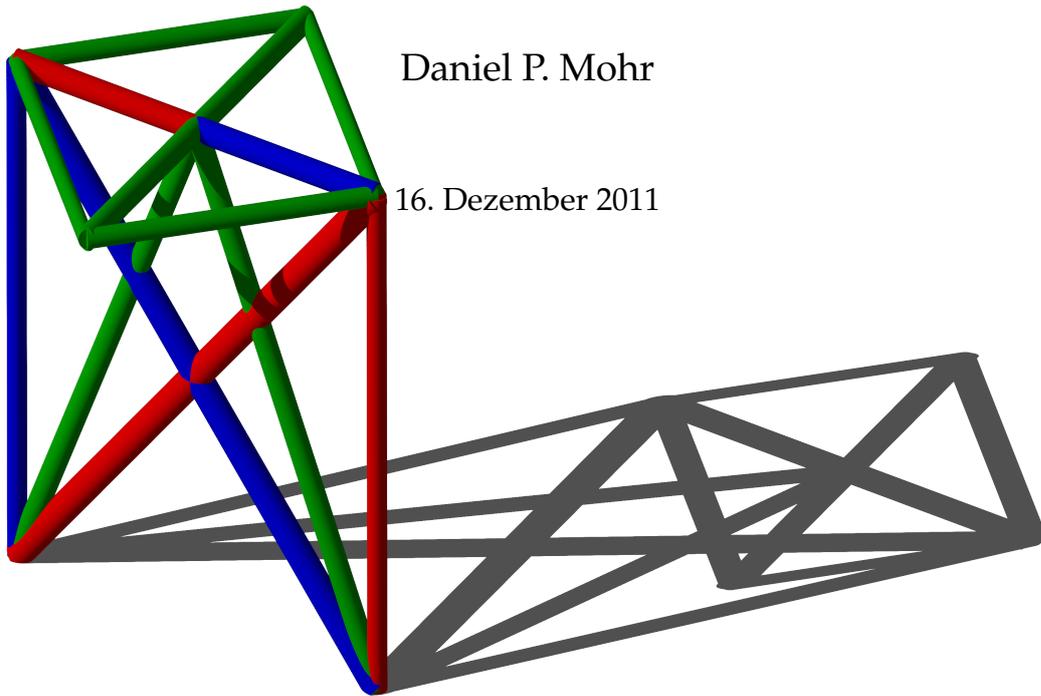


# Redundante Topologieoptimierung

Dissertation

Daniel P. Mohr

16. Dezember 2011



*der Bundeswehr*  
**Universität**  **München**

Institut für Mathematik und Rechneranwendung  
Fakultät für Luft- und Raumfahrttechnik  
Universität der Bundeswehr München



Universität der Bundeswehr München  
Fakultät für Luft- und Raumfahrttechnik  
Institut für Mathematik und Rechneranwendung

## **Redundante Topologieoptimierung**

Dipl.-Math. Daniel P. Mohr

Vollständiger Abdruck der bei der  
Fakultät für Luft- und Raumfahrttechnik  
der Universität der Bundeswehr München  
zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

eingereichten Dissertation

Vorsitzender: Prof. Dr.-Ing. Rapp

1. Berichterstatter: Prof. Dr. sc. math. habil. Joachim Gwinner
2. Berichterstatter: Prof. Dr.-Ing. habil. Fabian Duddeck

Diese Dissertation wurde am 23.09.2011 bei der Universität der Bundeswehr München, 85577 Neubiberg eingereicht und durch die Fakultät für Luft- und Raumfahrttechnik am 26.10.2011 angenommen.

Tag der Prüfung: 07.12.2011



ANGABEN IN DEN PFLICHTEXEMPLAREN  
DER DISSERTATION

UNIVERSITÄT DER BUNDESWEHR MÜNCHEN  
FAKULTÄT FÜR Luft- und Raumfahrttechnik

Thema der Dissertation: Redundante Topologieoptimierung

Verfasser: Daniel Paul Mohr  
(Vorname) (Name)

Promotionsausschuss:

Vorsitzender: Prof. Rapp

1. Berichterstatter: Prof. Gwinner

2. Berichterstatter: Prof. Duddeck

3. Berichterstatter: .....

4. Berichterstatter: .....

Tag der Prüfung: 07.12.2011

Mit der Promotion erlangter akademischer Grad: Doktor-Ingenieurs

(Dr. -Ing...... )

Neubiberg, den 07.12.2011



## Kurzbeschreibung

In dieser Dissertation wird das interdisziplinäre Gebiet der Topologieoptimierung aus der Strukturoptimierung gekreuzt mit dem mathematischen Gebiet der Informations- und Kodierungstheorie, um für RAMS (Reliability, Availability, Maintainability, Safety) eine Lösung zu erhalten, die der Sicherheitstechnik fail-safe genügt. Nach einer neuen, verallgemeinerten Definition der Redundanz wird diese verwendet um redundante Strukturen zu berechnen.

In dieser Arbeit werden zwei grundlegend verschiedene Methoden der Topologieoptimierungen benutzt. Im einen Fall wird von einer kontinuierlichen Struktur ausgegangen. Dazu wird der Bauraum vollständig in einzelne Elemente aufgeteilt. Da nicht die Topologieoptimierung an sich Gegenstand dieser Arbeit ist, sondern das Finden redundanter Strukturen, wurde hier – um größtmögliche Vergleichbarkeit zu gewährleisten – der erfolgreichste und verbreitetste Ansatz SIMP gewählt. Im anderen Fall wird der Bauraum durch diskrete Elemente abgedeckt. Um die notwendige Rechenzeit gering zu halten, wurden die einfachsten Elemente hierfür gewählt. Der Bauraum wurde also mit Stäben überdeckt und aus diesen werden die sinnvollsten ausgewählt. Dies entspricht der ground structure method. Zur Bewertung wurde die matrix force method verwendet.

Aktuelles Forschungsinteresse ist die Berücksichtigung von möglichen Schäden – vielerorts wird dies durch die Vermeidung von Schäden durchgeführt. In dieser Arbeit soll nicht vermieden werden, daß ein Schaden eintritt. Statt dessen soll die Lösung auch nach Eintritt eines Schadens noch ihre Aufgabe erfüllen. Es wird also eine redundante Struktur gesucht.

Eine neue Definition der Redundanz in der vorliegenden Arbeit ermöglicht eine Anwendung dieses Begriffs in der Topologieoptimierung um letztlich relevante Schäden an der optimierten Struktur zuzulassen und somit redundante Strukturen numerisch zu berechnen.



# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>I</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Schadenstoleranz</b>	<b>7</b>
<b>3 Redundanz</b>	<b>11</b>
3.0.1 Beispiel: Festplatten . . . . .	12
3.1 Def. Redundanz mittels Automaten . . . . .	15
3.1.1 Beispiel: RAID 1 aus 2 Festplatten . . . . .	16
3.1.2 Beispiel: das Bremsen eines Fahrrades . . . . .	18
3.2 Def. Redundanz mittels Funktionen . . . . .	21
3.2.1 Beispiel: RAID 1 aus 2 Festplatten . . . . .	22
3.2.2 Beispiel: RAID 5 aus 3 Festplatten . . . . .	23
3.2.3 Beispiel: Fahrrad soll bremsen . . . . .	26
3.3 Problem der Redundanz . . . . .	27
<b>4 Schadenstoleranz in der Topologieoptimierung</b>	<b>29</b>
4.1 Schadenstoleranz durch Redundanz im SIMP-Ansatz . . . . .	32
4.1.1 Idee . . . . .	32
4.1.2 Redundanz von $1 - \frac{1}{n}$ . . . . .	34
4.1.3 Redundanz von $\frac{1}{n}$ . . . . .	47
4.1.4 Ergebnisse . . . . .	61
4.2 Schadenstoleranz durch Redundanz im Fachwerk . . . . .	62
4.2.1 Idee . . . . .	62
4.2.2 Fachwerk mit Redundanz $1 - \frac{1}{n}$ als Lineares Programm . . . . .	65
4.2.3 Fachwerk mit Redundanz $\frac{1}{n}$ . . . . .	69
4.2.4 Greedy-Algorithmus . . . . .	73

## INHALTSVERZEICHNIS

---

4.2.5	Ergebnisse . . . . .	78
<b>5</b>	<b>Schlussbemerkungen</b>	<b>85</b>
<b>A</b>	<b>Endlicher Automat</b>	<b>89</b>
<b>B</b>	<b>Informations- und Kodierungstheorie</b>	<b>91</b>
<b>C</b>	<b>Topologieoptimierung</b>	<b>97</b>
C.1	SIMP . . . . .	98
C.2	Nummerierung der Elemente beim SIMP-Ansatz . . . . .	106
C.2.1	2 D . . . . .	106
C.2.2	3 D . . . . .	109
C.3	Topologieoptimierung beim Fachwerk . . . . .	114
C.3.1	Topologieoptimierung beim Fachwerk analog zum SIMP-Ansatz . . . . .	114
C.3.2	Topologieoptimierung beim Fachwerk als Volu- mensminimierung (plastic design) . . . . .	119
C.3.3	Geometriematrix . . . . .	120
C.4	Vernetzung des Bauraums durch Stäbe beim Fachwerk . . . . .	123
C.4.1	2 D - jeder mit jedem . . . . .	123
C.4.2	3 D - jeder mit jedem . . . . .	126
C.5	Entwurfsmethode von Michell . . . . .	133
<b>D</b>	<b>Stabwerk-Interpretation</b>	<b>135</b>
<b>E</b>	<b>Optimierung unter binärer Ganzzahligkeitsbedingung</b>	<b>137</b>
<b>F</b>	<b>Vergleich der Redundanz <math>\frac{1}{n}</math> und <math>1 - \frac{1}{n}</math> beim Stabwerk</b>	<b>141</b>
F.1	2-dim. Strukturen . . . . .	141
F.2	3-dim. Strukturen . . . . .	147
F.2.1	Redundanz $\frac{1}{n}$ vgl. $1 - \frac{1}{n}$ : kran1x1 lastfall1 . . . . .	147
F.2.2	Redundanz $\frac{1}{n}$ vgl. $1 - \frac{1}{n}$ : kran1x1 lastfall2 . . . . .	148

## INHALTSVERZEICHNIS

---

F.2.3	Redundanz $\frac{1}{n}$ vgl. $1 - \frac{1}{n}$ : kran1x1 lastfall3 . . . . .	149
<b>G</b>	<b>Trivialitäten</b>	<b>151</b>
	<b>Literatur</b>	<b>160</b>
	<b>Tabellenverzeichnis</b>	<b>161</b>
	<b>Abbildungsverzeichnis</b>	<b>165</b>

## INHALTSVERZEICHNIS

---

---

# 1 Einleitung

In dieser Dissertation wird das interdisziplinäre Gebiet der Topologieoptimierung aus der Strukturoptimierung gekreuzt mit dem mathematischen Gebiet der Informations- und Kodierungstheorie, um für RAMS („Reliability, Availability, Maintainability, Safety“ nach Norm EN 50126) eine Lösung zu erhalten, die der Sicherheitstechnik fail-safe<sup>1</sup> genügt. Die beiden Begriffe der Redundanz aus der Zuverlässigkeitstheorie und aus der Informations- und Kodierungstheorie werden durch eine neue Definition verallgemeinert, um gegenüber Schäden tolerante Strukturen – also redundante Strukturen – zu berechnen.

Strukturoptimierung ist sicherlich spätestens seit der ersten durchschlagenden Arbeit [Mic04] von 1904 ein hochinteressantes und multidisziplinäres Forschungsgebiet. Prinzipiell wird zwischen

- Materialoptimierung (optimization of material)
- Gestaltoptimierung (shape optimization)
- Dimensionierung (sizing optimization)
- Topologieoptimierung (topology optimization)

unterschieden ([MG06]).

Materialoptimierung beschreibt ‚einfach‘ die Suche nach einem optimalen Material. Bei komplexeren Strukturen können auch einzelne Teile unterschiedliche Materialien aufweisen (z. B. [Che11]). Somit ist eine diskrete Optimierung nötig – hierbei liegt also keine kontinuierliche Zielfunktion zu Grunde, wodurch die Komplexität der Berechnung NP-vollständig<sup>2</sup> wird. Weiterhin fällt im weitesten Sinne auch die Optimierung der Faserwinkel bei Verbundwerkstoffen in dieses Gebiet. Der Ansatz *free material optimization* in [ZKB97] [BtKNZ97] sucht das optimale Material durch

---

<sup>1</sup>Ein System, welches nach fail-safe ausgelegt ist, darf bei möglichen Schäden nicht komplett versagen, bevor es außer Betrieb genommen werden kann. Weiterhin muss der Schaden hierfür erkennbar und bewertbar sein.

<sup>2</sup>Die Komplexitätsklasse P umfasst alle Probleme mit polynomiellen Lösungsverfahren. In der Klasse NP liegen alle Probleme, deren Lösung sich in polynomieller Zeit überprüfen lassen. Obwohl Lösungen von NP-vollständigen Problemen in polynomieller Zeit überprüft werden können, ist für NP-vollständige Probleme bisher kein polynomieller Lösungsalgorithmus bekannt. Wenn für ein NP-vollständiges Problem ein polynomieller Lösungsalgorithmus gefunden würde, so wären die Komplexitätsklassen P und NP identisch.

die Wahl des besten Elastizitätstensors bei gleichzeitig optimaler Materialverteilung. Dabei wird der Elastizitätstensor kontinuierlich variiert und entspricht damit nicht zwangsweise einem real existierenden Material.

Gestaltoptimierung [AH01] sucht nach der optimalen Oberfläche einer Struktur. Häufig wird in jeder Iteration der Optimierung mittels Finite-Elemente-Methode eine Approximation der Von-Mises-Vergleichsspannung berechnet und dadurch ein Update der Oberfläche ermöglicht (z. B. [KKÖ10]).

Dimensionierung ergibt die notwendigen Abmessungen der gewünschten Einzelbauteile. Man geht hierbei also von einer bereits strukturellen Zusammensetzung aus klar definierten Bauteilen aus. Für jedes Bauteil sind nur sehr wenige Freiheitsgrade gesucht – z. B. die Durchmesser von Balken einer gewissen Ausführung.

Topologieoptimierung ([BS04], [EO01], [CGM73]) stellt die größtmögliche Freiheit zur Verfügung. Man sollte sich klar sein, dass eine Kaffeetasse (wegen ihres Henkels) die gleiche Topologie ([CKR06]) aufweist wie ein Donut (Torus). In diesem Sinn liegen diese beiden grundsätzlich verschiedenen Strukturen in der selben Äquivalenzklasse einer Topologie. Bei der Suche der optimalen Topologie muss daher eine Bewertung zu Grunde gelegt werden, die das optimale Element aus allen möglichen der selben Äquivalenzklasse bewertet. Bei der Suche nach der besten Topologie für eine mechanische Struktur sind die typischen Bewertungskriterien das Volumen oder die Steifigkeit (bzw. die umgekehrt proportionale Nachgiebigkeit (Compliance)). Um diese Kriterien bewerten zu können, muss allerdings eine gesuchte Topologie (bestehend aus vielen möglichen Strukturen, die in einer Äquivalenzklasse zusammengefasst sind) auch in der besten Form sein – auch bei gleichem Material verhalten sich die Struktur einer Kaffeetasse oder eines Donuts komplett unterschiedlich. Weiterhin bedeutet die beste Form auch eine geeignete Dimensionierung der Einzelkomponenten. Die Frage nach einem besten Material hierfür stellt sich ebenfalls, wird allerdings bisher in der Literatur größtenteils getrennt behandelt. In diesem Sinn umfasst die Topologieoptimierung die anderen grundlegenden Strukturoptimierungsansätze und stellt daher die Königsdisziplin dar.

In dieser Arbeit werden zwei grundlegend verschiedene Methoden der Topologieoptimierungen verwendet. Im ersten Ansatz wird von einer kontinuierlichen Struktur (Kapitel 4.1, Anhang C.1) ausgegangen. Dazu wird der Bauraum vollständig in einzelne Elemente aufgeteilt. Da nicht die Topologieoptimierung an sich Gegenstand dieser Arbeit ist, sondern

---

das Finden redundanter Strukturen unter Verwendung dieser Methoden, wurde hier – um größtmögliche Vergleichbarkeit zu gewährleisten – der erfolgreichste und verbreitetste Ansatz SIMP [BS04], basierend auf der Homogenisierungsmethode [BK88], gewählt; aber prinzipiell wären auch andere Ansätze, wie beispielsweise die *bubble method* [EKS94], anwendbar. Im zweiten Ansatz wird der Bauraum durch diskrete Elemente (Kapitel 4.2, Anhang C.3.2) abgedeckt. Um die notwendige Rechenzeit gering zu halten, wurden die einfachsten Elemente hierfür gewählt. Der Bauraum wurde also mit Stäben überdeckt und aus diesen werden die sinnvollsten ausgewählt. Dies entspricht der Grundstrukturmethode (ground structure method) [DGG64]. Zur Bewertung wurde die Matrix-Kraft-Methode (matrix force method [Prz68]) verwendet – vgl. Anhang C.3.3, [MG06], [Mar03] und [OAH96].

Auf aktuelles Forschungsinteresse stößt die Berücksichtigung von möglichen Schäden. Das verbreitete Vorgehen hierbei ist die Berücksichtigung von Unsicherheiten ([BS07], [Mar03], [CD08]). In der Topologieoptimierung finden sich die Schlagwörter *reliability-based design* [MF03] und *robust design* [DK04], welche bis heute ein intensiv bearbeitetes Forschungsgebiet darstellen (z. B. [BS07], [CD08], [CCL10], [MSMK11]). All diesen Ansätzen liegt zu Grunde, dass man versucht, Schäden zu vermeiden. Dies entspricht letztlich dem klassischen safe-life<sup>3</sup> Ansatz. Prinzipbedingt ist dies ein sehr teurer Ansatz und ist nur gerechtfertigt, da eine Optimierung stattfindet.

In dieser Arbeit soll zugelassen werden, dass ein Schaden eintritt. Die Lösung soll jedoch auch nach Eintritt eines Schadens noch ihre (evtl. reduzierte) Aufgabe erfüllen. Vergleichbares findet sich in der Strukturoptimierung bei Ermüdungsschäden, welche durch Lebensdaueranalyse Berücksichtigung finden können (z. B. [Ozt11]). Allerdings wird hierbei wiederum versucht, die Entstehung dieser Schäden zu verhindern und ein Schaden wird abermals nicht zugelassen.

Um einen strukturellen Ausfall (Schaden) zulassen zu können, muss unter normalen Bedingungen (kein Schaden) etwas Überflüssiges vorhanden sein. Diese einfache Überlegung führt unmittelbar zu Redundanz. Im technischen Sinn ([GF07], [BLZ09], [BMPR11]) wird hierbei meist das mehrfache Vorhandensein einer Funktionseinheit als Redundanz verstanden. Dahingegen wird in der Informations- und Kodierungstheorie ([Sha48], [HQ95], [Rez61], [Tod84]) feiner unterschieden und auch ein Bruchteil

---

<sup>3</sup>Ein System, welches nach safe-life ausgelegt wird, darf unter geforderten Bedingungen nicht versagen (vgl. Kapitel 2).

mehr als nötig wird als Redundanz verstanden. Im technischen Sinn ist es schwierig, die Redundanz zu messen. In [HY02] werden verschiedene Möglichkeiten (u. a. auch die auf Shannons Entropie-Begriff aus der Informations- und Kodierungstheorie basierende von [Ž00]) angegeben, um die Redundanz von Systemen zu vergleichen. In [KBH11] wird zwar auf diese Möglichkeiten verwiesen, aber dennoch ist es dort nötig, eine weitere spezielle Methode der Bewertung einzuführen, um damit die Redundanz von Stabwerken vergleichen zu können. Dagegen ist in der Informations- und Kodierungstheorie die Redundanz ([Sha48]) eindeutig durch ein Maß bewertbar.

Die meisten dieser Bewertungen und Maße der Redundanz basieren auf dem Spezialfall, dass Wahrscheinlichkeiten von entsprechenden Ereignissen bekannt sind. In der vorliegenden Arbeit wird der allgemeinere Fall, in dem Wahrscheinlichkeiten nicht berücksichtigt werden, zu Grunde gelegt. Dies führt hier zu einer neuen und allgemeinen Definition der Redundanz. Erst diese allgemeine Definition ermöglicht in der Topologieoptimierung letztlich relevante Schäden an der optimierten Struktur zuzulassen und somit redundante Strukturen numerisch zu berechnen.

Da es praktisch keine Vergleichsmöglichkeiten bzgl. der Berücksichtigung von tatsächlichen Schäden bei der Topologieoptimierung in der Literatur gibt, musste auf Vergleiche bei der Ergebnisinterpretation weitestgehend verzichtet werden.

Die vorliegende Arbeit ist wie folgt strukturiert:

In Kapitel 2 wird die grundsätzliche Problemstellung erläutert und somit die Arbeit motiviert.

Im Kapitel 3 wird eine neue Definition des Begriffs der Redundanz gegeben. Dieser verbindet den unpräzisen Begriff der Redundanz aus der Ingenieurswelt mit dem sehr speziellen, aber präzisen Begriff der Redundanz in der Informations- und Kodierungstheorie. Diese neue Definition umfasst also diese beiden bekannten Begriffe. Dazu wird erst sehr allgemein mittels eines deterministischen endlichen Automaten erklärt, was Redundanz ist, um dann in Kapitel 3.2 eine präzise Definition zu geben, die auch ein Gütekriterium für die Redundanz liefert; auch dies ist konsistent mit dem Bekannten aus der Informations- und Kodierungstheorie.

Das Kapitel 4 zeigt mit neuen Modellierungen die Anwendung der Redundanz für mechanische Strukturen. Bereits im Vorentwurf kann nun bei der Topologieoptimierung der Wunsch einer Ausfallsicherheit beachtet werden. Die Güte der Ausfallsicherheit kann hierbei frei gewählt wer-

---

den. Dieses bietet die Grundlage um im weiteren Entwicklungsprozess fail-safe oder sogar damage-tolerance<sup>4</sup> zu ermöglichen.

Im Anhang finden sich sowohl die als bekannt vorausgesetzten Grundlagen, als auch notwendige Konzepte zur Umsetzung der numerischen Vorgehensweisen (vgl. Anhänge C.2 und C.4) und weitergehende Ausführungen (vgl. Anhänge C.3.3, D und F).

---

<sup>4</sup>Dieser Begriff wird in Kapitel 2 erklärt.

## 1 Einleitung

---

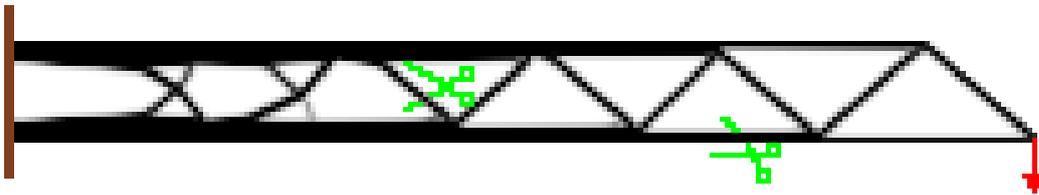


Abbildung 2.1: In diesem Bild wird eine mit dem SIMP-Ansatz (vgl. Anhang C.1) berechnete 2-dimensionale Struktur dargestellt. Jedes erkennbare Pixel repräsentiert ein finites Element. Die linke Seite ist gelagert und rechts unten wirkt eine Kraft nach unten. Was passiert, wenn die Scheren die Struktur an der einen oder der anderen Stelle zerschneiden?

## 2 Schadenstoleranz

Schadenstoleranz ist die Toleranz gegenüber Schäden.

In Abb. 2.1 ist das typische Problem von gewichts- oder steifigkeitsoptimalen Strukturen dargestellt. Ein Zerstören der Struktur an gewissen Stellen (durch die Scheren symbolisiert) führt zum Versagen der Struktur unter der Belastung (Pfeil rechts im Bild), für die die Struktur ausgelegt ist. Die linke Schere stellt noch einen verhältnismäßig geringen Schaden dar: Für die Belastung, für die die Struktur ausgelegt ist, versagt die Struktur; aber geringere Belastungen vermag sie noch zu bewältigen. Obwohl auch die rechte Schere lediglich einen ‚Stab‘ entfernt, stellt dies einen erheblich größeren Schaden dar: Die Struktur überträgt die Kraft auf das Lager (Wand auf der linken Seite) nur noch an einem ‚Punkt‘. Dies führt unweigerlich zu einem Totalversagen der Struktur. Derartige optimale Strukturen tolerieren also überhaupt keinen Schaden. Wie kann eine Struktur einen Schaden tolerieren? Wie findet man solche Strukturen?

In [GF07, S. F 16 ff.; S. Q 104 ff.] werden für allgemeine technische Systeme verschiedene Sicherheitstechniken<sup>5</sup> genannt, die auch allgemein anerkannte Herangehensweisen sind:

- safe-life  
Das System wird hierbei für eine Lebensdauer ausgelegt und innerhalb dieser Zeit soll unter geforderten Bedingungen kein Versagen des Systems vorkommen.

<sup>5</sup>Eine Sicherheitstechnik ist eine Herangehensweise. Manchmal wird in der Literatur statt Sicherheitstechnik auch der Begriff Designphilosophie verwendet.

## 2 Schadenstoleranz

---

- fail-safe  
Hierbei dürfen durchaus Schäden vorkommen. Diese dürfen aber nicht zum kompletten Versagen des Systems führen, bevor dieses außer Betrieb genommen werden kann. Weiterhin muss der Schaden hierfür erkennbar und bewertbar sein.
- damage-tolerance  
Dies wird im Besonderen in der Luftfahrt gefordert. Dabei wird nun der gleiche Anspruch wie bei fail-safe gestellt und weiterhin werden nicht erkennbare Schäden zugelassen, die wiederum innerhalb eines Wartungsintervalls nicht zu einem kritischen Versagen führen.

Zum besseren Verständnis seien hier noch ein paar Beispiele genannt:

- safe-life: Beim Ausgehen nimmt man meist genügend Geld mit, um hinterher in der Kneipe die Zeche auch garantiert begleichen zu können. Dabei wird „aus Sicherheit“ mehr als notwendig viel Geld mitgenommen.
- safe-life: Der Rosinenbomber Douglas DC-3 wurde nach safe-life ausgelegt. [GF07, S. Q 106 ff.]
- fail-safe: Ein Zugführer überfährt ein Haltsignal und der Zug wird daraufhin automatisch angehalten. Trotz des Fehlers (Schaden) passiert kein Unfall.
- fail-safe: Lokführer in modernen Zügen müssen durch unregelmäßige Betätigung eines Schalters (Knopf oder Pedal) nach automatisierter Aufforderung ihre Funktionsfähigkeit nachweisen (Totmanneinrichtung); andernfalls hält der Zug automatisch an. Also selbst bei einem plötzlichen Versterben des Zugführers werden die Passagiere nicht gefährdet.
- fail-safe: Unter dem Gesichtspunkt fail-safe fliegt die Douglas DC-3 heute noch. [GF07, S. Q 106 ff.]

Nach [VDI07, S. 5] versteht man unter Funktionszuverlässigkeit die Fähigkeit einer Betrachtungseinheit, eine geforderte Funktionalität unter gegebenen Bedingungen für ein gegebenes Zeitintervall zu erfüllen. Dies entspricht dem obigen Begriff safe-life.

---

Wenn man in der Praxis ein Produkt (Betrachtungseinheit) kauft, so bekommt man vom Hersteller eine Zusicherung (Gewährleistung) auf die Funktionsfähigkeit (Produkt macht was es soll.) unter gewissen Einsatzbedingungen (Bedingungen wie „über 10 °C“ und „unter 40 °C“) für einen gewissen Zeitraum (Zeitintervall wie z. B. 2 Jahre).

Die Funktionszuverlässigkeit sowie safe-life bezieht sich nicht auf Schäden, sondern fordert lediglich die Funktionsfähigkeit in einem gewissen Zeitintervall – unabhängig davon, ob ein Schaden eintritt oder nicht.

In [Rüe94, S. 38 ff.] wird Zuverlässigkeit eines Gerätes definiert als die Wahrscheinlichkeit, dass es während einer bestimmten Zeitspanne korrekt funktioniert. Dabei wird lediglich zwischen funktionstüchtig und nicht funktionstüchtig unterschieden. Wieder wird nicht vom Schaden gesprochen. Allerdings wird das zu untersuchende System als in einzelne Komponenten zerlegt betrachtet und jede dieser kann nun funktionieren oder eben nicht. Mittels Ausfallwahrscheinlichkeiten können nun von einfachen Serien- bzw. Parallelsystemen Zuverlässigkeiten des Gesamtgerätes berechnet werden. Durch die vereinfachende Annahme der Unabhängigkeit der Einzelkomponenten können dann diese bedingten Ausfallwahrscheinlichkeiten berechnet werden. Neuerdings gibt es auch Bestrebungen, abhängige Komponenten zu betrachten ([BMPR11]).

Man kann also bei der Betrachtungsweise nach [Rüe94] zumindest zu dem Schluss kommen, dass Parallelsysteme tolerant gegenüber dem Ausfall eines Teilsystems sind – dies ist also eine Toleranz gegenüber gewisser Schäden und erfüllt damit die Forderungen bei fail-safe.

Da Gegenstand dieser Arbeit mechanische Strukturen sind, möchte ich nun die obigen Sicherheitstechniken nochmals unter diesem speziellen Blickwinkel beleuchten:

Bei safe-life muss die mechanische Struktur für die Lebensdauer allen Belastungen trotzen. Dies erfordert offensichtlich eine meist überdimensionierte Struktur. In [GF07, S. Q 106 ff.] wird beispielsweise die Douglas DC-3 genannt, die nach safe-life ausgelegt wurde, aber auf Grund von größerem Wissen, welches nach der Fertigstellung erworben wurde, unter der Betrachtungsweise fail-safe über das ursprüngliche Zeitfenster hinaus weiterbetrieben wird.

Bei damage-tolerance muss nun für jeden erdenklichen Schaden an der mechanischen Struktur eine Lebensdaueranalyse für mindestens ein Wartungsintervall durchgeführt werden. Damage-tolerance ist nicht Gegenstand dieser Arbeit.

In dieser Arbeit soll aus dem Nichts mittels Topologieoptimierung eine Struktur erstellt werden, die gewisse Schäden tolerieren kann. Auf Grund des angedeuteten Aufwands bei damage-tolerance und dem enormen Rechenaufwand bereits bei der klassischen Topologieoptimierung [BS04] bleibt nichts anderes übrig, als sich auf fail-safe zurückzuziehen.

Weiterhin vertrete ich die Meinung, dass die Topologieoptimierung zur Findung einer mechanischen Struktur ohnehin nur einen Vorentwurf darstellt, der meist unter weiteren Randbedingungen weiter verbessert und verarbeitet werden muss. In diesem Zuge kann unter Umständen damage-tolerance nachgewiesen werden – hierzu ist es allerdings sinnvoll, bereits den Vorentwurf mittels fail-safe zu gestalten.

In [Haf83] versteht Haftka unter ‚damage tolerant‘ die Toleranz gegenüber ‚global damage‘; dies wiederum definiert er als Zerstörung einer wesentlichen strukturellen Komponente im Gegensatz zu einem lokalen Schaden wie ein Riss.

Schadenstoleranz ist also die Fähigkeit einer Betrachtungseinheit, auch unter gewissen Schadensfällen ihre (evtl. eingeschränkte) Funktionalität zu erfüllen. Hierbei wird Schadenstoleranz als eigenständiger neuer Begriff verstanden und somit unterschieden zu damage-tolerance.

Bei einer mechanischen Struktur kann man sicherlich bei so etwas wie einem Riss von einem Schaden reden, aber auch der Ausfall einer wesentlichen strukturellen Komponente ist nichts anderes als ein Schaden.

Ein **relevanter Schaden** einer Betrachtungseinheit ist der Ausfall eines wesentlichen Teils der Betrachtungseinheit.

Damit eine Betrachtungseinheit auch funktionieren kann, wenn ein wesentlicher Teil ausfällt, muss dieser Teil durch den Rest ersetzt werden können – zumindest insoweit, damit die gewünschte Funktionalität erfüllt wird. Es existiert also eine gewisse Redundanz innerhalb der Betrachtungseinheit bezüglich einer geforderten Funktionalität.

Immer wenn ein Teil einer Betrachtungseinheit ausfallen kann, ohne ein Versagen zu Verursachen, ist dieser Teil für die reine Funktionalität überflüssig. Bei welchen Teilen man von Redundanz sprechen möchte und welche Teile lediglich etwas Überflüssiges darstellen ist philosophischer Natur. Hier in dieser Arbeit wird erst dann von Redundanz gesprochen, sobald ein Teil, der ausfallen darf, einen wesentlichen Teil darstellt.

Nur durch Redundanz lässt sich Schadenstoleranz gegen den Ausfall einer wesentlichen strukturellen Komponente erreichen.

---

## 3 Redundanz

Unter Redundanz versteht man im Allgemeinen etwas Überflüssiges. Dies ist aber nicht notwendiger Weise schlecht, sondern erfüllt sogar einen Zweck – nämlich den Zweck der Ausfallsicherheit wie folgende Beispiele verdeutlichen:

- Serversysteme werden häufig mit 2 Netzteilen ausgestattet. Dabei ist offensichtlich 1 Netzteil überflüssig, es ermöglicht aber eine Ausfallsicherheit gegen Versagen eines Netzteils.
- Kernkraftwerke verfügen ebenfalls über redundante Systeme, um beim Ausfall (z. B. primäres Kühlsystem inkl. Energieverwertung) weiterhin die Sicherheit (z. B. durch Notkühlung) gewährleisten zu können.
- (Kraft-)Fahrzeuge besitzen unabhängige Bremssysteme, um Sicherheit auch beim Versagen eines Bremssystems zu gewährleisten.
- RAID<sup>6</sup>-Systeme aus Festplatten dienen ggf.<sup>7</sup> auch dazu die Ausfallsicherheit zu erhöhen:
  - RAID 1: Die Daten werden auf einer Festplatte gespeichert und auf  $n - 1$  weiteren gespiegelt. Die Daten sind also  $n$ -mal vorhanden und es können  $n - 1$  Festplatten versagen – ohne Datenverlust zu erleiden.
  - RAID 5:  $n$  Festplatten werden zu einem Verbund zusammengeschlossen. Die verwendbare Datenkapazität beträgt  $n - 1$  Festplatten. Beim Versagen von maximal einer Festplatte bleiben alle Daten erhalten.

Das letzte Beispiel macht deutlich, dass Redundanz keineswegs immer eine Verdoppelung oder Vervielfachung des Verwendeten sein muss. Zwar werden beim RAID 1 statt einer  $n$  Festplatten benötigt, allerdings beim RAID 5 werden statt  $n - 1$  Festplatten nur eine weitere benötigt. Offensichtlich erhält man dabei nicht die gleiche Güte der Redundanz.

---

<sup>6</sup>RAID = Redundant Array of Independent Disks

<sup>7</sup>RAID 0 dient lediglich dem Beschleunigen des Datenzugriffs und erhöht sogar die Ausfallwahrscheinlichkeit.

Dies macht es naheliegend, vom Grad der Redundanz zu reden. Während beim RAID 1 der Grad der Redundanz  $\frac{n-1}{n} \in \mathbb{Q}$  ist, ist der Grad der Redundanz beim RAID 5 nur  $\frac{1}{n} \in \mathbb{Q}$  (vgl. [PGK87], Kapitel 3.2).

Offensichtlich müssen wir unterscheiden worauf sich die Ausfallsicherheit bezieht. Während in der Technik im allgemeinen die Redundanz das mehrfache Vorhandensein vergleichbarer Ressourcen bzgl. einer Funktionalität meint, stellt in der Informations- oder Kodierungstheorie (vgl. [HQ95], [Rez61]) die mehrfach vorhandene Information die Redundanz dar.

Eine vorhandene Redundanz ermöglicht also das Lösen einer Aufgabenstellung durch ein System im Schadensfall. Und der Grad der Redundanz misst somit das Verhalten eines Systems bzgl. einer Aufgabenstellung unter Schadensfällen.

Redundanz kann durch mehrfache Auslegung zu Mehrkosten führen – etwas Überflüssiges hat eben auch Nachteile. Aber auch die Schadenswahrscheinlichkeit kann sich erhöhen, wie wir bei den Beispielen in Kapitel 3.0.1 sehen werden.

Die Schadenswahrscheinlichkeit einer Betrachtungseinheit wird häufig angegeben mit MTBF (Mean Time Between Failures), also der Erwartungswert der Betriebsdauer zwischen 2 Ausfällen. Damit ergibt sich bei einer üblicherweise angenommenen Exponentialverteilung die Wahrscheinlichkeit  $p$ , dass ein Schaden innerhalb der Zeit  $T$  eintritt, zu:

$$p = 1 - e^{-\frac{T}{\text{MTBF}}}$$

Als Beispiel sei hier die typische MTBF bei Festplatten von  $1,2 \cdot 10^6$  Stunden genannt. Dies ergibt dann innerhalb der vorgesehenen Laufzeit von 5 Jahren (typischer Gewährleistungszeitraum des Herstellers) eine Ausfallwahrscheinlichkeit von ungefähr 3,6 %.

#### 3.0.1 Beispiel: Festplatten

Im folgenden werden Eigenschaften von Festplattenverbänden genauer ausgeführt, um die grundlegenden Eigenschaften von RAID aufzuzeigen und damit den Sinn von Redundanz aufzuzeigen.

Beispielsweise RAID 5 aus  $n$  Festplatten mit der Ausfallwahrscheinlichkeit  $p \ll 1$  einer einzelnen Platte ist bezüglich der Datenkapazität ver-

---

gleichbar mit  $n - 1$  Festplatten, die alle gemeinsam genutzt werden. Damit ergibt sich nun die Wahrscheinlichkeit, dass das RAID 5 funktioniert zu:

$$\begin{aligned}
\binom{n}{n-1}(1-p)^{n-1}p + \binom{n}{n}(1-p)^n &= n(1-p)^{n-1}p + (1-p)^n \\
&= (1-p)^{n-1}(1 + (n-1)p) \\
&\approx (1 - (n-1)p)(1 + (n-1)p) \\
&= 1 - (n-1)^2p^2
\end{aligned}$$

Die Wahrscheinlichkeit, dass ein JBOD<sup>8</sup> aus  $n - 1$  Festplatten funktioniert, liegt dagegen bei:

$$(1-p)^{n-1} \approx 1 - (n-1)p$$

Somit ist also die Wahrscheinlichkeit, dass das RAID 5 funktioniert, höher. Allerdings ergibt sich die Wahrscheinlichkeit, dass beim RAID 5 ein Schaden vorliegt, zu:

$$\begin{aligned}
\sum_{i=1}^n \binom{n}{i}(1-p)^{n-i}p^i &= 1 - \binom{n}{0}(1-p)^n \\
&\approx 1 - (1 - np) \\
&= np
\end{aligned}$$

Während sich die Wahrscheinlichkeit, dass das JBOD-System aus  $n - 1$  Festplatten einen Schaden hat, nur zu folgendem ergibt:

$$\begin{aligned}
\sum_{i=1}^{n-1} \binom{n-1}{i}(1-p)^{n-1-i}p^i &= 1 - \binom{n-1}{0}(1-p)^{n-1} \\
&\approx 1 - (1 - (n-1)p) \\
&= (n-1)p
\end{aligned}$$

In Tabelle 3.1 sind die verschiedenen Wahrscheinlichkeiten für verschiedene Verwendungsarten von Festplatten nochmals übersichtlich zusammengefasst.

---

<sup>8</sup>JBOD = Just a Bunch Of Disks

### 3 Redundanz

Verwendungsart	RAID 1	RAID 5	JBOD	
Plattenzahl	$n$	$n$	$n - 1$	1
Kapazität	1	$n - 1$	$n - 1$	1
Maximalschaden	$n - 1$	1	0	0
Funktionswahrscheinlichkeit	$1 - p^n$	$1 - (n - 1)^2 p^2$	$(1 - p)^{n-1} \approx 1 - (n - 1)p$	$1 - p$
Schadenswahrscheinlichkeit	$np$	$np$	$(n - 1)p$	$p$

Tabelle 3.1: Vergleich der Funktions- und Ausfallwahrscheinlichkeiten verschiedener Möglichkeiten, die Datenkapazität von Festplatten zu nutzen. Redundanz reduziert dabei nicht die Schadenswahrscheinlichkeit, sondern erhöht nur die Funktionswahrscheinlichkeit.

Die höhere Wahrscheinlichkeit der Funktionsfähigkeit beim RAID 5 aus  $n$  Festplatten gegenüber dem System mit der selben Kapazität aus  $n - 1$  gleichzeitig genutzten Festplatten wird also mit einer höheren Schadenswahrscheinlichkeit erkaufte – allerdings sind auf Grund der Redundanz beim RAID 5 nicht alle Schäden Totalschäden, sondern es existieren akzeptable Schäden, die die Funktionsfähigkeit<sup>9</sup> eben nicht beeinträchtigen. In der Praxis können nun derartige akzeptable Schäden repariert werden bevor ein Totalschaden entsteht.

Wenn aber die Ausfallwahrscheinlichkeit (fast) ausschließlich vom Verschleiß abhängt und dieser auf Grund einer mehrfachen Auslegung einer Ressource im reziproken Verhältnis reduziert wird, so erhöht sich die Gesamtausfallwahrscheinlichkeit trotz Redundanz nicht. Dies ist beispielsweise in etwa beim Fahrrad mit Vorder- und Hinterbremse gegeben. Allerdings unterliegen die Bauteile in der Praxis Alterungserscheinungen, wodurch doch eine leicht erhöhte Ausfallwahrscheinlichkeit entsteht. Weiterhin entstehen durch diese mehrfache Auslegung einer Ressource auch Mehrkosten im selben Verhältnis.

<sup>9</sup>Bei entsprechender Betrachtungsweise der Funktion existiert unter Umständen durchaus eine Einschränkung durch ein langsames System. Hier wird aber als Funktion das Speichern von Daten gesehen und diese Funktionalität wird nicht beeinträchtigt.

**Fazit** Es können also durch Redundanz keine Vorteile ohne Nachteile erkaufte werden.

### 3.1 Definition der Redundanz (mit Hilfe eines Automaten)

Das System aus Betrachtungseinheit, gegebenen Bedingungen und der geforderten Funktionalität kann modelliert werden durch einen endlichen, deterministischen Automaten  $A = (S, T, s_0, S_z, \delta)$  aus folgenden Bestandteilen: (vgl. Anhang A, [Bro98])

- endliche Menge  $S$  von Zuständen
- endliche Menge  $T$  von Steuerungen (Eingangszeichen)
- Anfangszustand<sup>10</sup>  $s_0 \in S$
- Menge  $S_z \subseteq S$  von gewünschten Endzuständen
- Übergangsfunktion  $\delta : S \times T \rightarrow S$

Häufig wird ein neutrales Element  $\exists \varepsilon \in T : \forall s \in S : \delta(s, \varepsilon) = s$  gefordert. In den Beispielen in dieser Arbeit werden allerdings nur  $\varepsilon$ -freie Automaten betrachtet. Aber um die Allgemeinheit der Definition in diesem Kapitel nicht einzuschränken sei das neutrale Element  $\varepsilon$  genannt.

Die Modellierung der Zustände als Menge  $S$  schließt auch alle Schadenszustände  $D \subset S$  des System ein. Die prinzipielle **Aufgabe** besteht darin, mittels geeigneter Steuerungen aus  $T$  von einem Anfangszustand  $s_0$  zu einem gewünschten Endzustand aus  $S_z$  zu gelangen. Die möglichen Übergänge zwischen den Zuständen werden durch die Übergangsfunktion  $\delta$  modelliert.

Eine **Lösung** ist eine Folge von Eingangszeichen aus  $T$ , die einen **Weg**<sup>11</sup> von dem Anfangszustand  $s_0$  zu einem Endzustand aus  $S_z$  beschreibt. Genauer ist die Lösung also eine endliche Folge  $\{t_i\}_{i=1,2,\dots,n_l}$  von Eingangs-

---

<sup>10</sup>Theoretisch könnte man sich vorstellen, statt nur 1 Anfangszustand eine ganze Menge von Anfangszuständen zu betrachten. Dies würde aber bedeuten, dass die Maschine, die durch den Automaten modelliert wird, keinen definierten Zustand zu einem gewählten Anfangszeitpunkt hat. Darauf aufbauend ist kein deterministisches Vorgehen möglich. Bei einem nichtdeterministischen Automaten werden immerhin die Folgezustände nach einer Steuerung in einem bekannten Zustand offen gelassen.

<sup>11</sup>Ein Automat kann als Graph dargestellt werden; dabei stellt jeder Zustand einen Knoten und jede Übergangsmöglichkeit eine gerichtete Kante dar. (vgl. Anhang A)

zeichen aus  $T$ , so dass mittels wiederholter Anwendung der Übergangsfunktion, gesteuert durch die Eingangszeichen, von  $s_0$  in  $S_z$  abgebildet wird:

$$s_{j_0} := \delta(s_0, t_1)$$

$$s_{j_k} := \delta(s_{j_{k-1}}, t_k) \text{ für } k = 1, 2, \dots, n_l$$

$$s_{j_{n_l}} = \delta(\dots(\delta(\delta(s_0, t_1), t_2), \dots), t_{n_l}) \in S_z$$

Ein Zustand ist **erreichbar** von einem Zustand, wenn ein Weg vom ersten zum letzteren existiert.

