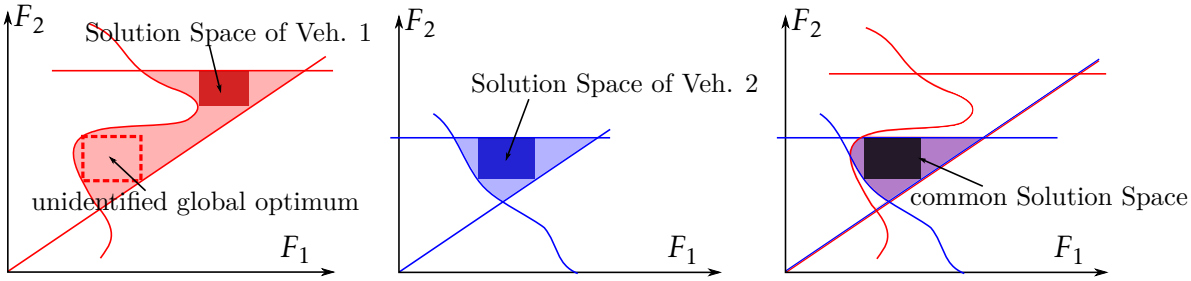
7.2.2 Complexity after predictive elimination

The reduction rate Formula (5.34) indicates that the rest of the configurations is counted as $A(P_e, m) \pm \mathcal{O}(A(P_e - i, m - i))$, in which $i \geq 2$, which means theoretically, the rest number of configurations can still have the same complexity as $B(P_e)$. This can be interpreted in the practice that if there exist not enough bad genes regarding the critical size of Solution Space, the elimination process may lead to test almost all the configurations exhaustively.

Moreover, the remaining configurations of each single type of component are combined in a full factorial manner to form the configurations of an entire vehicle architecture. If the rest number of some component types cannot be sufficiently reduced, the rest number of the configurations for vehicle architecture to be tested remains large.

7.2.3 Strong non-linearity and non-convexity in Solution Space evaluation

The established predictive elimination process is rather effective when the Solution Spaces used to evaluate the configurations are subjected to linear constraints. Which is the case shown in Figure 5.4. Here, the increasing commonality requirements lead to monotonous decreasing of Solution Space size. However, if the constraints of Solution Spaces are strongly nonlinear and non-convex, the size of Solution Space without commonality constraints can have several unidentified local optima, shown in Figure 7.3(a). In this exemplary case for Veh. 1, the optimizer fails to detect the global optimum. When the Veh. 1 is set to be common with Veh. 2, whose Solution Space is depicted in Figure 7.3(b), the commonality constraints may redirect the optimizer to find another even better local optimum, shown in Figure 7.3(c). This results in non-monotonously decreasing of the size of Solution Space with increasing commonality index, i.e. the Solution Space in Figure 7.3(c) with higher commonality is larger than that with no commonality in Figure 7.3(a). As a consequence, the predictive elimination process may unexpectedly eliminate feasible configuration.



(a) The Solution Space of Veh. 1 subjected strongly non-linear constraint. The optimizer is only able to identify one local optimum.

(b) The Solution Space of Veh. 2.

(c) The common Solution Space fulfilling the constraints of Veh. 1 and Veh. 2.

Figure 7.3: The Solution Space identification problem of the vehicle architecture caused by the strong non-linear constraint(s) and non-convex design space.

7.2.4 Genetic algorithms to control the computational cost

In order to overcome the two problems stated above, the genetic algorithms can be adopted to enhance the heuristic search. Each configuration of the set partition can be transferred into binary string with the help of rook placement model given in Definition A.2. The procedure is to be discussed in the future work.

The critical reflections stated above are considered to guide the further research projects as well as to improve the practicality of the established approach.

"When you reach the end of your rope, tie a knot in it and hang on."[a]

Franklin D. Roosevelt

[a] "Franklin D. Roosevelt Quotes." BrainyQuote.com. BrainyMedia Inc, 2018. 13 November 2018. https://www.brainyquote.com/quotes/franklin_d_roosevelt_101840

8

Conclusions

8.1 Conclusions

In this work, an approach to develop vehicle architectures considering in particular commonality constraints for early phase crashworthiness design is established for the first time for a load case with deformable barrier. This enables the decision making to the commonality configuration in the early design phase: the infeasible commonality configurations of the vehicle architecture with respect to design flexibility are eliminated. The predictive elimination process requires much less evaluations than the number of eventually eliminated infeasible configurations. The complexity of the original NP-hard problem is therefore strongly reduced. This heuristic elimination approach can be applied to other product commonality design cases as long as a certain commonality configuration can be evaluated efficiently.

The efficient assessment of the commonality configuration in this thesis is achieved by identifying the Solution Space of the structure design parameters (i.e. force-deformation characteristics of the crash relevant components) for the given commonality configuration. Namely, the size of the Solution Space assesses directly the feasibility of the commonality configuration. The Solution Space identification approach is thus extended to be applied on a load case which includes a deformable barrier. This enhanced Solution Space approach can also be applied independently of the commonality evaluation to enable an efficient V-model-based

structure design for rear impacts in the early design phase.

Aiming on extend the Solution Space to handle load cases with deformable barriers, a low-fidelity model to describe vehicle-barrier interaction is proposed. The method is based on an initial extraction of relevant load paths of the structure in a rear impact (here FMVSS 301). In the load path analysis, the internal energy information and load transfer through structure assembly of comparable vehicles is taken into account complemented by expert knowledge. Then, the impacted contact areas between barrier and vehicle are estimated for each load path (position, size, and shape). This allows to approximate the barrier reaction forces, which are then used as boundary conditions for the load paths design in the vehicle. Then, a simplified model for barrier–vehicle interaction is proposed where the vehicle is represented by a non-linear lumped mass-spring system. The thus established low-fidelity model allows to derive the upper and lower force–deformation curves for components in each load path, which define the corridors for the Solution Space method. These corridors decouple and modularize the development of the crash relevant components – if the force-deformation curve of each component locates within its given corridor, the overall vehicle crashworthiness criteria are fulfilled. In early design phases, it is advantageous if these corridors are as wide as possible (at least where this is needed) to obtain a flexibility concerning design changes realized later in the design process. Hence, a new optimization scheme accounting

for the nonlinearities due to the barrier–vehicle interaction is proposed. This set of maximized corridors is called optimal Solution Space.

To summarize, this work presents (i) a new development process to design the commonality of vehicle architectures with the help of (ii) a new simplified modeling for rear impact together with (iii) a new optimization algorithm enabling the usage of this approach within the established Solution Space approach. Through this, we believe to contribute to the highly efficient development of vehicle architectures with respect to crashworthiness in early design stages.

A

Appendix

The visualization approach – Line integral convolution mentioned in Section 7.1.1 and applied in Figure 7.1 is defined as follows:

Definition A.1: Line integral convolution (Cabral & Leedom 1993)

Let $\mathbf{u}$ be the vector field. Then a streamline parametrized by arc length can be defined as $\frac{d\sigma(s)}{ds} = \frac{\mathbf{u}(\sigma(s))}{|\mathbf{u}(\sigma(s))|}$. Let $\sigma_{\mathbf{r}}(s)$ be the streamline that passes through the point $\mathbf{r}$ for $s = 0$. Then the image color at $\mathbf{r}$ can be set to

$$D(\mathbf{r}) = \int_{-L/2}^{L/2} k(s)N(\sigma_{\mathbf{r}}(s))ds \tag{A.1}$$

where $k(s)$ is the convolution kernel, $N(\mathbf{r})$ is the noise image, and L the length of streamline that is followed.