Ein **Schaden** ist eine Teilmenge  $D_i \subset S$ , die zwar von  $s_0$  erreichbar ist, aber von der Anfangszustand  $s_0$  nicht erreichbar ist. Eine Reparatur des Systems ist also durch  $\delta$  nicht modelliert. Weiterhin ist innerhalb von  $D_i$  jeder Knoten durch jeden anderen erreichbar. Dies grenzt die Schäden untereinander ab. Am Beispiel 3.1.2 werden wir sehen, dass ein Schaden wirklich aus einer mehrelementigen Menge bestehen kann.

Schäden, von denen aus kein gewünschter Endzustand erreicht werden kann, also **Totalschäden**, müssen in dieser Definition der Redundanz ausgeschlossen werden. Die restlichen Schäden sind die **akzeptablen Schäden**.

**Redundanz** liegt vor, wenn für jede dieser Teilmengen  $D_i$  mit  $i = 1, \dots, n$  und  $n > 1$ , die einen akzeptablen Schaden repräsentieren, ein Weg existiert, der zu einem gewünschten Endzustand aus  $S_z$  führt.

Um Redundanz des Systems zu untersuchen, muss also ein Weg zu einem Schaden  $D_i \subset S$  und von dort ein Weg zu einem Endzustand aus  $S_z$  gefunden werden; dazu müssen alle Schäden  $D_i$  entweder bekannt sein oder bestimmt werden.

#### 3.1.1 Beispiel: RAID 1 aus 2 Festplatten

2 Festplatten  $P_1$  und  $P_2$  bilden ein RAID-1-Verbund. Die Menge der Zustände  $S := \{s_0, s_1, s_2, s_3\}$  mit dem Anfangszustand  $s_0$  ergibt sich aus:

- $s_0$  = „Daten können von  $P_1$  und  $P_2$  gelesen werden.“
- $s_1$  = „Daten können nur von  $P_1$  gelesen werden;  $P_2$  ist defekt.“

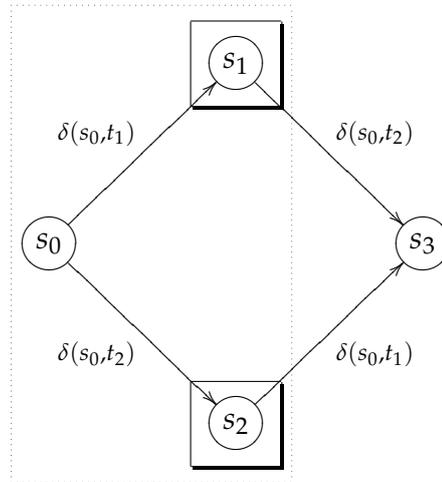


Abbildung 3.1: Automat zu RAID 1 als Graph

- $s_2 =$  „Daten können nur von  $P_2$  gelesen werden;  $P_1$  ist defekt.“
- $s_3 =$  „Daten können weder von  $P_1$  noch von  $P_2$  gelesen werden;  $P_1$  und  $P_2$  sind defekt.“

Die gewünschten Endzustände sind  $S_z = \{s_0, s_1, s_2\}$ , das gestrichelte Rechteck in Abb. 3.1.

Die Menge der Steuerungen  $T = \{t_1, t_2\}$  ergibt sich aus:

- $t_1 =$  „ $P_1$  fällt aus.“
- $t_2 =$  „ $P_2$  fällt aus.“

Die Übergangsfunktion  $\delta$  kann dem Graphen in Abb. 3.1 entnommen werden.

Die schattierten Rechtecke in Abb. 3.1 zeigen die akzeptablen Schäden an. Also ist  $D_1 = \{s_1\}$  und  $D_2 = \{s_2\}$ .

Trivialerweise gehört jeder Zustand eines Schadens bereits zu einem Endzustand aus  $S_z$ . Somit liegt Redundanz vor.

### 3.1.2 Beispiel: das Bremsen eines Fahrrades

Ein Fahrrad besteht unter anderem aus 2 Bremsen. Die Aufgabe besteht darin ein fahrendes Fahrrad zum Stehen zu bringen. Um einfacher die Zustände  $s_i$  zu definieren, seien folgende Zustandsmengen  $Z_i$  eingeführt:

- $Z_1 = \{s_i : \text{„Fahrrad fährt.“}\}$
- $Z_2 = \{s_i : \text{„Fahrrad steht.“}\}$
- $Z_3 = \{s_i : \text{„Vorderbremse funktioniert.“}\}$
- $Z_4 = \{s_i : \text{„Hinterbremse funktioniert.“}\}$

Damit ergibt sich die Menge aller Zustände  $S := Z_1 \cup Z_2 \cup Z_3 \cup Z_4 := \{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$ . Der Anfangszustand  $s_0$  und die restlichen 7 Zustände ergeben sich zu: (Die im Folgenden angegebenen Verknüpfung der Mengen  $Z_i$  sind jeweils genau 1-elementig.)

- $s_0 = \text{„Fahrrad fährt und alles ist OK.“} \in Z_1 \cap Z_3 \cap Z_4$
- $s_1 = \text{„Fahrrad fährt, Vorderbremse ist OK, Hinterbremse ist defekt.“} \in Z_1 \cap Z_3 \setminus Z_4$
- $s_2 = \text{„Fahrrad fährt, Vorderbremse ist defekt, Hinterbremse ist OK.“} \in Z_1 \cap Z_4 \setminus Z_3$
- $s_3 = \text{„Fahrrad fährt, Vorderbremse ist defekt, Hinterbremse ist defekt.“} \in Z_1 \setminus Z_3 \setminus Z_4$
- $s_4 = \text{„Fahrrad steht und alles ist OK.“} \in Z_2 \cap Z_3 \cap Z_4$
- $s_5 = \text{„Fahrrad steht, Vorderbremse ist OK, Hinterbremse ist defekt.“} \in Z_2 \cap Z_3 \setminus Z_4$
- $s_6 = \text{„Fahrrad steht, Vorderbremse ist defekt, Hinterbremse ist OK.“} \in Z_2 \cap Z_4 \setminus Z_3$
- $s_7 = \text{„Fahrrad steht, Vorderbremse ist defekt, Hinterbremse ist defekt.“} \in Z_2 \setminus Z_3 \setminus Z_4$

Die gewünschten Endzustände sind  $S_z = Z_2 = \{s_4, s_5, s_6, s_7\}$ , das gestrichelte Rechteck in Abb. 3.2.

Die Menge der Steuerungen  $T = \{t_1, t_2, t_3, t_4, t_5\}$  ergibt sich aus:

### 3.1 Def. Redundanz mittels Automaten

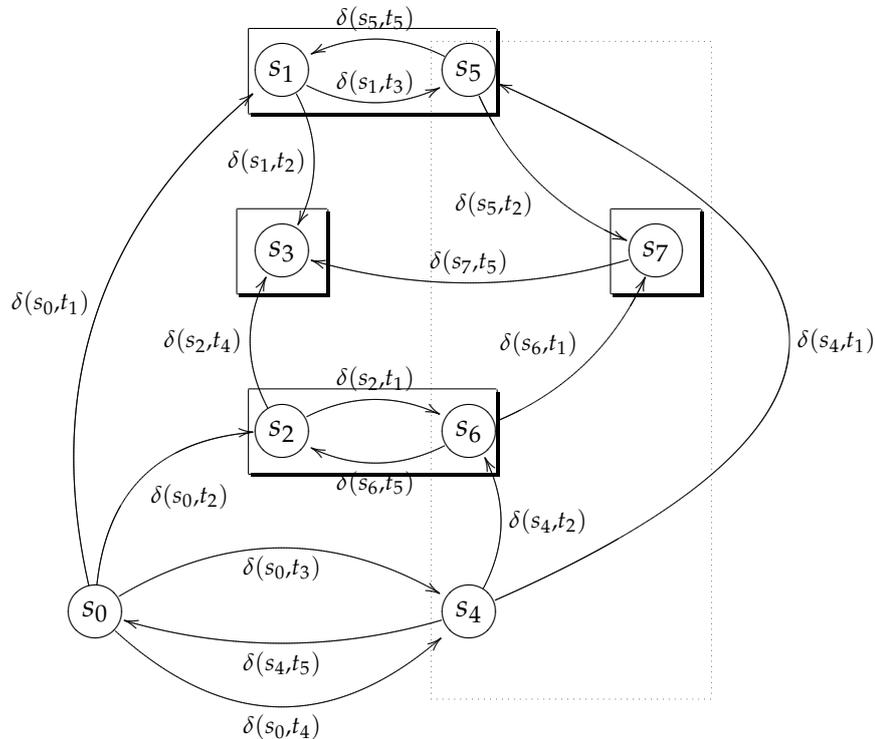


Abbildung 3.2: Automat zu Fahrrad soll bremsen

- $t_1$  = „Hinterbremse fällt aus.“
- $t_2$  = „Vorderbremse fällt aus.“
- $t_3$  = „Vorderbremse bremsst.“
- $t_4$  = „Hinterbremse bremsst.“
- $t_5$  = „Fahrrad beschleunigt.“

Die Übergangsfunktion  $\delta$  kann dem Graphen in Abb. 3.2 entnommen werden.

Die schattierten Rechtecke in Abb. 3.2 zeigen die akzeptablen Schäden an. Die Schadensfälle ergeben sich also zu  $D_1 = \{s_1, s_5\}$ ,  $D_2 = \{s_2, s_6\}$ ,  $D_3 = \{s_3\}$  und  $D_4 = \{s_7\}$ . Offensichtlich kann aus  $D_1$  und  $D_2$  ein gewünschter Endzustand in  $S_z$  erreicht werden.  $D_3$  und  $D_4$  sind als Total Schäden zu interpretieren in denen keinerlei Hoffnung auf Funktionalität besteht; trotzdem ist  $D_4$  ein gewünschter Endzustand. Unter Ausschluss

### 3 Redundanz

---

der Totalschäden liegt also für die akzeptablen Schäden  $D_1$  und  $D_2$  Redundanz vor.

### 3.2 Definition der Redundanz (mit Hilfe von Funktionen)

Die Modellierung besteht aus der Angabe aller Zustände  $Z$ , des Anfangszustandes  $z_A \in Z$ , der gewünschten Endzustände  $Z_B \subset Z$  und der Angabe der Schadenszustände  $Z_S \subset Z$ . Die Aufgabe  $P$  ist wiederum, eine Abbildung von  $z_A$  in  $Z_B$  zu finden. Alle Lösungen werden durch die Menge  $L := \{f_i : Z \rightarrow Z \mid f_i(z_A) \in Z_B\}$  mit der Indexmenge der Funktionen  $I := \{i \mid f_i \in L\}$  beschrieben.

Die Menge  $L_J := \{f_i \in L : i \in J\}$  mit den Indizes  $J \subset I$  löst die Aufgabe  $P$  mit **Redundanz**, wenn

$$\forall i \in J : \exists j \in J : f_i(z_A) \neq f_j(z_A) \in Z_S.$$

Mit anderen Worten: Wenn ein System  $i$  ausfällt, dann existiert mindestens ein anderes<sup>12</sup> System  $j$ , das die Aufgabe  $P$  in einem Schadenszustand aus  $Z_S$  löst, also abbildet in  $Z_S \cap Z_B$ .

Es genügt bei dieser Definition nicht,  $f_i \neq f_j$  zu fordern: Der Ausfall eines Systems  $i$  bedeutet, dass die Lösung  $f_i$  nicht mehr funktioniert, da der Schaden  $f_i(z_A)$  den Grund für den Ausfall darstellt. Somit wird das Gesamtsystem unter einem Schaden betrachtet, womit auch ein gewünschter Endzustand ohne Schaden nicht mehr existiert. Daher muss eine Ersatzlösung  $f_j(z_A) \in Z_S$  ein Schadenszustand sein.

Die Redundanz besteht also in der Existenz mehr als nötiger Abbildungen  $f_i$ . Die Funktionen  $f_i$  stellen die Wege im vorhergehenden Abschnitt dar.

Die Menge  $L_J := \{f_i \in L : i \in J\}$  mit den Indizes  $J \subset I$  löst die Aufgabe  $P$  mit Redundanz vom **absoluten Grad**  $g$ , wenn

$$\forall J_g \subset J \text{ mit } |J_g| = g : \exists j \in J : \forall i \in J_g : f_i(z_A) \neq f_j(z_A) \in Z_S.$$

Mit anderen Worten: Für alle möglichen  $g$ -fachen Ausfälle von Systemen  $i \in J_g$  existiert jeweils mindestens ein anderes<sup>13</sup> System  $j$ , das die Aufgabe  $P$  weiterhin in einem Schadenszustand aus  $Z_S$  löst, also abermals abbildet in  $Z_S \cap Z_B$ .

Der (relative) **Grad**  $r$  der Redundanz ist  $r = \frac{g}{|J|-1}$ ; die Subtraktion im Nenner entfernt die zu viel gezählte Abbildung, die die Lösung ohne Schaden darstellt. Es gilt offensichtlich  $0 \leq r \leq 1$ .

<sup>12</sup>falls  $j = i$ , dann wäre  $f_i(a) = f_j(a)$

<sup>13</sup>falls  $j = i$ , dann wäre  $f_i(a) = f_j(a)$

Dabei bedeutet der Wert  $r = 0$  keine Redundanz – er ist gleichbedeutend mit  $J_g = \emptyset$  bzw.  $g = 0$ . Dies trifft beispielsweise ein RAID 1 aus 1 Festplatte zu.

Der theoretische Wert  $r = 1$  bedeutet eine extrem gute Redundanz. Beispielsweise bei einem RAID 1 mit  $n$  Festplatten strebt  $r$  gegen 1 für  $n$  gegen unendlich:  $r = 1 - \frac{1}{n} \xrightarrow{n \rightarrow \infty} 1$

Bereits  $r = \frac{1}{2}$  stellt eine sehr gute Redundanz dar. Sie ist beispielsweise bei einem RAID 1 aus 2 Festplatten gegeben – also die Hälfte der verfügbaren Bauteile darf ausfallen, ohne Totalversagen zu riskieren.

Der Wert  $1 - r \in [0; 1]$  heißt **Grad der Effizienz**. Ein Grad der Effizienz von 1, also eine Redundanz von 0, bedeutet eine effektive Nutzung ohne überflüssige Teile.

Es ergibt sich die prägnante Formel:

$$\text{„Redundanz“} + \text{„Effizienz“} = 1$$

Die hier vorgestellten verallgemeinerten Begriffe „Grad der Redundanz“ und „Grad der Effizienz“ sind, sofern die Nachrichten gleichwahrscheinlich sind, konsistent zu den bekannten Begriffen „Redundanz“ und „Effizienz“ in der Informations- und Kodierungstheorie (vgl. Anhang B) im Bezug auf Codierungen. Der geneigte Leser vermag dies leicht einzusehen. In Kapitel 3.2.2 wird dies auch exemplarisch vorgeführt.

#### 3.2.1 Beispiel: RAID 1 aus 2 Festplatten

2 Festplatten  $P_1$  und  $P_2$  bilden ein RAID-1-Verbund. Die Menge aller Zustände  $Z := \{z_A, z_1, z_2, z_3\}$  mit dem Anfangszustand  $z_A$  ergibt sich aus:

- $z_A =$  „Daten können von  $P_1$  und  $P_2$  gelesen werden.“
- $z_1 =$  „Daten können nur von  $P_1$  gelesen werden;  $P_2$  ist defekt.“
- $z_2 =$  „Daten können nur von  $P_2$  gelesen werden;  $P_1$  ist defekt.“
- $z_3 =$  „Daten können weder von  $P_1$  noch von  $P_2$  gelesen werden;  $P_1$  und  $P_2$  sind defekt.“

Die gewünschten Endzustände sind  $Z_B = \{z_A, z_1, z_2\}$  und die Schadenszustände sind  $Z_S = \{z_1, z_2, z_3\}$ . Es gibt 3 Abbildungen von  $z_A$  in  $Z_B$  die alle Lösungen  $L := \{f_1, f_2, f_3\}$  darstellen:

- $f_1(z_A) = z_A$
- $f_2(z_A) = z_1$
- $f_3(z_A) = z_2$

$L_J = \{f_1, f_2, f_3\} = L$  löst nun die Aufgabe mit Redundanz vom Grad  $g = 1$  und  $r = \frac{1}{2}$ . Für eine größere Ausfallmenge  $J_g$  muss keine alternative Lösung mehr existieren, die in einen Schadensfall aus  $Z_S$  abbildet – z. B. für  $J_g = \{2, 3\}$  gibt es keine Abbildung in  $(L_J \setminus \{f_i | i \in J_g\}) \cap Z_S$ .

### 3.2.2 Beispiel: RAID 5 aus 3 Festplatten

3 Festplatten  $P_1, P_2$  und  $P_3$  bilden ein RAID-5-Verbund; d. h. Daten, die gespeichert werden sollen, werden in 2 Blöcke zerlegt und auf 2 verschiedenen Platten gespeichert, auf der dritten Platte wird eine XOR-Verknüpfung<sup>14</sup> beider Datenblöcke gespeichert. Somit können beim Ausfall einer beliebigen Platte die Daten gelesen werden – entweder direkt, oder der nicht lesbare Datenblock kann aus den Paritätsdaten der XOR-Verknüpfung rekonstruiert werden. Die Menge aller Zustände  $Z := \{z_A, z_1, z_2, z_3, z_4, z_5, z_6, z_7\}$  mit dem Anfangszustand  $z_A$  ergibt sich aus:

- $z_A =$  „Daten können gelesen werden, keine Platte defekt.“
- $z_1 =$  „Daten können von  $\{P_2, P_3\}$  gelesen werden;  $P_1$  ist defekt.“
- $z_2 =$  „Daten können von  $\{P_1, P_3\}$  gelesen werden;  $P_2$  ist defekt.“
- $z_3 =$  „Daten können von  $\{P_1, P_2\}$  gelesen werden;  $P_3$  ist defekt.“
- $z_4 =$  „Daten können nicht gelesen werden;  $\{P_1, P_2\}$  sind defekt.“
- $z_5 =$  „Daten können nicht gelesen werden;  $\{P_1, P_3\}$  sind defekt.“
- $z_6 =$  „Daten können nicht gelesen werden;  $\{P_2, P_3\}$  sind defekt.“
- $z_7 =$  „Daten können nicht gelesen werden;  $\{P_1, P_2, P_3\}$  sind defekt.“

Die gewünschten Endzustände sind  $Z_B = \{z_A, z_1, z_2, z_3\}$  und die Schadenszustände sind  $Z_S = \{z_4, z_5, z_6, z_7\}$ . Es gibt 4 Abbildungen von  $z_A$  in  $Z_B$  die alle Lösungen  $L := \{f_1, f_2, f_3, f_4\}$  darstellen:

<sup>14</sup>Die XOR-Verknüpfung wird auch manchmal ausschließliches Oder genannt. Eine Verknüpfungstabelle ist implizit in Tabelle 3.2 enthalten.

### 3 Redundanz

---

- $f_1(z_A) = z_A$
- $f_2(z_A) = z_1$
- $f_3(z_A) = z_2$
- $f_4(z_A) = z_3$

$L_J = \{f_1, f_2, f_3, f_4\} = L$  löst nun die Aufgabe mit Redundanz vom Grad  $g = 1$  und  $r = \frac{1}{3}$ . Es darf also nur eine Festplatte ausfallen.

Aus der Sicht der Informations- und Kodierungstheorie kann dies ebenfalls beleuchtet werden. Dazu sollen nun exemplarisch 2 Bits auf einem RAID 5 aus 3 Festplatten gespeichert werden. Die 2 Bits, die gespeichert werden sollen, entsprechen also einer 4-nären Quelle mit gleichen Wahrscheinlichkeiten. Die Codierung  $C$  speichert nun die Daten in 3 Bits auf den 3 Festplatten.<sup>15</sup>

$$C(x, y) := (x, y, x \underline{\vee} y) \in Q := \{0, 1\}^3 \text{ mit } x, y \in \{0, 1\}$$

Der Ausgang der Codierung bildet nun eine 8-näre Quelle. Als Wertetabelle für die Codierung  $C$  mit den zugehörigen gleichverteilten Wahrscheinlichkeiten ergibt sich Tabelle 3.2.

Die restlichen Wörter aus  $Q$  treten mit der Wahrscheinlichkeit 0 auf. Somit ergibt sich für die Entropie der Codewörter

$$H_2 = \sum_{s \in Q} p(s) \text{lb} \frac{1}{p(s)} = \sum_{x \in \{0,1\}} \sum_{y \in \{0,1\}} p(C(x, y)) \text{lb} \frac{1}{p(C(x, y))} = 2$$

während die maximale Entropie einer 8-nären Quelle über dem Alphabet  $Q$

$$H_0 = \sum_{i=1}^8 \frac{1}{8} \text{lb} 8 = \text{lb} 8 = 3$$

beträgt. Somit ergibt sich aus Sicht der Informations- und Kodierungstheorie die absolute Redundanz  $R$  ebenfalls zu  $g$ :

$$R = H_0 - H_2 = 3 - 2 = 1$$

---

<sup>15</sup>  $\underline{\vee}$  steht für die logische XOR-Verknüpfung.

### 3.2 Def. Redundanz mittels Funktionen

$(x, y)$	$C(x, y)$	Wahrscheinlichkeit $p(C(x, y))$
$(0, 0)$	$(0, 0, 0)$	$\frac{1}{4}$
$(0, 1)$	$(0, 1, 1)$	$\frac{1}{4}$
$(1, 0)$	$(1, 0, 1)$	$\frac{1}{4}$
$(1, 1)$	$(1, 1, 0)$	$\frac{1}{4}$

Tabelle 3.2: Wertetabelle der Codierung bei einem RAID 5 aus 3 Festplatten mit den zugehörigen Wahrscheinlichkeiten

Gleiches gilt für die relative Redundanz:<sup>16</sup>

$$r = \frac{R}{H_0} = \frac{1}{3}$$

Die Einschränkung auf das Speichern von nur 2 Bits kann natürlich fallen gelassen werden. Zum Speichern eines Datenblockes aus  $m$  Bits<sup>17</sup> muss dieser trotzdem in 2 Teile  $x$  und  $y$  zerlegt werden. Um die XOR-Verknüpfung zu ermöglichen, gehen wir davon aus, dass diese Teile gleiche Größe  $\frac{m}{2} \in \mathbb{N}$  besitzen – andernfalls muss ein informationsloses Bit (Irrelevanz) zugefügt werden. Damit ergibt sich die Codierung zu:

$$C(x, y) := (x, y, x \underline{\vee} y) \in Q := \{0, 1\}^{\frac{3m}{2}} \text{ mit } x, y \in \{0, 1\}^{\frac{m}{2}}$$

Die Codewörter bilden wiederum eine  $2^{\frac{3m}{2}}$ -näre Quelle, wobei nur  $2^m$  Wörter als Codewörter mit gleichen Wahrscheinlichkeiten  $\frac{1}{2^m}$  auftreten.

<sup>16</sup>Die Wahl der Basis 2 für den Logarithmus, also  $\text{lb}$ , war beliebig; allerdings angepasst an das Bit. Für die beliebige Wahl  $a$  als Basis ergibt sich der selbe Wert  $r = \frac{\log_a 2}{\log_a 8} = \log_8 2 = \frac{1}{3}$ .

<sup>17</sup>In der technischen Umsetzung orientiert man sich natürlich an der Größe der Datenblöcke, die eine Festplatte in dem verwendeten Dateisystem speichert. Ein typischer Wert hierfür ist 32 Kilobytes als chunk size.

Als Entropie  $H_2$  der Codierung, als maximale Entropie  $H_0$  einer  $2^{\frac{3m}{2}}$ -nären Quelle und als relative Redundanz  $r$  ergeben sich:

$$H_2 = \sum_{i=1}^{2^m} \frac{1}{2^m} \log 2^m = m$$

$$H_0 = \log 2^{\frac{3m}{2}} = \frac{3m}{2}$$

$$r = \frac{H_0 - H_2}{H_0} = 1 - \frac{2m}{3m} = \frac{1}{3}$$

Somit wurde hier in diesem Beispiel gezeigt, dass die Begriffe Redundanz und Effizienz aus Kapitel 3.2 im wesentlichen mit denen in der Informations- und Kodierungstheorie übereinstimmen.

Diese RAID-Beispiele können sehr einfach auf  $n$  Festplatten erweitert werden und nun kann die Tabelle 3.1 um die Begriffe Grad der Redundanz und Grad der Effizienz erweitert werden. Dies ist in Tabelle 3.3 dargestellt.

#### 3.2.3 Beispiel: Fahrrad soll bremsen

Auch das Beispiel aus Kapitel 3.1.2 können wir aufgreifen: Die Menge aller Zustände  $Z := \{z_A, z_1, z_2, z_3, z_4, z_5, z_6, z_7\}$  mit dem Anfangszustand  $z_A$  ergibt sich aus (Analog zu  $s_i$  aus Kapitel 3.1.2):

- $z_A =$  „Fahrrad fährt, kein Schaden.“
- $z_1, z_2, z_3 =$  „Fahrrad fährt mit Schäden.“
- $z_4 =$  „Fahrrad steht, kein Schaden“
- $z_5 =$  „Fahrrad steht, Vorderbremse ist OK, Hinterbremse ist defekt“
- $z_6 =$  „Fahrrad steht, Vorderbremse ist defekt, Hinterbremse ist OK“
- $z_7 =$  „Fahrrad steht, Vorderbremse ist defekt, Hinterbremse ist defekt“

Die gewünschten Endzustände sind  $Z_B = \{z_4, z_5, z_6, z_7\}$  und die Schadenszustände sind  $Z_S = \{z_1, z_2, z_3, z_5, z_6, z_7\}$ . Es gibt 3 Abbildungen von  $z_A$  in  $Z_B$ , die alle Lösungen  $L := \{f_1, f_2, f_3\}$  darstellen – auf  $z_7$  kann nicht direkt aus fahrenden Zuständen abgebildet werden; schließlich stellt ein  $f_i$  ein Bremsen dar, und kein Bremsen führt zu  $z_7$ :

Verwendungsart	RAID 1	RAID 5	JBOD	
Plattenzahl	$n$	$n$	$n - 1$	1
Kapazität	1	$n - 1$	$n - 1$	1
Maximalschaden	$n - 1$	1	0	0
Funktionswahrscheinlichkeit	$1 - p^n$	$1 - (n - 1)^2 p^2$	$(1 - p)^{n-1} \approx 1 - (n - 1)p$	$1 - p$
Schadenswahrscheinlichkeit	$np$	$np$	$(n - 1)p$	$p$
Grad der Redundanz	$1 - \frac{1}{n}$	$\frac{1}{n}$	0	0
Grad der Effizienz	$\frac{1}{n}$	$1 - \frac{1}{n}$	1	1

Tabelle 3.3: Vergleich der Funktions- und Ausfallwahrscheinlichkeiten verschiedener Möglichkeiten die Datenkapazität von Festplatten zu nutzen. Redundanz reduziert dabei nicht die Schadenswahrscheinlichkeit, sondern erhöht nur die Funktionswahrscheinlichkeit. Grad der Redundanz und Grad der Effizienz sind entsprechend angegeben.

- $f_1(z_A) = z_4$
- $f_2(z_A) = z_5$
- $f_3(z_A) = z_6$

$L_J = \{f_1, f_2, f_3\} = L$  löst nun die Aufgabe mit Redundanz vom Grad  $g = 1$  und  $r = \frac{1}{2}$ . Es darf also nur eine Bremse ausfallen.

### 3.3 Problem der Redundanz

Das Problem der Redundanz besteht nun darin, einen redundanten Lösungssatz zu finden, also eine geeignete Indexmenge  $J$ .

Um eine Indexmenge  $J \subset I$  zu finden muss die Ausgangsmenge  $I$  aufgeteilt werden auf  $J$  und  $I \setminus J$ . Dies ist das NP-vollständige Partitionsproblem [Kar72].

### 3 Redundanz

---

Diese Indexmenge  $J$  macht auch deutlich, dass der Begriff Redundanz eine Eigenschaft von Betrachtungseinheiten aus diskreten Komponenten ist. Wenn keine diskreten Komponenten vorhanden sind, die einen wesentlichen Teil darstellen, kann auch nicht von einer Redundanz gesprochen werden.

---

## 4 Schadenstoleranz in der Topologieoptimierung

Um nun die beiden Begriffe Schadenstoleranz (aus Kapitel 2) und Topologieoptimierung (im Anhang C findet sich ein Überblick) zusammenzuführen, können wir vorwärts und rückwärts denken. Versuchen wir erst mal den etwas klareren Weg nach vorne:

In der Topologieoptimierung ist nun unter gewissen Bedingungen  $\varphi \in \Phi$  die optimale<sup>18</sup> Dichteverteilung  $\varphi : \Omega \rightarrow \{0;1\}$  auf einem Bauraum  $\Omega \subset \mathbb{R}^d$  gesucht – also ein Optimierungsproblem:

suche optimales  $\varphi$      unter  $\varphi \in \Phi$

Wenn eine Lösung  $\tilde{\varphi} : \Omega \rightarrow \{0;1\}$  nun Schadenstoleranz aufweisen soll, so muss die Struktur  $\Sigma := \{x \in \Omega : \tilde{\varphi}(x) = 1\}$  redundant gegenüber den Belastungen sein. Sie muss also die Lasten auf mehr als eine Art und Weise abtragen können. Beispielsweise könnte die Last  $f_1$  durch einen Lastpfad in dem Bereich  $\Sigma_{1,A} \subset \Sigma$  oder durch einen weiteren Bereich  $\Sigma_{1,B} \subset \Sigma$  mit  $\Sigma_{1,A} \neq \Sigma_{1,B}$  getragen werden. Dies bedeutet nun, dass aus der einen Lösung  $\tilde{\varphi}$  zwei Lösungen

$$\tilde{\varphi}_1 : \Sigma_{1,A} \rightarrow \{0;1\} \text{ und } \tilde{\varphi}_2 : \Sigma_{1,B} \rightarrow \{0;1\}$$

mit

$$\tilde{\varphi}_1 = \tilde{\varphi} \Big|_{\Sigma_{1,A}} \text{ und } \tilde{\varphi}_2 = \tilde{\varphi} \Big|_{\Sigma_{1,B}}$$

werden. Sicherlich wäre  $\Sigma_{1,A} \cap \Sigma_{1,B} = \emptyset$  wünschenswert<sup>19</sup>, aber es gibt Bereiche, wie die Stellen für die Einleitung der Kraft, die sowohl zu  $\Sigma_{1,A}$  als auch zu  $\Sigma_{1,B}$  gehören müssen.

Wenn nun für jede Last  $f_i$  mit  $i \in I$  mindestens 2 derartige Bereiche  $\Sigma_{i,A}$  und  $\Sigma_{i,B}$  existieren, so ist die Struktur tolerant gegenüber allen (akzeptablen) Schäden  $\Sigma_{i,j} \setminus \Sigma_{i,k}$  mit den Indizes  $i \in I$  und  $j, k \in \{A, B\}$ .

---

<sup>18</sup>Die Optimalität wird natürlich durch eine reellwertige Zielfunktion gemessen. Auf diese wird hier noch nicht eingegangen, da sie zum Einen bei der folgenden Betrachtung keine Rolle spielt und zum Anderen sich je nach Methode der Topologieoptimierung unterscheidet.

<sup>19</sup>Ist die Redundanz in der Gesamtaufgabenstellung nur für einen Teil der Aufgabenstellung (also z. B. ein Teil des Bauraums mit entsprechenden Krafteinleitungen und Lagern) gefordert, so wird hier nur diese Teilaufgabe als eigentliche Aufgabe betrachtet.

Mit anderen Worten, es existiert eine 3-elementige Lösungsmenge  $L_J := \{\Sigma_{i,A}, \Sigma_{i,B}, \Sigma_{i,A} \cup \Sigma_{i,B}\}$  für das Problem und für jede 1-elementige Auswahl aus der Lösungsmenge  $L_J$  existiert eine andere Lösung des Problems – also ergibt sich ein absoluter Grad der Redundanz von  $g = 1$  und ein (relativer) Grad der Redundanz von  $r = \frac{1}{3-1} = \frac{1}{2}$ .

Dabei ist natürlich wünschenswert, dass sowohl die Vereinigung aller akzeptablen Schäden  $\bigcup_{i,j,k} (\Sigma_{i,j} \setminus \Sigma_{i,k})$  als auch jeder akzeptable Schaden für sich, d. h.  $\Sigma_{i,j} \setminus \Sigma_{i,k}$  mit  $i \in I$  und  $j, k \in \{A, B\}$ , möglichst groß sind. Mehr als  $\bigcup_{i,j,k} (\Sigma_{i,j} \setminus \Sigma_{i,k}) = \Sigma$  ist dabei nicht erreichbar. Und alle wesentlichen strukturellen Komponenten sollten als akzeptabler Schaden  $\Sigma_{i,j} \setminus \Sigma_{i,k}$  mit geeigneten Indizes  $i, j, k$  darstellbar sein.

Damit wäre das Ziel einer Schadenstoleranz erreicht. Aber wie soll man am Anfang sicher stellen, dass man am Ende eine geeignete Zerlegung in  $A := \bigcup_i \Sigma_{i,A}$  und  $B := \bigcup_i \Sigma_{i,B}$  von  $\Sigma = A \cup B$  erhält?

Wenn man nun gedanklich mit dem Wunsch dieser Zerlegung anfängt, so soll die gewünschte Struktur beim Eintreten von jedem erdenklichen Schaden ihre Aufgabe ‚Halten‘ erfüllen – nicht bei jedem Schaden, sondern nur bei den akzeptablen Schäden; dabei soll natürlich auch das Versagen jeder wesentlichen strukturellen Komponente berücksichtigt sein.

Das Problem: Wie erkennt man eine wesentliche strukturelle Komponente? Und wie beschreibt man eine Komponente, wenn man überhaupt keine Struktur hat?

In [Rüe94, Kapitel 2.5] wurde die Zuverlässigkeit sehr einfach durch Parallelsysteme beschrieben: In Abb. 4.1 ist ein Zuverlässigkeitsschaltbild aus 3 Elementen (Komponenten) zu sehen.  $\Sigma_1$  bildet mit  $\Sigma_2$  ein Parallelsystem  $P_1$ , welches beim Ausfall eines Elementes weiterhin funktioniert, während  $\Sigma_3$  mit dem Parallelsystem  $P_1$  ein Seriensystem bildet, welches schon beim Ausfall eines der Seriensystemelemente versagt.

Das Gesamtsystem aus Abb. 4.1 toleriert somit einen Ausfall von einem akzeptablen Schaden  $\Sigma_1$  oder  $\Sigma_2$ .  $\Sigma_3$  stellt einen Totalschaden dar, dies wäre beispielsweise der Bereich des Bauraumes an dem die Last angreift.

Dies kann hier so leicht erkannt werden, da das Gesamtsystem (die gesamte Struktur)  $\Sigma$  bereits geeignet in parallele bzw. serielle Bereiche zerlegt wurde.

In der Topologieoptimierung sucht man nun die Gesamtstruktur. Aber selbst wenn man diese hätte, so ist die Aufspaltung in parallele Teilstruk-

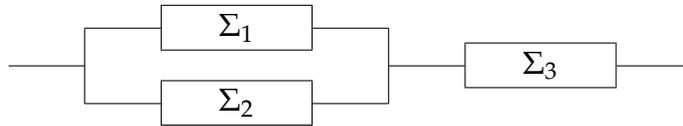


Abbildung 4.1: Beispiel eines Zuverlässigkeitsschaltbild:  $\Sigma_1$  und  $\Sigma_2$  bilden ein Parallelsystem  $P_1$ .  $P_1$  bildet mit  $\Sigma_3$  ein Seriensystem.

turen immer noch das nach [Kar72] NP-vollständige Partitionsproblem:

$$\begin{aligned} \text{Eingabe: } & (c_1, c_2, \dots, c_s) \in \mathbb{Z}^s \\ & S := \{1, 2, \dots, s\} \\ \text{Problem: } & \exists I \subset S : \sum_{h \in I} c_h = \sum_{h \in S \setminus I} c_h \end{aligned}$$

Die Komplexitätsklasse NP umfasst alle Probleme, deren Lösung in polynomieller Zeit überprüfbar ist und damit auch alle Probleme mit polynomiellen Lösungsverfahren, die in der Komplexitätsklasse P zusammengefasst sind. Wenn nun ein Unterschied zwischen den Komplexitätsklassen NP und P existiert, so liegt das Problem, eine redundante Struktur aus dem Nichts zu bauen, nicht mehr in NP, da ja bereits die Überprüfung der Lösung NP-vollständig und daher nicht in P ist – NP-vollständige Probleme liegen nur dann in P, wenn das Gegenteil der Annahme, also  $NP = P$ , gilt.

Unter der heute weit verbreiteten Annahme  $NP \neq P$  ist also das Problem, eine redundante Struktur zu berechnen, zu aufwendig.

Es ist aber denkbar, die Aufspaltung in parallele Systeme bereits am Anfang im Bauraum vorzunehmen. Also  $\Omega = \Omega_1 \cup \Omega_2$ , und nun ist in  $\Omega_1$  bzw.  $\Omega_2$  jeweils eine optimale Struktur zu finden.

Da keinerlei Information über  $\Omega_1$  bzw.  $\Omega_2$  vorliegt, ist eine triviale Wahl  $\Omega = \Omega_1 = \Omega_2$ . Nun bleibt zu hoffen, dass die optimierten Strukturen  $\Sigma_1 \subset \Omega_1$  und  $\Sigma_2 \subset \Omega_2$  nicht identisch sind, da als Ergebnis nur  $\Sigma = \Sigma_1 \cup \Sigma_2$  in Frage kommt und die Schadenstoleranz in  $\Sigma \setminus (\Sigma_1 \cap \Sigma_2)$  liegt. Um nicht nur vom Glauben und der Hoffnung zu leben, sollte dies geeignet formuliert werden.

### 4.1 Schadenstoleranz durch Redundanz im SIMP-Ansatz

In diesem Kapitel 4.1 wird die grundlegende Idee, Redundanz in der Topologieoptimierung zu modellieren, am Beispiel des SIMP-Ansatzes (vgl. Anhang C.1) umgesetzt. Dazu wird in Kapitel 4.1.1 die Idee in einfacher Form präsentiert. In Kapitel 4.1.2 wird dann die hohe Redundanz von  $1 - \frac{1}{n}$  umgesetzt. Um die Effizienz zu steigern wird im Anschluss in Kapitel 4.1.3 der Grad der Redundanz  $\frac{1}{n}$  auf die gleiche Art und Weise modelliert und somit eine Effizienz von  $1 - \frac{1}{n}$  erzielt. Allerdings ergibt sich dabei eine erheblich schlechtere Zielfunktion – Konvexität ist nicht einmal mehr näherungsweise vorhanden – als in Kapitel 4.1.2. Ein Umschreiben in andere Designvariablen ergibt wieder die gleiche näherungsweise Konvexität wie in Kapitel 4.1.2. Zum Schluss werden in Kapitel 4.1.4 die numerisch berechneten Ergebnisse beleuchtet. Dabei zeigt sich die Schwäche bei diesem Zugang, was Kapitel 4.2 motiviert.

#### 4.1.1 Idee

Die am Anfang von Kapitel 4 erklärte Zerlegung der Gesamtstruktur  $\Sigma$  in  $\Sigma_1$  und  $\Sigma_2$  liefert unmittelbar eine Redundanz bzgl. eines Schadens in  $\Sigma \setminus (\Sigma_1 \cap \Sigma_2)$ . Basierend auf dem SIMP-Ansatz (vgl. Anhang C.1) wird nun eine Modellierung dieser Zerlegung gegeben.

Gesucht ist die Gesamtstruktur, dargestellt durch die diskretisierte Dichteverteilung  $\rho \in \mathbb{R}^N$ , welche in die beiden Teilstrukturen  $\rho_1$  und  $\rho_2$  zerlegt sei, also  $\rho = \rho_1 + \rho_2$ . Die zugehörigen Steifigkeitsmatrizen  $K_{\rho_1}$ ,  $K_{\rho_2}$  und  $K_\rho$  ergeben sich analog – wie beim SIMP-Ansatz wird hier wiederum eine Penalty-Potenz in der Assemblierung der Elementsteifigkeitsmatrizen  $K_e$  verwendet; daher gilt im Allgemeinen  $K_{\rho_1} + K_{\rho_2} \neq K_\rho$ . Die Gleichgewichtsgleichungen  $K_r u_r = f_r$ ,  $r \in \{\rho, \rho_1, \rho_2\}$  werden – wie üblich (vgl. Anhang C.1) – hier nicht als Nebenbedingung aufgefasst, sondern als Teil der Zielfunktion. Also die Verschiebungen  $u_r$ ,  $r \in \{\rho, \rho_1, \rho_2\}$  werden aus dem linearen Gleichungssystem (4.8) bestimmt, um damit die Zielfunktion (4.1) auszuwerten. Damit das Gleichungssystem (4.8) eindeutig lösbar ist, muss die künstliche Unterschranke  $\rho_{lb}$  eingeführt werden. Dabei wird ein  $\alpha$  eingeführt, um die Teilstrukturen einzeln geringer zu belasten. Ein  $\alpha = 0.8$  bedeutet also, dass eine Teilstruktur 80 % der Gesamt-Last tragen soll. Die Zielfunktion setzt sich als Summe der 3 Compliance-Werte für die 3 Strukturen zusammen. (Im Anhang C.1 wird für die Penalty-Potenz  $p = 1$  die Konvexität der Compliance als Funktion der Dichtever-

teilung gezeigt; daher ist zumindest für  $p = 1$  die Zielfunktion als Summe von Compliance-Funktionen konvex.) Somit bleiben nur die vom SIMP-Ansatz bekannten Nebenbedingungen. Dies führt nun direkt zu einer sehr einfachen Formulierung, welche numerisch gut handzuhaben ist. Damit lautet die Problemstellung:

$$\text{Zielfunktion: } z(\rho_1, \rho_2) = f^T u_\rho + f^T u_{\rho_1} + f^T u_{\rho_2} \longrightarrow \min_{\rho_1, \rho_2} \quad (4.1)$$

$$\text{unter: } \|\rho\|_1 = \sum_{e=1}^N \rho(e) \leq V \quad (4.2)$$

$$0 < \rho_{lb} \leq \rho_1, \rho_2, \rho \leq 1 \quad (4.3)$$

$$\text{mit: } \rho := \rho_1 + \rho_2 \quad (4.4)$$

$$K_{\rho_1} := \sum_{e=1}^N (\rho_1(e))^p K_e \quad (4.5)$$

$$K_{\rho_2} := \sum_{e=1}^N (\rho_2(e))^p K_e \quad (4.6)$$

$$K_\rho := \sum_{e=1}^N (\rho(e))^p K_e \quad (4.7)$$

$$\begin{pmatrix} K_\rho & 0 & 0 \\ 0 & K_{\rho_1} & 0 \\ 0 & 0 & K_{\rho_2} \end{pmatrix} \begin{pmatrix} u_\rho \\ u_{\rho_1} \\ u_{\rho_2} \end{pmatrix} = \begin{pmatrix} f \\ \alpha f \\ \alpha f \end{pmatrix} \quad (4.8)$$

Auf Grund der – in Abb. 4.2 dargestellten – für  $p > 1$  nichtlinearen Abhängigkeit der Compliance von dem Volumen wird durch die Zielfunktion eine Partitionierung in  $\rho_1$  und  $\rho_2$  erzwungen: Das zur Verfügung stehende Volumen wird wegen der Nebenbedingung (4.3) und der Identität (4.4) auf  $\rho_1$  und  $\rho_2$  verteilt. Werte von  $\frac{1}{2}$  für die  $e$ -te Komponente von  $\rho_i$ , also  $\rho_i(e)$ , werden somit in der Nebenbedingung (4.2) im Verhältnis hoch gewichtet, während diese Werte in der Compliance gering gewichtet sind. Dies ist also die selbe Idee, die der Penalty-Potenz zu Grunde liegt.

Diese Aufteilung in  $\rho_1$  und  $\rho_2$  kann auf  $n$  Teile erweitert werden. Diese Partitionierung in  $n$  Teilstrukturen kann nun in unterschiedlichster Weise geschehen.