The rook placement representation adopted to transfer the set partition configurations into binary strings is defined as follows:

Definition A.2: Rook placement representation (Mansour 2012)

The n-th triangular board is the board consisting of $n-1$ columns with with $n-i$ cells in the i-th row, (first column is the leftmost column and first row is the top row). For convenience, we also join pending edges at the right of the first row and at the top of the first column. A rook placement is a way of placing non-attacking rooks on such a board, that is, putting no two rooks in the same row or column. Let $R_{n,k}$ be the set of all rook placement of n – k rooks on the n-triangular shape, where a rook is indicated by a black disk.

Figure A.1 shows an example of the rook replacement taken from (Mansour 2012):

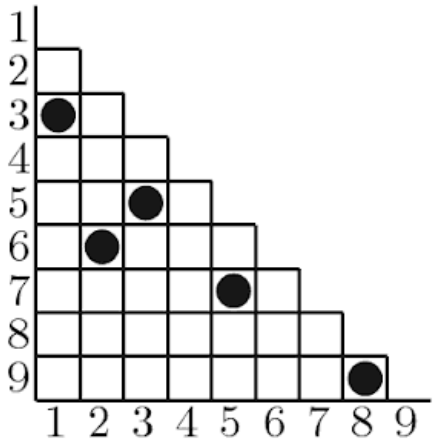


Figure A.1: A rook placement representation of 1357/26/4/89 (Mansour 2012).

References

Balakrishnan, P. V. & Jacob, V. S. (1996), 'Genetic algorithms for product design', *Management Science* **42**(8), 1105–1117. doi: 10.1287/mnsc.42.8.1105.

Bell, E. T. (1934), 'Exponential polynomials', *Annals of Mathematics* **35**(2), 258–277. doi: 10.2307/1968431.

Berend, D. & Tassa, T. (2010), 'Improved bounds on Bell numbers and on moments of sums of random variables', *Probability and Mathematical Statistics* **30**(2), 185–205. doi: 10.1111/j.1532-5415.2004.52225.x.

Berman, J. & Idziak, P. M. (2005), *Generative complexity in algebra*, Vol. 175, Memoirs of the American Mathematical Society. doi: 10.1090/memo/0828.

Bois, P. D., Chou, C. C., Fileta, B. B., Khalil, T. B., King, A. I., Mahmood, H. F., Mertz, H. J., Wismans, J., Prasad, P. & Belwafa, J. E. (2004), *VEHICLE CRASHWORTHINESS AND OCCUPANT PROTECTION*, American Iron and Steel Institute.
URL: *http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.208.2312*

Boström, O., Svensson, M., Aldman, B., Hansson, H., Häaland, Y., Lövsund, P., Seeman, T., Suneson, A., Säljö, A. & Örtengren, T. (1996), A new neck injury criterion candidate-based on injury findings in the cervical spinal ganglia after experimental neck extension trauma, *in* 'Proc. IRCOBI Conf.', Dublin, Ireland.

Cabral, B. & Leedom, L. C. (1993), Imaging vector fields using line integral convolution, *in* 'Proc. of the 20th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '93', Anaheim, CA, USA, pp. 263–270. doi: 10.1145/166117.166151.

Chinneck, J. W. (2008), *Feasibility and Infeasibility in Optimization: Algorithms and Computational Methods*, Vol. 118 of *Int. Series in Operations Research and Management Science*, Springer. doi: 10.1007/978-0-387-74932-7.

Collier, D. A. (1981), 'The measurement and operating benefits of component part commonality', *Decision Sciences* **12**(1), 85–96. doi: 10.1111/j.1540-5915.1981.tb00063.x.

Dean, W. (2016), Computational complexity theory, *in* E. N. Zalta, ed., 'The Stanford Encyclopedia of Philosophy', winter 2016 edn, Metaphysics Research Lab, Stanford University, Stanford, CA, USA.

Dobiński, G. (1877), 'Summieirung der Reihe $\sum \frac{n^m}{n!}$ für $m = 1, 2, 3, 4, 5 \ldots$', *Archiv der Mathematik und Physik* **61**, 333–336.