## 4 Schadenstoleranz in der Topologieoptimierung

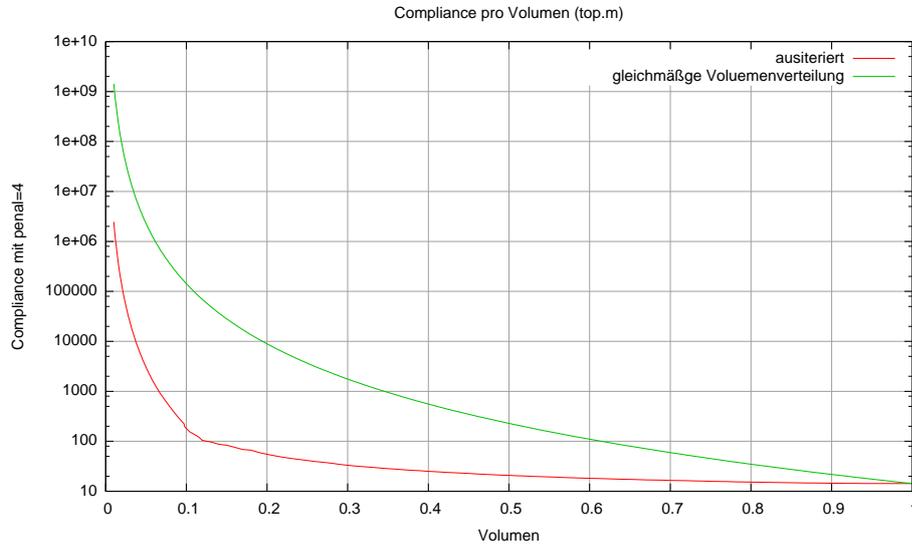


Abbildung 4.2: Nichtlineare Abhängigkeit der Compliance vom Volumen

Wie in Kapitel 4.1.3 und 4.1.2 gezeigt wird, ist dadurch eine Redundanz sowohl von  $\frac{1}{n}$  als auch von  $1 - \frac{1}{n}$  möglich; also können folgende Werte (in Abb. 4.3 visualisiert) erreicht werden:

$$0, \dots, \frac{1}{12'}, \frac{1}{11'}, \frac{1}{10'}, \frac{1}{9'}, \frac{1}{8'}, \frac{1}{7'}, \frac{1}{6'}, \frac{1}{5'}, \frac{1}{4'}, \frac{1}{3'}, \frac{1}{2'}, \frac{2}{3'}, \frac{3}{4'}, \frac{4}{5'}, \frac{5}{6'}, \frac{6}{7'}, \frac{7}{8'}, \frac{8}{9'}, \frac{9}{10'}, \frac{10}{11'}, \frac{11}{12'} \dots$$

In Kapitel 5 wird auch kurz angesprochen wie das Vorgehen in dieser Arbeit auf alle rationalen Werte zwischen 0 und 1 erweitert werden kann. Abgesehen davon, dass dies extrem aufwendig wäre, erscheint hier in dieser Arbeit, bei der erstmaligen Berechnung von redundanten Strukturen der Aufwand dafür auch nicht gerechtfertigt. Weiterhin liefern die Werte  $\frac{1}{n}$  und  $1 - \frac{1}{n}$  in einem Vorentwurf bereits vielfältige redundante Lastpfade, die somit einem Anwender zur Verfügung stehen um eine reale Struktur zu entwickeln.

### 4.1.2 Redundanz von $1 - \frac{1}{n}$

Die Grundidee aus Kapitel 4.1.1 kann durch eine Aufteilung in  $n \in \mathbb{N}$  Teilstrukturen  $\rho_i$  verallgemeinert werden und somit eine sehr hohe Redundanz von  $1 - \frac{1}{n}$  erreicht werden. Nach Kapitel 3.2 ergibt sich die niedrige Effizienz von  $\frac{1}{n'}$ , welche hier in Kauf genommen werden muss. Dadurch

## 4.1 Schadenstoleranz durch Redundanz im SIMP-Ansatz

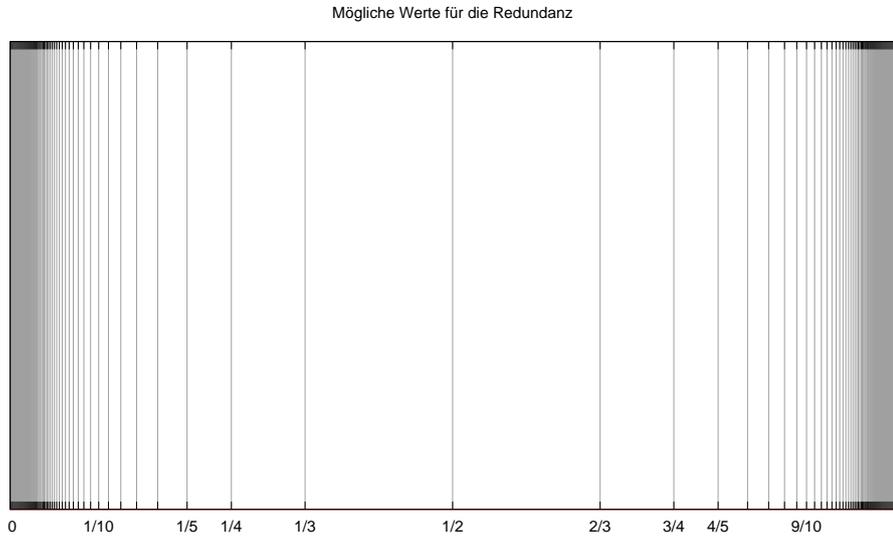


Abbildung 4.3: Mögliche Werte für die Redundanz

kommen wir zu folgender Aufgabenstellung:

$$\text{Zielfunktion: } z(\rho_1, \rho_2, \dots, \rho_n) = f^T u_\rho + \sum_{i=1}^n f^T u_{\rho_i} \longrightarrow \min_{\rho_i} \quad (4.9)$$

$$\text{unter: } \|\rho\|_1 = \sum_{e=1}^N \rho(e) \leq V \quad (4.10)$$

$$0 < \rho_{lb} \leq \rho \leq 1 \quad (4.11)$$

$$\forall i \in \{1, 2, \dots, n\} : 0 < \rho_{lb} \leq \rho_i \leq 1 \quad (4.12)$$

$$\text{mit: } \rho := \sum_{i=1}^n \rho_i \quad (4.13)$$

$$K_\rho := \sum_{e=1}^N (\rho(e))^p K_e \quad (4.14)$$

$$\forall i \in \{1, 2, \dots, n\} : K_{\rho_i} := \sum_{e=1}^N (\rho_i(e))^p K_e \quad (4.15)$$

$$K_\rho u_\rho = f \quad (4.16)$$

$$\forall i \in \{1, 2, \dots, n\} : K_{\rho_i} u_{\rho_i} = \alpha f \quad (4.17)$$

Für  $n = 2$  ist diese Aufgabenstellung identisch zu dem in Kapitel 4.1.3.

Der Parameter  $\alpha$  reduziert die Last, welche von einer Teilstruktur getragen werden soll. Selbstverständlich kann man sich zu jeder einzelnen Teilstruktur ein eigenes  $\alpha_i$  vorstellen. Da aber keine Teilstruktur irgendwelche Vorzüge gegenüber einer anderen hat, gibt dies wenig Sinn und wird hier nicht betrachtet. Weiterhin kann durch einfache Umformung  $\alpha$  aus der linearen Gleichung  $K_{\rho_i} u_{\rho_i} = \alpha f$  als Gewichtungsfaktor  $\alpha$  in die Zielfunktion geschrieben werden. Somit ist also  $\alpha$  lediglich ein Verfahrensparameter, der die Gewichtungen der Compliancewerte in der Zielfunktion realisiert.

Um später die Eigenschaften der Problemstellung untersuchen zu können und um letztlich numerische Berechnungen durchführen zu können, werden die Ableitungen benötigt – durch implizite Differentiation ist dies möglich:

Die Gleichgewichtsgleichung  $K_{\rho} u_{\rho} = f$  wird auf beiden Seiten nach  $\rho_i(e)$  differenziert und nach der Ableitung von  $u_{\rho}$  aufgelöst:

$$K_{\rho} u_{\rho} = f \implies \left( \frac{\partial K_{\rho}}{\partial \rho_i(e)} \right) u_{\rho} + K_{\rho} \left( \frac{\partial u_{\rho}}{\partial \rho_i(e)} \right) = 0 \quad (4.18)$$

$$\implies \frac{\partial u_{\rho}}{\partial \rho_i(e)} = -K_{\rho}^{-1} \left( \frac{\partial K_{\rho}}{\partial \rho_i(e)} \right) u_{\rho} \quad (4.19)$$

Genauso erhält man:

$$\frac{\partial u_{\rho_j}}{\partial \rho_i(e)} = -K_{\rho_j}^{-1} \left( \frac{\partial K_{\rho_j}}{\partial \rho_i(e)} \right) u_{\rho_j}$$

Ableiten der Identität  $K_{\rho}^{-1} K_{\rho} = I$  ergibt

$$\frac{\partial K_{\rho}^{-1}}{\partial \rho_i(e)} = -K_{\rho}^{-1} \frac{\partial K_{\rho}}{\partial \rho_i(e)} K_{\rho}^{-1},$$

bzw. analog

$$\frac{\partial K_{\rho_j}^{-1}}{\partial \rho_i(e)} = -K_{\rho_j}^{-1} \frac{\partial K_{\rho_j}}{\partial \rho_i(e)} K_{\rho_j}^{-1}.$$

Im folgenden wird das Kronecker-Delta

$$\delta_{a,b} := \begin{cases} 1 & \text{falls } a = b \\ 0 & \text{sonst} \end{cases}$$

verwendet. Weitere wissenswerte Kleinigkeiten finden sich in Anhang G.

Bei den obigen impliziten Ableitungen und im folgenden treten ein paar einfache partielle Ableitungen der Steifigkeitsmatrizen auf, die sich wie folgt ergeben:

$$\begin{aligned}
 \frac{\partial K_\rho}{\partial \rho_i(e)} &= \frac{\partial}{\partial \rho_i(e)} \sum_{\hat{e}=1}^N (\rho(\hat{e}))^p K_{\hat{e}} \\
 &= p \cdot (\rho(e))^{p-1} \cdot \underbrace{\left( \frac{\partial \rho(e)}{\partial \rho_i(e)} \right)}_{=1, \text{ da } \rho = \sum_i \rho_i} K_e \\
 &= p (\rho(e))^{p-1} K_e
 \end{aligned}$$

$$\frac{\partial^2 K_\rho}{\partial \rho_i(e) \partial \rho_j(\hat{e})} = \delta_{e,\hat{e}} p (p-1) (\rho(e))^{p-2} K_e$$

Analog:

$$\begin{aligned}
 \frac{\partial K_{\rho_j}}{\partial \rho_i(e)} &= \frac{\partial}{\partial \rho_i(e)} \sum_{\hat{e}=1}^N (\rho_j(\hat{e}))^p K_{\hat{e}} \\
 &= p \cdot (\rho_i(e))^{p-1} \cdot \delta_{i,j} K_e
 \end{aligned}$$

$$\frac{\partial^2 K_{\rho_k}}{\partial \rho_i(e) \partial \rho_j(\hat{e})} = \delta_{e,\hat{e}} \delta_{i,k} \delta_{j,k} p (p-1) (\rho_i(e))^{p-2} K_e$$

Mit  $u_r(e)$  wird im Folgenden die  $e$ -te Komponente von  $u_r$  bezeichnet. Damit ergeben sich nun die ersten partiellen Ableitungen der einzelnen Summanden der Zielfunktion (4.9) mit Hilfe von (4.19) zu:

$$\begin{aligned}
 \frac{\partial}{\partial \rho_i(e)} f^T u_\rho &= f^T \left( -K_\rho^{-1} \left( \frac{\partial K_\rho}{\partial \rho_i(e)} \right) u_\rho \right) \\
 &= -u_\rho^T \left( \frac{\partial K_\rho}{\partial \rho_i(e)} \right) u_\rho \\
 &= -p \cdot (\rho(e))^{p-1} \cdot (u_\rho(e))^T \cdot K_e \cdot u_\rho(e)
 \end{aligned}$$

#### 4 Schadenstoleranz in der Topologieoptimierung

---

$$\begin{aligned}\frac{\partial}{\partial \rho_i(e)} f^T u_{\rho_j} &= -\delta_{i,j} \frac{1}{\alpha} u_{\rho_i}^T \left( \frac{\partial K_{\rho_i}}{\partial \rho_i(e)} \right) u_{\rho_i} \\ &= -\frac{\delta_{i,j} \cdot p}{\alpha} \cdot (\rho_i(e))^{p-1} \cdot (u_{\rho_i(e)})^T \cdot K_e \cdot u_{\rho_i(e)}\end{aligned}$$

Die zweiten partiellen Ableitungen der Zielfunktion ergeben sich analog:  
Für den ersten Summanden der Zielfunktion gilt:

$$\begin{aligned}\frac{\partial}{\partial \rho_i(e)} \frac{\partial}{\partial \rho_j(\hat{e})} f^T u_{\rho} &= \frac{\partial}{\partial \rho_i(e)} \frac{\partial}{\partial \rho_j(\hat{e})} f^T K_{\rho}^{-1} f \\ &= -u^T K_{\rho} \left( \frac{\partial K_{\rho}^{-1}}{\partial \rho_i(e)} \frac{\partial K_{\rho}}{\partial \rho_j(\hat{e})} u + K_{\rho}^{-1} \frac{\partial}{\partial \rho_i(e)} \left( \frac{\partial K_{\rho}}{\partial \rho_j(\hat{e})} u \right) \right) K_{\rho} u \\ &= f^T K_{\rho}^{-1} \\ &\quad \left( 2 \frac{\partial K_{\rho}}{\partial \rho_i(e)} K_{\rho}^{-1} \frac{\partial K_{\rho}}{\partial \rho_j(\hat{e})} - \frac{\partial^2 K_{\rho}}{\partial \rho_i(e) \partial \rho_j(\hat{e})} \right) \\ &\quad K_{\rho}^{-1} f\end{aligned}\tag{4.20}$$

Auf gleiche Weise ergibt sich für die restlichen Summanden der Zielfunktion:

$$\begin{aligned}\frac{\partial}{\partial \rho_i(e)} \frac{\partial}{\partial \rho_j(\hat{e})} f^T u_{\rho_k} &= \alpha \delta_{i,k} \delta_{j,k} f^T K_{\rho_k}^{-1} \\ &\quad \left( 2 \frac{\partial K_{\rho_k}}{\partial \rho_i(e)} K_{\rho_k}^{-1} \frac{\partial K_{\rho_k}}{\partial \rho_j(\hat{e})} - \frac{\partial^2 K_{\rho_k}}{\partial \rho_i(e) \partial \rho_j(\hat{e})} \right) K_{\rho_k}^{-1} f \\ &= \alpha \delta_{i,k} \delta_{j,k} f^T K_{\rho_i}^{-1} \\ &\quad \left( 2 \frac{\partial K_{\rho_i}}{\partial \rho_i(e)} K_{\rho_i}^{-1} \frac{\partial K_{\rho_i}}{\partial \rho_i(\hat{e})} - \frac{\partial^2 K_{\rho_i}}{\partial \rho_i(e) \partial \rho_i(\hat{e})} \right) K_{\rho_i}^{-1} f\end{aligned}$$

Die Hessematrix des ersten Summanden der Zielfunktion  $f^T u_{\rho}$  ergibt sich zu der Blockmatrix:

$$H_S := \left( \begin{array}{c|ccc} \left( \frac{\partial}{\partial \rho_1(e)} \frac{\partial}{\partial \rho_1(\hat{e})} f^T u_{\rho} \right)_{e,\hat{e}} & \cdots & \left( \frac{\partial}{\partial \rho_1(e)} \frac{\partial}{\partial \rho_n(\hat{e})} f^T u_{\rho} \right)_{e,\hat{e}} \\ \vdots & \vdots & \vdots \\ \left( \frac{\partial}{\partial \rho_n(e)} \frac{\partial}{\partial \rho_1(\hat{e})} f^T u_{\rho} \right)_{e,\hat{e}} & \cdots & \left( \frac{\partial}{\partial \rho_n(e)} \frac{\partial}{\partial \rho_n(\hat{e})} f^T u_{\rho} \right)_{e,\hat{e}} \end{array} \right)$$

Hierbei läuft  $e$  und  $\hat{e}$  in allen  $n^2$  Blöcken jeweils in den Grenzen  $1 \leq e, \hat{e} \leq N$ .

**Fall  $p = 1$ :** Für  $p = 1$  fallen die gemischten Terme  $\frac{\partial^2 K_\rho}{\partial \rho_i(e) \partial \rho_j(\hat{e})}$  und  $\frac{\partial^2 K_{\rho_k}}{\partial \rho_i(e) \partial \rho_j(\hat{e})}$  in den partiellen Ableitungen weg. Beidseitige Multiplikation der Hessematrix  $H_S$  mit einem beliebigen Vektor  $v := (v_1^T, v_2^T, \dots, v_n^T)^T \neq 0$  ergibt mit (4.20):

$$\begin{aligned} v^T H_S v &= \sum_{i=1}^n \sum_{j=1}^n v_i^T \left( \frac{\partial}{\partial \rho_i(e)} \frac{\partial}{\partial \rho_j(\hat{e})} f^T u_\rho \right)_{1 \leq e, \hat{e} \leq N} v_j \\ &= \sum_{i=1}^n \sum_{j=1}^n \sum_{e=1}^N \sum_{\hat{e}=1}^N v_i(e) \left( \frac{\partial}{\partial \rho_i(e)} \frac{\partial}{\partial \rho_j(\hat{e})} f^T u_\rho \right) v_j(\hat{e}) \\ &= 2f^T K_\rho^{-1} \left( \sum_{i=1}^n \sum_{j=1}^n \sum_{e=1}^N \sum_{\hat{e}=1}^N v_i(e) K_e K_\rho^{-1} K_{\hat{e}} v_j(\hat{e}) \right) K_\rho^{-1} f \end{aligned}$$

Mit  $H_i := \sum_{e=1}^N v_i(e) K_e$  ergibt sich nun:

$$\begin{aligned} v^T H_S v &= 2f^T K_\rho^{-1} ( H_1 \quad H_2 \quad \dots \quad H_n ) \\ &\quad \begin{pmatrix} K_\rho^{-1} & K_\rho^{-1} & \dots & K_\rho^{-1} \\ K_\rho^{-1} & K_\rho^{-1} & \dots & K_\rho^{-1} \\ \vdots & \vdots & \vdots & \vdots \\ K_\rho^{-1} & K_\rho^{-1} & \dots & K_\rho^{-1} \end{pmatrix} \begin{pmatrix} H_1 \\ H_2 \\ \vdots \\ H_n \end{pmatrix} K_\rho^{-1} f \end{aligned}$$

Mit  $H := \begin{pmatrix} H_1 \\ H_2 \\ \vdots \\ H_n \end{pmatrix}$  ergibt sich nun wegen der positiven Definitheit von

$K_\rho^{-1}$ :

$$v^T H_S v = 2 \left( H K_\rho^{-1} f \right)^T \begin{pmatrix} K_\rho^{-1} & K_\rho^{-1} & \dots & K_\rho^{-1} \\ K_\rho^{-1} & K_\rho^{-1} & \dots & K_\rho^{-1} \\ \vdots & \vdots & \vdots & \vdots \\ K_\rho^{-1} & K_\rho^{-1} & \dots & K_\rho^{-1} \end{pmatrix} \left( H K_\rho^{-1} f \right) \geq 0$$

#### 4 Schadenstoleranz in der Topologieoptimierung

---

Für  $p = 1$  ist also der erste Summand der Zielfunktion  $f^T u_\rho$  konvex.

Analoges Vorgehen für die restlichen  $n$  Summanden liefert für den  $i$ -ten:

$$v^T H_{S_i} v = 2\alpha \left( HK_{\rho_i}^{-1} f \right)^T \begin{pmatrix} 0 & \dots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \dots & 0 \\ 0 & K_{\rho_i}^{-1} & 0 \\ 0 & \dots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \dots & 0 \end{pmatrix} \left( HK_{\rho_i}^{-1} f \right) \geq 0$$

Für  $p = 1$  sind also alle Summanden der Zielfunktion  $f^T u_{\rho_i}$  konvex und als Funktion von  $\rho_i$  der entsprechende Summand sogar strikt konvex.

Somit ist für  $p = 1$  die Zielfunktion (4.9) als Summe konvexer Funktionen selber konvex.

Weiterhin bildet  $\sum_{e=1}^N \rho(e)$  eine Hyperebene und somit beschreibt die Nebenbedingung (4.10) einen entsprechenden Halbraum. Die Nebenbedingungen (4.12)  $0 < \rho_{lb} \leq \rho_i \leq 1$  beschreiben einfache Boxbedingungen. Und die Bedingung (4.11)  $0 < \rho_{lb} \leq \rho \leq 1$  beschreibt als lineare Nebenbedingung ein konvexes Polyeder – sogar ein konvexes, beschränktes Polyeder.

Also bilden die Nebenbedingungen ein konvexes, beschränktes Polyeder.

Somit liegt für  $p = 1$  ein konvexes Optimierungsproblem vor.

Falls  $(\tilde{\rho}_1^T, \tilde{\rho}_2^T, \dots, \tilde{\rho}_n^T)^T$  eine Lösung des Optimierungsproblem (4.9) ist, so ist auf Grund der Symmetrie der Aufgabenstellung  $(\tilde{\rho}_2^T, \tilde{\rho}_1^T, \tilde{\rho}_3^T, \dots, \tilde{\rho}_n^T)^T$  und jede andere Permutation ebenfalls eine Lösung – hat den selben Zielfunktionswert und ist gleichsam zulässig.

Aus  $z(\tilde{\rho}_1, \tilde{\rho}_2, \tilde{\rho}_3, \dots, \tilde{\rho}_n) = z(\tilde{\rho}_2, \tilde{\rho}_1, \tilde{\rho}_3, \dots, \tilde{\rho}_n)$  folgt durch die Konvexität mit  $\rho_{\tilde{1},2} := \frac{1}{2}\tilde{\rho}_1 + \frac{1}{2}\tilde{\rho}_2$ :

$$z(\tilde{\rho}_1, \tilde{\rho}_2, \tilde{\rho}_3, \dots, \tilde{\rho}_n) = z(\rho_{\tilde{1},2}, \rho_{\tilde{1},2}, \tilde{\rho}_3, \dots, \tilde{\rho}_n)$$

Damit folgt für  $p = 1$  die Existenz einer Lösung

$$\left( \hat{\rho}_1^T, \hat{\rho}_2^T, \dots, \hat{\rho}_n^T \right)^T$$

mit der Gleichheit aller  $\hat{\rho}_1 = \hat{\rho}_2 = \dots = \hat{\rho}_i = \dots = \hat{\rho}_n$ , wobei

$$\hat{\rho}_1 = \frac{1}{n} \sum_{i=1}^n \tilde{\rho}_i.$$

Da dies keinerlei Redundanz bzgl. physikalischer Teilstrukturen liefern würde, ist eine derartige Lösung uninteressant. Daher ist also für die Redundanz ganz wesentlich  $p > 1$  zu fordern.

**Fall  $p > 1$ :** Für  $\rho_i = \rho_j$  folgt auch für  $p > 1$ :

$$\frac{\partial}{\partial \rho_i(e)} f^T u_{\rho_i} = \frac{\partial}{\partial \rho_j(e)} f^T u_{\rho_j}$$

Daher muss ein Gradientenverfahren bereits mit einem geeigneten, „unsymmetrischen“ Startwert loslaufen. Mit  $\mathbb{1}$  dem Vektor passender Dimension der nur aus den Einträgen 1 besteht, wäre beispielsweise ein Anfangswert:

$$\rho_i = \mathbb{1} \cdot \frac{i \cdot \frac{V}{N}}{\sum_{j=1}^n j} = \mathbb{1} \cdot \frac{2i \cdot \frac{V}{N}}{n(n+1)}$$

Im Fall  $p > 1$  ergibt sich bei beidseitiger Multiplikation der Hessematrix  $H_S$  des ersten Summanden der Zielfunktion  $f^T u_\rho$  mit dem Vektor  $v$

$$\begin{aligned} v^T H_S v &= \sum_{i=1}^n \sum_{j=1}^n v_i^T \left( \frac{\partial}{\partial \rho_i(e)} \frac{\partial}{\partial \rho_j(\hat{e})} f^T u_\rho \right)_{1 \leq e, \hat{e} \leq N} v_j \\ &= f^T K_\rho^{-1} \left( \sum_{i=1}^n \sum_{j=1}^n \sum_{e=1}^N \sum_{\hat{e}=1}^N v_i(e) S_{i,j,e,\hat{e}} v_j(\hat{e}) \right) K_\rho^{-1} f, \end{aligned}$$

wobei

$$S_{i,j,e,\hat{e}} := 2p^2 (\rho_i(e))^{p-1} K_e K_\rho^{-1} (\rho_i(\hat{e}))^{p-1} K_{\hat{e}} - \delta_{e,\hat{e}} p (p-1) (\rho_i(e))^{p-2} K_e.$$

#### 4 Schadenstoleranz in der Topologieoptimierung

---

Für die restlichen  $n$  Summanden ergibt sich für  $p > 1$  analog für den  $k$ -ten:

$$\begin{aligned}
 v^T H_{S_k} v &= \sum_{i=1}^n \sum_{j=1}^n v_i^T \left( \frac{\partial}{\partial \rho_i(e)} \frac{\partial}{\partial \rho_i(\hat{e})} f^T u_{\rho_k} \right)_{1 \leq e, \hat{e} \leq N} v_j \\
 &= f^T K_{\rho_k}^{-1} \left( \sum_{i=1}^n \sum_{j=1}^n \sum_{e=1}^N \sum_{\hat{e}=1}^N v_i(e) \delta_{i,k} \delta_{j,k} S_{i,j,e,\hat{e}} v_j(\hat{e}) \right) K_{\rho_k}^{-1} f \\
 &= f^T K_{\rho_k}^{-1} \left( \sum_{e=1}^N \sum_{\hat{e}=1}^N v_k(e) S_{i,j,e,\hat{e}} \right) v_k(\hat{e}) K_{\rho_k}^{-1} f
 \end{aligned}$$

Die gemischten Terme  $\frac{\partial^2 K_\rho}{\partial \rho_i(e) \partial \rho_j(\hat{e})}$  und  $\frac{\partial^2 K_{\rho_k}}{\partial \rho_i(e) \partial \rho_j(\hat{e})}$  fallen also nicht nur für  $p = 1$ , sondern auch für  $e \neq \hat{e}$  weg.

Ein Vergleich der Hessematrizen in den beiden Fällen  $p = 1$  und  $p > 1$  zeigt: Der Unterschied und damit die Nicht-Konvexität im Fall  $p > 1$  liegt nur an den paar negativen Summanden in  $S_{i,j,e,\hat{e}}$  mit  $\delta_{e,\hat{e}} = 1$ .

Falls also  $h_l \in \mathbb{R}^{N \times N}$ , mit

$$\forall 1 \leq l \leq n : h_l := 2p (\rho_l(e))^{p+1} K_e K_\rho^{-1} K_e - (p-1) \rho_l(e) K_e \geq 0,$$

positiv semidefinit ist, ist auch für  $p > 1$  die Zielfunktion konvex.

Offensichtlich ist also  $h_l$  für  $\rho_l(e) = 0$  positiv semidefinit.

Für einen beliebigen Vektor  $v \neq 0$  und  $\rho_l(e) > 0$  gilt mit  $N$  Verschiebungen  $u_\rho$ :

$$\begin{aligned}
 \frac{v^T h_l v}{2p \rho_l(e)} &= \sum_{k=1}^N \sum_{j=1}^N \left( \sum_{i_1=1}^N \sum_{i_2=1}^N \right. \\
 &\quad \left. (\rho_l(e))^p v_{i_1} (K_e)_{i_1,k} (K_\rho^{-1})_{k,j} v_{i_2} (K_e)_{j,i_2} - \frac{1 - \frac{1}{p}}{2N^2} v_k (K_e)_{k,j} v_j \right)
 \end{aligned}$$

Hierbei und im weiteren bedeutet  $(K_e)_{i_1,k}$  die Komponente der Matrix  $K_e$  in der  $i_1$ -ten Zeile und  $k$ -ten Spalte; analog für die anderen Matrizen.

Für einen ungünstigen Vektor  $v \neq 0$  mit  $\forall i \neq k : v_i = 0$ ,  $\rho_l(e) > 0$  und  $(K_e)_{k,k} \neq 0$  gilt:

$$\frac{2N^2 v^T h_l v}{2p \rho_l(e) v_k^2 (K_e)_{k,k}} = 2N^2 (\rho_l(e))^p (K_e)_{k,k} (K_\rho^{-1})_{k,k} + \frac{1}{p} - 1$$

Es ist daher offensichtlich<sup>20</sup>  $h_l$  nicht überall positiv semidefinit und somit die Zielfunktion nicht überall konvex. Man kann aber doch an sinnvollen Stellen ( $\forall e : \rho(e) \approx 0 \vee \rho(e) \approx 1$ ) eine lokal-konvexe Zielfunktion erwarten – also die negativen Summanden in  $S_{i,j,e,\hat{e}}$  überwiegen oder  $h_l$  positiv semidefinit ist. Dies hat sich bei numerischen Experimenten auch gezeigt.

Eine an [Sig01] angelehnte Implementierung – allerdings mit einer anderen Optimierungsvariante – als MATLAB-Funktion `toprm8.m` führte zu den Ergebnissen in Abb. 4.4 und 4.5. Eine Interpretation der Bilder aus Abb. 4.4 findet sich für  $n = 1$ ,  $n = 2$  und  $n = 3$  in Abb. 4.6.

Die unterschiedlichen Strukturen bei verschiedenen Auflösungen entstehen durch die gleiche Wahl von dem Verfahrensparameter  $r_{min}$  für die Gewichtung des Gradienten durch Nachbarelemente (vgl. Anhang C.1). Bei angepasster Wahl von  $r_{min}$  erhält man weiterhin eine Unabhängigkeit von der Diskretisierung.

Im Besonderen an den Bildern mit der feinen Diskretisierung (Abb. 4.5) kann man die Notwendigkeit für stärkere Verästelung bei höherer Redundanz sehen.

---

<sup>20</sup>Auch wenn  $M$  typischerweise sehr groß ist, könnte doch  $(K_\rho^{-1})_{k,k} = 0$  gelten und somit  $\frac{2M^2}{2p\rho_l(e)v_k^2(K_e)_{k,k}} = \frac{1}{p} - 1 < 0 \Rightarrow v^T h_l v < 0$ .

#### 4 Schadenstoleranz in der Topologieoptimierung

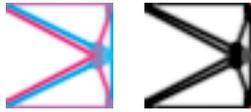
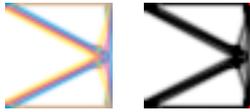
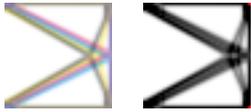
$n = 1$	$n = 2$	$n = 3$	$n = 4$
 53.3 3949.4535	 55.6 39435.1224	 73.6 178534.2013	 74.5 735142.3998
 178.6 11133.6406	 296.7 121791.1674	 341.3 826973.7125	 772.0 3302173.9609
 121.4 7351.3227	 144.8 70999.7563	 586.9 483720.0411	 703.9 1937229.7541

Abbildung 4.4: Verschiedene Strukturen mit zugehörigen Compliance-Werten mit unterschiedlichen Redundanzgraden von  $1 - \frac{1}{n}$  für die angegebenen  $n$ . Unter den Grafiken stehen jeweils 2 Zahlenwerte. Der obere gibt das Volumen  $\sum_{e=1}^N \rho(e)$  an und der untere den Compliance-Wert  $f^T u_\rho$ . Die Bilder der ersten Zeile haben einen einfachen Lastfall unten rechts mit einer Kraft nach unten. In der zweiten Bildzeile liegt ein 2-facher Lastfall unten rechts mit einer Kraft nach unten und oben rechts mit einer Kraft nach oben. Die letzte Bildzeile ist ebenfalls ein 2-facher Lastfall mit Kräften nach unten die unten rechts und unten in der Mitte wirken. In allen Beispielen ist die linke Seite gelagert.

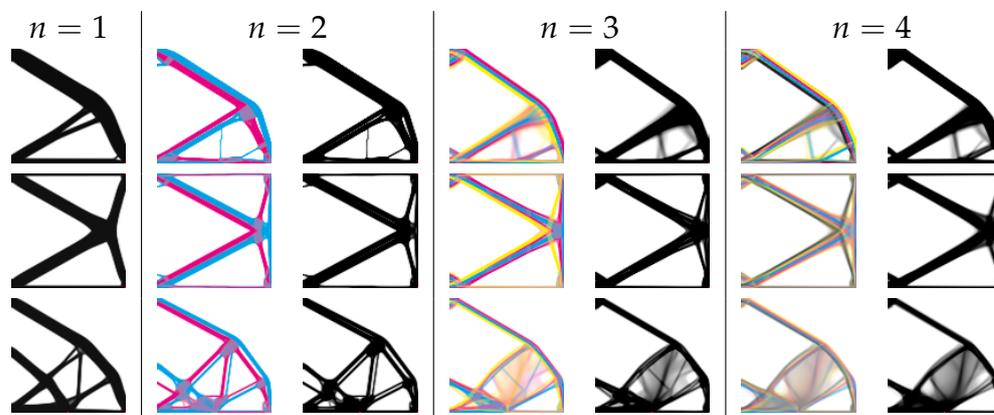


Abbildung 4.5: Verschiedene Strukturen mit unterschiedlichen Redundanzgraden von  $1 - \frac{1}{n}$  für die angegebenen  $n$ ; wegen der feinen Auflösung wurde nicht fertig ausiteriert. Die Bilder der ersten Zeile stellen einen einfachen Lastfall dar. Die beiden anderen Bildzeilen sind jeweils ein 2-facher Lastfall. In Abb. 4.4 werden die Lastfälle genau beschrieben.

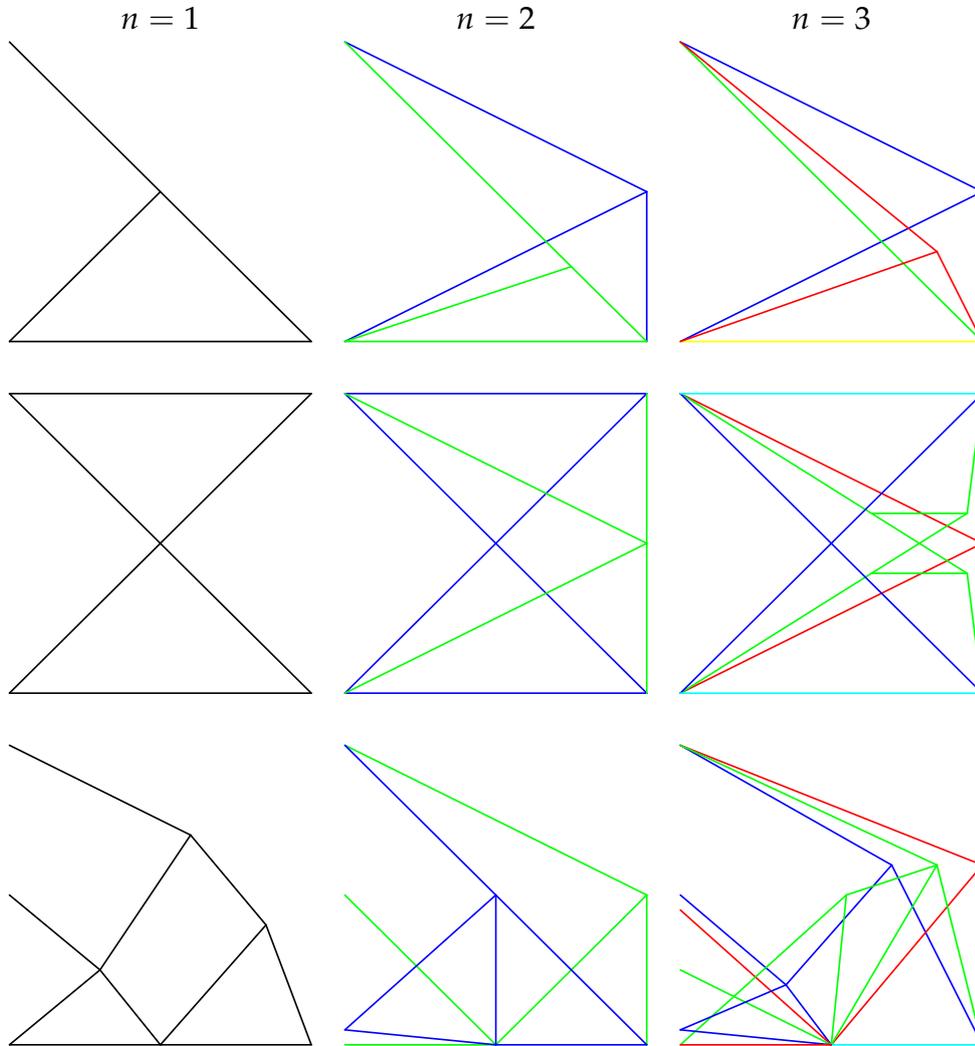


Abbildung 4.6: Einige Strukturen aus Abb. 4.4 mit unterschiedlichen Redundanzgraden wurden hier als Stabwerke interpretiert: Die Farben **Rot**, **Grün** und **Blau** kennzeichnen die einzelnen Teilstrukturen. Die Farbe **Gelb** gehört dementsprechend als Farbmischung zur **roten** und **grünen** Teilstruktur. Analog gehört **Türkis** zur **grünen** und **blauen** Teilstruktur (vgl. Anhang D).

### 4.1.3 Redundanz von $\frac{1}{n}$

Durch eine Aufteilung in  $n$  Teilstrukturen, von denen jeweils  $n - 1$  zusammen die Aufgabe bei reduzierter Last erfüllen sollen, kann eine geringere Redundanz als  $\frac{1}{2}$  von  $\frac{1}{n}$  erzielt werden. Hierbei ist  $n = 1$  natürlich nicht möglich, da dies eine Redundanz von 1 bedeuten würde. Daher muss  $n > 1$  gelten – natürlich erst ab  $n > 2$  gilt  $\frac{1}{n} < \frac{1}{2}$ , aber die folgende Modellierung ist ab  $n > 1$  gültig. Somit kann eine höhere Effizienz  $1 - \frac{1}{n}$  als in Kapitel 4.1.2 erzielt werden. Damit kommen wir zu folgender Aufgabenstellung:

$$\text{Zielfunktion: } z(\rho_1, \rho_2, \dots, \rho_n) = f^T u_\rho + \sum_{i=1}^n f^T u_{\bar{\rho}_i} \longrightarrow \min_{\rho_i} \quad (4.21)$$

$$\text{unter: } \|\rho\|_1 := \sum_{e=1}^N \rho(e) \leq V \quad (4.22)$$

$$0 < \rho_{lb} \leq \rho \leq 1 \quad (4.23)$$

$$\forall i \in \{1, 2, \dots, n\} : 0 < \rho_{lb} \leq \rho_i \leq 1 \quad (4.24)$$

$$\text{mit: } \rho := \sum_{i=1}^n \rho_i \quad (4.25)$$

$$\bar{\rho}_i := \sum_{j=1, j \neq i}^n \rho_j = \rho - \rho_i \quad (4.26)$$

$$K_\rho := \sum_{e=1}^N (\rho(e))^p K_e \quad (4.27)$$

$$\forall i \in \{1, 2, \dots, n\} : K_{\bar{\rho}_i} := \sum_{e=1}^N (\bar{\rho}_i(e))^p K_e \quad (4.28)$$

$$K_\rho u_\rho = f \quad (4.29)$$

$$\forall i \in \{1, 2, \dots, n\} : K_{\bar{\rho}_i} u_{\bar{\rho}_i} = \alpha f \quad (4.30)$$

Eigentlich sind die hier interessanten Teilstrukturen nun nicht mehr  $\rho_i$ , sondern  $\bar{\rho}_i$  und somit sollte  $0 < \rho_{lb} \leq \bar{\rho}_i \leq 1$  gelten. Aber dies wird durch die Nebenbedingung (4.24)  $0 < \rho_{lb} \leq \rho_i \leq 1$  automatisch erreicht – sogar noch mehr:

$$\bar{\rho}_i = \sum_{j=1, j \neq i}^n \rho_j \geq (n-1)\rho_{lb} \geq \rho_{lb}$$

$$\bar{\rho}_i = \rho - \rho_i \leq 1 - \rho_i \leq 1 - \rho_{lb} < 1$$

#### 4 Schadenstoleranz in der Topologieoptimierung

---

Also:

$$0 < \rho_{lb} \leq (n-1)\rho_{lb} \leq \bar{\rho}_i \leq 1 - \rho_{lb} < 1$$

Auf der anderen Seite ist die Forderung an  $\rho_i$  durchaus notwendig: Die obere Schranke  $\rho_i \leq 1$  ergibt sich unmittelbar aus  $\rho \leq 1$ . Die untere Schranke entstammt der Bedeutung von  $\rho_i$ :  $K_{\bar{\rho}_i} u_{\bar{\rho}_i} = \alpha f$  beschreibt das Halten/Funktionieren von  $\bar{\rho}_i$  und somit den Ausfall von  $\rho_i$ . Wenn  $\rho_i$  nun negativ wäre, dann würde dies einem Ausfall (ein Wegnehmen) von einer negativen relativen Dichte bedeuten; dies wiederum entspricht dem Hinzugeben einer positiven relativen Dichte – also eigentlich kein Ausfall/Schaden mehr!

Für  $n = 2$  ist dieser Ansatz identisch zu dem in Kapitel 4.1.2 und liefert daher gleichfalls eine Redundanz von  $\frac{1}{2}$ .