Eichberger, A., Steffan, H., Geigl, B., Svensson, M., Boström, O., Leinzinger, P. & Darok, M. (1998), Evaluation of the applicability of the neck injury criterion (NIC) in rear end impacts on the basis of human subject tests, *in* 'Proc. IRCOBI Conf.', Vol. 1998-13-00, Göteborg, Swenden, pp. 321–333.

Fellini, R., Kokkolaras, M., Papalambros, P. Y. & Perez-Duarte, A. (2002), 'Platform selection under performance loss constraints in optimal design of product families', *ASME 2002 Design Engineering Technical Conferences and Computer and Information in Engineering Conference* (734), 613–621. doi: 10.1115/DETC2002/DAC-34099.

Fender, J., Duddeck, F. & Zimmermann, M. (2014), 'On the calibration of simplified vehicle crash models', *Structural and Multidisciplinary Optimization* **49**(3), 455–469. doi: 10.1007/s00158-013-0977-7.

Fender, J., Duddeck, F. & Zimmermann, M. (2017), 'Direct computation of solution spaces', *Structural and Multidisciplinary Optimization* **55**(5), 1787–1796. doi: 10.1007/s00158-016-1615-y.

Fender, J. H. W. (2014), Solution spaces for vehicle crash design, PhD thesis, Technische Universität München. ISBN 9783844025507.

Gausemeier, J. & Moehringer, S. (2002), 'VDI 2206- A New Guideline for the Design of Mechatronic Systems', *IFAC Proceedings Volumes* **35**(2), 785–790. **URL:** *https://www.sciencedirect.com/science/article/pii/S1474667017340351*

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st edn, Addison-Wesley Longman Publishing Co., Inc. doi: 10.5860/CHOICE.27-0936.

Green, P. E. & Krieger, A. M. (1985), 'Models and heuristics for product line selection', *Marketing Science* **4**(1), 1–19. doi: 10.1287/mksc.4.1.1.

Hilton Holloway (2013), 'How BMW and MINI will sell a million cars per year off one platform', https://www.autocar.co.uk/car-news/new-cars/how-bmw-and-mini-will-sell-million-cars-year-one-platform. [Online; accessed 4-11-2018].

Ho, T.-H. & Tang, C. S. (1998), *Product Variety Management: Research Advances*, Int. Series in Operations Research & Management Science, Springer US. ISBN 978-1-4615-5579-7.

Holtta, K. M. M. & Salonen, M. P. (2003), Comparing three different modularity methods, *in* 'ASME 2003 Int. Design Engineering Technical Conferences and Computers and Information in Engineering Conference', Vol. 3b: 15th International Conference on Design Theory and Methodology, American Society of Mechanical Engineers, Chicago, Illinois, USA, pp. 533 – 541. doi: 10.1115/DETC2003/DTM-48649.

Hou, W., Shan, C., Yu, Y., Hu, P. & Zhang, H. (2017), 'Modular platform optimization in conceptual vehicle body design via modified graph-based decomposition algorithm and cost-based priority method', *Structural and Multidisciplinary Optimization* **55**(6), 2087 – 2097. doi: 10.1007/s00158-016-1629-5.

Jiao, J., Simpson, T. W. & Siddique, Z. (2007), 'Product family design and platform-based product development: A state-of-the-art review', *Journal of Intelligent Manufacturing* **18**(1), 5–29. doi: 10.1007/s10845-007-0003-2.

Jiao, J. & Tseng, M. M. (2000), 'Understanding product family for mass customization by developing commonality indices', *Journal of Engineering Design* **11**(3), 225–243. doi: 10.1080/095448200750021003.

Jose, A. & Tollenaere, M. (2005), 'Modular and platform methods for product family design: Literature analysis', *Journal of Intelligent Manufacturing* **16**(3), 371–390. doi: 10.1007/s10845-005-7030-7.

Kamal, M. M. (1970), 'Analysis and simulation of vehicle to barrier impact', *SAE Technical Paper* pp. 1498–1503. doi: 10.4271/700414.

Kaul, A. & Rao, V. R. (1995), 'Research for product positioning and design decisions: An integrative review', *International Journal of Research in Marketing* **12**(4), 293–320. doi: 10.1016/0167-8116(94)00018-2.

Kelly, D., Reidsema, C., Bassandeh, A., Pearce, G. & Lee, M. (2011), 'On interpreting load paths and identifying a load bearing topology from finite element analysis', *Finite Elements in Analysis and Design* **47**(8), 867–876. doi: 10.1016/j.finel.2011.03.007.

Kelly, D. W., Hsu, P. & Asudullah, M. (2001), 'Load paths and load flow in finite element analysis', *Engineering Computations* **18**(1/2), 304–313. doi: 10.1108/02644400110365923.

Kelly, D. W., Pearce, G., Ip, M. & Bassandeh, A. (2011), Plotting load paths from vectors of finite element stress results, *in* 'NAFEMS World Congress 2011', Boston, USA.

Kelly, D. W., Reidsema, C. A. & Lee, M. C. W. (2011), 'An algorithm for defining load paths and a load bearing topology in finite element analysis', *Engineering Computations* **28**(2), 196–214. doi: 10.1108/02644401111109231.

Kim, C. H., Mijar, A. R. & Arora, J. S. (2001), 'Development of simplified models for automotive crashworthiness simulation and design using optimization', *Structural Multidisciplinary Optimization* **22**(4), 307–321. doi: 10.1007/PL00013285.

Kota, S., Sethuraman, K. & Miller, R. (2000), 'A metric for evaluating design commonality in product families', *Journal of Mechanical Design* **122**(4), 403. doi: 10.1115/1.1320820.

Kovalyov, M. Y. & Pesch, E. (2010), 'A generic approach to proving NP-hardness of partition type problems', *Discrete Applied Mathematics* **158**(17), 1908–1912. doi: 10.1016/j.dam.2010.08.001.

Krafft, M., Kullgren, A., Tingvall, C., Boström, O. & Fredriksson, R. (2000), 'How crash severity in rear impacts influences short- and long-term consequences to the neck', *Accident Analysis and Prevention* **32**(2), 187–195. doi: 10.1016/S0001-4575(99)00083-4.

Lange, V. A., Fender, J., Song, L. & Duddeck, F. (2018), 'Early phase modeling of frontal impacts for crashworthiness: From lumped mass–spring models to deformation space models', *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* . doi: 10.1177/0954407018814034.

Lechner, R., Hailer, I., Horion, S. & Steffan, H. (2017), Validation of the causality of influencing seat design parameters, identified by using multivariate analysis methods, on the BioRID-II ATD kinematics in low-speed rear-end impacts, *in* '2017 IRCOBI Conference Proceedings', Antwerp, Belgium, pp. 613–622.

Lust, R. (1992), 'Structural optimization with crashworthiness constraints', *Structure Optimization* **4**(2), 85–89. doi: 10.1007/BF01759921.

Mansour, T. (2012), *Combinatorics of set partitions*, Discrete Mathematics and its Applications, Taylor & Francis. ISBN 9781439863336.

Martin, M. V. & Ishii, K. (1996), Design for variety - a methodology for understanding the costs of product proliferation, *in* 'Proceedings of the ASME 1996 Design Engineering Technical Conferences and Computers in Engineering Conference', Irvine, CA, USA.

Martin, M. V. & Ishii, K. (1997), Design for variety: development of complexity indices and design charts, *in* 'Proceedings of DETC '97 1997 ASME Design Engineering Technical Conferences', Sacramento, CA, USA.

Messac, A., Martinez, M. P. & Simpson, T. W. (2002), 'Introduction of a product family penalty function using physical programming', *Journal of Mechanical Design* **124**(2), 164 – 172. doi: 10.1115/1.1467602.