Um die Ableitungen bestimmen zu können, müssen durch Differentiation einzelne Terme berechnet werden:

$$\begin{aligned} \frac{\partial K_{\bar{\rho}_j}}{\partial \rho_i(e)} &= \frac{\partial}{\partial \rho_i(e)} \sum_{\hat{e}=1}^N (\rho(\hat{e}) - \rho_j(\hat{e}))^p K_{\hat{e}} \\ &= p \cdot (\rho(e) - \rho_j(e))^{p-1} \cdot (1 - \delta_{i,j}) K_e \\ &= p \cdot (\bar{\rho}_j(e))^{p-1} \cdot (1 - \delta_{i,j}) K_e \end{aligned}$$

Differentiation der Gleichung  $K_\rho u_\rho = f$  auf beiden Seiten nach  $\rho_i(e)$  liefert:

$$\begin{aligned} K_\rho u_\rho = f &\implies \left( \frac{\partial K_\rho}{\partial \rho_i(e)} \right) u_\rho + K_\rho \left( \frac{\partial u_\rho}{\partial \rho_i(e)} \right) = 0 \\ &\implies \frac{\partial u_\rho}{\partial \rho_i(e)} = -K_\rho^{-1} \left( \frac{\partial K_\rho}{\partial \rho_i(e)} \right) u_\rho \end{aligned} \quad (4.31)$$

Analog erhält man:

$$\frac{\partial u_{\bar{\rho}_j}}{\partial \rho_i(e)} = -K_{\bar{\rho}_j}^{-1} \left( \frac{\partial K_{\bar{\rho}_j}}{\partial \rho_i(e)} \right) u_{\bar{\rho}_j}$$

Beidseitiges Ableiten der Identitäten  $K_\rho^{-1} K_\rho = I$  bzw.  $K_{\bar{\rho}_j}^{-1} K_{\bar{\rho}_j} = I$  liefert:

$$\frac{\partial K_\rho^{-1}}{\partial \rho_i(e)} = -K_\rho^{-1} \frac{\partial K_\rho}{\partial \rho_i(e)} K_\rho^{-1}$$

$$\frac{\partial K_{\bar{\rho}_j}^{-1}}{\partial \rho_i(e)} = -K_{\bar{\rho}_j}^{-1} \frac{\partial K_{\bar{\rho}_j}}{\partial \rho_i(e)} K_{\bar{\rho}_j}^{-1}$$

Die ersten partiellen Ableitungen der Zielfunktion ergeben sich nun zu

$$\frac{\partial}{\partial \rho_i(e)} f^T u_\rho \stackrel{(4.31)}{=} -u_\rho^T \left( \frac{\partial K_\rho}{\partial \rho_i(e)} \right) u_\rho = -p \cdot (\rho(e))^{p-1} \cdot (u_\rho(e))^T \cdot K_e \cdot u_\rho(e)$$

und

$$\begin{aligned} \frac{\partial}{\partial \rho_i(e)} f^T u_{\bar{\rho}_j} &= -f^T K_{\bar{\rho}_j}^{-1} \left( \frac{\partial K_{\bar{\rho}_j}}{\partial \rho_i(e)} \right) u_{\bar{\rho}_j} = -\frac{1}{\alpha} u_{\bar{\rho}_j}^T \left( \frac{\partial K_{\bar{\rho}_j}}{\partial \rho_i(e)} \right) u_{\bar{\rho}_j} \\ &= -\frac{(1 - \delta_{i,j}) \cdot p}{\alpha} \cdot (\bar{\rho}_j(e))^{p-1} \cdot (u_{\bar{\rho}_j}(e))^T \cdot K_e \cdot u_{\bar{\rho}_j}(e). \end{aligned}$$

Die zweiten partiellen Ableitungen der Zielfunktion ergeben sich analog:

Für den ersten Summanden der Zielfunktion gilt mit  $u_\rho = K_\rho^{-1} f$ :

$$\frac{\partial}{\partial \rho_i(e)} \frac{\partial}{\partial \rho_j(\hat{e})} f^T u_\rho = f^T K_\rho^{-1} \left( 2 \frac{\partial K_\rho}{\partial \rho_i(e)} K_\rho^{-1} \frac{\partial K_\rho}{\partial \rho_j(\hat{e})} - \frac{\partial^2 K_\rho}{\partial \rho_i(e) \partial \rho_j(\hat{e})} \right) K_\rho^{-1} f$$

Für die restlichen Summanden der Zielfunktion ergibt sich analog:

$$\begin{aligned} \frac{\partial}{\partial \rho_i(e)} \frac{\partial}{\partial \rho_j(\hat{e})} f^T u_{\bar{\rho}_k} &= \frac{1}{\alpha} u_{\bar{\rho}_k}^T \left( 2 \frac{\partial K_{\bar{\rho}_k}}{\partial \rho_j(\hat{e})} K_{\bar{\rho}_k}^{-1} \frac{\partial K_{\bar{\rho}_k}}{\partial \rho_i(e)} - \frac{\partial^2 K_{\bar{\rho}_k}}{\partial \rho_i(e) \partial \rho_j(\hat{e})} \right) u_{\bar{\rho}_k} \\ &= \alpha (1 - \delta_{j,k}) (1 - \delta_{i,k}) f^T K_{\bar{\rho}_k}^{-1} \\ &\quad \left( 2 \frac{\partial K_{\bar{\rho}_k}}{\partial \rho_j(\hat{e})} K_{\bar{\rho}_k}^{-1} \frac{\partial K_{\bar{\rho}_k}}{\partial \rho_i(e)} - \frac{\partial^2 K_{\bar{\rho}_k}}{\partial \rho_i(e) \partial \rho_j(\hat{e})} \right) K_{\bar{\rho}_k}^{-1} f \end{aligned}$$

Die Hessematrix des ersten Summanden der Zielfunktion  $f^T u_\rho$  ergibt sich zu der Blockmatrix:

$$H_S := \left( \begin{array}{c|c|c} \left( \frac{\partial}{\partial \rho_1(e)} \frac{\partial}{\partial \rho_1(\hat{e})} f^T u_\rho \right)_{e,\hat{e}} & \cdots & \left( \frac{\partial}{\partial \rho_1(e)} \frac{\partial}{\partial \rho_n(\hat{e})} f^T u_\rho \right)_{e,\hat{e}} \\ \hline \vdots & \vdots & \vdots \\ \hline \left( \frac{\partial}{\partial \rho_n(e)} \frac{\partial}{\partial \rho_1(\hat{e})} f^T u_\rho \right)_{e,\hat{e}} & \cdots & \left( \frac{\partial}{\partial \rho_n(e)} \frac{\partial}{\partial \rho_n(\hat{e})} f^T u_\rho \right)_{e,\hat{e}} \end{array} \right)$$

#### 4 Schadenstoleranz in der Topologieoptimierung

---

**Fall  $p = 1$ :** Für  $p = 1$  fällt der gemischte Term  $\frac{\partial^2 K_\rho}{\partial \rho_i(e) \partial \rho_j(\hat{e})}$  in den partiellen Ableitungen weg. Beidseitige Multiplikation der Hessematrix  $H_S$  mit einem beliebigen Vektor  $v := (v_1^T, v_2^T, \dots, v_n^T)^T \neq 0$  ergibt:

$$\begin{aligned} v^T H_S v &= \sum_{i=1}^n \sum_{j=1}^n v_i^T \left( \frac{\partial}{\partial \rho_i(e)} \frac{\partial}{\partial \rho_j(\hat{e})} f^T u_\rho \right)_{1 \leq e, \hat{e} \leq N} v_j \\ &= \sum_{i=1}^n \sum_{j=1}^n \sum_{e=1}^N \sum_{\hat{e}=1}^N v_i(e) \frac{\partial}{\partial \rho_i(e)} \frac{\partial}{\partial \rho_j(\hat{e})} f^T u_\rho v_j(\hat{e}) \\ &= 2f^T K_\rho^{-1} \left( \sum_{i=1}^n \sum_{j=1}^n \sum_{e=1}^N \sum_{\hat{e}=1}^N v_i(e) K_e K_\rho^{-1} K_{\hat{e}} v_j(\hat{e}) \right) K_\rho^{-1} f \end{aligned}$$

Mit  $H_i := \sum_{e=1}^N v_i(e) K_e$  ergibt sich nun:

$$\begin{aligned} v^T H_S v &= 2f^T K_\rho^{-1} \begin{pmatrix} H_1 & H_2 & \dots & H_n \end{pmatrix} \\ &\quad \begin{pmatrix} K_\rho^{-1} & K_\rho^{-1} & \dots & K_\rho^{-1} \\ K_\rho^{-1} & K_\rho^{-1} & \dots & K_\rho^{-1} \\ \vdots & \vdots & \vdots & \vdots \\ K_\rho^{-1} & K_\rho^{-1} & \dots & K_\rho^{-1} \end{pmatrix} \begin{pmatrix} H_1 \\ H_2 \\ \vdots \\ H_n \end{pmatrix} K_\rho^{-1} f \end{aligned}$$

Aufgrund der positiven Definitheit von  $K_\rho^{-1}$  folgt mit  $H := \begin{pmatrix} H_1 \\ H_2 \\ \vdots \\ H_n \end{pmatrix}$ :

$$v^T H_S v = 2 \left( H K_\rho^{-1} f \right)^T \begin{pmatrix} K_\rho^{-1} & K_\rho^{-1} & \dots & K_\rho^{-1} \\ K_\rho^{-1} & K_\rho^{-1} & \dots & K_\rho^{-1} \\ \vdots & \vdots & \vdots & \vdots \\ K_\rho^{-1} & K_\rho^{-1} & \dots & K_\rho^{-1} \end{pmatrix} \left( H K_\rho^{-1} f \right) \geq 0$$

Für  $p = 1$  ist also der erste Summand der Zielfunktion  $f^T u_\rho$  konvex.

Analoges Vorgehen für die restlichen  $n$  Summanden liefert für den  $i$ -ten: (Dabei steht ★ für irgendwelche Einträge, die wegen der Multiplikation

mit 0 keine Rolle spielen.)

$$\begin{aligned}
 v^T H_{S_i} v &= 2\alpha f^T K_{\bar{\rho}_i}^{-1} ( H_1 \dots H_{i-1} \star H_{i+1} \dots H_n ) \cdot \\
 &\quad \begin{pmatrix} K_{\bar{\rho}_i}^{-1} & \dots & K_{\bar{\rho}_i}^{-1} & 0 & K_{\bar{\rho}_i}^{-1} & \dots & K_{\bar{\rho}_i}^{-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ K_{\bar{\rho}_i}^{-1} & \dots & K_{\bar{\rho}_i}^{-1} & 0 & K_{\bar{\rho}_i}^{-1} & \dots & K_{\bar{\rho}_i}^{-1} \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ K_{\bar{\rho}_i}^{-1} & \dots & K_{\bar{\rho}_i}^{-1} & 0 & K_{\bar{\rho}_i}^{-1} & \dots & K_{\bar{\rho}_i}^{-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ K_{\bar{\rho}_i}^{-1} & \dots & K_{\bar{\rho}_i}^{-1} & 0 & K_{\bar{\rho}_i}^{-1} & \dots & K_{\bar{\rho}_i}^{-1} \end{pmatrix} \begin{pmatrix} H_1 \\ \vdots \\ H_{i-1} \\ \star \\ H_{i+1} \\ \vdots \\ H_n \end{pmatrix} K_{\bar{\rho}_i}^{-1} f \\
 &= 2\alpha (HK_{\rho_i}^{-1} f)^T \begin{pmatrix} K_{\bar{\rho}_i}^{-1} & \dots & K_{\bar{\rho}_i}^{-1} & 0 & K_{\bar{\rho}_i}^{-1} & \dots & K_{\bar{\rho}_i}^{-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ K_{\bar{\rho}_i}^{-1} & \dots & K_{\bar{\rho}_i}^{-1} & 0 & K_{\bar{\rho}_i}^{-1} & \dots & K_{\bar{\rho}_i}^{-1} \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ K_{\bar{\rho}_i}^{-1} & \dots & K_{\bar{\rho}_i}^{-1} & 0 & K_{\bar{\rho}_i}^{-1} & \dots & K_{\bar{\rho}_i}^{-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ K_{\bar{\rho}_i}^{-1} & \dots & K_{\bar{\rho}_i}^{-1} & 0 & K_{\bar{\rho}_i}^{-1} & \dots & K_{\bar{\rho}_i}^{-1} \end{pmatrix} \cdot \\
 &\quad (HK_{\rho_i}^{-1} f) \\
 &\geq 0
 \end{aligned}$$

Für  $p = 1$  sind also alle Summanden der Zielfunktion  $f^T u_{\bar{\rho}_i}$  konvex.

Somit ist für  $p = 1$  die Zielfunktion als Summe konvexer Funktionen  $f^T u_{\rho}$  und  $f^T u_{\bar{\rho}_i}$  selber konvex.

Die Nebenbedingungen (4.22)-(4.24) bilden ein konvexes, beschränktes Polyeder.

Für  $p = 1$  liegt also ein konvexes Optimierungsproblem vor.

Falls  $(\tilde{\rho}_1^T, \tilde{\rho}_2^T, \dots, \tilde{\rho}_n^T)^T$  eine Lösung des Optimierungsproblem (4.21)-(4.30) ist, so ist auf Grund der Symmetrie der Aufgabenstellung  $(\tilde{\rho}_2^T, \tilde{\rho}_1^T, \tilde{\rho}_3^T, \dots, \tilde{\rho}_n^T)^T$  und jede andere Permutation ebenfalls eine Lösung – hat also den selben Zielfunktionswert und ist gleichsam zulässig.

Aus  $z(\tilde{\rho}_1, \tilde{\rho}_2, \tilde{\rho}_3, \dots, \tilde{\rho}_n) = z(\tilde{\rho}_2, \tilde{\rho}_1, \tilde{\rho}_3, \dots, \tilde{\rho}_n)$  folgt durch die Konvexität mit  $\rho_{1,2} := \frac{1}{2}\tilde{\rho}_1 + \frac{1}{2}\tilde{\rho}_2$ :

$$z(\tilde{\rho}_1, \tilde{\rho}_2, \tilde{\rho}_3, \dots, \tilde{\rho}_n) = z(\rho_{1,2}, \rho_{1,2}, \tilde{\rho}_3, \dots, \tilde{\rho}_n)$$

## 4 Schadenstoleranz in der Topologieoptimierung

---

Damit folgt für  $p = 1$  die Existenz einer Lösung

$$\left(\hat{\rho}_1^T, \hat{\rho}_2^T, \dots, \hat{\rho}_n^T\right)^T$$

mit der Gleichheit aller  $\hat{\rho}_1 = \hat{\rho}_2 = \dots = \hat{\rho}_n$ , wobei

$$\hat{\rho}_1 = \frac{1}{n} \sum_{i=1}^n \tilde{\rho}_i.$$

Da dies keinerlei Redundanz bzgl. physikalischer Teilstrukturen liefern würde, ist eine derartige Lösung uninteressant. Daher ist also für die Redundanz ganz wesentlich  $p > 1$  zu fordern.

**Fall  $p > 1$ :** Für  $\bar{\rho}_i = \bar{\rho}_j$  und  $i \neq k \neq j \neq l \neq i$  folgt

$$\frac{\partial}{\partial \rho_k(e)} f^T u_{\bar{\rho}_i} = \frac{\partial}{\partial \rho_l(e)} f^T u_{\bar{\rho}_j}$$

auch für  $p > 1$ .

Daher muss ein Gradientenverfahren bereits mit einem geeigneten, „unsymmetrischen“ Startwert loslaufen.

Beispielsweise:

$$\rho_i = \mathbb{1} \cdot \frac{i \cdot \frac{V}{N}}{\sum_{j=1}^n j} = \mathbb{1} \cdot \frac{2i \cdot \frac{V}{N}}{n(n+1)}$$

Im Fall  $p > 1$  ergibt sich allerdings keine so schöne Hessematrix wie in Kapitel 4.1.2. Die Konvexität ist hier viel stärker gestört. Ein Ersetzen der jetzigen Designvariablen  $\rho_i$  durch die eigentlich bereits verwendeten  $\bar{\rho}_i$  liefert auch für  $p > 1$  wieder eine diagonaldominante Hessematrix wie in Kapitel 4.1.2. Somit lässt sich das Problem (4.21)-(4.30) äquivalent wie folgt

formulieren:

$$\text{Zielfunktion: } z(\bar{\rho}_1, \bar{\rho}_2, \dots, \bar{\rho}_n) = f^T u_\rho + \sum_{i=1}^n f^T u_{\bar{\rho}_i} \longrightarrow \min_{\rho_i} \quad (4.32)$$

$$\text{unter: } \|\rho\|_1 = \frac{1}{n-1} \sum_{e=1}^N \left( \sum_{i=1}^n \bar{\rho}_i(e) \right) \leq V \quad (4.33)$$

$$0 < \rho_{lb} \leq \rho \leq 1 \quad (4.34)$$

$$\forall i \in \{1, 2, \dots, n\} : 0 < \rho_{lb} \leq \rho_i \quad (4.35)$$

$$\text{mit: } \rho := \frac{1}{n-1} \sum_{i=1}^n \bar{\rho}_i = \sum_{i=1}^n \rho_i \quad (4.36)$$

$$\rho_i := \rho - \bar{\rho}_i \quad (4.37)$$

$$K_\rho := \sum_{e=1}^N (\rho(e))^p K_e \quad (4.38)$$

$$\forall i \in \{1, 2, \dots, n\} : K_{\bar{\rho}_i} := \sum_{e=1}^N (\bar{\rho}_i(e))^p K_e \quad (4.39)$$

$$K_\rho u_\rho = f \quad (4.40)$$

$$\forall i \in \{1, 2, \dots, n\} : K_{\bar{\rho}_i} u_{\bar{\rho}_i} = \alpha f \quad (4.41)$$

Die Einzelstrukturen finden sich hinterher in  $\rho_i := \rho - \bar{\rho}_i$ . Da diese ausfallen können sollen, also einen Schaden bilden, müssen diese die Bedingung  $0 < \rho_{lb} \leq \rho_i \leq 1$  erfüllen; hierbei ist die rechte Seite  $\rho_i \leq 1$  implizit oben enthalten:

$$\rho_i = \rho - \sum_{j=1, j \neq i}^n \rho_j \leq 1 - \sum_{j=1, j \neq i}^n \rho_j \leq 1 - (n-1)\rho_{lb} < 1$$

Die ebenfalls gewünschte Boxbedingung  $0 < \bar{\rho}_i \leq 1$  ist ebenfalls bereits enthalten:

$$\bar{\rho}_i = \sum_{j=1, j \neq i}^n \rho_j \geq (n-1)\rho_{lb} \geq \rho_{lb} > 0$$

$$\bar{\rho}_i = \rho - \rho_i \leq 1 - \rho_i \leq 1 - \rho_{lb} < 1$$

Also:

$$0 < \rho_{lb} \leq (n-1)\rho_{lb} \leq \bar{\rho}_i \leq 1 - \rho_{lb} < 1 \quad (4.42)$$

#### 4 Schadenstoleranz in der Topologieoptimierung

---

Weiterhin folgt aus  $\rho_{lb} \leq \rho_i$  die schärfere Bedingung

$$0 < n\rho_{lb} \leq \sum_{i=1}^n \rho_i = \rho$$

als in (4.34) angegeben.

Daher sollte  $n\rho_{lb}$  klein genug sein.

Um die Ableitungen bestimmen zu können, müssen wiederum durch implizite Differentiation einzelne Terme berechnet werden:

Für die partiellen Ableitungen der Steifigkeitsmatrizen ergibt sich:

$$\frac{\partial K_{\bar{\rho}_j}}{\partial \bar{\rho}_i(e)} = \frac{\partial}{\partial \bar{\rho}_i(e)} \sum_{\hat{e}=1}^N (\bar{\rho}_j(\hat{e}))^p K_{\hat{e}} = \delta_{i,j} p \cdot (\bar{\rho}_i(e))^{p-1} K_e$$

$$\begin{aligned} \frac{\partial K_{\rho}}{\partial \bar{\rho}_i(e)} &= \frac{\partial}{\partial \bar{\rho}_i(e)} \sum_{\hat{e}=1}^N (\rho(\hat{e}))^p K_{\hat{e}} = \sum_{\hat{e}=1}^N p (\rho(\hat{e}))^{p-1} \frac{\delta_{e,\hat{e}}}{n-1} K_{\hat{e}} \\ &= \frac{p (\rho(e))^{p-1}}{n-1} K_e \end{aligned}$$

Weiterhin ergibt sich durch Ableiten der Gleichungen  $K_{\rho} u_{\rho} = f$  und  $K_{\bar{\rho}_j} u_{\bar{\rho}_j} = \alpha f$ :

$$\begin{aligned} \frac{\partial u_{\rho}}{\partial \bar{\rho}_i(e)} &= -K_{\rho}^{-1} \left( \frac{\partial K_{\rho}}{\partial \bar{\rho}_i(e)} \right) u_{\rho} \\ \frac{\partial u_{\bar{\rho}_j}}{\partial \bar{\rho}_i(e)} &= -\delta_{i,j} K_{\bar{\rho}_j}^{-1} \left( \frac{\partial K_{\bar{\rho}_j}}{\partial \bar{\rho}_i(e)} \right) u_{\bar{\rho}_j} \end{aligned}$$

Weiterhin ergibt sich durch Ableiten der Identitäten  $K_{\rho}^{-1} K_{\rho} = I$  und  $K_{\bar{\rho}_j}^{-1} K_{\bar{\rho}_j} = I$ :

$$\begin{aligned} \frac{\partial K_{\rho}^{-1}}{\partial \bar{\rho}_i(e)} &= -K_{\rho}^{-1} \frac{\partial K_{\rho}}{\partial \bar{\rho}_i(e)} K_{\rho}^{-1} \\ \frac{\partial K_{\bar{\rho}_j}^{-1}}{\partial \bar{\rho}_i(e)} &= -K_{\bar{\rho}_j}^{-1} \frac{\partial K_{\bar{\rho}_j}}{\partial \bar{\rho}_i(e)} K_{\bar{\rho}_j}^{-1} \end{aligned}$$

Damit ergeben sich die ersten partiellen Ableitungen der Zielfunktion zu:

$$\begin{aligned} \frac{\partial}{\partial \bar{\rho}_i(e)} f^T u_\rho &= -u_\rho^T \left( \frac{\partial K_\rho}{\partial \bar{\rho}_i(e)} \right) u_\rho \\ &= \frac{-p \cdot (\rho(e))^{p-1}}{n-1} \cdot u_{\rho,e}^T \cdot K_e \cdot u_{\rho,e} \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial \bar{\rho}_i(e)} f^T u_{\bar{\rho}_j} &= -\frac{1}{\alpha} u_{\bar{\rho}_j}^T \left( \frac{\partial K_{\bar{\rho}_j}}{\partial \bar{\rho}_i(e)} \right) u_{\bar{\rho}_j} \\ &= \frac{-\delta_{i,j} \cdot p \cdot (\bar{\rho}_j(e))^{p-1}}{\alpha} \cdot \left( u_{\bar{\rho}_j(e)} \right)^T \cdot K_e \cdot u_{\bar{\rho}_j(e)} \end{aligned}$$

Analog folgen die zweiten partiellen Ableitungen der Zielfunktion:

Für den ersten Summanden der Zielfunktion gilt:

$$\frac{\partial}{\partial \bar{\rho}_i(e)} \frac{\partial}{\partial \bar{\rho}_j(\hat{e})} f^T u_\rho = f^T K_\rho^{-1} \left( 2 \frac{\partial K_\rho}{\partial \bar{\rho}_i(e)} K_\rho^{-1} \frac{\partial K_\rho}{\partial \bar{\rho}_j(\hat{e})} - \frac{\delta_{e,\hat{e}} \partial^2 K_\rho}{\partial \bar{\rho}_i(e) \partial \bar{\rho}_j(\hat{e})} \right) K_\rho^{-1} f$$

Die restlichen Summanden der Zielfunktion sind dann:

$$\begin{aligned} \frac{\partial}{\partial \bar{\rho}_i(e)} \frac{\partial}{\partial \bar{\rho}_j(\hat{e})} f^T u_{\bar{\rho}_k} &= \frac{1}{\alpha} u_{\bar{\rho}_k}^T \left( 2 \frac{\partial K_{\bar{\rho}_k}}{\partial \bar{\rho}_j(\hat{e})} K_{\bar{\rho}_k}^{-1} \frac{\partial K_{\bar{\rho}_k}}{\partial \bar{\rho}_i(e)} - \frac{\partial^2 K_{\bar{\rho}_k}}{\partial \bar{\rho}_i(e) \partial \bar{\rho}_j(\hat{e})} \right) u_{\bar{\rho}_k} \\ &= \alpha \delta_{j,k} \delta_{i,k} f^T K_{\bar{\rho}_k}^{-1} \\ &\quad \left( 2 \frac{\partial K_{\bar{\rho}_k}}{\partial \bar{\rho}_j(\hat{e})} K_{\bar{\rho}_k}^{-1} \frac{\partial K_{\bar{\rho}_k}}{\partial \bar{\rho}_i(e)} - \frac{\partial^2 K_{\bar{\rho}_k}}{\partial \bar{\rho}_i(e) \partial \bar{\rho}_j(\hat{e})} \right) K_{\bar{\rho}_k}^{-1} f \end{aligned}$$

Die Hessematrix des ersten Summanden der Zielfunktion  $f^T u_\rho$  ergibt sich zu der Blockmatrix:

$$H_S := \left( \begin{array}{c|c|c} \left( \frac{\partial}{\partial \bar{\rho}_1(e)} \frac{\partial}{\partial \bar{\rho}_1(\hat{e})} f^T u_\rho \right)_{e,\hat{e}} & \cdots & \left( \frac{\partial}{\partial \bar{\rho}_1(e)} \frac{\partial}{\partial \bar{\rho}_n(\hat{e})} f^T u_\rho \right)_{e,\hat{e}} \\ \vdots & \vdots & \vdots \\ \left( \frac{\partial}{\partial \bar{\rho}_n(e)} \frac{\partial}{\partial \bar{\rho}_1(\hat{e})} f^T u_\rho \right)_{e,\hat{e}} & \cdots & \left( \frac{\partial}{\partial \bar{\rho}_n(e)} \frac{\partial}{\partial \bar{\rho}_n(\hat{e})} f^T u_\rho \right)_{e,\hat{e}} \end{array} \right)$$

#### 4 Schadenstoleranz in der Topologieoptimierung

---

Der gemischte Term  $\frac{\partial^2 K_\rho}{\partial \bar{\rho}_i(e) \partial \bar{\rho}_j(\hat{e})}$  fällt für  $p = 1$  wiederum in den partiellen Ableitungen weg. Beidseitige Multiplikation der Hessematrix  $H_S$  mit einem beliebigen Vektor  $v := (v_1^T, v_2^T, \dots, v_n^T)^T \neq 0$  ergibt:

$$\begin{aligned} v^T H_S v &= \sum_{i=1}^n \sum_{j=1}^n v_i^T \left( \frac{\partial}{\partial \bar{\rho}_i(e)} \frac{\partial}{\partial \bar{\rho}_j(\hat{e})} f^T u_\rho \right)_{1 \leq e, \hat{e} \leq N} v_j \\ &= \sum_{i=1}^n \sum_{j=1}^n \sum_{e=1}^N \sum_{\hat{e}=1}^N v_i(e) \frac{\partial}{\partial \bar{\rho}_i(e)} \frac{\partial}{\partial \bar{\rho}_j(\hat{e})} f^T u_\rho v_j(\hat{e}) \\ &= \frac{2}{(n-1)^2} f^T K_\rho^{-1} \left( \sum_{i=1}^n \sum_{j=1}^n \sum_{e=1}^N \sum_{\hat{e}=1}^N v_i(e) K_e K_\rho^{-1} K_{\hat{e}} v_j(\hat{e}) \right) K_\rho^{-1} f \end{aligned}$$

Mit  $H_i := \sum_{e=1}^N v_i(e) K_e$  ergibt sich nun:

$$\begin{aligned} v^T H_S v &= \frac{2}{(n-1)^2} f^T K_\rho^{-1} ( H_1 \ H_2 \ \dots \ H_n ) \\ &\quad \begin{pmatrix} K_\rho^{-1} & K_\rho^{-1} & \dots & K_\rho^{-1} \\ K_\rho^{-1} & K_\rho^{-1} & \dots & K_\rho^{-1} \\ \vdots & \vdots & \vdots & \vdots \\ K_\rho^{-1} & K_\rho^{-1} & \dots & K_\rho^{-1} \end{pmatrix} \begin{pmatrix} H_1 \\ H_2 \\ \vdots \\ H_n \end{pmatrix} K_\rho^{-1} f \end{aligned}$$

Mit  $H := \begin{pmatrix} H_1 \\ H_2 \\ \vdots \\ H_n \end{pmatrix}$  ergibt sich nun wegen der positiven Definitheit von  $K_\rho^{-1}$ :

$$\begin{aligned} v^T H_S v &= \frac{2}{(n-1)^2} (HK_\rho^{-1}f)^T \begin{pmatrix} K_\rho^{-1} & K_\rho^{-1} & \dots & K_\rho^{-1} \\ K_\rho^{-1} & K_\rho^{-1} & \dots & K_\rho^{-1} \\ \vdots & \vdots & \vdots & \vdots \\ K_\rho^{-1} & K_\rho^{-1} & \dots & K_\rho^{-1} \end{pmatrix} (HK_\rho^{-1}f) \\ &\geq 0 \end{aligned}$$

Für  $p = 1$  ist also der erste Summand der Zielfunktion  $f^T u_\rho$  konvex.

Für die restlichen  $n$  Summanden liefert ein analoges Vorgehen für den  $i$ -ten: (Dabei steht  $\star$  für irgendwelche Einträge, die wegen der Multiplikation mit 0 keine Rolle spielen.)

$$\begin{aligned}
 v^T H_{S_i} v &= 2\alpha f^T K_{\bar{\rho}_i}^{-1} ( \star \dots \star H_i \star \dots \star ) \cdot \\
 &\quad \begin{pmatrix} 0 & \dots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \dots & 0 \\ 0 & K_{\bar{\rho}_i}^{-1} & 0 \\ 0 & \dots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} \star \\ \vdots \\ \star \\ H_i \\ \star \\ \vdots \\ \star \end{pmatrix} K_{\bar{\rho}_i}^{-1} f \\
 &= 2\alpha (HK_{\rho_i}^{-1} f)^T \begin{pmatrix} 0 & \dots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \dots & 0 \\ 0 & K_{\bar{\rho}_i}^{-1} & 0 \\ 0 & \dots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \dots & 0 \end{pmatrix} (HK_{\rho_i}^{-1} f) \\
 &\geq 0
 \end{aligned}$$

Für  $p = 1$  sind also alle Summanden der Zielfunktion  $f^T u_{\bar{\rho}_i}$  konvex, und die Zielfunktion als Summe konvexer Funktionen  $f^T u_{\rho}$  und  $f^T u_{\bar{\rho}_i}$  selber konvex.

Weiterhin bildet

$$\frac{1}{n-1} \sum_{e=1}^N \left( \sum_{i=1}^n \bar{\rho}_i(e) \right) = V$$

eine Hyperebene und somit beschreibt die Nebenbedingung (4.33) in der Form

$$\sum_{e=1}^N \left( \sum_{i=1}^n \bar{\rho}_i(e) \right) \leq (n-1)V$$

einen entsprechenden Halbraum. Die Nebenbedingung (4.34) (bzw. Boxbedingung für  $\rho$ ) bildet zusammen mit (4.35) in der Form (4.42) einfache

#### 4 Schadenstoleranz in der Topologieoptimierung

---

Boxbedingungen für die Designvariablen  $\bar{\rho}_i$ . Alleine bildet die Nebenbedingung (4.34) in der Form

$$\rho_{lb} \leq \sum_{i=1}^n \bar{\rho}_i \leq n - 1$$

als lineare Nebenbedingung für die  $\bar{\rho}_i$  ein konvexes Polyeder.

Also bilden die Nebenbedingungen ein konvexes, beschränktes Polyeder.

Für  $p = 1$  liegt also ein konvexes Optimierungsproblem vor.

Falls  $(\tilde{\rho}_1^T, \tilde{\rho}_2^T, \dots, \tilde{\rho}_n^T)^T$  eine Lösung des Optimierungsproblem (4.32)-(4.41) ist, so ist auf Grund der Symmetrie der Aufgabenstellung  $(\tilde{\rho}_2^T, \tilde{\rho}_1^T, \tilde{\rho}_3^T, \dots, \tilde{\rho}_n^T)^T$  und jede andere Permutation ebenfalls eine Lösung – hat also den selben Zielfunktionswert und ist gleichsam zulässig.

Aus  $z(\tilde{\rho}_1, \tilde{\rho}_2, \tilde{\rho}_3, \dots, \tilde{\rho}_n) = z(\tilde{\rho}_2, \tilde{\rho}_1, \tilde{\rho}_3, \dots, \tilde{\rho}_n)$  folgt durch die Konvexität mit  $\rho_{1,2} := \frac{1}{2}\tilde{\rho}_1 + \frac{1}{2}\tilde{\rho}_2$ :

$$z(\tilde{\rho}_1, \tilde{\rho}_2, \tilde{\rho}_3, \dots, \tilde{\rho}_n) = z(\rho_{1,2}, \rho_{1,2}, \tilde{\rho}_3, \dots, \tilde{\rho}_n)$$

Damit folgt für  $p = 1$  die Existenz einer Lösung

$$\left(\hat{\rho}_1^T, \hat{\rho}_2^T, \dots, \hat{\rho}_n^T\right)^T$$

mit der Gleichheit aller  $\hat{\rho}_1 = \hat{\rho}_2 = \dots = \hat{\rho}_n$ , wobei

$$\hat{\rho}_1 = \frac{1}{n} \sum_{i=1}^n \tilde{\rho}_i.$$

Da dies keinerlei Redundanz bzgl. physikalischer Teilstrukturen liefern würde, ist eine derartige Lösung uninteressant. Daher ist also für die Redundanz ganz wesentlich  $p > 1$  zu fordern.

Für  $\bar{\rho}_i = \bar{\rho}_j$  und  $i \neq k \neq j \neq l \neq i$  folgt auch für  $p > 1$ :

$$\frac{\partial}{\partial \rho_k(e)} f^T u_{\bar{\rho}_i} = \frac{\partial}{\partial \rho_l(e)} f^T u_{\bar{\rho}_j}$$

Daher muss ein Gradientenverfahren abermals bereits mit einem geeigneten, „unsymmetrischen“ Startwert loslaufen. Beispielsweise:

$$\rho_i = \mathbb{1} \cdot \frac{i \cdot \frac{V}{N}}{\sum_{j=1}^n j} = \mathbb{1} \cdot \frac{2i \cdot \frac{V}{N}}{n(n+1)}$$

Das Problem (4.32)-(4.41) hat nun die gleiche Struktur der Hessematrix wie das Problem (4.9)-(4.17). Damit ist das Problem (4.32)-(4.41) für eine praktische Rechnung dem Problem (4.21)-(4.30) überlegen und somit vorzuziehen.

Eine Implementierung als MATLAB-Funktion führte zu Ergebnissen, die exemplarisch in Abb. 4.7 dargestellt sind. Es wurde nicht nur die Gesamtstruktur  $\rho$  mit einer Einfärbung der einzelnen Teilstrukturen  $\rho_i$  abgebildet, sondern auch das, was bei einem Ausfall einer Teilstruktur übrigbleibt – nämlich  $\bar{\rho}_i$ . Die Teilstrukturen  $\rho_i$  sind durch die einzelnen Farben repräsentiert.

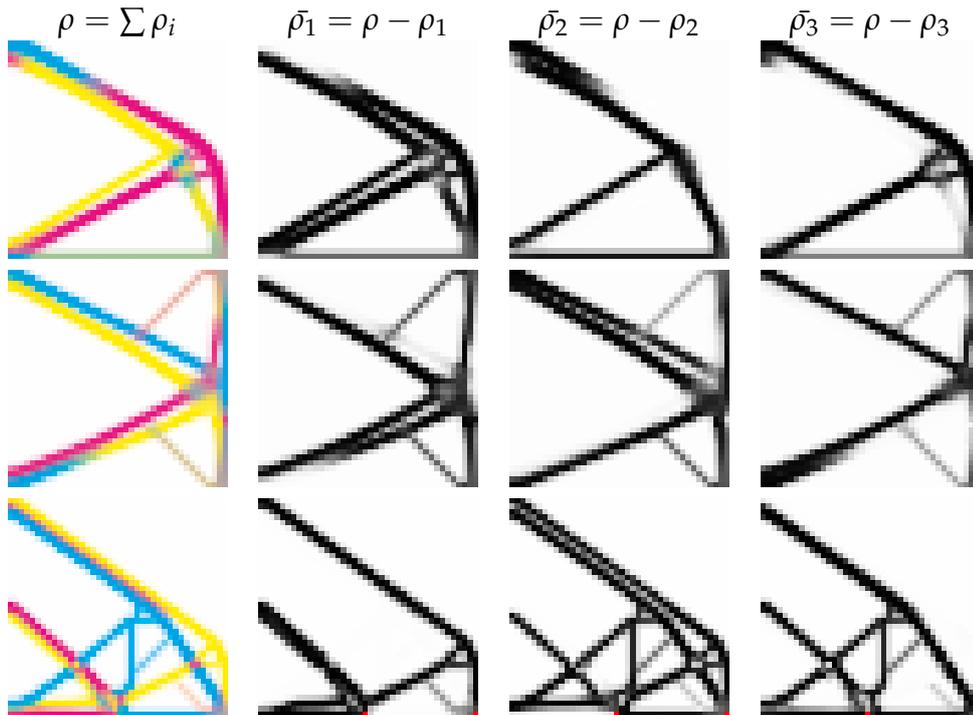


Abbildung 4.7: Verschiedene Strukturen mit einem Redundanzgrad von  $\frac{1}{3}$ . Es wurde nicht nur die Gesamtstruktur  $\rho$  mit einer Einfärbung der einzelnen Teilstrukturen  $\rho_i$  abgebildet, sondern auch das, was bei einem Ausfall einer Teilstruktur übrigbleibt:  $\bar{\rho}_i$ . In allen Beispielen ist die linke Seite gelagert. Die Bilder der ersten Zeile haben einen einfachen Lastfall unten rechts mit einer Kraft nach unten. In der zweiten Bildzeile liegt ein 2-facher Lastfall unten rechts mit einer Kraft nach unten und oben rechts mit einer Kraft nach oben. Die letzte Bildzeile ist ebenfalls ein 2-facher Lastfall mit Kräften nach unten die unten rechts und unten in der Mitte wirken. Blau repräsentiert  $\rho_1$ , rot  $\rho_2$  und gelb  $\rho_3$ .

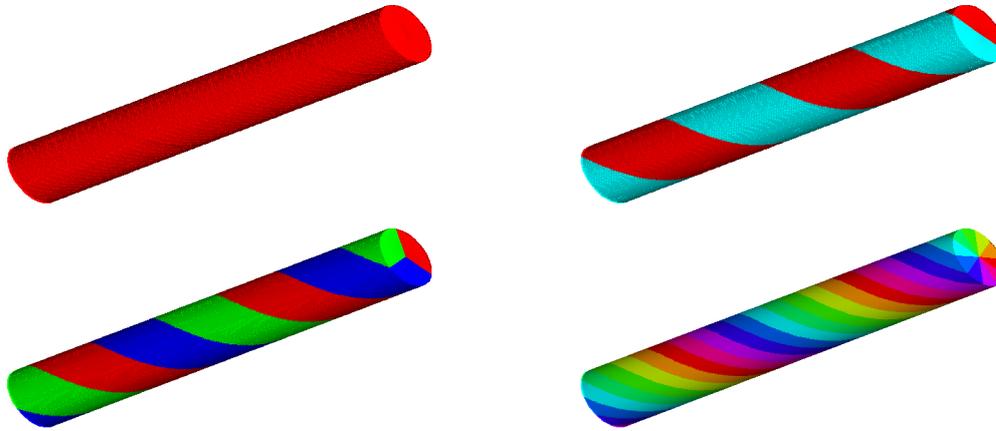


Abbildung 4.8: Gedankenexperiment: Keine Trennung der Einzelstrukturen

### 4.1.4 Ergebnisse

Die Ergebnisse aus den Abbildungen 4.4, 4.5 und 4.7 zeigen deutlich das Problem dieses Ansatzes hier im Kapitel 4.1. Die Strukturen kleben zusammen und ein Ausfall eines Bauteils (z. B. Stab) sorgt unter Umständen für den Ausfall mehrerer Teilstrukturen. Dieses Phänomen ist besonders bei der Redundanz von  $\frac{1}{n}$  in Abb. 4.7 deutlich.

Noch deutlicher erkennt man dies in Abb. 4.8. Jede Farbe repräsentiert hier eine Teilstruktur; die Anzahl der Teilstrukturen/Farben kann beliebig gesteigert werden. Der 3-dimensionale Stab ist somit redundant bzgl. des Ausfalls jeder als Farbe repräsentierten Teilstruktur. Je nach Auslegung könnte entweder jede einzelne eingefärbte Teilstruktur die Aufgabe des Stabes übernehmen, oder aber jede einzelne Teilstruktur, repräsentiert durch eine Farbe, darf ausfallen und die restliche Struktur hält immer noch. Eine Trennung der Teilstrukturen ist räumlich nicht mehr gegeben. Auf Grund der ineinander verdrehten Teilstrukturen kann nun praktisch an jeder Stelle auch eine weitere Struktur passend zu einer Farbe angeschlossen werden. Man kann sich leicht vorstellen, dass mit diesem Grundkonzept auch komplexere Strukturen aufgebaut werden können und weiterhin keinerlei räumliche Trennung der einzelnen Teilstrukturen existiert. Damit wurde das Ziel also nicht erreicht.

### 4.2 Schadenstoleranz durch Redundanz im Fachwerk

Im Kapitel 4.1 wurde versucht, auf Grundlage des SIMP-Ansatzes Redundanz (vgl. Kapitel 3) als Forderung zu modellieren. Da der SIMP-Ansatz von einer kontinuierlichen Struktur in seiner Modellierung ausgeht, scheiterte dieses Vorgehen in gewisser Weise. Dieses Problem des „Zusammenklebens“ wurde in 4.1.4 dargelegt, was damit dieses Kapitel 4.2 motiviert.

Hier wird nun eine diskrete Überdeckung des Bauraums mittels Stäben (Grundstruktur) zu Grunde gelegt und diese Modellierung führt unter Umständen zu einem linearen Programm. In Kapitel 4.2.1 wird die Idee der Redundanz in der Topologieoptimierung aus Kapitel 4.1 in nachvollziehbarer Art und Weise auf die Topologieoptimierung im Fachwerk übertragen. Damit kann in Kapitel 4.2.2 die Redundanz von  $1 - \frac{1}{n}$  und somit eine geringe Effizienz von  $\frac{1}{n}$  als lineares Optimierungsproblem erzielt werden. Eine höhere Effizienz wird in Kapitel 4.2.3 modelliert.

In beiden Fällen ergibt sich ein hochdimensionales gemischt-ganzzahliges lineares Programm. Für diese Art von Problemen kann durch den Greedy-Algorithmus in Kapitel 4.2.4 effizient eine approximative Lösung gefunden werden; es müssen nämlich nur einmal von der Wurzel bis zu einem Blatt  $nN$  Optimierungsprobleme betrachtet werden, statt den  $2^{nN}$  Optimierungsproblemen in den Blättern des binären Suchbaums. Diese approximative Lösung kann als Oberschranke und somit zur Reduzierung des Suchbaums verwendet werden.<sup>21</sup>

Dadurch berechnete, nicht-approximierte Lösungen zu verschiedenen Beispielen werden in Kapitel 4.2.5 vorgestellt und beleuchtet.

#### 4.2.1 Idee

Das Vorgehen aus Kapitel 4.1.2 basierend auf der Idee, die in Kapitel 4.1.1 erklärt wurde, kann nun auch auf die Topologieoptimierung durch den Fachwerksansatz nach Anhang C.3.2 (vgl. [DGG64], [OAH96], [BS04])

---

<sup>21</sup>Die Greedy-Lösung stellt natürlich auch einen zulässigen Punkt dar und kann daher als speziellen Startwert für einen Simplex-Algorithmus gewählt werden. Dies spielt aber für einen Branch-and-Bound-Algorithmus keine Rolle, da dieser für ein spezielles relaxiertes Problem eine Startlösung benötigt, und dies bietet die Greedy-Lösung nicht zwangsweise. Das Vorgehen des Greedy-Algorithmus sollte daher direkt als Tiefensuche in ein Branch-and-Bound-Vorgehen eingebettet werden.

übertragen werden. Dabei ermöglicht dieser Ansatz einen relativ geringen Berechnungsaufwand, der letztlich trotzdem noch sehr groß ist.

Hier wird der Bauraum durch die Grundstruktur [DGG64, ground structure] aller  $N$  potentieller Stäbe überdeckt. Diese Gesamtstruktur wird nun in  $n$  Teilstrukturen  $s_i$  zerlegt;  $s_i \in \mathbb{R}^N$  stellt den Vektor der Stabquerschnitte für die  $i$ -te Teilstruktur dar. Für jede dieser Teilstrukturen muss die Gleichgewichtsbedingung „ $Cw = f$ “ nach der Matrix-Kraft-Methode (vgl. [Prz68, matrix force method]) gelten; damit ergeben sich die Belastungen  $w_i \in \mathbb{R}^N$  für jede Teilstruktur.

Die bekannte Topologieoptimierung durch den Fachwerksansatz mit Volumensminimierung (vgl. Anhang C.3.2) ist ein rein lineares Optimierungsproblem. Die Aufteilung in Teilstrukturen wurde in Kapitel 4.1.2 durch die nichtlineare Zielfunktion (vgl. Kapitel 4.1.1) des SIMP-Ansatzes erzielt. Hier müssen, nun in Anlehnung an die  $\rho_i$ , künstliche Variablen  $x_i \in [0;1]^N$  eingeführt werden, die eine Aufteilung jedes einzelnen Stabes zu einer Teilstruktur beschreiben. Die auch hier notwendige Nichtlinearität für die gewünschte Partitionierung wird wiederum durch eine Penalty-Potenz erreicht.

Dies führt nun direkt auf folgende Aufgabenstellung, die allerdings nicht mehr linear ist:

$$\text{Zielfunktion: } V(x_1, x_2, \dots, x_n, s) = \sum_{i=1}^n s_i l^T x_i \rightarrow \min_{x_i, w_i, s_i} \quad (4.43)$$

$$\text{unter: } \forall i = 1, \dots, n : \quad Cw_i = \alpha f \quad (4.44)$$

$$\forall j = 1, \dots, N : \quad \sigma_{\min} s_i \left(x_i^{(j)}\right)^p \leq w_i^{(j)} \leq \sigma_{\max} s_i \left(x_i^{(j)}\right)^p \quad (4.45)$$

$$0 \leq x_i \leq 1 \quad (4.46)$$

$$0 \leq \sum_{i=1}^n x_i \leq 1 \quad (4.47)$$

$$Cw = f \quad (4.48)$$

$$\sigma_{\min} \sum_{i=1}^n s_i \left(x_i^{(j)}\right)^p \leq w \leq \sigma_{\max} \sum_{i=1}^n s_i \left(x_i^{(j)}\right)^p \quad (4.49)$$

Zur Übersicht finden sich in folgender Tabelle die Bedeutungen der einzelnen Variablen – sowohl die Daten des Anwenders ( $n, N, m, l, C, f, \sigma_{\min}, \sigma_{\max}, p, \alpha$ ), als auch die Modellierungsvariablen ( $V, x_i, w_i, w, s_i$ ):

#### 4 Schadenstoleranz in der Topologieoptimierung

---

$n \in \mathbb{N}$	Anzahl Teilstrukturen
$N \in \mathbb{N}$	Anzahl Stäbe (finiter Elemente)
$m \in \mathbb{N}$	Anzahl Freiheitsgrade (Knotenzahl * Raumdimension)
$l \in \mathbb{R}^N$	Stablängen
$C \in \mathbb{R}^{m \times N}$	Geometriematrix / Gleichgewichtsmatrix (vgl. Anhang C.3.3)
$f \in \mathbb{R}^m$	vordefinierte Last
$\sigma_{\min} \in \mathbb{R}^-$	physikalische Unterschranke (z. B. Fließgrenze)
$\sigma_{\max} \in \mathbb{R}^+$	physikalische Oberschranke (z. B. Fließgrenze)
$p$	Penalty-Potenz
$\alpha \in ]0, 1] \subset \mathbb{R}$	Reduzierung der Last
$V = V(x_1, x_2, \dots, x_n, s)$	Volumen als reellwertige Zielfunktion
$x_i \in [0; 1]^N$	Stabauswahl
$w_i, w \in \mathbb{R}^N$	Reaktionskräfte / Belastungen der einzelnen Stäbe
$s_i \in \mathbb{R}^{+N}$	Stabquerschnitte der $i$ -ten Teilstruktur, Designvariablen

Die Reduzierung der Last durch den Parameter  $\alpha$  ändert nicht die Topologie: Division der Gleichung (4.44) durch  $\alpha$  liefert

$$C\hat{w}_i = f$$

mit  $\hat{w}_i = \frac{w_i}{\alpha}$ . Wegen (4.45) gilt für  $\hat{w}_i$

$$\sigma_{\min} \frac{s_i}{\alpha} \left(x_i^{(j)}\right)^p \leq \hat{w}_i^{(j)} \leq \sigma_{\max} \frac{s_i}{\alpha} \left(x_i^{(j)}\right)^p.$$

Also werden bei einer Reduzierung der Last die Stabquerschnitte  $\frac{s_i}{\alpha}$ , die die Last  $f$  tragen, lediglich zu kleineren Stabquerschnitten  $s_i$ . Dies ändert aber nicht die Topologie.