Miao, C., Du, G., Xia, Y. & Wang, D. (2016), 'Genetic algorithm for mixed integer nonlinear bilevel programming and applications in product family design', *Mathematical Problems in Engineering* **2016**(Article ID 1379315). doi: 10.1155/2016/1379315.

Nayak, R. U., Chen, W. & Simpson, T. W. (2002), 'A variation-based method for product family design', *Engineering Optimization* **34**(1), 65–81. doi: 10.1080=03052150290004621.

Newcomb, P. J., Bras, B. & Rosen, D. W. (1998), 'Implications of modularity on product design for the life cycle', *Journal of Mechanical Design* **120**(3), 483–490. doi: 10.1115/1.2829177.

Ni, C. M. & Song, J. (1986), Computer-aided design analysis methods for vehicle structural crashworthiness, *in* 'Proc. of Symposium on Vehicle Crashworthiness Including Impact Biomechanics', ASME Applied Mechanics Division, pp. 125–139.

N.N. (2008), Laboratory test procedure for FMVSS 305, electric powered vehicles: Electrolytespillage and electrical shock protection., Technical Report TP-305-01, U.S. Department of Transportation NHTSA. Retrieved 04-11-2018, from https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/tp-305-01.pdf.

N.N. (2011), RCAR Low-speed structural crash test protocol, Technical report, Research Council for Automobile Repairs. Retrieved from https://www.rcar.org/Papers/Procedures/rcar{_}LowSpeedCrashTest2{_}2.pdf on 2018-11-04.

N.N. (2018*a*), 'Bell number — Wikipedia, the free encyclopedia', https://en.wikipedia.org/w/index.php?title=Bell_number&oldid=860278212. [Online; accessed 13-10-2018].

N.N. (2018*b*), 'Cactus graph — Wikipedia, the free encyclopedia', https://en.wikipedia.org/w/index.php?title=Cactus_graph&oldid=841677886. [Online; accessed 13-10-2018].

N.N. (2018*c*), 'Car platform — Wikipedia, the free encyclopedia', https://en.wikipedia.org/w/index.php?title=Car_platform&oldid=865400619. [Online; accessed 4-11-2018].

N.N. (2018*d*), 'List of GM platforms — Wikipedia, the free encyclopedia', https://en.wikipedia.org/w/index.php?title=List_of_GM_platforms&oldid=839939763. [Online; accessed 4-11-2018].

Nocedal, J. & Wright, S. J. (2006), *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering„ 2 edn, Springer. ISBN 978-0-387-40065-5.

Pai, J.-E. (2014), Evaluation of FMVSS no. 301, fuel system integrity, as upgraded in 2005 to 2009, Technical Report No. DOT HS812 038, National Highway Traffic Safety Administration (NHTSA), Washington DC, USA.

Parsons, G. G. (1990), Motor vehicle fires in traffic crashes and the effects of the fuel system integrity standard, Technical Report No. DOT HS807 675, National Highway Traffic Safety Administration (NHTSA), Washington DC, USA.

Pine II, B. J. (1993), 'Mass customizing products and services', *Planning Review* **21**(4), 6–55. doi: 10.1108/eb054420.

Reus, B. (2016), 'Limits of computation – from a programming perspective'. ISBN 978-3-319-27889-6.

Rosanwo, O., Petz, C., Prohaska, S., Hege, H.-C. & Hotz, I. (2009), Dual streamline seeding, *in* '2009 IEEE Pacific Visualization Symposium', pp. 9–16. doi: 10.1109/PACIFICVIS.2009.4906832.

Seghal, B. & Gorai, P. (2012), 'Platform strategy will shape future of oems- flexibility to drive growth', *Evaluserve, White Paper* (January), 13. Retrieved 04-11-2018 from http://sandhill.com/wp-content/files_mf/evalueservewhitepaperplatformstrategywillshapefutureofoems.pdf.

Siddique, Z. & Rosen, D. W. (2001), 'On combinatorial design spaces for the configuration design of product families', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **15**(2), 91–108. doi: 10.1017/S0890060401152029.