Weiterhin liefert die Summation der Gleichung (4.44)

$$\sum_{i=1}^n Cw_i = C \sum_{i=1}^n w_i = n\alpha f.$$

Somit hält ohnehin die Gesamtstruktur das  $n$ -fache einer Teilstruktur. Erst eine extreme Reduzierung der Last durch  $\alpha < \frac{1}{n}$  würde somit ein anderes

Ergebnis liefert. Für  $\alpha \geq \frac{1}{n}$  sind daher die Nebenbedingungen (4.48) und (4.49) überflüssig.

Da also im Besonderen die Topologie unabhängig von dem Parameter  $\alpha$  ist, wird dieser im Folgenden nicht mehr betrachtet oder implizit  $\alpha = 1$  festgelegt.

Falls  $\tilde{a}$  eine Lösung von der Standardaufgabenstellung (C.7) im Anhang C.3.2 ist, so ist  $\tilde{x}_1 = \frac{1}{\tilde{s}_1} \tilde{a}$  mit  $\tilde{s}_1 = \max \{\tilde{a}\}$  für  $n = 1$  eine Lösung von (4.43)-(4.49). Allerdings wurde bei dieser Formulierung (4.43) der Redundanz die Linearität von (C.7) zunichte gemacht.

#### 4.2.2 Fachwerk mit Redundanz $1 - \frac{1}{n}$ als Lineares Programm

Die Linearität kann erhalten bleiben, wenn man eine Ganzzahligkeitsbedingung  $x_i \in \{0;1\}^N$  einführt:

$$\text{Zielfunktion: } V(s_1, s_2, \dots, s_n) = \sum_{i=1}^n l^T s_i \rightarrow \min_{x_i, w_i, s_i} \quad (4.50)$$

$$\text{unter: } \forall i = 1, \dots, n :$$

$$C w_i = f \quad (4.51)$$

$$\sigma_{\min} s_{\max} x_i \leq w_i \leq \sigma_{\max} s_{\max} x_i \quad (4.52)$$

$$\sigma_{\min} s_i \leq w_i \leq \sigma_{\max} s_i \quad (4.53)$$

$$0 \leq \sum_{i=1}^n x_i \leq 1 \quad (4.54)$$

$$x_i \in \{0;1\}^N \quad (4.55)$$

Weiterhin wird nun ein maximaler Stabquerschnitt  $s_{\max}$  benötigt. Dieser kann aber in der Praxis so groß gewählt werden, dass dieser Wert keine Einschränkung vorgibt.

In folgender Tabelle finden sich die Bedeutungen der einzelnen Variablen – sowohl die Daten des Anwenders ( $n, N, m, l, C, f, \sigma_{\min}, \sigma_{\max}, s_{\max}$ ), als auch die Modellierungsvariablen ( $V, x_i, w_i, s_i$ ):

$n \in \mathbb{N}$	Anzahl Teilstrukturen
$N \in \mathbb{N}$	Anzahl Stäbe (finiter Elemente)
$m \in \mathbb{N}$	Anzahl Freiheitsgrade (Knotenzahl * Raumdimension)
$l \in \mathbb{R}^N$	Stablängen

#### 4 Schadenstoleranz in der Topologieoptimierung

---

$C \in \mathbb{R}^{m \times N}$	Geometriematrix / Gleichgewichtsmatrix
$f \in \mathbb{R}^m$	vordefinierte Last
$s_{\max}$	maximaler Stabquerschnitt
$\sigma_{\min} \in \mathbb{R}^-$	Unterschranke (z. B. physikalische Fließgrenze)
$\sigma_{\max} \in \mathbb{R}^+$	Oberschranke (z. B. physikalische Fließgrenze)
$V(x_1, x_2, \dots, x_n) \in \mathbb{R}$	Volumen als Zielfunktion
$x_i \in \{0; 1\}^N$	Stabauswahl
$w_i \in \mathbb{R}^N$	Reaktionskräfte / Belastungen der einzelnen Stäbe
$s_i \in \mathbb{R}^{+N}$	Stabquerschnitte, Designvariablen

Im Falle der Relaxation von (4.50)-(4.55) in der Ganzzahligkeitsbeschränkung müssen folgende Bedingungen als Ersatz von (4.55) hinzugenommen werden:

$$\forall i = 1, 2, \dots, n : 0 \leq x_i \leq 1 \quad (4.56)$$

Die Aufgabenstellung (4.50)-(4.55) ist wie (C.7) ein lineares Programm.

Dieser Ansatz kann auf einfache Weise auf den mehrfachen Lastfall übertragen werden. Dazu dienen nun die Hochindizes. Statt also einen Lastfall  $f$  zu betrachten werden nun die  $n_l$  Lastfälle  $f^{(1)}, f^{(2)}, \dots, f^{(n_l)}$  betrachtet. Zu jedem Lastfall müssen natürlich nun auch geeignete Kräfte  $w_i^{(k)}$  betrachtet werden:

$$\text{Zielfunktion: } V(s_1, s_2, \dots, s_n) = \sum_{i=1}^n l^T s_i \rightarrow \min_{x_i, w_i^{(k)}, s_i} \quad (4.57)$$

$$\text{unter: } \forall k = 1, \dots, n_l : \forall i = 1, \dots, n :$$

$$C w_i^{(k)} = f^{(k)} \quad (4.58)$$

$$\sigma_{\min} s_{\max} x_i \leq w_i^{(k)} \leq \sigma_{\max} s_{\max} x_i \quad (4.59)$$

$$\sigma_{\min} s_i \leq w_i^{(k)} \leq \sigma_{\max} s_i \quad (4.60)$$

$$0 \leq \sum_{i=1}^n x_i \leq 1 \quad (4.61)$$

$$x \in \{0; 1\}^{nN} \quad (4.62)$$

$$s \geq 0 \quad (4.63)$$

## 4.2 Schadenstoleranz durch Redundanz im Fachwerk

---

Diese Aufgabenstellung (4.57)-(4.63) ist wiederum ein lineares Programm.

Mit  $x = (x_1^T, x_2^T, \dots, x_n^T)^T \in \mathbb{R}^{nN}$ ,

$$w = \left( w_1^{(1)T}, w_2^{(1)T}, \dots, w_n^{(1)T}, w_1^{(2)T}, w_2^{(2)T}, \dots, w_n^{(2)T}, \dots, \right. \\ \left. w_1^{(n_1)T}, w_2^{(n_1)T}, \dots, w_n^{(n_1)T} \right)^T \in \mathbb{R}^{nm_1N}$$

und  $s = (s_1^T, s_2^T, \dots, s_n^T)^T \in \mathbb{R}^{nN}$  ergibt sich folgende (4.64)-(4.72) zu (4.57)-(4.63) äquivalente Aufgabenstellung:

$$\text{Zielfunktion: } (l^T, l^T, \dots, l^T) s \rightarrow \min_{x, w, s} \quad (4.64)$$

$$\text{unter: Gleichung aus Abb. 4.9} \quad (4.65)$$

$$\text{Ungleichung aus Abb. 4.10} \quad (4.66)$$

$$s_1, s_2, \dots, s_n \geq 0 \quad (4.67)$$

$$x \in \{0; 1\}^{nN} \quad (4.68)$$

$$\text{mit: } x_{\min} := \sigma_{\min} s_{\max} I \quad (4.69)$$

$$x_{\max} := \sigma_{\max} s_{\max} I \quad (4.70)$$

$$\sigma_{\min} = \sigma_{\min} I \quad (4.71)$$

$$\sigma_{\max} = \sigma_{\max} I \quad (4.72)$$

In den Abbildungen 4.9 und 4.10 kann die wesentliche Struktur der Nebenbedingungsmatrizen erkannt werden. Die Matrizen sind Sparsematrizen und bieten keinerlei weitere Vereinfachungsmöglichkeiten.



### 4.2.3 Fachwerk mit Redundanz $\frac{1}{n}$

Auf analoge Weise kann auch die Redundanz  $\frac{1}{n}$  modelliert werden: Es findet eine Aufteilung in  $n > 1$  Teilstrukturen<sup>22</sup>  $s_i$  statt, wobei jeweils  $n - 1$  zusammen die Aufgabe „Halten“ erfüllen sollen.

Dadurch kann für  $n > 2$  eine geringere Redundanz und somit eine höhere Effizienz als in Kapitel 4.2.2 erzielt werden. Für  $n = 2$  ergibt sich wegen  $1 - \frac{1}{2} = \frac{1}{2}$  die gleiche Redundanz in beiden Fällen.

Das „Halten“ von  $n - 1$  Teilstrukturen entspricht dem Ausfall einer Teilstruktur. Somit können die  $n$  Fälle mit der Gesamtstruktur  $s = \sum_{i=1}^n s_i$  durch  $\bar{s}_i = s - s_i$  beschrieben werden. Gleiches ist für die Zuordnung der Stäbe zu den Teilstrukturen durch  $x_i$  bzw. die Zuordnung der Ausfälle durch  $\bar{x}_i$  notwendig. Das „Halten“ der Gesamtstruktur  $s$  muss im Gegensatz zu Kapitel 4.1.3 nicht explizit gefordert werden, sondern ergibt sich automatisch, z. B.:  $w_i \leq \sigma_{\max} \bar{s}_i \leq \sigma_{\max} (\bar{s}_i + s_i) = \sigma_{\max} s$ . Die Stabkräfte  $w_i$  werden nicht nur durch  $\bar{s}_i$  gehalten, sondern auch durch  $s$  – also  $\sigma_{\min} s \leq w_i \leq \sigma_{\max} s$ .

Man erhält:

$$\text{Zielfunktion: } V(s_1, s_2, \dots, s_n) = \sum_{i=1}^n l^T s_i \rightarrow \min_{x_i, w_i, s_i} \quad (4.73)$$

$$\text{unter: } \forall i = 1, 2, \dots, n :$$

$$C w_i = f \quad (4.74)$$

$$\sigma_{\min} s_{\max} \bar{x}_i \leq w_i \leq \sigma_{\max} s_{\max} \bar{x}_i \quad (4.75)$$

$$\sigma_{\min} \bar{s}_i \leq w_i \leq \sigma_{\max} \bar{s}_i \quad (4.76)$$

$$\sigma_{\min} s_{\max} x_i \leq \sigma_{\min} s_i \leq \sigma_{\max} s_i \leq \sigma_{\max} s_{\max} x_i \quad (4.77)$$

$$0 \leq \sum_{i=1}^n x_i \leq 1 \quad (4.78)$$

$$x_i \in \{0; 1\}^N \quad (4.79)$$

$$\text{mit: } \bar{x}_i = \sum_{j=1, j \neq i}^n x_j \quad (4.80)$$

$$\bar{s}_i = \sum_{j=1, j \neq i}^n s_j \quad (4.81)$$

<sup>22</sup>Für  $n = 1$  wäre die Redundanz 1 und die Effizienz somit 0. Dies ist natürlich nicht möglich.

#### 4 Schadenstoleranz in der Topologieoptimierung

---

Die Bedeutung der verwendeten Variablen findet sich in folgender Tabelle – sowohl die Daten des Anwenders ( $n, N, m, l, C, f, \sigma_{\min}, \sigma_{\max}, s_{\max}$ ), als auch die Modellierungsvariablen ( $V, x_i, \bar{x}_i, w_i, s_i$ ):

$n \in \mathbb{N}$	Anzahl Teilstrukturen
$N \in \mathbb{N}$	Anzahl Stäbe (finiter Elemente)
$m \in \mathbb{N}$	Anzahl Freiheitsgrade (Knotenzahl * Raumdimension)
$l \in \mathbb{R}^N$	Stablängen
$C \in \mathbb{R}^{m \times N}$	Geometriematrix / Gleichgewichtsmatrix
$f \in \mathbb{R}^m$	vordefinierte Last
$\sigma_{\min} \in \mathbb{R}^-$	Unterschranke (z. B. physikalische Fließgrenze)
$\sigma_{\max} \in \mathbb{R}^+$	Oberschranke (z. B. physikalische Fließgrenze)
$s_{\max}$	maximaler Stabquerschnitt
$V(x_1, x_2, \dots, x_n) \in \mathbb{R}$	Volumen als Zielfunktion
$x_i \in \{0; 1\}^N$	Stabauswahl der Gesamtstruktur
$\bar{x}_i \in \{0; 1\}^N$	Stabauswahl der Teilstrukturen
$w_i \in \mathbb{R}^N$	Reaktionskräfte / Belastungen der einzelnen Stäbe
$s_i \in \mathbb{R}^{+N}$	Stabquerschnitte, Designvariablen

Im Falle der Relaxation von (4.73)-(4.81) in der Ganzzahligkeitsbeschränkung (4.79) müssen folgende Bedingungen als Ersatz hinzu genommen werden.

$$\forall i = 1, 2, \dots, n : 0 \leq x_i, \bar{x}_i \leq 1 \quad (4.82)$$

Die Ungleichung (4.77) zerfällt in mehrere einzelne Bedingungen. Die mittlere Bedingung  $\sigma_{\min}s_i \leq \sigma_{\max}s_i$  ist lediglich eine Anforderung an sinnvolle Verfahrensparameter  $\sigma_{\min} \leq \sigma_{\max}$ . Die anderen beiden Bedingungen  $\sigma_{\min}s_{\max}x_i \leq \sigma_{\min}s_i$  und  $\sigma_{\max}s_i \leq \sigma_{\max}s_{\max}x_i$  bedeuten ( $\sigma_{\min} < 0$ ):

$$s_i \leq s_{\max}x_i$$

Auch diese Aufgabenstellung (4.73)-(4.81) ist ein lineares Programm. Mit  $x = (x_1^T, x_2^T, \dots, x_n^T)^T$ ,  $w = (w_1^T, w_2^T, \dots, w_n^T)^T$  und  $s = (s_1^T, s_2^T, \dots, s_n^T)^T$  ergibt sich das zu (4.73)-(4.81) äquivalente Programm (Hierbei wurde (4.82)

## 4.2 Schadenstoleranz durch Redundanz im Fachwerk

berücksichtigt; im nicht relaxierten Fall sind diese Beschränkungen nicht falsch, sondern nur überflüssig.):

$$\begin{aligned}
 \text{Zielfunktion: } & (l^T, l^T, \dots, l^T) s \rightarrow \min_{x, w_+, w_-, s, h_1, h_2, h_3, h_4} \\
 \text{unter: } & \begin{pmatrix} 0 & C_n & -C_n & 0 \end{pmatrix} \begin{pmatrix} x \\ w_+ \\ w_- \\ s \end{pmatrix} = f \\
 & \begin{pmatrix} \sigma_{\min} s_{\max} \bar{I} & -I & I & 0 \\ -\sigma_{\max} s_{\max} \bar{I} & I & -I & 0 \\ \hline 0 & -I & I & \sigma_{\min} \bar{I} \\ 0 & I & -I & \sigma_{\max} \bar{I} \\ \hline -s_{\max} I & 0 & 0 & I \\ \hline -\bar{I} & 0 & 0 & 0 \\ \bar{I} & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ w_+ \\ w_- \\ s \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \\
 & x, w_+, w_-, s \geq 0 \\
 & x \in \{0; 1\}^{nN}
 \end{aligned}$$

$$\begin{aligned}
 \text{mit: } & w = w_+ - w_- \\
 & \bar{I} = \begin{pmatrix} 0 & I & I & \dots & I \\ I & 0 & I & \dots & I \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ I & \dots & I & 0 & I \\ I & \dots & I & I & 0 \end{pmatrix} \\
 & C_n = \begin{pmatrix} C & 0 & 0 & \dots & 0 \\ 0 & C & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & C & 0 \\ 0 & \dots & 0 & 0 & C \end{pmatrix}
 \end{aligned}$$

Diese Problemstellung kann auf einfache Weise auf den mehrfachen Lastfall übertragen werden. Dazu dienen nun die Hochindizes. Statt also einen Lastfall  $f$  zu betrachten werden nun die  $n_l$  Lastfälle  $f^{(1)}, f^{(2)}, \dots, f^{(n_l)}$  betrachtet. Zu jedem Lastfall müssen natürlich nun auch geeignete Kräfte

#### 4 Schadenstoleranz in der Topologieoptimierung

---

$w_i^{(k)}$  betrachtet werden. Es ergibt sich folgendes lineares Programm:

$$\text{Zielfunktion: } V(s_1, s_2, \dots, s_n) = \sum_{i=1}^n l^T s_i \rightarrow \min_{x_i, w_i, s_i} \quad (4.83)$$

$$\text{unter: } \forall i = 1, 2, \dots, n :$$

$$s_i \leq s_{\max} x_i \quad (4.84)$$

$$\forall k = 1, 2, \dots, n_l :$$

$$C w_i^{(k)} = f^{(k)} \quad (4.85)$$

$$\sigma_{\min} s_{\max} \bar{x}_i \leq w_i^{(k)} \leq \sigma_{\max} s_{\max} \bar{x}_i \quad (4.86)$$

$$\sigma_{\min} \bar{s}_i \leq w_i^{(k)} \leq \sigma_{\max} \bar{s}_i \quad (4.87)$$

$$0 \leq \sum_{i=1}^n x_i \leq 1 \quad (4.88)$$

$$x \in \{0; 1\}^{nN}; s \geq 0 \quad (4.89)$$

$$\text{mit: } \bar{x}_i = \sum_{j=1, j \neq i}^n x_j \quad (4.90)$$

$$\bar{s}_i = \sum_{j=1, j \neq i}^n s_j \quad (4.91)$$

Dieser Ansatz (4.83)-(4.91) ist wiederum ein lineares Programm:

$$\text{Zielfunktion: } (l^T, l^T, \dots, l^T) s \rightarrow \min_{x, w, s} \quad (4.92)$$

$$\text{unter: } \text{Gleichung aus Abb. 4.11} \quad (4.93)$$

$$\text{Ungleichung aus Abb. 4.12} \quad (4.94)$$

$$s_1, s_2, \dots, s_n \geq 0 \quad (4.95)$$

$$x \in \{0; 1\}^{nN} \quad (4.96)$$

$$\text{mit: } x_{\min} = \sigma_{\min} s_{\max} \quad (4.97)$$

$$x_{\max} = \sigma_{\max} s_{\max} \quad (4.98)$$

Den Gleichungen und Ungleichungen in den Abbildungen 4.11 und 4.12 kann die Struktur der Matrizen entnommen werden: Die Sparsematrizen bieten keinerlei weitere Vereinfachungen.





Instanz beschrieben werden:

$$\begin{aligned}
 LP(I) &: c^T \begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \min_{x,y} \\
 &A \begin{pmatrix} x \\ y \end{pmatrix} = b \\
 &y \geq 0 \\
 &\forall i \in I : x_i = 1 \\
 &\forall i \in J \setminus I : x_i \in \{0, 1\}
 \end{aligned}$$

Damit ergibt sich folgender Algorithmus um eine approximative Greedy-Lösung zu finden:

- 1: setze  $i = 0$ ;  $I_i := \emptyset$
- 2: bestimme Lösung  $\tilde{x}^{(i)}, \tilde{y}^{(i)}$  von  $LP(I_i)$ 
  - 2:1: falls keine Lösung möglich: Heuristik hat versagt; Abbruch.
  - 2:2: sonst:
    - 2:2:1: bestimme nicht-binäre Variablen:
 
$$\bar{B}_i := \left\{ j \in J : 0 < \tilde{x}_j^{(i)} < 1 \right\}$$
      - 2:2:1:1: falls  $|\bar{B}_i| > 0$ :
        - bestimme geeigneten Index  $j_i \in \bar{B}_i$ ;
        - setze  $I_{i+1} = I_i \cup \{j_i\}$ ;  $i = i + 1$ ; gehe zu 2
      - 2:2:1:2: sonst: Lösung  $\tilde{x}^{(i)}, \tilde{y}^{(i)}$  gefunden

Jede Komponente von  $x$  wählt in 2:2:1:1 einen Stab für eine gewisse Teilstruktur aus. Es ist daher naheliegend, in 2:2:1:1 einen Index auszuwählen, der dem voluminösesten Stab einer nicht ganzzahligen, binären Variable entspricht. Falls nicht in 2:1 abgebrochen wird, führt dieses Vorgehen trivialerweise zu einer zulässigen Lösung. An Beispielen (vgl. Anhang F) wurden in Tabelle 4.4 die Zielfunktionswerte der Greedy-Heuristik mit der vollständigen Optimumssuche verglichen. Man sieht deutlich die geringe Abweichung der Greedy-Lösung von der Optimallösung.

Auch wenn nun diese Greedy-Heuristik nicht die Optimallösung liefert, ist der Aufwand sehr gering. Falls man in der Praxis damit nicht zufrieden ist, so eignet sich dieser zulässige Punkt weiterhin für das Abschneiden vieler Äste des Binärbaums. Im verbleibenden Baum kann dann mittels

#### 4 Schadenstoleranz in der Topologieoptimierung

Redundanz		$\frac{1}{5}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$	$1 - \frac{1}{2}$	$1 - \frac{1}{3}$	$1 - \frac{1}{4}$	$1 - \frac{1}{5}$
4 Kn.,	greedy	700	700	700	700	700			
6 St.	Optimum	700	700	700	700	700	0	0	0
9 Kn.,	greedy	600	750	600	600	600	1083		
28 St.	Optimum	512	512	525	600	600	1083	0	
16 Kn.,	greedy	526	503	554	572	635	936	1563	2233
86 St.	Optimum		446	468	572	572	936	1356	1952
25 Kn.,	greedy	414	443	448	558	559	894		
200 St.	Optimum		409	444	558	558	888	1228	1609
36 Kn.,	greedy		396	434	553	553	872		
418 St.	Optimum		392	429	553	553	859	1176	
49 Kn.,	greedy			421	552	552	842		
748 St.	Optimum			421	551	551	842		
64 Kn.,	greedy				549	567	839		
1282 St.	Optimum			417	549	549	834		
81 Kn.,	greedy			420	549	549	835		
2040 St.	Optimum			416	548	548	832		

Tabelle 4.4: Die Probleme zu unterschiedlichen Diskretisierungen und unterschiedlichen Graden der Redundanz wurden sowohl mit einem Greedy-Algorithmus, als auch mit einer vollständigen Optimumssuche gelöst. Die Zielfunktionswerte (Volumen) der Ergebnisse sind hier in der Tabelle aufgelistet. Man erkennt deutlich die geringe Abweichung der Greedy-Lösung von der Optimallösung.

einem Branch-and-Bound-Algorithmus ([Mak10], [KV08]) die Optimallösung gefunden werden.

Genau dieses kombinierte Vorgehen lieferte auch die optimalen Ergebnisse in Kapitel 4.2.5 bzw. im Anhang F. Auf diese Weise konnten Binärbäume mit bis zu  $2^{18636} \approx 10^{5610}$  Blättern nach der Optimallösung abgesucht werden – z. B. im Anhang F.2.1 für 6212 Stäbe, einer Redundanz von  $1 - \frac{1}{3}$  und 55908 Unbekannten.

Da (4.57) und (4.83) lineare Programme sind, muss nur in der Wurzel des Suchbaums eine Startlösung (Ecke) gefunden werden. In weiteren Schritten erhält man durch die Fixierung einer Variable auf 0 oder 1 lediglich eine Zusatzbedingung, so dass die Lösung von davor immer noch dual zulässig ist. Somit kann dann mittels eines dualen Simplexalgorithmus in wenigen Iterationen eine zulässige Lösung gefunden werden, die dann

mittels eines revidierten primalen Simplexalgorithmus in wenigen Iterationen zu einem neuen Optimum führt. Nur durch dieses Vorgehen ist es überhaupt möglich, den Baum effizient abzusuchen – nur der Simplexalgorithmus mit seinen verschiedenen Ausführungen ermöglicht die geringen Änderungen von einem Knoten zum nächsten auszunutzen [Mak10].

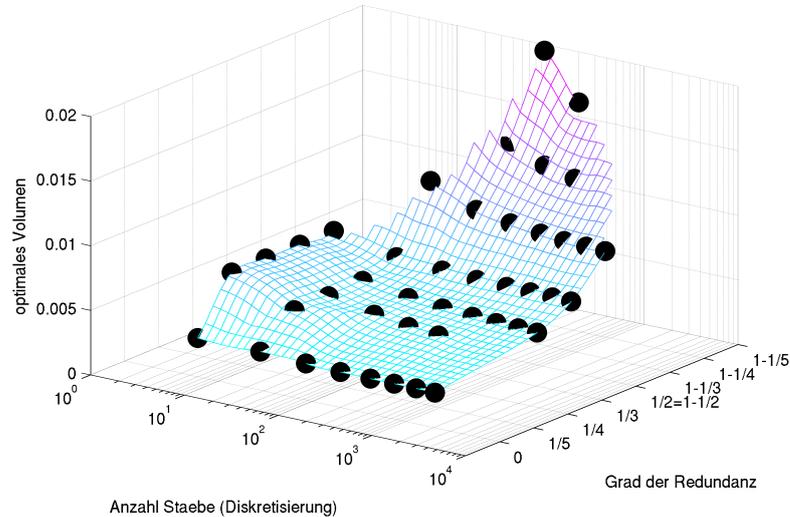


Abbildung 4.13: Vergleich des Grades der Redundanz und der Diskretisierung im Bezug zum Volumen: Die Kugeln stellen gerechnete Beispiele dar. Zur besseren Vergleichbarkeit wurde eine Fläche durch diese Stützstellen gelegt – die Anzahl der Stäbe kann aber natürlich nicht beliebig gewählt werden, sondern muss zumindest ganzzahlig und positiv sein; analoges gilt für den Grad der Redundanz – also  $\frac{1}{n}$  bzw.  $1 - \frac{1}{n}$ ,  $n \in \mathbb{N}$ . Für die gerechneten Grade der Redundanz überstieg ab einer gewissen Feinheit der Diskretisierung der technische Rechenaufwand meine Möglichkeiten. Die „fehlenden“ Ergebnisse für höhere Grade der Redundanz bei wenigen Stäben sind nicht lösbare Problemstellungen – weil zu wenige Stäbe vorhanden sind, können diese nicht auf die geforderte Anzahl Teilstrukturen aufgeteilt werden.

### 4.2.5 Ergebnisse

Die Beispiele aus Anhang F.1 wurden in Abb. 4.13 genutzt, um den Grad der Redundanz und die Diskretisierungsfeinheit im Bezug zum Volumen exemplarisch zu untersuchen. Man sieht deutlich das qualitative Verhalten dieser Werte zueinander. Annäherungsweise wie  $\frac{1}{x}$  verhält sich das Volumen zur Anzahl Stäbe und ebenfalls zum Grad der Redundanz. Bei wenig Freiheiten, also einer groben Diskretisierung, müssen Lösungen mit einem großen Volumen akzeptiert werden. Eine hohe Redundanz ist bei zu grober Diskretisierung gar nicht möglich. Bei einer Erhöhung der Stabzahl ändert sich das Volumen ab einem gewissen Wert kaum noch.

## 4.2 Schadenstoleranz durch Redundanz im Fachwerk

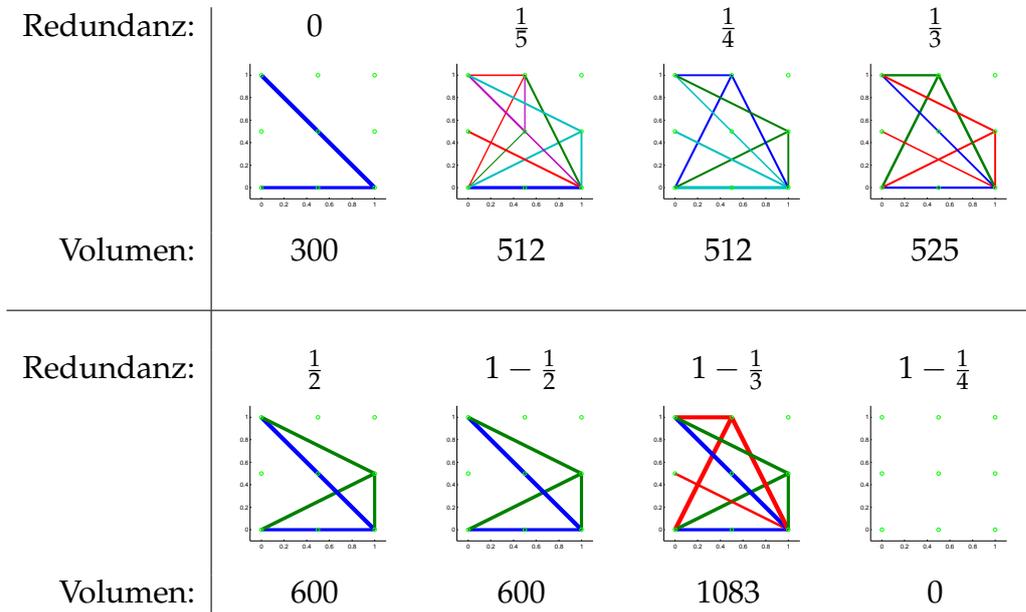


Abbildung 4.14: 2-dimensionales Beispiel eines Fachwerks mit verschiedenen Graden der Redundanz bei einer Grundstruktur aus 28 Stäben

Ein einfaches 2-dimensionales Beispiel eines Kranarms in einem quadratischen Bauraum findet sich in Abb. 4.14. Hierbei zieht eine Last unten rechts nach unten, während die linke Seite gelagert ist. Bei einer Wahl von 3 Knoten in horizontale und in vertikale Richtung ergeben sich 28 potentielle Stäbe durch die Verbindung von jedem Knoten mit jedem anderen (vgl. Anhang C.4), wobei Stäbe über andere Stäbe hinweg nicht betrachtet werden.

Für die Optimallösung ohne Redundanz ist also ein Volumen von 300 nötig. Die beiden Grade der Redundanz von  $\frac{1}{2}$  und  $1 - \frac{1}{2}$  sind natürlich identisch, beziehen sich aber auf die beiden verschiedenen Ansätze aus Kapitel 4.2.2 und 4.2.3. Somit konnte damit die Korrektheit der Implementierungen wie folgt bestätigt werden.

Bei dem Grad der Redundanz von  $\frac{1}{2}$  haben beide Teilstrukturen (blau und grün) ein Volumen von 300 und stellen einzeln betrachtet jeweils eine Optimallösung ohne Redundanz dar – dies sind 2 Ecken des Polyeders des linearen Programms. Da jede einzelne Struktur halten muss, ist dies offensichtlich die beste Lösung.

## 4 Schadenstoleranz in der Topologieoptimierung

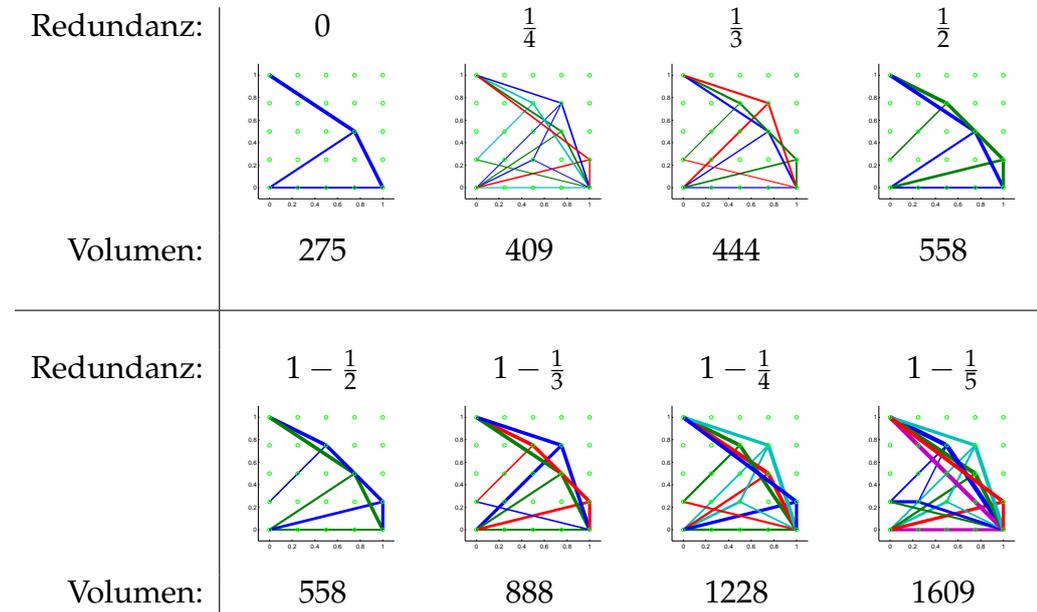


Abbildung 4.15: 2-dimensionales Beispiel eines Fachwerks mit verschiedenen Graden der Redundanz bei einer Grundstruktur aus 200 Stäben

Die Strukturen für die Grade der Redundanz von  $\frac{1}{3}$  und  $1 - \frac{1}{3}$  haben die gleiche Topologie, jedoch unterschiedliche Stabquerschnitte und Volumen.

Die Lösung zu einem Grad der Redundanz von  $\frac{1}{4}$  ist nur theoretisch eine Lösung. Es kann zwar jede einzelne eingefärbte Teilstruktur wie gewünscht ausfallen, aber leider besteht die rote Teilstruktur aus keinem einzigen Stab. Ein Grad der Redundanz von  $1 - \frac{1}{4}$  oder höher ist bei dieser Anzahl der Stäbe nicht möglich.

Bei einer Erhöhung der Stabzahl durch 5 Knoten in jede Richtung auf 200 potentielle Stäbe erkennt man in Abb. 4.15 deutlich die mangelnde Freiheit im 2-dimensionalen, wie im folgenden noch weiter diskutiert wird.

Wenn durch den mehrfachen Lastfall (Kraft unten rechts nach unten oder Kraft oben rechts nach oben) in Abb. 4.16 mehr Lastpfade nötig sind, fehlen die Freiheiten gleichfalls.

Im 3-dimensionalen sind deutlich mehr Freiheiten in der Ausbildung verschiedener Lastpfade vorhanden. Im Anhang F.2 sind einige Beispiele dargestellt.

## 4.2 Schadenstoleranz durch Redundanz im Fachwerk

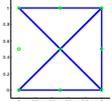
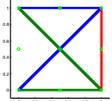
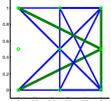
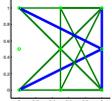
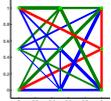
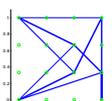
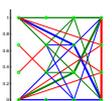
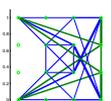
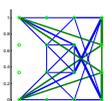
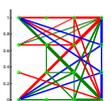
Redundanz:	0	$\frac{1}{3}$	$\frac{1}{2}$	$1 - \frac{1}{2}$	$1 - \frac{1}{3}$
9 Kn., 28 St. Volumen:	 350	 700	 850	 850	 1742
16 Kn., 86 St. Volumen:	 333	 637	 780	 780	 1388

Abbildung 4.16: 2-dimensionales Beispiel eines Fachwerks unter mehrfachen Lastfall mit verschiedenen Graden der Redundanz und verschiedenen Diskretisierungen

Als einfaches Beispiel sei hier der einfache Lastfall mit einer Kraft in  $x$ -Richtung bei  $(0;0;5;1)$  und der Lagerung in der  $x$ - $y$ -Ebene genannt. Bei 3 Knoten in jede Richtung ergeben sich 274 potentielle Stäbe, die nicht in beiden Knoten gelagert sind.

In Abb. 4.17 sind verschiedene Grade der Redundanz dargestellt. Bei einem Grad der Redundanz von 0 oder  $\frac{1}{2}$  wird die zusätzliche Freiheit der dritten Raumdimension nicht benötigt und es ergibt sich eine 2-dimensionale Struktur. An diesem Beispiel einer Struktur wird auch deutlich, dass Redundanz üblicherweise keinerlei Robustheit bzgl. Unsicherheiten der Kräfte (vgl. [MSMK11]) erzielen muss.

Bei einer Redundanz von  $\frac{1}{3}$  in Abb. 4.17(c) werden nun zusätzlich die nach vorne verlaufenden grün eingefärbten Stäbe benötigt, wobei die anderen Stäbe dünner sein können als bei einer Redundanz von  $\frac{1}{2}$  in Abb. 4.17(d). Dadurch wird ein geringeres Volumen erzielt. Die grünen Stäbe bilden hier eine Teilstruktur, die offensichtlich alleine nicht halten kann. Bemerkenswert ist auch, dass die Gesamtstruktur nicht symmetrisch ist.

Während sich im 2-dimensionalen häufig für eine Redundanz von  $\frac{1}{3}$  und  $1 - \frac{1}{3}$  die gleiche Topologie ergeben hat, ergibt sich hier nun deutlich eine andere. Für  $1 - \frac{1}{3}$  besteht die Gesamtstruktur in Abb. 4.17(e) aus den Teilstrukturen in Abb. 4.18.

Die Mächtigkeit der in dieser Arbeit entwickelten redundanten Strukturen zeigt sich sehr eindrucksvoll an dem 3-dimensionalen Beispiel in Abb. 4.19 eines mehrfachen Lastfalls. Die  $x$ - $y$ -Ebene ist gelagert und an den Knoten

## 4 Schadenstoleranz in der Topologieoptimierung

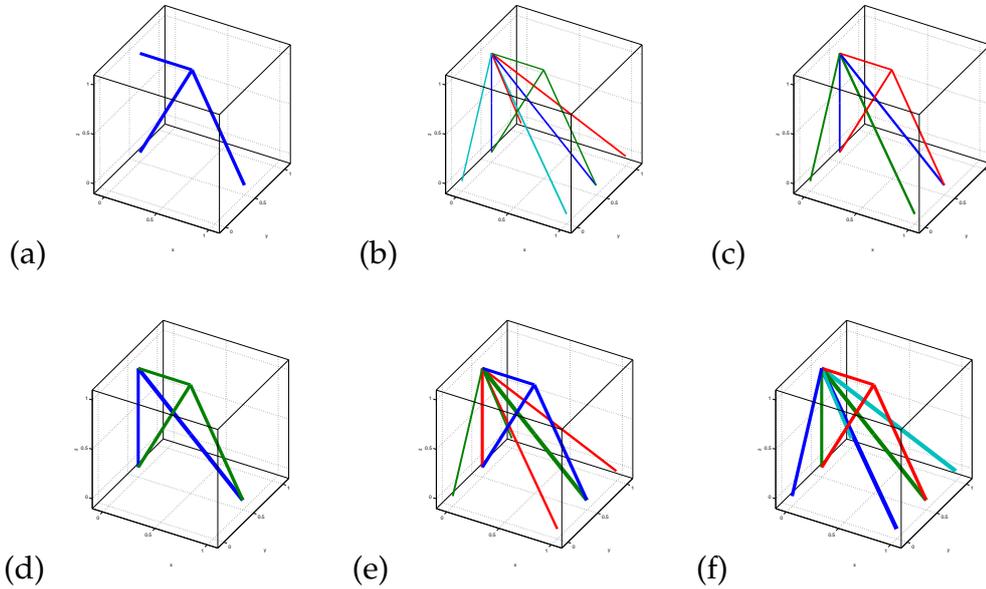


Abbildung 4.17: Der Boden ist gelagert und oben links in der Mitte wirkt eine horizontale Last. Gezeigt sind verschiedene Grade der Redundanz:

Grad der Redundanz	0	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2} = 1 - \frac{1}{2}$	$1 - \frac{1}{3}$	$1 - \frac{1}{4}$
Bild	(a)	(b)	(c)	(d)	(e)	(f)
Volumen	300	433	475	600	950	1300

$(0;0,5;1)$  bzw.  $(1;0,5;1)$  wirken abwechselnd horizontale Kräfte in  $-x$ - bzw.  $x$ -Richtung.

Das Außergewöhnliche sind die Teilstrukturen 4.19(b) und 4.19(d). Diese sind nicht zusammenhängend und können somit erst im Zusammenwirken mit mindestens einer der anderen Teilstrukturen die Aufgabe erfüllen. Bemerkenswert ist auch, dass die (nur) grüne Teilstruktur 4.19(c) die dritte Raumdimension nutzt.

Die Optimallösung dieser Vorgaben ohne Redundanz kommt mit einem Volumen von 350 aus. Ein Grad der Redundanz von  $\frac{1}{2}$  kann mit einem Volumen von 800 erreicht werden. Für die Redundanz  $\frac{1}{3}$  in Abb. 4.19 ist lediglich ein Volumen von 625 nötig.

## 4.2 Schadenstoleranz durch Redundanz im Fachwerk

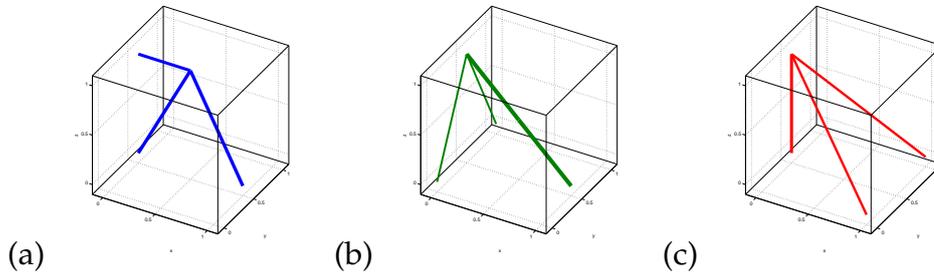


Abbildung 4.18: Teilstrukturen von Abb. 4.17(e)

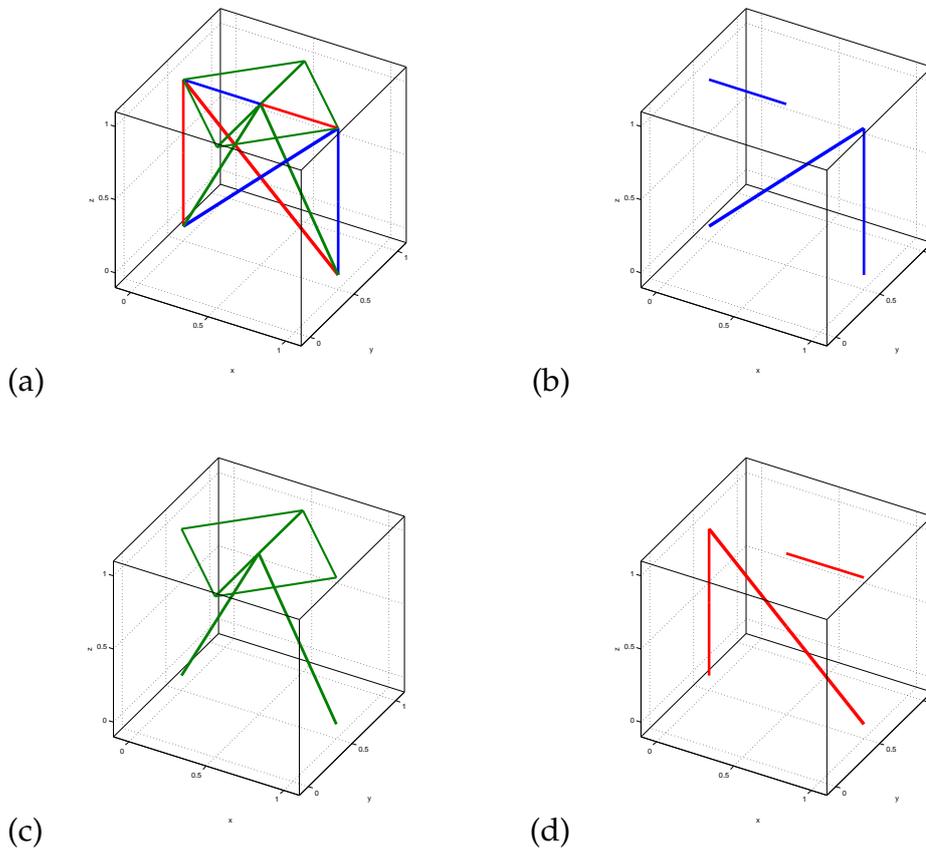


Abbildung 4.19: Der Boden ist gelagert und oben links und rechts wirken abwechselnd Kräfte. In (a) ist die Gesamtstruktur mit einem Grad der Redundanz von  $\frac{1}{3}$  abgebildet und in (b), (c), (d) sind die Einzelstrukturen dargestellt. Die Volumina der Strukturen ergeben sich zu: 625, 175, 275 und 175.



---

## 5 Schlussbemerkungen

In Kapitel 3 wurde eine neue Definition des Begriffs Redundanz gegeben. Diese stellt eine Verallgemeinerung sowohl des Begriffs Redundanz aus der Technik, als auch des Begriffs Redundanz aus der Informations- und Kodierungstheorie dar. Damit war es nun erstmals möglich in Kapitel 4 Redundanz bereits im Vorentwurf durch die Topologieoptimierung zu erzielen. Dies ermöglicht im weiteren Entwicklungsprozess einer Struktur sowohl fail-safe als auch damage-tolerance.

Der hohe Berechnungsaufwand auf Grund des NP-vollständigen und hochdimensionalen Optimierungsproblems für die redundante Topologieoptimierung schreit nach effizienteren Optimierungsverfahren und leistungsfähigeren Rechnern! Eine momentane Abhilfe stellt das Greedy-Vorgehen aus Kapitel 4.2.4 dar. Dennoch ist auf diesem Gebiet sicherlich noch viel Arbeit zu leisten.

Die beiden grundlegenden Ansätze der Redundanz für  $\frac{1}{n}$  und  $1 - \frac{1}{n}$  ermöglichen bereits sehr viele Werte des Grades der Redundanz – vgl. Abb. 4.3. In beiden Ansätzen werden  $n$  Teilstrukturen erstellt, welche allerdings unterschiedlich verknüpft werden. Während bei dem Grad der Redundanz von  $1 - \frac{1}{n}$  jede der  $n$  Teilstrukturen die Aufgabe erfüllen soll, soll bei dem Grad der Redundanz von  $\frac{1}{n}$  jede Kombination von  $n - 1$  Teilstrukturen die Aufgabe erfüllen – dies ist gleichbedeutend mit dem Ausfall irgendeiner der  $n$  Teilstrukturen. Prinzipiell ist auch eine Verallgemeinerung denkbar, in der jede mögliche Auswahl von  $m$  Teilstrukturen die Funktionalität erfüllt. Damit wäre jeder beliebige rationale Wert des Grades der Redundanz zwischen einschließlich 0 und ausschließlich 1 möglich. Allerdings erhöht dieses Vorgehen nochmals die Anzahl der Nebenbedingungen und ist daher extrem aufwendig. Die praktische Notwendigkeit für diese feine Graduierung des Grades der Redundanz in einem Vorentwurf halte ich zwar für gering, dennoch möchte ich diese theoretische Erweiterung nicht verschweigen.

Auf Grund des hohen Berechnungsaufwandes wurden hier in dieser Arbeit lediglich einfache Fachwerke redundant ausgelegt. Dies reicht, um das Grundprinzip der Redundanz in der Topologieoptimierung darzustellen. Prinzipiell kann aber der Bauraum mit jedem diskreten Strukturelement überdeckt werden – z. B. auch mit Balkenelementen. Allerdings zeigen auch die Stäbe bereits redundante Lastpfade auf und ermöglichen somit dem Anwender daraufhin eine praxisnahe Struktur zu erstellen, die die gewünschte Redundanz aufweist. Damit die Redundanz nicht ver-

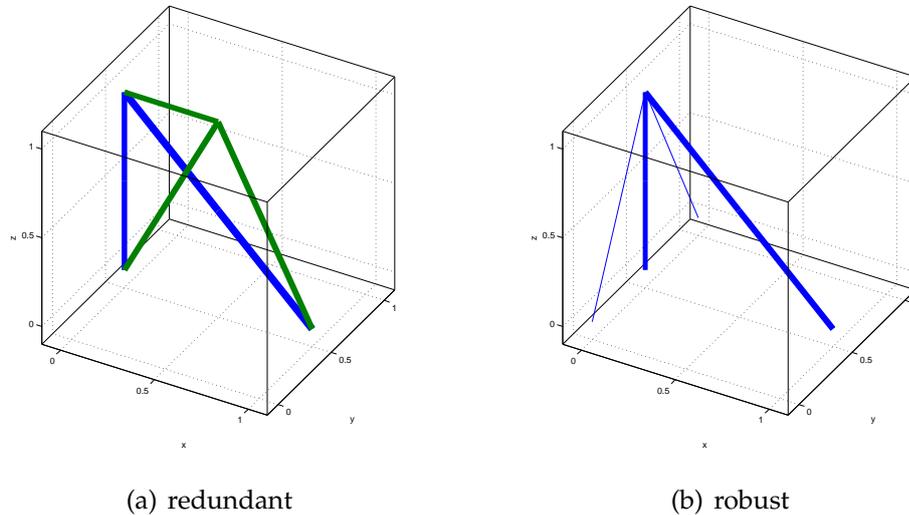


Abbildung 5.1: Vergleich von robuster und von redundanter Topologieoptimierung. (a) zeigt die redundante Topologieoptimierung anhand des Beispiels aus 4.17(d) und (b) zeigt die robuste Topologieoptimierung anhand des gleichen Beispiels, welches auch in [MSMK11] betrachtet wurde. Die redundante Struktur (a) ist nicht robust und die robuste Struktur (b) ist nicht redundant.

loren geht, kann und muss im weiteren Optimierungsprozess (z. B. Gestaltoptimierung oder Dimensionierung der gewählten Teilstrukturen) die im Vorentwurf (Topologieoptimierung) gewonnene Partitionierung weiter betrachtet werden. Das hier in dieser Arbeit entwickelte Konzept ermöglicht mit seiner Allgemeinheit eine triviale Übertragung auf diese nachfolgenden Optimierungen und Analysen. In der industriellen Anwendung wird unter anderem multidisziplinäre Optimierung des Gesamtsystems durchgeführt (z. B. [Dud08]).

Die verwendete Matrix-Kraft-Methode (matrix force method) auf einer Grundstruktur (ground structure) ermöglicht die redundante Topologieoptimierung als gemischt-ganzzahlige lineare Optimierung. Bei Verwendung einer (klassischen) Finiten Elemente Methode erhöht sich der Aufwand exponentiell und übersteigt damit die heute verfügbare Rechenkapazität bei praktischen Problemen. Weitere Entwicklungsarbeit auf dem Gebiet der gemischt-ganzzahligen nichtlinearen Optimierung bei vielen Freiheitsgraden ist daher angeraten.

---

Die heute weitverbreitete Vorgehensweise zur Vermeidung von Schäden durch die Betrachtung von Unsicherheiten der vorgegebenen Parameter (z. B. Kräfte) führt nach meiner Auffassung stets auf eine robuste Optimierung (vgl. [MSMK11], [BTN98], [Soy73]). Analytisch ist es leicht, dieses Vorgehen mit der Redundanz aus der vorliegenden Arbeit zu koppeln. Die Notwendigkeit hierfür wird klar, wenn man sich das Beispiel aus Abb. 4.17(d) bzw. Abb. 5.1(a) ansieht: Eine Variation der Krafrichtung führt zu einem Totalversagen dieser redundanten Struktur selbst im schadlosen Zustand. Das selbe Beispiel wird in [MSMK11] betrachtet und robust gegen Störungen der Kraftgröße/-richtung ausgelegt – in [Stö03] wird dieses Beispiel unter Berücksichtigung von stochastischen Unsicherheiten in der Kraft durch ein deterministisches Ersatzproblem<sup>23</sup> ebenfalls robust ausgelegt. Bei geeigneter Dimensionierung der Stabquerschnitte ist auch bei robuster Auslegung in [MSMK11, Abb. 5(b)] (ebenfalls in Abb. 5.1(b) dargestellt) das Versagen bestimmter Stäbe möglich, aber diese notwendige Dimensionierung ist nicht vorhanden. Also beispielsweise könnte jeder Stab in der  $y$ - $z$ -Ebene ausfallen, wenn die anderen Stäbe durch geeignete Durchmesser die Belastung aushalten. Dies ist allerdings sicherlich bei der Auslegung des vertikalen Stabes in der  $y$ - $z$ -Ebene durch die robuste Optimierung nicht gegeben. Der aus der  $y$ - $z$ -Ebene herausragende Stab ist sogar bei geeigneten Stabquerschnitten der übrigen Stäbe notwendig, um ein Totalversagen der Reststruktur zu vermeiden. Erst das Zusammenführen beider Ansätze, der redundanten und der robusten Optimierung, schafft Abhilfe.

Auch wenn es sicherlich äußerst ungewöhnlich ist, in den Schlussworten noch neue Formulierungen zu geben, möchte ich hier zeigen, wie einfach diese Ansätze zumindest analytisch zusammengeführt werden können, während auf Grund der Komplexität (hochdimensional, semi-infinit und gemischt-ganzzahlig) die praktische Umsetzung eine echte Herausforderung<sup>24</sup> darstellt: Robuste redundante Topologieoptimierung soll nun auf Grundlage von (4.57) und mit den dortigen Bezeichnungen modelliert werden – hierbei wird also ein Grad der Redundanz von  $1 - \frac{1}{n}$  erreicht. Der mehrfache Lastfall wird in der Menge  $F := \{f^{(1)}, f^{(2)}, \dots, f^{(n)}\}$  zusammengefasst. Nun ist eine Lösung gesucht, die robust gegen Störungen dieser Kräfte ist. Die Menge aller Störungen sei  $\hat{F}$ , wobei  $F \subset \hat{F}$ . Somit ergibt sich mit der Indexmenge  $I := \{1, 2, \dots, n\}$  folgendes hochdimensionales, semi-infinites und gemischt-ganzzahliges Optimierungsproblem, mit dessen Formulierung ich meine Arbeit - als Ausblick auf zu-

---

<sup>23</sup>Dieses Ersatzproblem wurde durch eine Diskretisierung der Unsicherheiten erreicht.

<sup>24</sup>Dies würde daher den Rahmen der vorliegenden Dissertation sprengen.

## 5 Schlussbemerkungen

---

künftige Forschung - abschließen möchte:

$$\text{Zielfunktion: } V(s_1, s_2, \dots, s_n) = \sum_{i=1}^n l^T s_i \rightarrow \min_{x_i, w_i, s_i} \quad (5.1)$$

$$\text{unter: } \forall f \in \hat{F} : \forall i \in I :$$

$$Cw_i(f) = f \quad (5.2)$$

$$\sigma_{\min} s_{\max} x_i \leq w_i(f) \leq \sigma_{\max} s_{\max} x_i \quad (5.3)$$

$$\sigma_{\min} s_i \leq w_i(f) \leq \sigma_{\max} s_i \quad (5.4)$$

$$0 \leq \sum_{i=1}^n x_i \leq 1 \quad (5.5)$$

$$x \in \{0; 1\}^{nN} \quad (5.6)$$

$$s \geq 0 \quad (5.7)$$

---

## A Endlicher Automat

In der Informatik wird häufig ein Algorithmus bzw. eine Problemlösung durch einen endlichen Automaten dargestellt. In [Bro98, Teil IV, Kapitel 1.3.2] wird wie folgt definiert:

Ein *endlicher Automat*  $A = (S, T, s_0, S_z, \delta)$  ist gegeben durch die folgenden Bestandteile:

- eine endliche Menge  $S$  von Zuständen,
- eine endliche Menge  $T$  von Eingangszeichen,
- ein Anfangszustand  $s_0 \in S$ ,
- eine Menge  $S_z \subseteq S$  von Endzuständen,
- eine Übergangsfunktion (bzw. -relation)  $\delta : S \times (T \cup \{\varepsilon\}) \rightarrow \mathcal{P}(S)$ .

Dabei stellt  $\mathcal{P}(S)$  die Potenzmenge von  $S$  dar. Sind die Werte der Übergangsfunktion höchstens einelementig und gilt für das ausgezeichnete neutrale Element  $\varepsilon$  die Aussage  $\delta(s, \varepsilon) \subseteq \{s\}$  für alle  $s \in S$ , so heißt der Automat *deterministisch*, sonst *nichtdeterministisch*.

Die Übergangsfunktion wird zwar meist „Funktion“ genannt, bedingt durch fehlende Strukturen in  $S$  und  $T$  ist dies mathematisch gesehen häufig nicht richtig. Typischerweise hat man in  $S$  bzw.  $T$  bestenfalls Gruppenstrukturen zur Verfügung.

Sehr übersichtlich lässt sich ein endlicher Automat als Graph darstellen. Die Elemente der Menge  $S$  von Zuständen bilden die Knoten. Jede Übergangsrelation, die einen Zustand  $s_i \in S$  durch eine Steuerung  $t \in T$  mit einem anderen Zustand  $s_j \in S \setminus \{s_i\}$  in Beziehung setzt, wird als gerichtete Kante  $(s_i, s_j)$  dargestellt:

$$(s_i, s_j) \text{ mit } s_i \neq s_j \text{ ist Kante} \Leftrightarrow \exists t \in T : \{s_j\} \in \delta(s_i, t)$$

Dabei ist offensichtlich  $t \neq \varepsilon$  nicht das neutrale Element. Die Übergangsfunktion bzw. -relation kann dann direkt dem Graphen entnommen werden. In Abb. A.1 wird ein Beispiel gezeigt. Auch durch Zustandsdiagramme oder -tabellen können Automaten dargestellt werden; dies wird hier allerdings nicht betrachtet.

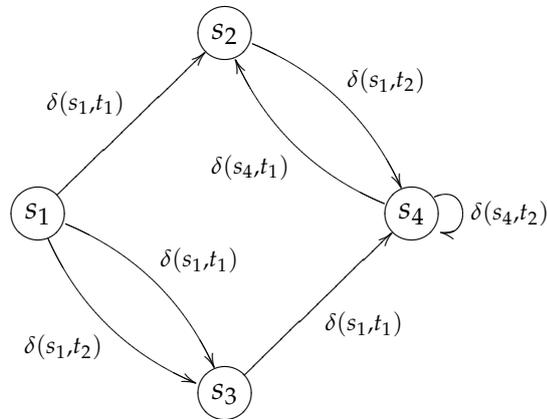


Abbildung A.1: Beispiel eines Automaten als Graph dargestellt

	$s_1$	$s_2$	$\dots$
$t_1$	$s_{i_{1,1}}$	$s_{i_{1,2}}$	$\dots$
$t_2$	$s_{i_{2,1}}$	$s_{i_{2,2}}$	$\dots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$

Tabelle A.1: Beispiel einer Übergangstabelle bestehend aus den Zuständen  $s_i \in S$  und den Eingangszeichen  $t_i \in T$

Da die Mengen  $S$  und  $T$  endlich sind, bietet es sich an, die Übergangsfunktion  $\delta$  tabellarisch als Übergangstabelle (vgl. Tabelle A.1) zu erfassen; dieses Konzept ist allerdings bei Übergangsrelationen oder vielen Zuständen bzw. Eingangszeichen nicht mehr so übersichtlich wie die graphentheoretische Interpretation.

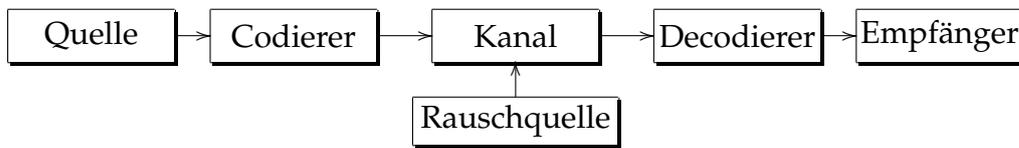


Abbildung B.1: Grundproblem der Informations- und Kodierungstheorie

## B Informations- und Kodierungstheorie

In der Informationstheorie wird die Kommunikation behandelt. Als zentraler Punkt wird **Information** von einer **Quelle** über einen **Kanal** zu einem **Empfänger** transportiert – vgl. Abb. B.1. In der Kodierungstheorie beschäftigt man sich mit dem Code an sich. Dieser soll z. B. durch die **Rauschquelle** verursachte Störungen im Kanal ausgleichen oder er soll die Datenmenge reduzieren. Für exakte Definitionen und weitergehendes Verständnis sei hier auf [HQ95], [Rez61] und [Sha48] verwiesen.

Ein **Alphabet**  $F = \{f_1, f_2, \dots, f_q\}$  ist eine  $q$ -elementige Menge, wobei die Elemente **Buchstaben** genannt werden. Die Elemente des Alphabets werden zu **Wörtern** zusammengesetzt. Ein Wort kann also als (endliche) Folge von Elementen des Alphabets dargestellt werden. Häufig wird eine derartige endliche Folge als Vektor (geordnetes Tupel) dargestellt. Die Menge aller derartiger Wörter sei  $\mathcal{F}$ .

Eine  $u$ -**näre Quelle** mit den Wörtern  $W = \{s_1, s_2, \dots, s_u\}$  ist ein Stichprobenraum  $(W, p)$  mit einer Wahrscheinlichkeitsverteilung  $p(W) = (p_1, p_2, \dots, p_u)$ . Der Einfachheit halber soll hier nun dieses Alphabet  $F$  auch den Codewörtern zu Grunde liegen – eine nicht notwendige Vereinfachung. Eine **Codierung**

$$C : W \rightarrow \mathcal{F} \setminus \{()\}$$

ist eine injektive Abbildung, die jeder **Nachricht** (also jedem Wort aus  $W$ ) ein nicht leeres **Codewort** aus  $\mathcal{F}$  zuordnet. Hierbei wird mit  $()$  das leere Codewort repräsentiert. Das Bild  $C(W)$  heißt **Code**. Ein Code heißt **Blockcode**, wenn alle Codewörter die gleiche Länge haben.

Beispielsweise ist eines der bekanntesten Codewörter die ISBN (International Standard Book Number) auf Büchern. Dem ISBN-Code<sup>25</sup>, der bis

---

<sup>25</sup>Ich beschränke mich hier auf den alten Code, da diesem wenigstens ein Körper aus 11 Elementen zu Grunde liegt – bei dem seit 2007 zu verwendenden liegt eine 10-elementige

2006 üblich war, liegt das 11-näre Alphabet  $\{0, 1, \dots, 9, X\}$  zu Grunde. Bezeichnen wir mit  $z_i, i \in \{1, 2, \dots, 10\}$  den  $i$ -ten Buchstaben

$$z_i \in \begin{cases} \{0, 1, \dots, 9, X\} & \text{für } i = 10 \\ \{0, 1, \dots, 9\} & \text{sonst} \end{cases}$$

eines Codewortes, so ergibt sich für den letzten Buchstaben

$$z_{10} = \left( \sum_{i=1}^9 iz_i \right) \bmod 11,$$

wobei beim numerischen Ergebnis 10 das Zeichen  $X$  für  $z_{10}$  gesetzt wird. Das letzte Zeichen ist also überflüssig und ist eine Prüfziffer.

Häufig wird noch nach **Quell-** und **Kanalcodierung** unterschieden. Hierbei bezieht man sich auf unterschiedliche Aufgaben dieser Codes; während eine Quellcodierung möglichst „effizient“ sein soll, soll die Kanalcodierung den Störungen im Kanal entgegenwirken.

Mit einem willkürlich gewählten  $1 < a \in \mathbb{R}$  ist die **mittlere Information**  $H_a(W)$  einer Quelle

$$H_a(W) := \sum_{s \in W} p(s) \log_a \frac{1}{p(s)}$$

wobei  $p(s)$  die Wahrscheinlichkeit der Information  $s$  angibt.  $H_a(W)$  ist ein Maß und heißt nach [Sha48] auch **Entropie**. Eine beliebige  $u$ -näre Quelle über dem Alphabet  $F$  mit den  $u$  Wörtern  $W$  kann keine höhere Entropie besitzen als:

$$H_0 := \log_a u$$

Dies wird genau dann erreicht, wenn alle Nachrichten gleich wahrscheinlich sind:

$$H_0 = \sum_{i=1}^u \frac{1}{u} \log_a u = \log_a u$$

Die Differenz

$$R := H_0 - H_a(W)$$

---

Menge vor. Somit kann die Rechenoperation  $\bmod$  durch die entsprechende Rechenoperation im Körper ersetzt werden.

---

beschreibt, wie eine Quelle die Möglichkeiten durch das Alphabet ausnutzt. Man bezeichnet  $R$  als **(absolute) Redundanz**. Die **relative Redundanz** (vgl. [Sha48])

$$r := \frac{R}{H_0}$$

ist skaliert und es gilt offensichtlich  $r \in [0; 1]$ . Ein Wert von 1 für die relative Redundanz bedeutet aber eine „wortlose“ Quelle und erscheint seltsam. Der Wert

$$E := \frac{H_a(W)}{H_0}$$

beschreibt ebenfalls die Ausnutzung des Alphabets durch eine Quelle und heißt **Effizienz**. Es gilt:

$$r + E = 1$$

Wenn man den Ausgang einer Codierung wiederum als Quelle interpretiert, können die Begriffe Redundanz und Effizienz direkt auf eine Codierung übertragen werden.

Durch das unnötige Zeichen im obigen Beispiel des ISBN-Codes ist klar, dass diese Codierung eine positive Redundanz aufweist. Die Effizienz muss damit kleiner 1 sein. Man könnte also die Information, die beispielsweise ein Buch eindeutig benennt, auch kürzer mit 9 Zeichen fassen. Aber das 10-te Zeichen (Prüfziffer) ermöglicht das Erkennen von kleinen Übertragungsfehlern. Ein geeigneter Decodierer (z. B. ein Buchhändler) kann damit die erneute Übertragung der Information beim Codierer (z. B. ein Kunde am Telefon) anfordern und somit den Fehler, der durch Rauschen (z. B. das durch IP-Telefonie verursachte Echo) verursacht wurde, korrigieren. Mit einer genügend großen Redundanz können kleine Fehler durch die Rauschquelle bei der Übertragung im Kanal vom Decodierer auch direkt korrigiert werden – man redet hierbei von **fehlerkorrigierenden Codes**.

Eine Codierung, die eine Abbildung von und in strukturlose Menge ist, muss durch eine Code-Tabelle beschrieben werden. Ein derartiges Vorgehen bedeutet für einen Codierer bzw. Decodierer ein permanentes Suchen in der Code-Tabelle. Eine wesentliche Erleichterung bietet eine Struktur in

der Urbild- und Bildmenge. Im einfachsten Fall bilden diese Mengen zumindest eine Halbgruppe oder eine Gruppe. Noch mehr Struktur kann erreicht werden, in dem diese Mengen mit geeigneten Verknüpfungen endliche Körper darstellen. Dieser Spezialfall wird nun genauer betrachtet.

Sei  $\mathbb{F}_q$  ein endlicher Körper mit  $q$  Elementen<sup>26</sup>. Weiter sei  $V_n(q) := \mathbb{F}_q^n$ ,  $n \in \mathbb{N}$  ein  $n$ -dimensionaler Vektorraum über dem endlichen Körper  $\mathbb{F}_q$  mit der **Hamming-Metrik**

$$\rho : V_n(q) \times V_n(q) \rightarrow \mathbb{N}_0, (x, y) \mapsto |\{i \in \{1, 2, \dots, n\} : x_i \neq y_i\}|.$$

Die Hamming-Metrik misst also in wievielen Komponenten sich die Vektoren  $x, y \in V_n(q)$  unterscheiden.

Die Elemente des Körpers  $\mathbb{F}_q$  repräsentieren ein  $q$ -näres Alphabet. Der Vektorraum  $V_n(q)$  repräsentiert alle Codewörter der Länge  $n$ .

Damit ist ein **linearer  $(n, k)$ -Code** über  $\mathbb{F}_q$  ein  $k$ -dimensionaler<sup>27</sup> Untervektorraum  $\mathbf{C}$  von  $V_n(q)$ . Offensichtlich ist ein linearer  $(n, k)$ -Code ein Blockcode.

Legen wir wieder das Alphabet  $\mathbb{F}_q$  auch einer  $u$ -nären Quelle mit den  $k$ -Buchstaben-langen Wörtern  $V_k(q)$  zu Grunde, so bildet eine **lineare  $(n, k)$ -Codierung**  $C : V_k(q) \rightarrow V_n(q)$  einen  $k$ -dimensionalen Vektor (eine Information als Wort der Quelle)

$$s_i = \begin{pmatrix} s_i^{(1)} \\ s_i^{(2)} \\ \vdots \\ s_i^{(k)} \end{pmatrix} \in V_k(q)$$

auf ein Vektorraum-Element (ein Codewort)

$$C(s_i) = \sum_{j=1}^k s_i^{(j)} b_j \in \mathbf{C} \subset V_n(q)$$

ab, wobei  $B := \{b_1, b_2, \dots, b_k\}$  eine Basis des Untervektorraums  $\mathbf{C}$  sei. Mit der **Generatormatrix**

$$G := ( b_1 \mid b_2 \mid \dots \mid b_k ) \in \mathbb{F}_q^{n \times k}$$

---

<sup>26</sup>Dazu muss  $q$  eine Potenz einer Primzahl sein und daher  $q \geq 2$ .

<sup>27</sup>also:  $k \in \mathbb{N}, k \leq n$

---

kann dies übersichtlich durch

$$Gs_i \in \mathbf{C}$$

bzw.

$$\mathbf{C} = \{c \in V_n(q) : c = Gs, s \in V_k(q)\}$$

dargestellt werden.

Zum Abschluss dieses Kapitels sei noch das Beispiel des Hamming-Codes [Ham50], der einen perfekten linearen fehlerkorrigierenden Blockcode  $\mathbf{C}$  über  $\mathbb{F}_2$  darstellt, genannt. Er wurde früher zur Fehlerkorrektur im Arbeitsspeicher von Großrechnern verwendet, wurde aber durch ECC ersetzt. Weiterhin kann dieser Code auch auf Festplatten übertragen werden und führt damit zum RAID 2 [PGK87], welches heute ebenfalls keine Anwendung mehr findet. Die Generatormatrix des linearen  $(7, 4)$ -Hamming-Codes lautet:

$$G := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \in \mathbb{F}_2^{7 \times 4}$$

Für die technische Anwendung als RAID 2 bedeutet dies, dass man  $7k$ ,  $k \in \mathbb{N}$  Festplatten einsetzt und die Datenkapazität von  $4k$  Festplatten nutzen kann. Die restlichen  $3k$  Festplatten dienen der Redundanz und dienen zum Speichern der Paritätsinformationen.

Alle Wörter  $s \in V_4(2) = \mathbb{F}_2^4$ , die gespeichert/codiert werden sollen, sind gleichwahrscheinlich und werden durch  $G$  auf  $V_7(2) = \mathbb{F}_2^7$  abgebildet. Damit ergibt sich, wegen der linearen Unabhängigkeit der Spalten von  $G$ , die mittlere Information des Ausgangs der Codierung  $\mathbf{C}$ , gemessen bezüglich der Basis  $2 = |\mathbb{F}_2|$ , zu:

$$H_2\left(\mathbf{C}\left(\mathbb{F}_2^4\right)\right) = \sum_{c \in \mathbf{C}\left(\mathbb{F}_2^4\right)} p(c) \log_2 \frac{1}{p(c)} = \sum_{i=1}^{2^4} \frac{1}{2^4} \log 2^4 = 4$$

Die maximale Entropie ist

$$H_0 = \log_2 2^7 = 7.$$

Somit ergibt sich die relative Redundanz

$$r = \frac{H_0 - H_2(C(\mathbb{F}_2^4))}{H_0} = \frac{7 - 4}{7} = \frac{3}{7}$$

und die Effizienz

$$E = 1 - r = \frac{4}{7}.$$

---

## C Topologieoptimierung

In diesem Kapitel des Anhangs werden Standardmodellierungen und -methoden zur Topologieoptimierung wiedergegeben, damit auch ein hiermit nicht vertrauter Leser einen Einblick bekommen kann.

Unter Topologieoptimierung versteht man zunächst die Suche nach einer optimalen Topologie einer statischen Struktur. Der Begriff der Topologie ([CKR06]) bezieht sich hier auf den geometrischen Körper. Eine Kugel und ein Zylinder haben hierbei die gleiche Topologie – ebenfalls ein Torus und eine Kaffeetasse; d. h. diese Objekte lassen sich jeweils paarweise durch gleiche Mengensysteme (Äquivalenzklassen) beschreiben. Bei der Suche der optimalen Topologie muss daher eine Bewertung zu Grunde gelegt werden, die das optimale Element aus allen möglichen der selben Äquivalenzklasse bewertet. Die Suche nach dem optimalen Element innerhalb einer Äquivalenzklasse lässt sich durch Gestaltoptimierung oder Dimensionierung bewerkstelligen. Diese Trennung findet sich beispielsweise bei [EKS94, bubble method].

Die verbreiteten Bewertungskriterien in der Topologieoptimierung sind das Volumen und die Steifigkeit (bzw. die umgekehrt proportionale Compliance). Diese Kriterien müssen nun auf die beste Struktur in einer Äquivalenzklasse angewendet werden, um diese Menge zu bewerten. Es wird stets ein linear elastisches Materialverhalten zu Grunde gelegt.

Man unterscheidet 2 grundsätzliche verschiedene Methoden zur Topologieoptimierung, die für diese Arbeit verwendet wurden:

- kontinuierliche Topologieoptimierung (vgl. Kapitel [C.1 SIMP](#))
- diskrete Topologieoptimierung (vgl. Kapitel [C.3 Topologieoptimierung beim Fachwerk](#))

Diese Unterscheidung erfolgt anhand der physikalischen Bedeutung der Strukturen. Daher ist hier der Begriff „diskret“ nicht bezüglich Optimierung zu verstehen. Die Grundidee, nach der optimalen Topologie zu fragen, ist immer ein diskretes Problem – nämlich die Suche nach einem geeigneten Mengensystem.

Die kontinuierliche Topologieoptimierung sucht nach einer optimalen Materialverteilung. Dies findet sich bereits bei [Was60]. Die diskrete Topologieoptimierung erzeugt die Topologie einer Struktur durch die Wahl der Dicke gewisser gewählter diskreter Teilstrukturen, die den Bauraum

überdecken. Im Gegensatz zur Dimensionierung werden explizit auch Dicken von 0 zugelassen sind. Dies kann bereits in [RT73] gefunden werden. Diese Teilstrukturen können durch eine finite Elemente Methode behandelt werden (vgl. Kapitel C.3.1). In Kapitel C.3.2 wird ein andere Ansatz, nämlich mit der Matrix-Kraft-Methode, verwendet.

Während der Ansatz in Kapitel C.1 bei einem vorgegebenen Volumen die Compliance minimiert und dadurch die Steifigkeit maximiert, werden in Kapitel C.3 sowohl die Compliance als auch das Volumen als Bewertungskriterien verwendet.

### C.1 SIMP: Solid Isotropic Material with Penalization

Die erfolgreichste Methode der Topologieoptimierung ist die Homogenisierungsmethode mit der Compliance als Zielfunktion. Um nach der Relaxation weiterhin eine möglichst diskrete Dichteverteilung zu erhalten wird der folgende SIMP-Ansatz (vgl. [BS04]) verwendet. Die Herleitung der Steifigkeitsmatrix  $K = K(x) = \sum_{e=1}^n K_e x_e^p$  als Summe der Elementsteifigkeitsmatrizen findet sich in Standardwerken wie [Prz68] oder [KB97]. Als wichtige Eigenschaft sei hier genannt, dass  $K$  symmetrisch und für  $x > 0$  positiv definit ist.

Der SIMP-Ansatz wird durch folgendes Optimierungsproblem gestellt:

$$\begin{aligned} \text{Zielfunktion:} \quad & c(x) = u^T f \rightarrow \min_x & (C.1) \\ \text{unter:} \quad & \|x\|_1 \leq V \\ \text{mit:} \quad & Ku = f \end{aligned}$$

Zur Übersicht finden sich in folgender Tabelle die Bedeutungen der einzelnen Variablen – sowohl die Daten des Anwenders ( $n, m, K_e, f, V, p$ ), als auch die Modellierungsvariablen ( $x, u, c, K$ ):

$n \in \mathbb{N}$	Anzahl finiter Elemente
$m \in \mathbb{N}$	Anzahl Freiheitsgrade (Knotenzahl * Raumdimension)
$K_e \in \mathbb{R}^{m \times m}$	Elementsteifigkeitsmatrix des Elementes $e$ in globalen Koordinaten
$f \in \mathbb{R}^m$	vordefinierte Last
$V \in \mathbb{R}^+$	vordefiniertes maximales Volumen
$p \in \mathbb{N}$	Penaltypotenz, Verfahrensparameter

$x = (x_1, x_2, \dots, x_n)^T \in [0; 1]^n$	relative Dichte der einzelnen finiten Elemente, Designvariablen
$u = u(x) \in \mathbb{R}^m$	Verschiebungen
$c = c(x) \in \mathbb{R}$	Compliance als Zielfunktion
$K = K(x) = \sum_{e=1}^n K_e x_e^p \in \mathbb{R}^{m \times m}$	Steifigkeitsmatrix der Struktur

Eigentlich sollte  $x \in \{0; 1\}^n$  sein, doch eine Relaxation hat zu  $x \in [0; 1]^n$  geführt. Um trotzdem im Ergebnis möglichst Werte aus  $\{0; 1\}^n$  zu haben, wurde die Penaltypotenz  $p$  eingeführt – damit diese wirkt, muss  $p > 1$  sein.

Wie später gezeigt wird, ist für  $p = 1$  sowohl die Zielfunktion als auch der Zulässigkeitsbereich konvex. Für  $p > 1$  ist die Zielfunktion nur noch näherungsweise konvex.

Damit das Gleichungssystem  $Ku = f$  lösbar ist, muss  $K$  nicht singulär sein. Dies wird durch positive Dichten erreicht – dazu wird ein weiterer Verfahrensparameter  $\epsilon \in \mathbb{R}^+$  sowie eine weitere Restriktion an die Designvariablen eingeführt:

$$0 < \epsilon \leq x \leq 1 \text{ bzw. } x \in [\epsilon; 1]^n \text{ mit } \epsilon > 0, \text{ wobei } \epsilon \text{ klein ist.}$$

Da  $\epsilon^p \approx \epsilon$  und  $1^p = 1$ , ändert weder die  $\epsilon$ -Beschränkung noch die Penaltypotenz im gewünschten Optimum die Werte.

Somit stellt  $Ku = f$  keine Nebenbedingung dar, sondern ist Teil der Zielfunktion (vgl. [BS04] [Sig01]). Das Gleichungssystem wird nach  $u$  aufgelöst und damit kann dann die Zielfunktion ausgerechnet werden. Formal ergibt sich somit die Zielfunktion zu:

$$c(x) = u^T f = f K^{-1} f$$

Da dies numerisch das falsche Vorgehen<sup>28</sup> ist und diese Beschreibung des SIMP-Ansatzes auch ungewöhnlich wäre, bleibe ich hier bei der üblichen Konvention.

Offensichtlich definieren die echten Nebenbedingungen  $\|x\|_1 \leq V$  und  $x_{lb} \leq x \leq 1$  eine konvexe Menge:  $x_{lb} \leq x \leq 1$  ist eine Box-Bedingung;  $\|x\|_1 \leq V$  ist die Begrenzung durch eine Hyperfläche in dieser Box.

<sup>28</sup>Numerisch die Inverse einer  $n \times n$ -Matrix  $A$  zu berechnen erfolgt durch  $n$ -malige Lösung eines linearen Gleichungssystems:  $A = I \Leftrightarrow A^{-1}A = A^{-1}I = A^{-1}$ . Daher ist es effizienter das Gleichungssystem nur einmal zu lösen. Das Berechnen der Inversen einer Matrix ist eine „überraschend seltene Aufgabe“ [Sto89, Kapitel 4.2].

Damit ist der Zulässigkeitsbereich sogar ein konvexes, beschränktes Polyeder.

Zur Bestimmung der Ableitungen der Zielfunktion  $c(x)$  nach den Designvariablen  $x$  müssen ein paar einfache Identitäten nach  $x_e$  abgeleitet werden:

Die Identitäten

$$Ku = f$$

und

$$K^{-1}K = I$$

werden auf beiden Seiten partiell nach  $x_e$  abgeleitet:

$$Ku = f \implies \left( \frac{\partial K}{\partial x_e} \right) u + K \left( \frac{\partial u}{\partial x_e} \right) = 0 \quad (\text{C.2})$$

$$\begin{aligned} K^{-1}K = I &\implies \frac{\partial K^{-1}}{\partial x_e} K + K^{-1} \frac{\partial K}{\partial x_e} = 0 \\ &\implies \frac{\partial K^{-1}}{\partial x_e} = -K^{-1} \frac{\partial K}{\partial x_e} K^{-1} \end{aligned} \quad (\text{C.3})$$

Damit ergibt sich die erste Ableitung von  $c(x)$  komponentenweise zu einer nicht positiven Zahl:

$$\begin{aligned} \frac{\partial c(x)}{\partial x_e} &= \frac{\partial f^T u}{\partial x_e} \\ &= f^T \frac{\partial u}{\partial x_e} \\ &= u^T K \frac{\partial u}{\partial x_e} \\ &\stackrel{(\text{C.2})}{=} u^T K \left( -K^{-1} \left( \frac{\partial K}{\partial x_e} \right) u \right) \\ &= u^T \left( -\frac{\partial K}{\partial x_e} u \right) \\ &= -u^T \frac{\partial K}{\partial x_e} u \\ &\leq 0 \end{aligned}$$

Für die zweite Ableitung von  $c(x)$  ergibt sich komponentenweise:

$$\begin{aligned}
 \frac{\partial}{\partial x_e} \frac{\partial}{\partial x_{\hat{e}}} f^T u &= \frac{\partial}{\partial x_e} \frac{\partial}{\partial x_{\hat{e}}} f^T K^{-1} f \\
 &\stackrel{\text{(C.3)}}{=} -u^T K \left( \frac{\partial K^{-1}}{\partial x_e} \frac{\partial K}{\partial x_{\hat{e}}} u + K^{-1} \frac{\partial}{\partial x_e} \left( \frac{\partial K}{\partial x_{\hat{e}}} u \right) \right) K u \\
 &= f^T K^{-1} \left( \frac{\partial K}{\partial x_e} K^{-1} \frac{\partial K}{\partial x_{\hat{e}}} - \frac{\partial^2 K}{\partial x_e \partial x_{\hat{e}}} + \frac{\partial K}{\partial x_{\hat{e}}} K^{-1} \frac{\partial K}{\partial x_e} \right) K^{-1} f
 \end{aligned}$$

**Fall  $p = 1$ :** Für  $p = 1$  fällt der gemischte Term  $\frac{\partial^2 K}{\partial x_e \partial x_{\hat{e}}}$  weg. Beidseitige Multiplikation von der Hessematrix  $\left( \frac{\partial^2 f^T u}{\partial x_e \partial x_{\hat{e}}} \right)$  mit einem beliebigen Vektor  $v \neq 0$  ergibt:

$$\begin{aligned}
 v^T \left( \frac{\partial^2 f^T u}{\partial x_e \partial x_{\hat{e}}} \right) v &= \sum_{e=1}^N \sum_{\hat{e}=1}^N v_e \frac{\partial^2 f^T u}{\partial x_e \partial x_{\hat{e}}} v_{\hat{e}} \\
 &= \sum_{e=1}^N \sum_{\hat{e}=1}^N f^T K^{-1} \left( K_e K^{-1} K_{\hat{e}} + K_{\hat{e}} K^{-1} K_e \right) K^{-1} f v_e v_{\hat{e}} \\
 &= 2 \sum_{e=1}^N \sum_{\hat{e}=1}^N f^T K^{-1} (v_{\hat{e}} K_{\hat{e}}) K^{-1} (v_e K_e) K^{-1} f
 \end{aligned}$$

Mit  $H := \sum_{e=1}^N v_e K_e$  ergibt sich dann wegen der positiven Definitheit von  $K^{-1}$ :

$$\begin{aligned}
 v^T \left( \frac{\partial^2 f^T u}{\partial x_e \partial x_{\hat{e}}} \right) v &= 2 f^T K^{-1} H K^{-1} H K^{-1} f \\
 &= 2 \left( H K^{-1} f \right)^T K^{-1} \left( H K^{-1} f \right) \\
 &\geq 0
 \end{aligned} \tag{C.4}$$

Für  $p = 1$  ist also  $c(x)$  konvex und somit (C.1) ein konvexes Optimierungsproblem.

**Fall  $p > 1$ :** Für  $p > 1$  fällt der gemischte Term  $\frac{\partial^2 K}{\partial x_e \partial x_{\hat{e}}}$  nicht weg. Beidseitige Multiplikation von der Hessematrix  $\left( \frac{\partial^2 f^T u}{\partial x_e \partial x_{\hat{e}}} \right)$  mit einem beliebigen

Vektor  $v \neq 0$  ergibt:

$$\begin{aligned} v^T \left( \frac{\partial^2 f^T u}{\partial x_e \partial x_{\hat{e}}} \right) v &= \sum_{e=1}^N \sum_{\hat{e}=1}^N v_e \frac{\partial^2 f^T u}{\partial x_e \partial x_{\hat{e}}} v_{\hat{e}} \\ &= 2p^2 \sum_{e=1}^N \sum_{\hat{e}=1}^N f^T K^{-1} \left( v_{\hat{e}} x_{\hat{e}}^{p-1} K_{\hat{e}} \right) K^{-1} \left( v_e x_e^{p-1} K_e \right) K^{-1} f - \\ &\quad p(p-1) \sum_{e=1}^N f^T K^{-1} v_e^2 K_e x_e^{p-2} K^{-1} f \end{aligned}$$

Für  $p > 1$  bleibt allerdings eine näherungsweise Konvexität im nachstehenden Sinn: Für 2 verschiedene zulässige Punkte  $x^{(1)}$  und  $x^{(2)}$  ist mit  $0 < \alpha < 1$  und

$$x := \alpha x^{(1)} + (1 - \alpha) x^{(2)}$$

die Zwischenstelle  $x$  ebenfalls zulässig. Weiter gilt:

$$\begin{aligned} K(x) &= \sum_{e=1}^N K_e x_e = \sum_{e=1}^N K_e \left( \alpha x_e^{(1)} + (1 - \alpha) x_e^{(2)} \right) \\ &= \alpha K \left( x^{(1)} \right) + (1 - \alpha) K \left( x^{(2)} \right) \end{aligned}$$

Die Verschiebungen seien wie folgt definiert:

$$\begin{aligned} K \left( x^{(1)} \right) u^{(1)} &:= f \Leftrightarrow u^{(1)} := K \left( x^{(1)} \right)^{-1} f \\ K \left( x^{(2)} \right) u^{(2)} &:= f \Leftrightarrow u^{(2)} := K \left( x^{(2)} \right)^{-1} f \\ K(x) u &:= f \Leftrightarrow u := K(x)^{-1} f \end{aligned}$$

In grober Näherung können dabei die Inversen der stark diagonaldominanten Bandmatrizen durch inverse Diagonalmatrizen dargestellt werden. Dazu seien die Diagonalmatrizen  $S$  und  $T$  mit positiven Diagonalelementen<sup>29</sup> definiert:

$$S := \approx K \left( x^{(1)} \right)$$

---

<sup>29</sup>Wegen der positiven Definitheit müssen die Diagonalelemente ebenfalls positiv sein.

bzw.

$$\begin{aligned} K(x^{(1)})^{-1} &\approx: S^{-1} \\ &= \begin{pmatrix} s_1 & & \\ & s_2 & \\ & & \ddots \end{pmatrix}^{-1} = \begin{pmatrix} \frac{1}{s_1} & & \\ & \frac{1}{s_2} & \\ & & \ddots \end{pmatrix} \end{aligned}$$

Und:

$$T \approx: K(x^{(2)})$$

bzw.

$$\begin{aligned} K(x^{(2)})^{-1} &\approx: T^{-1} \\ &= \begin{pmatrix} t_1 & & \\ & t_2 & \\ & & \ddots \end{pmatrix}^{-1} = \begin{pmatrix} \frac{1}{t_1} & & \\ & \frac{1}{t_2} & \\ & & \ddots \end{pmatrix} \end{aligned}$$

Und damit ergibt sich für die Zwischenstelle die Näherung:

$$\begin{aligned} K(x)^{-1} &= \left( \alpha K(x^{(1)}) + (1 - \alpha) K(x^{(2)}) \right)^{-1} \\ &\approx (\alpha S + (1 - \alpha) T)^{-1} \\ &= \begin{pmatrix} \alpha s_1 + (1 - \alpha) t_1 & & \\ & \alpha s_2 + (1 - \alpha) t_2 & \\ & & \ddots \end{pmatrix}^{-1} \\ &= \begin{pmatrix} \frac{1}{\alpha s_1 + (1 - \alpha) t_1} & & \\ & \frac{1}{\alpha s_2 + (1 - \alpha) t_2} & \\ & & \ddots \end{pmatrix} \end{aligned}$$

Nun gilt für die Zielfunktionen

$$\begin{aligned}
 f^T u &= f^T K(x)^{-1} f \\
 &\approx \sum_{e=1}^N f_i^2 \frac{1}{\alpha s_i + (1-\alpha)t_i} \\
 \stackrel{\text{(C.5)}}{\leq} &\sum_{e=1}^N f_i^2 \alpha \frac{1}{s_i} + \sum_{e=1}^N f_i^2 (1-\alpha) \frac{1}{t_i} \\
 &\approx \alpha f^T K(x^{(1)})^{-1} f + (1-\alpha) f^T K(x^{(2)})^{-1} f \\
 &= \alpha f^T u^{(1)} + (1-\alpha) u^{(2)},
 \end{aligned}$$

da in den einzelnen Komponenten wegen der positiven Definitheit von  $K$  und damit von  $S$  und  $T$  gilt:

$$\begin{aligned}
 \frac{1}{\alpha s_i + (1-\alpha)t_i} &\leq \alpha \frac{1}{s_i} + (1-\alpha) \frac{1}{t_i} \Leftrightarrow \\
 0 &\leq \alpha^2 s_i t_i + \alpha(1-\alpha)t_i^2 + \alpha(1-\alpha)s_i^2 + (1-\alpha)^2 s_i t_i - s_i t_i \\
 &= (\alpha^2 + (1-\alpha)^2 - 1) s_i t_i + \alpha(1-\alpha)s_i^2 + \alpha(1-\alpha)t_i^2 \\
 &= 2\alpha(\alpha-1)s_i t_i + \alpha(1-\alpha)s_i^2 + \alpha(1-\alpha)t_i^2 \\
 &= \alpha(1-\alpha) \left( -2s_i t_i + s_i^2 + t_i^2 \right) \\
 &= \alpha(1-\alpha) (s_i - t_i)^2 \tag{C.5}
 \end{aligned}$$

Weiterhin gilt in mindestens einer Komponente keine Gleichheit, da  $S^{-1} \neq T^{-1}$  für 2 verschiedene Punkte  $x^{(1)}$  und  $x^{(2)}$ .