Siddique, Z., Rosen, D. W. & Wang, N. (1998), On the applicability of product variety design concepts to automotive platform commonality, *in* 'Proceedings of the 1996 ASME Design Engineering Technical Conferences', Atlanta, USA.

Simpson, T. W. & D'Souza, B. S. (2004), 'Assessing variable levels of platform commonality within a product family using a multiobjective genetic algorithm', *Concurrent Engineering* **12**(2), 119–129. doi: 10.1177/1063293X04044383.

Simpson, T. W., Siddique, Z. & Jiao, J. (2006), *Product platform and product family design – Methods and applications*, Springer. ISBN 978-0-387-29197-0.

Song, L., Fender, J. & Duddeck, F. (2015), A semi-analytical approach to identify solution spaces for crashworthiness in vehicle architectures, *in* '24th International Technical Conference on the Enhanced Safety of Vehicles', Göteborg, Sweden, pp. 1–12.

Song, L., Pabst, M., Duddeck, F. & Fender, J. (2017), 'A simplified model for barrier–vehicle interaction in a rear crash for early phase development and solution spaces', *International Journal of Crashworthiness* pp. 1–14. doi: 10.1080/13588265.2017.1350091.

Srinivas, N. & Deb, K. (1994), 'Multiobjective optimization using nondominated sorting in genetic algorithms', *Evolutionary Computation* **2**(3), 221–248. doi: 10.1017/CBO9781107415324.004.

Steward, D. V. (1981), 'Design structure system: A method for managing the design of complex systems.', *IEEE Transactions on Engineering Management* .

Tarasewich, P. & Nair, S. K. (2001), 'Designer-moderated product design', *IEEE Transactions on Engineering Management* **48**(2), 175–188. doi: 10.1109/17.922477.

Thevenot, H. J. & Simpson, T. W. (2006), 'Commonality indices for assessing product families', *Simpson T.W., Siddique Z., Jiao J. R. (eds) Product Platform and Product Family Design: Methods and Applications* pp. 107–129. doi: 10.1007/0-387-29197-0_7.

Trella, T. J., Gabler, H. C., Kanianthra, J. N. & Wagner, J. J. (1991), 'Side impact crashworthiness design: Evaluation of padding characteristics through mathematical simulations', *SAE Technical Paper 912900* . doi: 10.4271/912900.

Ulrich, K. (1995), 'The role of product architecture in the manufacturing firm', *Research Policy* **24**(3), 419–440. doi: 10.1016/0048-7333(94)00775-3.

Ulrich, K. T. & Eppinger, S. D. (2011), *Product Design and Development*, 5 edn, McGraw Hill Higher Education.

Venn, J. (1880), 'On the diagrammatic and mechanical representation of propositions and reasonings', *Philosophical Magazine Series 5* **10**(59), 1–18. doi: 10.1080/14786448008626877.

Volz, K. H. (2011), Physikalisch begründete Ersatzmodelle für die Crashoptimierung von Karosseriestrukturen in frühen Projektphasen, PhD thesis, Technische Universität München, Munich, Germany.

Wacker, J. G. & Treleven, M. (1986), 'Component part standardization: An analysis of commonality sources and indices', *Journal of Operations Management* **6**(2), 219–244. doi: 10.1016/0272-6963(86)90026-4.

White, F. M. (2015), *Fluid Mechanics*, 8 edn, McGraw-Hill, New York. ISBN-13 978-0073398273.

Zimmermann, M., Königs, S., Niemeyer, C., Fender, J., Zeherbauer, C., Vitale, R. & Wahle, M. (2017), 'On the design of large systems subject to uncertainty', *Journal of Engineering Design* **28**(4), 233–254. doi: 10.1080/09544828.2017.1303664.

Zimmermann, M. & von Hoessle, J. E. (2013), 'Computing solution spaces for robust design', *International Journal for Numerical Methods in Engineering* **94**(3), 290–307. doi: 10.1002/nme.4450.