Somit ist für  $p > 1$  die Zielfunktion  $c(x)$  näherungsweise konvex.

Die Penalty-Potenz  $p > 1$  wird eingeführt, damit jedes  $x$  eine Ecke des zulässigen konvexen, beschränkten Polyeder ist – also entweder minimale Dichte oder maximale Dichte in jeder Komponente  $x_e \in \{x_{lb}, 1\}$  – da Zwischendichten physikalisch nicht interpretierbar, nicht produzierbar und damit nicht sinnvoll sind. Da allerdings  $1^p = 1$  und  $x_{lb}^p \approx 0$  gilt, ist zumindest in der Nähe eines gewünschten Optimums die Zielfunktion auch für  $p > 1$  strikt konvex. Vorgehensweisen zur Vermeidung eines Schachbrettmusters – wie Gradientenbeschränkungen oder lokale Filterungen des Gradienten – verhindern allerdings die gewünschte komplette 0-1-Belegung.

Von mir wurde die von O. Sigmund eingeführte und in [BS04] unter der Überschrift „Filtering the sensitivities“ beschriebene Filterung verwendet. Dabei wird in jedem Iterationsschritt eines gradientenbasierten Optimierungsalgorithmus der Gradient der Zielfunktion entsprechend seinen Nachbarelementen gewichtet:

$$\frac{\widehat{\partial c}}{\partial x_k} = \frac{1}{x_k \sum_{i=1}^N \widehat{H}_{i,k}} \sum_{i=1}^N \widehat{H}_{i,k} x_i \frac{\partial c}{\partial x_i}$$

mit  $\widehat{H}_{i,k} = \max \{0, r_{\min} - \text{dist}(k, i)\}$

Hierbei werden für alle Elemente  $k$  alle anderen Elemente  $i$  betrachtet, deren Abstand  $\text{dist}(k, i)$  im Bauraum kleiner als der Verfahrensparameter  $r_{\min}$  ist.

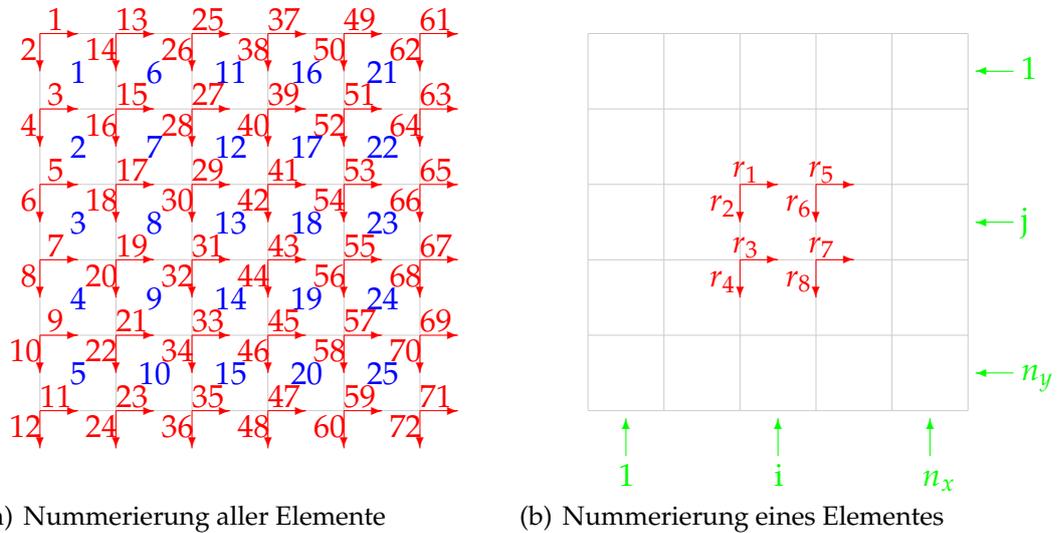


Abbildung C.1: Nummerierung der Elemente

## C.2 Nummerierung der Elemente beim SIMP-Ansatz

Wie bei finiten Elemente Methoden müssen auch beim SIMP-Ansatz die Elemente, Knoten und die Knotenrichtungen nummeriert werden. Im Folgenden wird eine explizite Nummerierung angegeben. Dadurch muss nicht langwierig über eine Tabelle wiederholt nachgeschlagen werden, sondern es kann durch einfache Formeln unmittelbar und somit effizienten auf die gewünschten Elemente, Knoten oder Knotenrichtungen zugegriffen werden. Weiterhin wird mit der im Folgenden gewählten Nummerierung eine schmale Bandbreite der globalen Steifigkeitsmatrix in globalen Koordinaten erzielt. Somit entfällt eine aufwendige Vorsortierung.

### C.2.1 2 D

In diesem Kapitel werden die 2-dimensionalen Elemente Quadrate geeignet nummeriert, um eine schmale Bandbreite der globalen Steifigkeitsmatrix in globalen Koordinaten zu erzielen.

Dazu werden, wie in Abb. C.1(a) dargestellt, die Elemente  $e(i, j)$  Spaltenweise durchnummeriert. Auf analoge Weise werden die Knotenrichtungen  $r(i, j)$  nummeriert. Die Zuordnung kann Abb. C.1(b) entnommen werden.

$$e(i, j) := (i - 1)n_y + j$$

$$r(i, j) := \begin{pmatrix} r_1(i, j) \\ r_2(i, j) \\ r_3(i, j) \\ r_4(i, j) \\ r_5(i, j) \\ r_6(i, j) \\ r_7(i, j) \\ r_8(i, j) \end{pmatrix} := \begin{pmatrix} 2 \cdot ((n_y + 1) \cdot (i - 1) + j) - 1 \\ 2 \cdot ((n_y + 1) \cdot (i - 1) + j) \\ 2 \cdot ((n_y + 1) \cdot (i - 1) + j) + 1 \\ 2 \cdot ((n_y + 1) \cdot (i - 1) + j) + 2 \\ 2 \cdot ((n_y + 1) \cdot (i - 1) + j + (n_y + 1)) - 1 \\ 2 \cdot ((n_y + 1) \cdot (i - 1) + j + (n_y + 1)) \\ 2 \cdot ((n_y + 1) \cdot (i - 1) + j + (n_y + 1)) + 1 \\ 2 \cdot ((n_y + 1) \cdot (i - 1) + j + (n_y + 1)) + 2 \end{pmatrix} =$$

$$\begin{pmatrix} 2 \cdot ((n_y + 1) \cdot (i - 1) + j) - 1 \\ 2 \cdot ((n_y + 1) \cdot (i - 1) + j) \\ 2 \cdot ((n_y + 1) \cdot (i - 1) + j) + 1 \\ 2 \cdot ((n_y + 1) \cdot (i - 1) + j) + 2 \\ 2 \cdot ((n_y + 1) \cdot i + j) - 1 \\ 2 \cdot ((n_y + 1) \cdot i + j) \\ 2 \cdot ((n_y + 1) \cdot i + j) + 1 \\ 2 \cdot ((n_y + 1) \cdot i + j) + 2 \end{pmatrix}; \quad r(1, 1) = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 2n_y + 3 \\ 2n_y + 4 \\ 2n_y + 5 \\ 2n_y + 6 \end{pmatrix}$$

$$n_x = n_y = 5: \quad r(1, 1) = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 13 \\ 14 \\ 15 \\ 16 \end{pmatrix}; \quad r(3, 2) = \begin{pmatrix} 27 \\ 28 \\ 29 \\ 30 \\ 39 \\ 40 \\ 41 \\ 42 \end{pmatrix}$$

In Abb. C.2 wurde die Struktur der Steifigkeitsmatrix angedeutet; nur die Nicht-Null-Elemente wurden als Fleck gezeichnet.

Es ist also nur belegt (teilweise sind sogar diese Elemente 0):

- Hauptdiagonale
- untere und obere Nebendiagonale: 1
- untere und obere Nebendiagonale: 2
- untere und obere Nebendiagonale: 3
- untere und obere Nebendiagonale:  $2n_y - 1$
- untere und obere Nebendiagonale:  $2n_y$

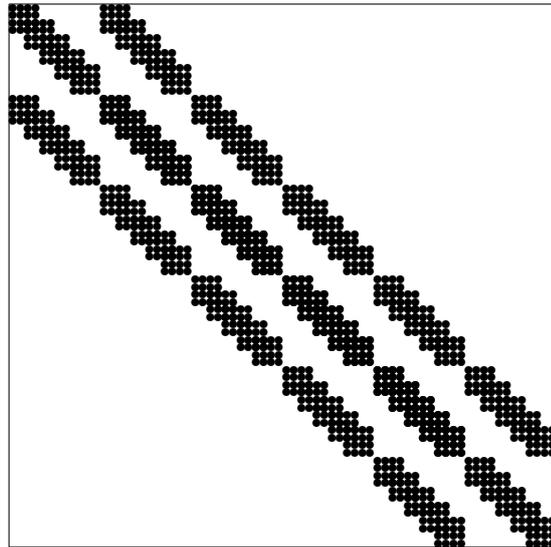


Abbildung C.2: Struktur der Steifigkeitsmatrix bei der Nummerierung nach Kapitel C.2.1 in 2 D

- untere und obere Nebendiagonale:  $2n_y + 1$
- untere und obere Nebendiagonale:  $2n_y + 2$
- untere und obere Nebendiagonale:  $2n_y + 3$
- untere und obere Nebendiagonale:  $2n_y + 4$
- untere und obere Nebendiagonale:  $2n_y + 5$

Also ergibt sich beispielsweise eine Bandmatrix  $K \in \mathbb{R}^{m \times m}$  mit

$$m = 2(n_x + 1)(n_y + 1)$$

und einer Bandbreite von

$$l = k_l + k_u + 1 = 2(2n_y + 5) + 1$$

bzw.

$$k_l = k_u = 2n_y + 5$$

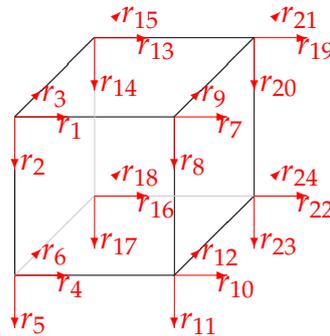


Abbildung C.3: Nummerierung eines Elementes in der  $k$ -ten Ebene,  $i$ -ten Spalte und  $j$ -ten Zeile in 3 D

untere und obere Nebendiagonalen. Mittels typischem Bandmatrixspeicherschema<sup>30</sup> kann dies in einer  $(l \times m)$ -Matrix  $KB$  gespeichert werden.

$n_x$	$n_y$	$K$	$KB$	Ersparnis
10	10	$242 \times 242$	$51 \times 242$	78.9 %
100	100	$20402 \times 20402$	$411 \times 20402$	98.0 %
1000	1000	$2004002 \times 2004002$	$4011 \times 2004002$	99.8 %

Bereits für  $n_x = n_y = 100$  benötigt man auf diese Weise rund 64 MB um  $KB$  als Double<sup>31</sup> zu speichern. Wenn man hingegen lediglich die 15 Diagonalen mit jeweils maximal  $m$  Elementen als Double speichert, benötigt man nur noch rund 2 MB. Allerdings benötigt man für eine Faktorisierung der Bandmatrix  $KB$  – beispielsweise zum Lösen eines linearen Gleichungssystems – den Speicherplatz aller Diagonalen, also auch der Nulldiagonalen.

### C.2.2 3 D

Wie in Kapitel C.2 werden die Elemente  $e(i, j, k)$  spaltenweise und ebenenweise von vorne nach hinten nummeriert. Auf analoge Weise werden die Knotenrichtungen  $r(i, j, k)$  durchnummeriert. Dies ist in Abb. C.3 dargestellt.

$$e(i, j, k) := (i - 1)n_y + (k - 1)n_x n_y + j$$

<sup>30</sup> $k_{i,j} = kb_{k_l+k_u+1+i-j}$  für  $\max(1, j - k_u) \leq i \leq \min(m, j + k_l)$  und  $k_{i,j} = 0$  sonst.

<sup>31</sup>64 Bit pro Element

$$m_x = n_x + 1; \quad m_y = n_y + 1$$

$$r(i, j, k) := \begin{pmatrix} r_1(i, j, k) \\ r_2(i, j, k) \\ r_3(i, j, k) \\ r_4(i, j, k) \\ r_5(i, j, k) \\ r_6(i, j, k) \\ r_7(i, j, k) \\ r_8(i, j, k) \\ r_9(i, j, k) \\ r_{10}(i, j, k) \\ r_{11}(i, j, k) \\ r_{12}(i, j, k) \\ r_{13}(i, j, k) \\ r_{14}(i, j, k) \\ r_{15}(i, j, k) \\ r_{16}(i, j, k) \\ r_{17}(i, j, k) \\ r_{18}(i, j, k) \\ r_{19}(i, j, k) \\ r_{20}(i, j, k) \\ r_{21}(i, j, k) \\ r_{22}(i, j, k) \\ r_{23}(i, j, k) \\ r_{24}(i, j, k) \end{pmatrix} := \begin{pmatrix} 3(m_y(i-1) + j) + 3(k-1)m_x m_y - 2 \\ 3(m_y(i-1) + j) + 3(k-1)m_x m_y - 1 \\ 3(m_y(i-1) + j) + 3(k-1)m_x m_y \\ 3(m_y(i-1) + j) + 3(k-1)m_x m_y + 1 \\ 3(m_y(i-1) + j) + 3(k-1)m_x m_y + 2 \\ 3(m_y(i-1) + j) + 3(k-1)m_x m_y + 3 \\ 3(m_y(i-1) + j + m_y) + 3(k-1)m_x m_y - 2 \\ 3(m_y(i-1) + j + m_y) + 3(k-1)m_x m_y - 1 \\ 3(m_y(i-1) + j + m_y) + 3(k-1)m_x m_y \\ 3(m_y(i-1) + j + m_y) + 3(k-1)m_x m_y + 1 \\ 3(m_y(i-1) + j + m_y) + 3(k-1)m_x m_y + 2 \\ 3(m_y(i-1) + j + m_y) + 3(k-1)m_x m_y + 3 \\ 3(m_y(i-1) + j) + 3k m_x m_y - 2 \\ 3(m_y(i-1) + j) + 3k m_x m_y - 1 \\ 3(m_y(i-1) + j) + 3k m_x m_y \\ 3(m_y(i-1) + j) + 3k m_x m_y + 1 \\ 3(m_y(i-1) + j) + 3k m_x m_y + 2 \\ 3(m_y(i-1) + j) + 3k m_x m_y + 3 \\ 3(m_y(i-1) + j + m_y) + 3k m_x m_y - 2 \\ 3(m_y(i-1) + j + m_y) + 3k m_x m_y - 1 \\ 3(m_y(i-1) + j + m_y) + 3k m_x m_y \\ 3(m_y(i-1) + j + m_y) + 3k m_x m_y + 1 \\ 3(m_y(i-1) + j + m_y) + 3k m_x m_y + 2 \\ 3(m_y(i-1) + j + m_y) + 3k m_x m_y + 3 \end{pmatrix}$$

Die Nicht-Null-Elemente der Steifigkeitsmatrix wurden in Abb. C.4 als Fleck gezeichnet um die Struktur abzubilden.

Es ist also nur belegt (teilweise sind sogar diese Elemente 0):

- Hauptdiagonale
- untere und obere Nebendiagonalen:

- |     |              |                             |                                    |
|-----|--------------|-----------------------------|------------------------------------|
|     |              | • $3(n_x + 1)(n_y + 1) - 4$ | • $3(n_x + 1)(n_y + 1) + 3n_y - 1$ |
|     |              | • $3(n_x + 1)(n_y + 1) - 3$ | • $3(n_x + 1)(n_y + 1) + 3n_y$     |
|     | • $3n_y - 1$ | • $3(n_x + 1)(n_y + 1) - 2$ | • $3(n_x + 1)(n_y + 1) + 3n_y + 1$ |
|     | • $3n_y$     | • $3(n_x + 1)(n_y + 1) - 1$ | • $3(n_x + 1)(n_y + 1) + 3n_y + 2$ |
|     | • $3n_y + 1$ | • $3(n_x + 1)(n_y + 1)$     | • $3(n_x + 1)(n_y + 1) + 3n_y + 3$ |
| • 1 | • $3n_y + 2$ | • $3(n_x + 1)(n_y + 1) + 1$ | • $3(n_x + 1)(n_y + 1) + 3n_y + 4$ |
| • 2 | • $3n_y + 3$ | • $3(n_x + 1)(n_y + 1) + 2$ | • $3(n_x + 1)(n_y + 1) + 3n_y + 5$ |
| • 3 | • $3n_y + 4$ | • $3(n_x + 1)(n_y + 1) + 3$ | • $3(n_x + 1)(n_y + 1) + 3n_y + 6$ |
| • 4 | • $3n_y + 5$ | • $3(n_x + 1)(n_y + 1) + 4$ | • $3(n_x + 1)(n_y + 1) + 3n_y + 7$ |
| • 5 | • $3n_y + 6$ | • $3(n_x + 1)(n_y + 1) + 5$ | • $3(n_x + 1)(n_y + 1) + 3n_y + 8$ |
|     | • $3n_y + 7$ | • $3(n_x + 1)(n_y + 1) + 6$ | • $3(n_x + 1)(n_y + 1) + 3n_y + 9$ |
|     | • $3n_y + 8$ |                             |                                    |
|     | • $3n_y + 9$ |                             |                                    |

Also ergibt sich beispielsweise eine Bandmatrix  $K \in \mathbb{R}^{m \times m}$  mit

$$m = 3(n_x + 1)(n_y + 1)(n_z + 1)$$

und einer Bandbreite von

$$l = k_l + k_u + 1 = 2(3(n_x + 1)(n_y + 1) + 3n_y + 9) + 1$$

bzw.

$$k_l = k_u = 3(n_x + 1)(n_y + 1) + 3n_y + 9$$

untere und obere Nebendiagonalen. Mittels typischem Bandmatrizenspeicherschema<sup>32</sup> kann dies in einer  $(l \times m)$ -Matrix  $KB$  gespeichert werden.

$n_x = n_y = n_z$	$K$	$KB$	Ersparnis
10	$3993 \times 3993$	$562 \times 3993$	85.9 %
100	$3090903 \times 3090903$	$41422 \times 3090903$	98.7 %
1000	$3009009003 \times 3009009003$	$4014022 \times 3009009003$	99.9 %

Bereits für  $n_x = n_y = n_z = 100$  benötigt man auf diese Weise rund 954 GB um  $KB$  als Double<sup>33</sup> zu speichern. Wenn man hingegen lediglich die 47 Diagonalen mit jeweils maximal  $m$  Elementen als Double speichert, benötigt man nur noch rund 1 GB. Allerdings benötigt man für eine Faktorisierung der Bandmatrix  $KB$  – beispielsweise zum Lösen eines linearen Gleichungssystems – den Speicherplatz aller Diagonalen, also auch der Nulldiagonalen.

---

<sup>32</sup> $k_{i,j} = kb_{k_l+k_u+1+i-j}$  für  $\max(1, j - k_u) \leq i \leq \min(m, j + k_l)$  und  $k_{i,j} = 0$  sonst.

<sup>33</sup>64 Bit pro Element

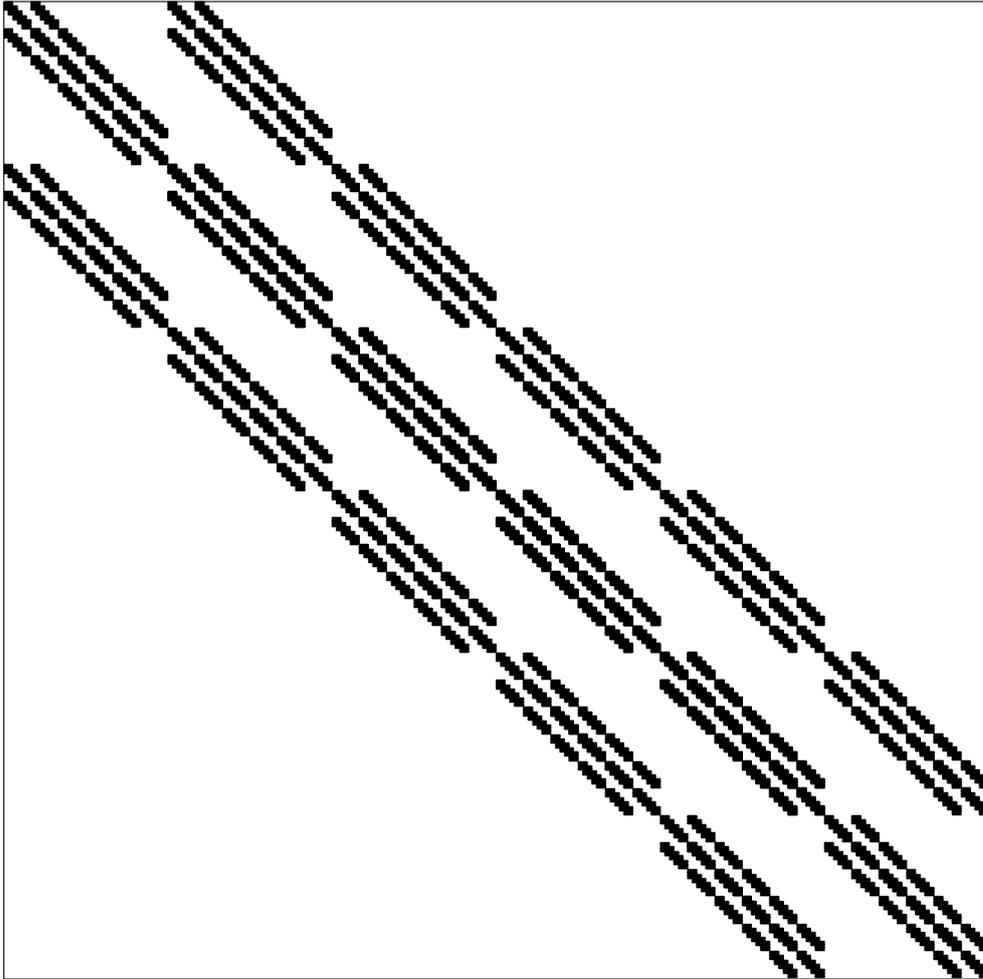


Abbildung C.4: Struktur der Steifigkeitsmatrix bei der Nummerierung nach Kapitel [C.2.2](#) in 3 D

### C.3 Topologieoptimierung beim Fachwerk

Um durch einen Fachwerkansatz eine Topologieoptimierung zu ermöglichen, wird der Bauraum mit Stäben überdeckt. Dieses Vorgehen wird als Grundstrukturmethode (ground structure method) bezeichnet ([DGG64]).

#### C.3.1 Topologieoptimierung beim Fachwerk analog zum SIMP-Ansatz

Analog zum SIMP-Ansatz (C.1) aus Anhang C.1 kann auch beim Fachwerk vorgegangen werden (vgl. [Ach93], [OAH96]). Eine Herleitung der Elementsteifigkeitsmatrix bzw. der Steifigkeitsmatrix findet sich in Standardliteratur wie [MG06] oder [Prz68]. Später in diesem Abschnitt wird die Elementsteifigkeitsmatrix auch explizit angegeben.

Allerdings sind die Designvariablen (Stabvolumen) von vornherein kontinuierlich zwischen 0 und einem vorgegebenen maximalen Stabvolumen. Dieses wird wieder auf 1 skaliert.

Da hierbei also keinerlei Relaxation notwendig ist, besteht auch keine Notwendigkeit für eine Penalty-Potenz.

Damit ergibt sich die Topologieoptimierung beim Fachwerk als folgendes Optimierungsproblem:

$$\begin{aligned} \text{Zielfunktion: } & z(x) = u^T f \rightarrow \min_x & (C.6) \\ \text{unter: } & Ku = f \\ & \|x\|_1 \leq V \end{aligned}$$

Wobei die Variablen folgende Bedeutungen haben – die Daten des Anwenders ( $n, m, K_e, f, V$ ), die Modellierungsvariablen ( $x, u, z, K$ ):

$n \in \mathbb{N}$	Anzahl Stäbe (finiter Elemente)
$m \in \mathbb{N}$	Anzahl Freiheitsgrade (Knotenzahl * Raumdimension)
$K_e \in \mathbb{R}^{m \times m}$	Elementsteifigkeitsmatrix des Stabes / Elementes $e$ in globalen Koordinaten (vgl. [Ach93, Kapitel 1.3.2], [MG06, Definition 2.5])
$f \in \mathbb{R}^m$	vordefinierte Last
$V \in \mathbb{R}^+$	vordefiniertes maximales Volumen
$x = (x_1, x_2, \dots, x_n)^T \in [0; 1]^n$	relative Volumen der einzelnen Stäbe, Designvariablen

$$\begin{array}{ll}
 u = u(x) \in \mathbb{R}^m & \text{Verschiebungen} \\
 z = z(x) \in \mathbb{R} & \text{Compliance als Zielfunktion} \\
 K = K(x) = \sum_{e=1}^n K_e x_e \in \mathbb{R}^{m \times m} & \text{Steifigkeitsmatrix des Stabwerkes}
 \end{array}$$

Im Gegensatz zu einer Aussage in [Ach93] ist hier die Zielfunktion als auch der Zulässigkeitsbereich konvex – wie im Kapitel C.1 für  $p = 1$  gezeigt wurde.

Auch alle weiteren Aussagen aus Anhang C.1 können analog übernommen werden.

Weiterhin wird in [Ach93, Theorem 2.11.1] festgestellt, dass – falls obiges Optimierungsproblem (C.6) lösbar ist mit dem Optimalwert  $C$  – ebenfalls folgendes lösbar ist mit dem Optimalwert  $V$  aus obiger Nebenbedingung:

$$\begin{array}{ll}
 \text{Zielfunktion:} & v(x) = \|x\|_1 \rightarrow \min_x \\
 \text{unter:} & Ku = f \\
 & u^T f \leq C
 \end{array}$$

Da bei der kontinuierlichen Topologieoptimierung im Allgemeinen kein fully stressed design vorliegt, ist dieser Zusammenhang nicht auf den SIMP-Ansatz (C.1) übertragbar. (Die Voraussetzung fully stressed design in [Ach93, Theorem 2.11.1] ist hierbei nicht erfüllt.)

Die Steifigkeitsmatrix stellt sich dabei wie folgt dar:

$$K = \sum_{e=1}^N x_e K_e$$

Ein Stab zwischen den Knoten  $(a_x, a_y)$  und  $(b_x, b_y)$  schließt mit der  $x$ -Achse den Winkel  $\phi$  ein und hat die Elementsteifigkeitsmatrix

$$K_e = \frac{E}{l^2} \begin{pmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{pmatrix}$$

wobei die Abkürzungen

$$\phi = \arctan \left( \frac{|a_x - b_x|}{|a_y - b_y|} \right); \quad c = \cos(\phi); \quad s = \sin(\phi);$$

verwendet wurden. Die Stablänge ist

$$l = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}$$

und für das Elastizitätsmodul wurde  $E = 1000$  verwendet.

Die Ableitung der Zielfunktion ergibt sich analog wie beim SIMP-Ansatz (vgl. Kapitel C.1).

Für die Durchführung der Optimierung und die Auswertung der Ableitung der Zielfunktion sind noch einige weitere Ableitungen notwendig, die hier abschließend angegeben werden:

$$\begin{aligned} \frac{\partial c^2}{\partial a_x} &= \frac{\partial}{\partial a_x} \cos \left( \arctan \left( \frac{|a_x - b_x|}{|a_y - b_y|} \right) \right) \\ &= \frac{-2(a_x - b_x)}{|a_y - b_y|^2 \left( \frac{|a_x - b_x|^2}{|a_y - b_y|^2} + 1 \right)^2} \\ &= \frac{-2(a_x - b_x) |a_y - b_y|^2}{\left( |a_x - b_x|^2 + |a_y - b_y|^2 \right)^2} \end{aligned}$$

$$\begin{aligned} \frac{\partial c^2}{\partial b_x} &= \frac{\partial}{\partial b_x} \cos \left( \arctan \left( \frac{|a_x - b_x|}{|a_y - b_y|} \right) \right) \\ &= -\frac{\partial c^2}{\partial a_x} \end{aligned}$$

$$\begin{aligned}
 \frac{\partial c^2}{\partial a_y} &= \frac{\partial}{\partial a_y} \cos \left( \arctan \left( \frac{|a_x - b_x|}{|a_y - b_y|} \right) \right) \\
 &= \frac{2 |a_x - b_x|^2}{(a_y - b_y)^3 \left( \frac{|a_x - b_x|^2}{|a_y - b_y|^2} + 1 \right)^2} \\
 &= \frac{2 |a_x - b_x|^2 (a_y - b_y)}{\left( |a_x - b_x|^2 + |a_y - b_y|^2 \right)^2}
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial c^2}{\partial b_y} &= \frac{\partial}{\partial b_y} \cos \left( \arctan \left( \frac{|a_x - b_x|}{|a_y - b_y|} \right) \right) \\
 &= -\frac{\partial c^2}{\partial a_y}
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial s^2}{\partial a_x} &= \frac{\partial}{\partial a_x} \sin \left( \arctan \left( \frac{|a_x - b_x|}{|a_y - b_y|} \right) \right) \\
 &= \frac{2(a_x - b_x)}{|a_y - b_y|^2 \left( \frac{|a_x - b_x|^2}{|a_y - b_y|^2} + 1 \right)} - \frac{2(a_x - b_x)^3}{|a_y - b_y|^4 \left( \frac{|a_x - b_x|^2}{|a_y - b_y|^2} + 1 \right)^2} \\
 &= \frac{2(a_x - b_x)}{|a_y - b_y|^2 \left( \frac{|a_x - b_x|^2}{|a_y - b_y|^2} + 1 \right)} - \frac{2(a_x - b_x)^3}{\left( |a_x - b_x|^2 + |a_y - b_y|^2 \right)^2}
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial s^2}{\partial b_x} &= \frac{\partial}{\partial b_x} \sin \left( \arctan \left( \frac{|a_x - b_x|}{|a_y - b_y|} \right) \right) \\
 &= -\frac{\partial s^2}{\partial a_x}
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial s^2}{\partial a_y} &= \frac{\partial}{\partial a_y} \sin \left( \arctan \left( \frac{|a_x - b_x|}{|a_y - b_y|} \right) \right) \\
 &= \frac{2|a_x - b_x|^4}{(a_y - b_y)^5 \left( \frac{|a_x - b_x|^2}{|a_y - b_y|^2} + 1 \right)^2} - \frac{2|a_x - b_x|^2}{(a_y - b_y)^3 \left( \frac{|a_x - b_x|^2}{|a_y - b_y|^2} + 1 \right)} \\
 &= \frac{2|a_x - b_x|^4}{(a_y - b_y) \left( |a_x - b_x|^2 + |a_y - b_y|^2 \right)^2} - \frac{2|a_x - b_x|^2}{(a_y - b_y)^3 \left( \frac{|a_x - b_x|^2}{|a_y - b_y|^2} + 1 \right)}
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial s^2}{\partial b_y} &= \frac{\partial}{\partial b_y} \sin \left( \arctan \left( \frac{|a_x - b_x|}{|a_y - b_y|} \right) \right) \\
 &= -\frac{\partial s^2}{\partial a_y}
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial cs}{\partial a_x} &= \frac{\partial}{\partial a_x} \cos \left( \arctan \left( \frac{|a_x - b_x|}{|a_y - b_y|} \right) \right) \sin \left( \arctan \left( \frac{|a_x - b_x|}{|a_y - b_y|} \right) \right) \\
 &= \frac{\operatorname{sgn}(a_x - b_x)}{|a_y - b_y| \left( \frac{|a_x - b_x|^2}{|a_y - b_y|^2} + 1 \right)} - \frac{2 \operatorname{sgn}(a_x - b_x) (a_x - b_x)^2}{|a_y - b_y|^3 \left( \frac{|a_x - b_x|^2}{|a_y - b_y|^2} + 1 \right)^2}
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial cs}{\partial b_x} &= \frac{\partial}{\partial b_x} \cos \left( \arctan \left( \frac{|a_x - b_x|}{|a_y - b_y|} \right) \right) \sin \left( \arctan \left( \frac{|a_x - b_x|}{|a_y - b_y|} \right) \right) \\
 &= -\frac{\partial cs}{\partial a_x}
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial cs}{\partial a_y} &= \frac{\partial}{\partial a_y} \cos \left( \arctan \left( \frac{|a_x - b_x|}{|a_y - b_y|} \right) \right) \sin \left( \arctan \left( \frac{|a_x - b_x|}{|a_y - b_y|} \right) \right) \\
 &= \frac{2 |a_x - b_x|^3 \operatorname{sgn}(a_y - b_y)}{|a_y - b_y|^4 \left( \frac{|a_x - b_x|^2}{|a_y - b_y|^2} + 1 \right)^2} - \frac{|a_x - b_x| \operatorname{sgn}(a_y - b_y)}{|a_y - b_y|^2 \left( \frac{|a_x - b_x|^2}{|a_y - b_y|^2} + 1 \right)} \\
 &= \frac{2 |a_x - b_x|^3 \operatorname{sgn}(a_y - b_y)}{\left( |a_x - b_x|^2 + |a_y - b_y|^2 \right)^2} - \frac{|a_x - b_x| \operatorname{sgn}(a_y - b_y)}{|a_y - b_y|^2 \left( \frac{|a_x - b_x|^2}{|a_y - b_y|^2} + 1 \right)}
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial cs}{\partial b_y} &= \frac{\partial}{\partial b_y} \cos \left( \arctan \left( \frac{|a_x - b_x|}{|a_y - b_y|} \right) \right) \sin \left( \arctan \left( \frac{|a_x - b_x|}{|a_y - b_y|} \right) \right) \\
 &= -\frac{\partial cs}{\partial a_y}
 \end{aligned}$$

### C.3.2 Topologieoptimierung beim Fachwerk als Volumensminimierung (plastic design)

In der Strukturoptimierung von Stabwerken verbreitet, zumindest aber in der stochastischen Fachwerksoptimierung (vgl. [MG06, Kapitel 9.2] [Mar03]) findet sich bis heute ein Ansatz, der direkt eine Volumensminimierung ermöglicht ([DGG64], [OAH96], [BS04, plastic design]). Das Kräftegleichgewicht  $Cw = f$  findet sich bereits in [Prz68, matrix force method] zusammen mit einer Herleitung der wesentlichen Teile der Geometriematrix. Die Geometriematrix wird in Kapitel C.3.3 vollständig hergeleitet.

$$\begin{aligned}
 \text{Zielfunktion: } & V(a) = l^T a \rightarrow \min_{a,w} & (C.7) \\
 \text{unter: } & Cw = f \\
 & \sigma_{\min} a \leq w \leq \sigma_{\max} a
 \end{aligned}$$

Die Bedeutungen der Variablen ist in folgender Tabelle übersichtlich zusammengestellt: (zuerst die Daten des Anwenders ( $n, m, l, C, f, \sigma_{\min}, \sigma_{\max}$ ) und dann die Modellierungsvariablen ( $V, a, w$ ))

$n \in \mathbb{N}$	Anzahl Stäbe (finiter Elemente)
$m \in \mathbb{N}$	Anzahl Freiheitsgrade (Knotenzahl * Raumdimension)
$l \in \mathbb{R}^n$	Stablängen
$C \in \mathbb{R}^{m \times n}$	Geometriematrix / Gleichgewichtsmatrix
$f \in \mathbb{R}^m$	vordefinierte Last
$\sigma_{\min} \in \mathbb{R}^-$	physikalische Unterschranke (z. B. Fließgrenze bei elastoplastischer Verformung)
$\sigma_{\max} \in \mathbb{R}^+$	physikalische Oberschranke (z. B. Fließgrenze bei elastoplastischer Verformung)
$V(a) \in \mathbb{R}$	Volumen als Zielfunktion
$a = (a_1, a_2, \dots, a_n)^T \in \mathbb{R}^{+n}$	Stabquerschnitte, Designvariablen
$w \in \mathbb{R}^n$	Reaktionskräfte / Belastungen der einzelnen Stäbe

$\sigma_{\min}$  und  $\sigma_{\max}$  könnten auch Diagonal-Matrizen sein, um jedem einzelnen Stab unterschiedliche Grenzen zuzuordnen. Als physikalische Grenzen macht dies natürlich nur bei verschiedenen Materialien Sinn. Dies müsste dann auch analog in der Zielfunktion berücksichtigt werden.

Dieser Ansatz (C.7) ist ein lineares Programm:

$$\begin{aligned} \text{Zielfunktion: } & l^T a \rightarrow \min_{a, w_+, w_-, s_1, s_2} \\ \text{unter: } & \begin{pmatrix} 0 & C & -C & 0 & 0 \\ \sigma_{\min} I & 0 & -I & I & 0 \\ -\sigma_{\max} I & I & 0 & 0 & I \end{pmatrix} \begin{pmatrix} a \\ w_+ \\ w_- \\ s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} f \\ 0 \\ 0 \\ 0 \end{pmatrix} \\ & a, w_+, w_-, s_1, s_2 \geq 0 \end{aligned}$$

### C.3.3 Geometriematrix

Die Geometriematrix wird in der Literatur praktisch nicht ausgerechnet. Daher ist dies hier nötig.

Die Geometriematrix beschreibt die Geometrie der Struktur. Dazu betrachten wir vorerst einen isolierten Stab im  $d$ -Dimensionalen,  $d \in \{2, 3\}$ . Die Knoten  $v^{(1)}, v^{(2)} \in \mathbb{R}^d$  werden durch einen Stab der Länge  $l = \left\| v^{(2)} - v^{(1)} \right\|_2$  verbunden. Eine statische Belastung des Stabes während

dieser in Ruhe bleibt, entspricht einer Kraftgröße  $q \in \mathbb{R}$  im Stab in Längsrichtung des Stabes. Die Kopplung der Stäbe im Fachwerk geschieht über Knotenkräfte. Daher wird nun dieser Kraft im Stab eine Kraft  $f^{(1)} \in \mathbb{R}^d$  im Knoten  $v^{(1)}$  zugeordnet:

$$f^{(1)} = \frac{1}{l} \left( v^{(1)} - v^{(2)} \right) q$$

Damit der Stab nun in Ruhe bleibt, muss statisches Gleichgewicht durch eine Kraft  $f^{(2)} \in \mathbb{R}^d$  im Knoten  $v^{(2)}$ , die  $f^{(1)}$  entgegenwirkt, gelten:

$$f^{(2)} = -f^{(1)}$$

Dieser Ansatz zeigt auch, dass durch die Knoten nur Kräfte in Längsrichtung des Stabes eingeleitet werden können.<sup>34</sup> Daher entstehen auch keine Drehmomente (also i. B. keine Biege- und Torsionsmomente) im Stab.

Mittels des Geometrievektors  $\tilde{\gamma} \in \mathbb{R}^{2d}$  kann dies kompakt geschrieben werden:

$$\frac{1}{l} \underbrace{\begin{pmatrix} v^{(1)} - v^{(2)} \\ v^{(2)} - v^{(1)} \end{pmatrix}}_{=:\tilde{\gamma}} q = \begin{pmatrix} f^{(1)} \\ f^{(2)} \end{pmatrix}$$

Wenn die Lage des Stabes im Raum durch seine Winkel zu den Koordinatenachsen gegeben ist, so kann der Geometrievektor  $\tilde{\gamma} \in \mathbb{R}^{2d}$  auch durch die Winkelfunktionen beschrieben werden. Beispielsweise für  $d = 2$  und einem Winkel  $\varphi$  des Stabes zur x-Achse ergibt sich:<sup>35</sup>

$$\tilde{\gamma} = \frac{1}{l} \begin{pmatrix} v^{(1)} - v^{(2)} \\ v^{(2)} - v^{(1)} \end{pmatrix} = \begin{pmatrix} -\cos \varphi \\ -\sin \varphi \\ \cos \varphi \\ \sin \varphi \end{pmatrix}$$

Da in der Praxis die Knoten gegeben sind, sollte man, um numerische Rundungsfehler zu vermeiden, nicht aus diesen Koordinaten die Winkel

<sup>34</sup>Eine Kraft  $f^{(1)}$ , die orthogonale auf den Stab wirkt, führt bei Multiplikation der Gleichung  $f^{(1)} = \frac{1}{l} \left( v^{(1)} - v^{(2)} \right) q$  mit  $f^{(1)}$  zu dem Wert 0 des Skalarproduktes auf der rechten Seite. Daher muss zwangsweise  $f^{(1)} = 0$  gelten.

<sup>35</sup>Im 3-Dimensionalen benötigt man natürlich 2 Winkel.

berechnen, um daraus dann mittels der Winkelfunktionen den Geometrievektor zu bestimmen – dies wäre auch viel aufwendiger als obige Definition. Die bei diesem schlechten Vorgehen resultierenden Fehler sind im Besonderen im 3-Dimensionalen deutlich.

Im Fachwerk müssen nun mehrere solche Stäbe zusammengefasst werden. Dazu werden die  $\frac{m}{d}$  Knoten und die  $n$  Stäbe global nummeriert. Für den  $i$ -ten Stab ergibt sich dann ein Geometrievektor  $\gamma^{(i)} \in \mathbb{R}^m$  in globalen Koordinaten und der oben definierte Geometrievektor  $\tilde{\gamma}^{(i)} \in \mathbb{R}^{2d}$  in lokalen Koordinaten. Jede Komponente von  $\gamma^{(i)}$  beschreibt nicht nur den zugeordneten Knoten, sondern auch die zugeordnete Raumrichtung. Für die  $j$ -te Komponente gilt:

$$\gamma_j^{(i)} = \begin{cases} \tilde{\gamma}_k^{(i)} & \text{falls Knoten } j \text{ mit dem Stab } i \text{ inzidiert und} \\ & k \text{ der Raumrichtung von } j \text{ entspricht,} \\ & \text{d. h.: } 2(i-1)d + k = j \\ 0 & \text{sonst} \end{cases}$$

Also falls der Stab  $i$  einen Endknoten  $j$  besitzt, so ist die Komponente  $\gamma_j^{(i)}$  des Geometrievektors in globalen Koordinaten gleich der entsprechenden Komponente des Geometrievektors  $\tilde{\gamma}^{(i)}$  in lokalen Koordinaten.

Die Geometriematrix ergibt sich durch Aneinanderreihen der  $n$  Geometrievektoren:

$$C := ( \gamma^{(1)} \mid \gamma^{(2)} \mid \dots \mid \gamma^{(n)} ) \in \mathbb{R}^{m \times n}$$

Somit kann das Kräftegleichgewicht mit der Gleichung

$$Cw = f$$

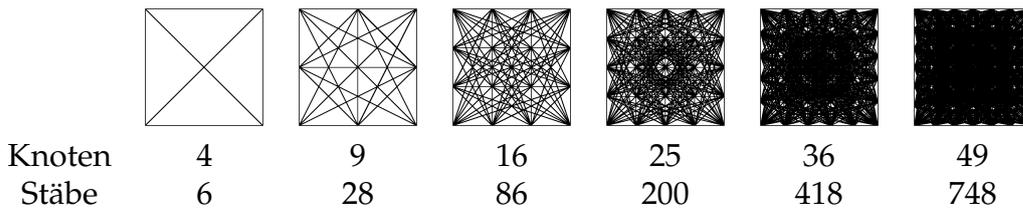
in den Stabkräften  $w \in \mathbb{R}^n$  und den Knotenkräften  $f \in \mathbb{R}^m$  global beschrieben werden. Dieses Vorgehen ist als matrix force method (vgl. [Prz68]) bekannt.

## C.4 Vernetzung des Bauraums durch Stäbe beim Fachwerk

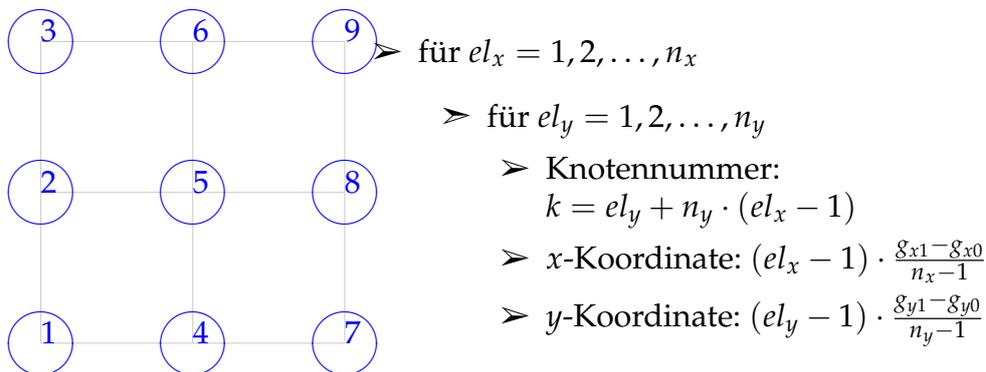
Prinzipiell ist die Vernetzung beliebig und kann dem Anwender überlassen werden. Um größtmögliche Freiheiten der Topologieoptimierung zu ermöglichen, kann jeder Knoten mit jedem anderen verbunden werden. Dabei müssen Stäbe, die durch mehrere andere Stäbe ersetzt werden können, nicht berücksichtigt werden. Andere Vernetzungen reduzieren die Freiheiten für die Optimierung.

### C.4.1 2 D - jeder mit jedem

Jeder Knoten soll mit jedem Knoten über einen Stab verbunden werden. Stäbe über einen anderen Knoten hinweg spielen keine Rolle.



Die Knoten werden auf einem Gitter erstellt:  $n_x$  Knoten in  $x$ -Richtung und  $n_y$  in  $y$ -Richtung. Dabei sind die Gitterkoordinaten  $(x, y)^T$  gleichmäßig verteilt:  $g_{x0} \leq x \leq g_{x1}$  und  $g_{y0} \leq y \leq g_{y1}$ . Weiterhin sind die Knoten nummeriert. Es wird davon ausgegangen, dass der  $x$ -Abstand 2-er benachbarter Knoten gleich dem  $y$ -Abstand ist; also:  $\frac{g_{x1}-g_{x0}}{n_x-1} = \frac{g_{y1}-g_{y0}}{n_y-1}$



Nun sollen zwischen diesen Knoten die Stäbe gesetzt werden. Dazu werden für jeden Knoten alle anderen Knoten sukzessive betrachtet, um dann ggf. einen Stab zwischen diesen zu setzen.

## C Topologieoptimierung

---

- für alle Knoten  $el_x = 1, 2, \dots, n_x$  und  $el_y = 1, 2, \dots, n_y$  mit  $k_1 = el_y + n_y \cdot (el_x - 1)$ 
  - $r_x = 1$  und  $r_y = 0$
  - falls  $el_x + r_x \leq n_x$ 
    - anderer Knoten  
 $k_2 = (el_y + r_y) + n_y \cdot ((el_x + r_x) - 1) = el_y + n_y \cdot el_x$
    - $(k_1, k_2)$  ist Stab (es gilt  $k_1 < k_2$ )
  - für  $r_x = -el_x + 1, -el_x + 2, \dots, -el_x + n_x$  und  $r_y = 1, 2, \dots, -el_y + n_y$ 
    - anderer Knoten  $k_2 = (el_y + r_y) + n_y \cdot ((el_x + r_x) - 1)$
    - falls kein anderer Knoten auf der Strecke zwischen  $k_1$  und  $k_2$  liegt
      - $(k_1, k_2)$  ist Stab

Es bleibt hierbei die Frage, wie man erkennt, dass kein weiterer Knoten zwischen 2-en liegt. Wir haben also

$$k_1 = el_y + n_y \cdot (el_x - 1)$$

und

$$k_2 = (el_y + r_y) + n_y \cdot ((el_x + r_x) - 1)$$

mit den Koordinaten

$$\begin{pmatrix} (el_x - 1) \cdot \frac{g_{x1} - g_{x0}}{n_x - 1} \\ (el_y - 1) \cdot \frac{g_{y1} - g_{y0}}{n_y - 1} \end{pmatrix}$$

und

$$\begin{pmatrix} (el_x + r_x - 1) \cdot \frac{g_{x1} - g_{x0}}{n_x - 1} \\ (el_y + r_y - 1) \cdot \frac{g_{y1} - g_{y0}}{n_y - 1} \end{pmatrix}.$$

Auf Grund der Annahme

$$\frac{g_{x1} - g_{x0}}{n_x - 1} = \frac{g_{y1} - g_{y0}}{n_y - 1}$$

ergeben sich die Positionen aus

$$\begin{pmatrix} el_x - 1 \\ el_y - 1 \end{pmatrix}$$

und

$$\begin{pmatrix} el_x + r_x - 1 \\ el_y + r_y - 1 \end{pmatrix} = \begin{pmatrix} el_x - 1 \\ el_y - 1 \end{pmatrix} + \begin{pmatrix} r_x \\ r_y \end{pmatrix}.$$

Für  $r_x = 0$  liegt nur für  $r_y = 1$  kein weiterer Knoten auf der Verbindung.

Für  $r_x \neq 0$  und  $r_y > 0$  ist die Richtung  $\begin{pmatrix} r_x \\ r_y \end{pmatrix}$  entscheidend. Falls ein weiterer Knoten auf der Verbindung liegt, so existiert ein  $\lambda > 1$  mit

$$\begin{pmatrix} |r_x| \\ r_y \end{pmatrix} = \lambda \begin{pmatrix} |\tilde{r}_x| \\ \tilde{r}_y \end{pmatrix}$$

wobei  $|r_x|, r_y, |\tilde{r}_x|, \tilde{r}_y \in \mathbb{N}$ .

Division beider Gleichungen liefert:

$$\frac{|r_x|}{r_y} = \frac{|\tilde{r}_x|}{\tilde{r}_y} \text{ wobei } |\tilde{r}_x| < |r_x| \text{ und } \tilde{r}_y < r_y$$

Mit anderen Worten: Falls  $\frac{r_x}{r_y}$  echt gekürzt werden kann, so liegt ein weiterer Knoten auf der Verbindung.

Also:

Falls der größte gemeinsame Teiler<sup>36</sup>  $\text{ggT}(|r_x|, r_y) = 1$ , so liegt kein weiterer Knoten auf der Verbindung.

Mittels dem Euklidischen Algorithmus kann dies effizient überprüft bzw. durchgeführt werden.

---

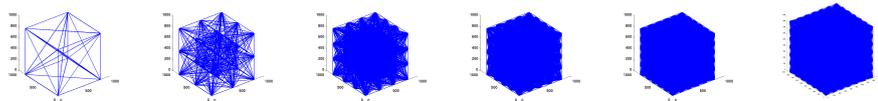
<sup>36</sup>vgl. Definition G.1 im Anhang G

Für  $n_x$  Knoten in  $x$ -Richtung und  $n_y$  in  $y$ -Richtung ergibt sich die Obergrenze der Anzahl Stäbe zu:

$$\begin{aligned}
 & \sum_{x=1}^{n_x-1} \sum_{y=1}^{n_y} 1 + \sum_{x=1}^{n_x} \sum_{y=1}^{n_y} 1 + \sum_{x=1}^{n_x} \sum_{y=1}^{n_y} \sum_{\substack{r_x=1-x, \\ r_x \neq 0}}^{n_x-x} \sum_{\substack{r_y=1, \\ \text{ggT}(r_x, r_y)=1}}^{n_y-y} 1 \\
 \leq & \sum_{x=1}^{n_x-1} \sum_{y=1}^{n_y} 1 + \sum_{x=1}^{n_x} \sum_{y=1}^{n_y} 1 + \sum_{x=1}^{n_x} \sum_{y=1}^{n_y} \sum_{r_x=1-x}^{n_x-x-1} \sum_{r_y=1}^{n_y-y} 1 \\
 = & (n_x - 1)n_y + n_x n_y + \frac{n_x n_y (n_x n_y - n_x - n_y + 1)}{2} \\
 = & \frac{n_x n_y (n_x n_y - n_x + 5 - n_y)}{2} \\
 \leq & \sum_{i=1}^{n_x n_y} i = \frac{n_x n_y (n_x n_y + 1)}{2}
 \end{aligned}$$

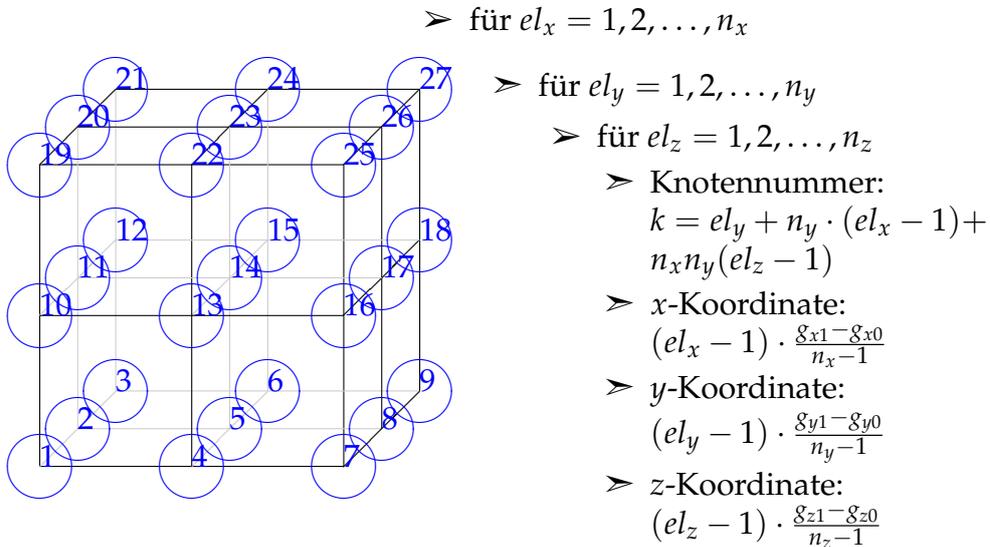
### C.4.2 3 D - jeder mit jedem

Jeder Knoten soll mit jedem Knoten über einen Stab verbunden werden. Stäbe über einen anderen Knoten hinweg spielen keine Rolle.



Knoten	8	27	64	125	216	343
Stäbe	28	302	1716	6412	19508	48426
Knoten	512	729	1000			
Stäbe	108868	220280	415716			

Die Knoten werden auf einem Gitter erstellt:  $n_x$  Knoten in  $x$ -Richtung,  $n_y$  in  $y$ -Richtung und  $n_z$  in  $z$ -Richtung. Dabei sind die Gitterkoordinaten  $(x, y, z)^T$  gleichmäßig verteilt:  $g_{x0} \leq x \leq g_{x1}$ ,  $g_{y0} \leq y \leq g_{y1}$  und  $g_{z0} \leq z \leq g_{z1}$ . Weiterhin sind die Knoten nummeriert. Es wird davon ausgegangen, dass der  $x$ -Abstand 2-er benachbarter Knoten gleich dem  $y$ -Abstand und auch gleich dem  $z$ -Abstand ist; also:  $\frac{g_{x1}-g_{x0}}{n_x-1} = \frac{g_{y1}-g_{y0}}{n_y-1} = \frac{g_{z1}-g_{z0}}{n_z-1}$



Nun sollen zwischen diesen Knoten die Stäbe gesetzt werden. Dazu werden alle Knoten durchgegangen und für jeden Knoten wieder alle anderen relevanten.

Dabei müssen ein paar Sonderfälle unterschieden werden:

- 1:  $r_x = 0, r_y = 0, r_z = 0$ : identischer Knoten, kein Stab
- 2:  $r_x = 0, r_y = 0, r_z \neq 0 \Rightarrow r_z = 1$
- 3:  $r_x = 0, r_y \neq 0, r_z = 0 \Rightarrow r_y = 1$
- 4:  $r_x \neq 0, r_y = 0, r_z = 0 \Rightarrow r_x = 1$
- 5:  $r_x \neq 0, r_y \neq 0, r_z = 0$ : analog zu 2 D
- 6:  $r_x \neq 0, r_y = 0, r_z \neq 0$ : analog zu 2 D
- 7:  $r_x = 0, r_y \neq 0, r_z \neq 0$ : analog zu 2 D
- 8:  $r_x \neq 0, r_y \neq 0, r_z \neq 0$ : eigentlicher 3 D Fall

Dies führt nun zu folgendem Algorithmus:

- > für alle Knoten  $el_x = 1, 2, \dots, n_x, el_y = 1, 2, \dots, n_y$  und  $el_z = 1, 2, \dots, n_z$   
 mit  $k_1 = el_y + n_y \cdot (el_x - 1) + n_x n_y (el_z - 1)$   
 >  $r_x = 0, r_y = 0$  und  $r_z = 1$ : falls  $el_z + r_z \leq n_z$

- anderer Knoten  $k_2 = (el_y + r_y) + n_y \cdot ((el_x + r_x) - 1) + n_x n_y ((el_z + r_z) - 1)$
  - $(k_1, k_2)$  ist Stab (es gilt  $k_1 < k_2$ )
- $r_x = 0, r_y = 1$  und  $r_z = 0$ : falls  $el_y + r_y \leq n_y$ 
  - anderer Knoten  $k_2 = (el_y + r_y) + n_y \cdot ((el_x + r_x) - 1) + n_x n_y ((el_z + r_z) - 1)$
  - $(k_1, k_2)$  ist Stab (es gilt  $k_1 < k_2$ )
- $r_x = 1, r_y = 0$  und  $r_z = 0$ : falls  $el_x + r_x \leq n_x$ 
  - anderer Knoten  $k_2 = (el_y + r_y) + n_y \cdot ((el_x + r_x) - 1) + n_x n_y ((el_z + r_z) - 1)$
  - $(k_1, k_2)$  ist Stab (es gilt  $k_1 < k_2$ )
- für  $r_x = -el_x + 1, -el_x + 2, \dots, -el_x + n_x; r_y = 1, 2, \dots, -el_y + n_y$  und  $r_z = 0$ 
  - anderer Knoten  $k_2 = (el_y + r_y) + n_y \cdot ((el_x + r_x) - 1) + n_x n_y ((el_z + r_z) - 1)$
  - falls  $r_x \neq 0$  und kein anderer Knoten auf der Strecke zwischen  $k_1$  und  $k_2$  liegt  
(ggT  $(r_x, r_y) = 1$ )
  - $(k_1, k_2)$  ist Stab
- für  $r_x = -el_x + 1, -el_x + 2, \dots, -el_x + n_x; r_y = 0$  und  $r_z = 1, 2, \dots, -el_z + n_z$ 
  - anderer Knoten  $k_2 = (el_y + r_y) + n_y \cdot ((el_x + r_x) - 1) + n_x n_y ((el_z + r_z) - 1)$
  - falls  $r_x \neq 0$  und kein anderer Knoten auf der Strecke zwischen  $k_1$  und  $k_2$  liegt  
(ggT  $(r_x, r_z) = 1$ )
  - $(k_1, k_2)$  ist Stab
- für  $r_x = 0; r_y = -el_y + 1, -el_y + 2, \dots, -el_y + n_y$  und  $r_z = 1, 2, \dots, -el_z + n_z$ 
  - anderer Knoten  $k_2 = (el_y + r_y) + n_y \cdot ((el_x + r_x) - 1) + n_x n_y ((el_z + r_z) - 1)$
  - falls  $r_y \neq 0$  und kein anderer Knoten auf der Strecke zwischen  $k_1$  und  $k_2$  liegt  
(ggT  $(r_y, r_z) = 1$ )
  - $(k_1, k_2)$  ist Stab

#### C.4 Vernetzung des Bauraums durch Stäbe beim Fachwerk

---

- für  $r_x = -el_x + 1, -el_x + 2, \dots, -el_x + n_x; r_y = -el_y + 1, -el_y + 2, \dots, -el_y + n_y$  und  $r_z = 1, 2, \dots, -el_z + n_z$ 
  - anderer Knoten  $k_2 = (el_y + r_y) + n_y \cdot ((el_x + r_x) - 1) + n_x n_y ((el_z + r_z) - 1)$
  - falls  $r_x \neq 0 \neq r_y$  und kein anderer Knoten auf der Strecke zwischen  $k_1$  und  $k_2$  liegt
    - $(k_1, k_2)$  ist Stab

Es bleibt hierbei die Frage, wie man erkennt, dass kein weiterer Knoten zwischen 2-en liegt. Wir haben also

$$k_1 = el_y + n_y \cdot (el_x - 1) + n_x n_y (el_z - 1)$$

und

$$k_2 = (el_y + r_y) + n_y \cdot ((el_x + r_x) - 1) + n_x n_y ((el_z + r_z) - 1)$$

mit den Koordinaten

$$\begin{pmatrix} (el_x - 1) \cdot \frac{g_{x1} - g_{x0}}{n_x - 1} \\ (el_y - 1) \cdot \frac{g_{y1} - g_{y0}}{n_y - 1} \\ (el_z - 1) \cdot \frac{g_{z1} - g_{z0}}{n_z - 1} \end{pmatrix}$$

und

$$\begin{pmatrix} (el_x + r_x - 1) \cdot \frac{g_{x1} - g_{x0}}{n_x - 1} \\ (el_y + r_y - 1) \cdot \frac{g_{y1} - g_{y0}}{n_y - 1} \\ (el_z + r_z - 1) \cdot \frac{g_{z1} - g_{z0}}{n_z - 1} \end{pmatrix}.$$

Auf Grund der Annahme

$$\frac{g_{x1} - g_{x0}}{n_x - 1} = \frac{g_{y1} - g_{y0}}{n_y - 1} = \frac{g_{z1} - g_{z0}}{n_z - 1}$$

ergeben sich die Positionen aus

$$\begin{pmatrix} el_x - 1 \\ el_y - 1 \\ el_z - 1 \end{pmatrix}$$

und

$$\begin{pmatrix} el_x + r_x - 1 \\ el_y + r_y - 1 \\ el_z + r_z - 1 \end{pmatrix} = \begin{pmatrix} el_x - 1 \\ el_y - 1 \\ el_z - 1 \end{pmatrix} + \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix}.$$

Die Fälle

- $r_x = 0$ ,
- $r_y = 0$  oder
- $r_z = 0$

sind bereits durch Analogie zum 2-D-Fall erledigt.

Für

- $r_x \neq 0, r_y \neq 0$  und  $r_z > 0$

ist wieder die Richtung  $\begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix}$  entscheidend. Falls ein weiterer Knoten auf der Verbindung liegt, so existiert ein  $\lambda > 1$  mit

$$\begin{pmatrix} |r_x| \\ |r_y| \\ r_z \end{pmatrix} = \lambda \begin{pmatrix} |\tilde{r}_x| \\ |\tilde{r}_y| \\ \tilde{r}_z \end{pmatrix}$$

wobei  $|r_x|, |r_y|, r_z, |\tilde{r}_x|, |\tilde{r}_y|, \tilde{r}_z \in \mathbb{N}$ .

Mit anderen Worten:

Falls  $\text{ggT}(|r_x|, |r_y|, r_z) = 1$ , so liegt kein weiterer Knoten auf der Verbindung.

Mittels dem Euklidischen Algorithmus kann effizient der ggT von 2 Zahlen gefunden werden. Wegen Satz [G.1](#) aus Anhang [G](#) gilt:

$$\begin{aligned} \text{ggT}(|r_x|, |r_y|, r_z) &= \text{ggT}(\text{ggT}(|r_x|, |r_y|), \text{ggT}(|r_x|, r_z)) \\ &= \text{ggT}(\text{ggT}(|r_x|, |r_y|), \text{ggT}(|r_y|, r_z)) \\ &= \text{ggT}(\text{ggT}(|r_x|, r_z), \text{ggT}(|r_y|, r_z)) \\ &= \text{ggT}(\text{ggT}(|r_x|, |r_y|), r_z) \end{aligned}$$

#### C.4 Vernetzung des Bauraums durch Stäbe beim Fachwerk

Für  $n_x$  Knoten in  $x$ -Richtung,  $n_y$  in  $y$ -Richtung und  $n_z$  in  $z$ -Richtung ergibt sich die Anzahl Stäbe zu (bzw. letztlich weniger als  $\frac{n_x n_y n_z (n_x n_y n_z + 1)}{2}$  Stäbe):

$$\begin{aligned}
 & \sum_{x=1}^{n_x} \sum_{y=1}^{n_y} \sum_{z=1}^{n_z} \left( \begin{array}{l} \text{„2. Fall + 3. Fall + 4. Fall + “} \\ \text{„5. Fall + 6. Fall + 7. Fall + “} \\ \text{„8. Fall“} \end{array} \right) \\
 = & \sum_{x=1}^{n_x} \sum_{y=1}^{n_y} \sum_{z=1}^{n_z-1} 1 + \sum_{x=1}^{n_x} \sum_{y=1}^{n_y-1} \sum_{z=1}^{n_z} 1 + \sum_{x=1}^{n_x-1} \sum_{y=1}^{n_y} \sum_{z=1}^{n_z} 1 + \\
 & \sum_{x=1}^{n_x} \sum_{y=1}^{n_y} \sum_{z=1}^{n_z} \left( \begin{array}{l} \sum_{\substack{r_x=1-x, \\ r_x \neq 0}}^{n_x-x} \left( \begin{array}{l} \sum_{r_y=1}^{n_y-y} 1 + \sum_{r_z=1}^{n_z-z} 1 \\ \text{ggT}(r_x, r_y)=1 \quad \text{ggT}(r_x, r_z)=1 \end{array} \right) + \sum_{\substack{r_y=1-y, \\ r_y \neq 0}}^{n_y-y} \left( \begin{array}{l} \sum_{r_z=1}^{n_z-z} 1 \\ \text{ggT}(r_y, r_z)=1 \end{array} \right) \end{array} \right) + \\
 & \sum_{x=1}^{n_x} \sum_{y=1}^{n_y} \sum_{z=1}^{n_z} \left( \begin{array}{l} \sum_{\substack{r_x=1-x, \\ r_x \neq 0}}^{n_x-x} \quad \sum_{\substack{r_y=1-y, \\ r_y \neq 0}}^{n_y-y} \quad \sum_{r_z=1}^{n_z-z} \\ \text{ggT}(r_x, r_y, r_z)=1 \end{array} \quad 1 \right) \\
 \leq & \sum_{x=1}^{n_x} \sum_{y=1}^{n_y} \sum_{z=1}^{n_z-1} 1 + \sum_{x=1}^{n_x} \sum_{y=1}^{n_y-1} \sum_{z=1}^{n_z} 1 + \sum_{x=1}^{n_x-1} \sum_{y=1}^{n_y} \sum_{z=1}^{n_z} 1 + \\
 & \sum_{x=1}^{n_x} \sum_{y=1}^{n_y} \sum_{z=1}^{n_z} \left( \sum_{r_x=1-x}^{n_x-x-1} \sum_{r_y=1}^{n_y-y} 1 + \sum_{r_x=1-x}^{n_x-x-1} \sum_{r_z=1}^{n_z-z} 1 + \sum_{r_y=1-y}^{n_y-y-1} \sum_{r_z=1}^{n_z-z} 1 \right) + \\
 & \sum_{x=1}^{n_x} \sum_{y=1}^{n_y} \sum_{z=1}^{n_z} \left( \sum_{r_x=1-x}^{n_x-x-1} \sum_{r_y=1-y}^{n_y-y-1} \sum_{r_z=1}^{n_z-z} 1 \right) \\
 = & \frac{n_x n_y n_z (n_x (n_y - 1) (n_z + 1) + n_y (n_z - 3) + (15 - n_z))}{4} - n_y n_z - n_x n_z - n_x n_y \\
 \leq & \sum_{i=1}^{n_x n_y n_z} i = \frac{n_x n_y n_z (n_x n_y n_z + 1)}{2}
 \end{aligned}$$

Die Qualität dieser letzten groben oberen Schranke kann Abb. C.5 entnommen werden.

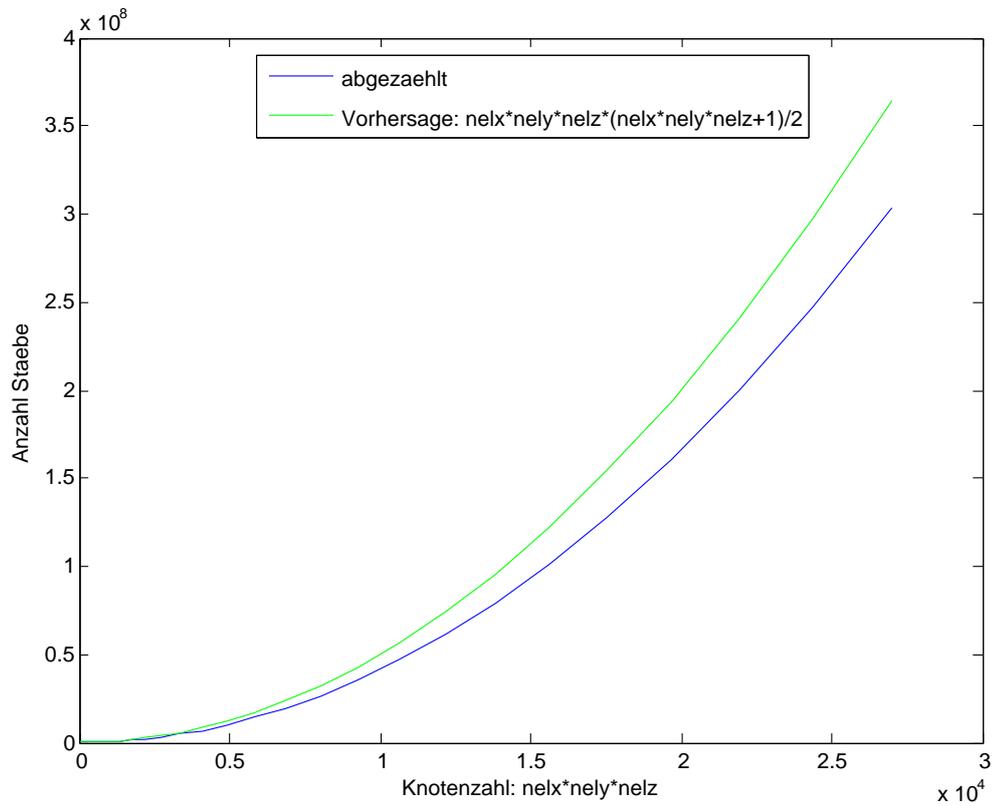


Abbildung C.5: Vergleich der Abschätzung der Anzahl Stäbe bei dieser Vernetzung und der exakten Anzahl

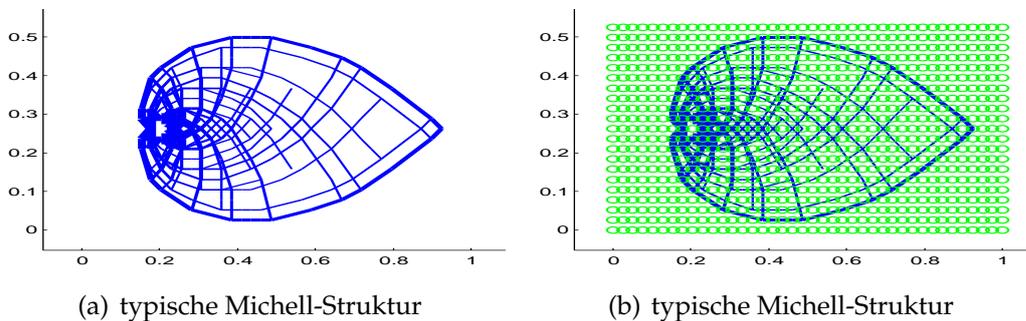


Abbildung C.6: Hier ist eine diskretisierte Lösung des Beispiels [Mic04, Fig. 1] dargestellt: eine typische Michell-Struktur. Sie wurde mit dem Vorgehen aus Kapitel C.3.2 berechnet und besteht aus 346 Stäben. In (b) sind zusätzlich die  $40 \cdot 21 = 840$  Knoten der Grundstruktur, die aus 215059 Stäben besteht, dargestellt. Rechts außen bei  $(0,9 \ 0,2625)^T$  wirkt eine Kraft nach unten und bei  $(0,2 \ 0,2625)^T$  ist ein verteiltes Festlager durch 5 gelagerte Knoten repräsentiert.

## C.5 Entwurfsmethode von Michell

Michell stellt in [Mic04] eine Methode zur Bestimmung einer ökonomischen Materialverteilung in einem „Fachwerk“<sup>37</sup> vor. Dieses prinzipielle Vorgehen wird heute als Entwurfsmethode von Michell (michell truss theory) bezeichnet und ist leider noch nicht vollständig verstanden [Sok11].

Michell hat sich unendlich viele Stäbe, die unendlich dünn sind, vorgestellt. Diese Stäbe müssen dann entlang eines Spannungsfeldes verlaufen. Dazu senkrechte Stäbe werden vernachlässigt. Als Zielfunktion in seinem Optimierungsproblem hat er das Volumen gewählt.

Dabei werden Materialeigenschaften unzureichend berücksichtigt (vgl. [Roz97], [Roz96], [Hem73]). Daher liefert diese Entwurfsmethode nur in einer sehr beschränkten Anzahl an Problemen die Optimallösung [Roz97].

Wenn man die unendlich vielen Stäbe diskretisiert und weitere Bedingungen von Hemp [Hem73] hinzu nimmt, so kann die Entwurfsmethode von Michell zu dem linearen Ansatz aus Kapitel C.3.2 weiterentwickelt werden (vgl. [Roz09], [Sve54], [Hem73], [DGG64]; Abb. C.6).

<sup>37</sup>Aus heutiger Sicht hat dies allerdings wenig mit einem Fachwerk zu tun.

Die Topologieoptimierung auf Basis der Homogenisierungsmethode (z. B. numerisch umgesetzt durch SIMP, vgl. Kapitel C.1) liefert bei geeignet niedrigen relativen Volumina (also geringes Volumen im Verhältnis zur Bauraumgröße) ebenfalls Michell-Strukturen; also das selbe Ergebnis wie die Entwurfsmethode von Michell (vgl. [BS04, Kapitel 3.3.3], [All01, Kapitel 4.2.3]).

Dazu wird in [All01, Kapitel 4.2.3] das Optimierungsproblem von Michell für einen Bauraum  $\Omega \subset \mathbb{R}^2$  für das Spannungsfeld  $\tau$  wie folgt dargestellt:

$$\inf_{\substack{\operatorname{div} \tau = 0 \text{ in } \Omega \\ \tau n = f \text{ auf } \partial\Omega}} \int_{\Omega} (|\tau_1| + |\tau_2|) dx$$

Hierbei sind  $\tau_i$  die Eigenwerte von  $\tau$  und  $f$  eine Last auf der Oberfläche. Die Zielfunktion ist also nicht das Volumen, aber unter gewissen Umständen zumindest proportional zum Volumen.

Die ursprüngliche Zielstellung war zwar die Beschreibung einer optimalen Struktur. Was aber bis heute an seinen Ergebnissen von Bedeutung ist, sind genau die Spannungsfelder, die Allaire in [All01, Kapitel 4.2.3] als die Entwurfstheorie von Michell wiedergibt.

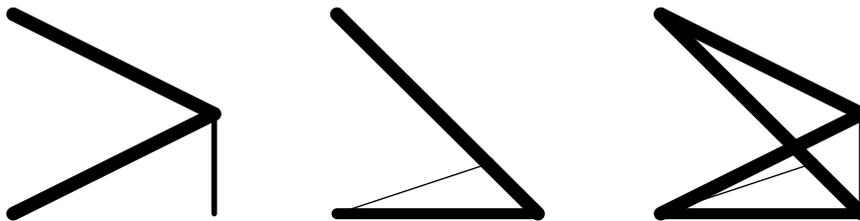
---

## D Stabwerk-Interpretation

Einige Strukturen aus Abb. 4.4 mit unterschiedlichen Redundanzgraden wurden in Abb. 4.6 als Stabwerk interpretiert. Diese interpretierten Strukturen sind nur augenscheinlich vergleichbar. Um konkrete Zahlenwerte als Maß zu finden, wurde die Compliance verschiedener Masseverteilungen berechnet – in jedem der 9 Bildern in Abb. D.1 wurde das gleiche Volumen und somit die gleiche Masse eingesetzt. ‚vol‘ steht für gleichmäßige Volumensverteilung auf jeden einzelnen Stab – ein kurzer Stab hat also die gleiche Masse wie ein langer Stab. ‚dicke‘ beschreibt die gleichmäßige Dicke aller Stäbe; gleichlange Stäbe haben somit die gleiche Masse. Bei ‚opt‘ wurde eine Sizing-Optimierung des Fachwerkes durchgeführt und – wie auch im Bild erkennbar – die Masse dementsprechend oberhalb einer Unterschranke optimal gewählt.

Man kann deutlich erkennen, dass keinerlei Vergleichbarkeit vorhanden ist. Bei der optimalen Materialverteilung wird die Redundanz in keiner Weise mehr berücksichtigt. Bei allen 3 Möglichkeiten der Masseverteilung werden die schlechten Knotenpositionen ersichtlich.

Eine Optimierung der ursprünglich einzelnen Teilstrukturen liefert auch keine Vergleichbarkeit:



- Die Optimierung bezieht sich lediglich auf eine Teilstruktur; das Zusammenwirken wird nicht berücksichtigt.
- Die Masseverteilung auf die beiden Teilstrukturen geschieht per Annahme zur Hälfte; dafür gibt es aber keinen Grund.
- Teilweise gibt es Stäbe, die zwangsweise zu mehr als einer Teilstruktur gehören.

## D Stabwerk-Interpretation

---

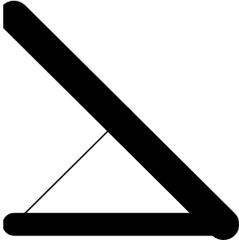
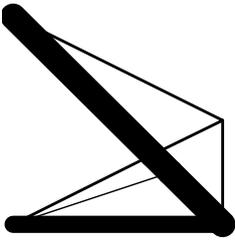
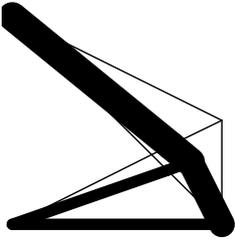
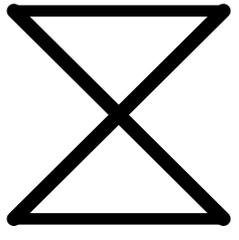
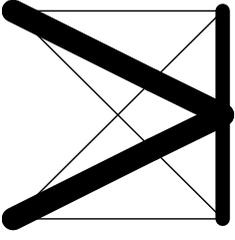
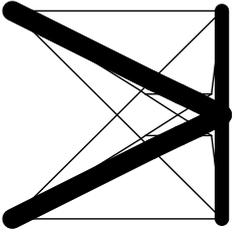
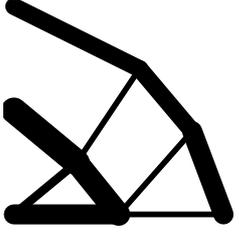
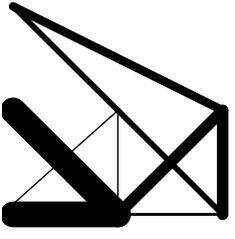
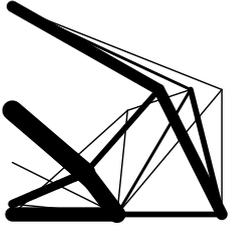
	$n = 1$	$n = 2$	$n = 3$
			
vol	0.00240	0.00257	0.00269
dicke	0.00239	0.00211	0.00178
opt	0.00181	0.00181	0.00155
			
vol	0.00720	0.00489	0.01178
dicke	0.00739	0.00451	0.00653
opt	0.00720	0.00415	0.00430
			
vol	0.00337	0.00546	0.00635
dicke	0.00345	0.00540	0.00369
opt	0.00285	0.00329	0.00299

Abbildung D.1: Verschiedene als Stabwerk interpretierte Strukturen in Abb. 4.6 aus Abb. 4.4 wurden hier mit unterschiedlichen Masseverteilungen optimal ausgelegt.

---

## E Optimierung unter binärer Ganzzahligkeitsbedingung

Das lineare Optimierungsproblem

$$\begin{aligned} \text{Zielfunktion: } & c^T \begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \min_x \\ \text{unter: } & A \begin{pmatrix} x \\ y \end{pmatrix} = b \end{aligned}$$

wird mit der binären Ganzzahligkeitsbedingung

$$y \in \{0, 1\}^{n_y}$$

zu einem NP-vollständigen Problem.

Am Beispiel eines allgemeinen Optimierungsproblems

$$\begin{aligned} \text{Zielfunktion: } & z(x, y) \rightarrow \min_x \\ \text{unter: } & g(x, y) = 0 \end{aligned}$$

mit der binären Ganzzahligkeitsbedingung

$$y \in \{0, 1\}^{n_y}$$

wird nun die Grundidee eines „Branch and Bound“-Algorithmus (vgl. [Dak65] oder [LD60]) erklärt. Dazu sei das Optimierungsproblem kurz als  $p(Y)$  mit  $Y = \{0, 1\}^{n_y}$  bezeichnet. Für verschiedene  $Y$  ergeben sich damit verschiedene Instanzen des selben Problems.

Das Aufstellen eines binären Entscheidungsbaums, also eines Binärbaums, ermöglicht alle zulässigen Punkte zu beschreiben. Dazu wird ab der Wurzel in jedem Knoten  $i$  ein Index  $j_i$  gewählt, zu dem  $y_{j_i}$  zu 0 oder 1 gewählt wird; auf dem Weg von der Wurzel zu einem Blatt werden dabei alle  $y_i$  gewählt. Alle  $2^{n_y}$  Blätter zusammen beschreiben also alle Möglichkeiten für die Belegung von  $y$  mit Werten (vgl. Beispiel in Abb. E.1). Für jedes Blatt kann nun eine entsprechende Instanz  $p$  betrachtet werden. Alle Blätter zu lösbaren Instanzen beschreiben nun alle Instanzen mit zulässigen Lösungen. Unter diesen kann durch direkten Vergleich der beste Wert gefunden werden – er stellt die Lösung des Ausgangsproblems  $p(\{0, 1\}^{n_y})$  dar.

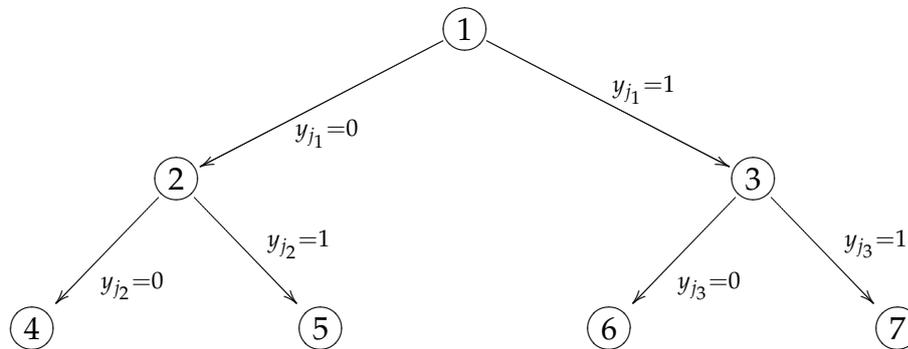


Abbildung E.1: Beispiel eines vollständigen Binärbaums der Tiefe 2

**Branch** Den gesamten Baum aufzustellen ist natürlich viel zu aufwendig. Dies geschieht rekursiv durch Lösen von Ersatzproblemen in jedem Knoten, angefangen bei der Wurzel. In der Wurzel wird das ganzzahlige Optimierungsproblem  $p(\{0, 1\}^{n_y})$  relaxiert zu  $p([0, 1]^{n_y})$ . In inneren Knoten ergibt der Weg von der Wurzel zu diesem bereits eine Einschränkung gewisser Komponenten von  $y$ ; die restlichen bleiben relaxiert im Intervall. Auf diese Weise wird jedem Knoten rekursiv eine Instanz von  $p$  zugewiesen.

- Im aktuellen Knoten wird versucht, die zugehörige Instanz zu lösen:
  - Falls eine Lösung möglich ist und im Optimalpunkt ein  $0 \neq y_{j_i} \neq 1$  existiert, so kann mit der Wahl 0 oder 1 dieser Komponente der Baum fortgesetzt werden und 2 weitere Knoten entstehen.
  - Falls eine Lösung möglich ist und im Optimalpunkt alle Komponenten  $y$  entweder 0 oder 1 sind, so ist eine zulässige Lösung mit entsprechendem Optimalwert gefunden. Der Baum muss ab diesem Knoten nicht mehr fortgesetzt werden – entweder es ist bereits ein Knoten der Tiefe  $n_y$  oder es ist bereits der Optimalwert aller Blätter des zugehörigen Unterbaums gefunden.
  - Falls keine Lösung möglich ist, so kann der Baum an diesem Knoten nicht mehr sinnvoll fortgesetzt werden.
- An einem anderen Knoten wird fortgesetzt.

Der so entstandene unvollständige Binärbaum beschreibt mit jedem Blatt entweder eine unzulässige Belegung von  $y$  oder einen zulässigen Punkt

---

mit einem Optimalwert. Unter diesen kann nun das Optimum gewählt werden.

**Bound** Nachdem mit dem beschriebenen Vorgehen ein zulässiger Punkt gefunden wurde, kann der zugehörige Zielfunktionswert im weiteren als Oberschranke genutzt werden und somit wird der abzusuchende Baum nochmals verkleinert.

**Cut** Bei linearen Optimierungsproblemen kann der Baum durch weitere Schranken aus Schnittebenen (z. B. die bekannten aber numerisch aufwendigen und daher instabilen Gomory-Schnitte nach [Gom58]) nach jedem lösbaren, relaxierten und unzulässigen Ersatzproblem weiter reduziert werden.

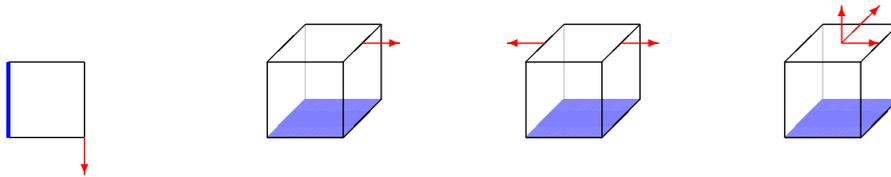
Aus der Kombination dieser 3 typischen Vorgehensweisen resultieren effiziente Algorithmen (vgl. [Mak10, GLPK]).



---

## F Vergleich der Redundanz $\frac{1}{n}$ und $1 - \frac{1}{n}$ beim Stabwerk

Im folgenden werden einige Ergebnisse von einfachen Beispielen, die berechnet wurden, wiedergegeben. Diese Beispiele sind leicht nachvollziehbar und ermöglichen somit die Redundanz zu erkennen. Folgende Bilder repräsentieren die Beispiele:



Als Bauraum wurde jeweils ein 2-dimensionaler bzw. ein 3-dimensionaler Würfel gewählt. Blau ist der Bereich gekennzeichnet, der gelagert ist. In rot wurden die Kräfte eingezeichnet. Mehrere Kraftpfeile stellen einen mehrfachen Lastfall dar. Auf Grund der Einfachheit dieser Beispiele kann leicht die Optimallösung eingesehen werden. Der zweifache Lastfall unterscheidet sich wesentlich von der Situation, in der beide Kräfte gleichzeitig wirken. Die maximalen Feinheiten der Grundstrukturn wurden an die verfügbare Rechenkapazität angepasst. Trotzdem beträgt die Gesamt-rechenzeit aller in diesem Anhang F wiedergegebenen Beispiele mehrere Jahre auf einem CPU-Kern mit 2,4 GHz. Der maximale Arbeitsspeicher-verbrauch lag bei ca. 50 GB für einzelne Beispiele.

### F.1 2-dim. Strukturen

In der folgenden Tabelle sind verschiedene Ergebnisse aus dem Ansatz von Kapitel 4.2 zu sehen. Jeweils unten rechts greift eine Last an, die senkrecht nach unten zieht. Die linke Seite ist gelagert. In der linken Spalte ist die Diskretisierung angegeben mit  $nel \times$ . Ein Wert von 2 bedeutet, dass 2 mal 2 Knoten gewählt wurden und dazwischen jeder Knoten mit jedem verbunden wurde. Die verschiedenen Farben stehen für verschiedene Teilstrukturen. Die Gesamtstruktur ist also jeweils die Überlagerung aller Farben. Unter den Bildern ist jeweils das Volumen angegeben. Für einen Vergleich dieser Strukturen untereinander spielt die Einheit bzw. Skalierung dieses Volumens keine Rolle.

Redundanz	0	$\frac{1}{5}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$	$1 - \frac{1}{2}$	$1 - \frac{1}{3}$	$1 - \frac{1}{4}$	$1 - \frac{1}{5}$
nelx=2 4 Kn., 6 St.	 300	 700	 700	 700	 700	 700	 0	 0	 0
nelx=3 9 Kn., 28 St.	 300	 512	 512	 525	 600	 600	 1083	 0	 1952
nelx=4 16 Kn., 86 St.	 283	 446	 468	 572	 572	 936	 1356	 1609	
nelx=5 25 Kn., 200 St.	 275	 409	 444	 558	 558	 888	 1228	 1609	
nelx=6 36 Kn., 418 St.	 273	 392	 429	 553	 553	 859	 1176	 1609	
nelx=7 49 Kn., 748 St.	 274	 421	 551	 551	 842	 1176	 1609	 1609	

Redundanz	0	$\frac{1}{5}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$	$1 - \frac{1}{2}$	$1 - \frac{1}{3}$	$1 - \frac{1}{4}$	$1 - \frac{1}{5}$
nelx=8 64 Kn., 1282 St.	 274			 417	 549	 549	 834		
nelx=9 81 Kn., 2040 St.	 273			 416	 548	 548	 832		

In der folgenden Tabelle ist jeweils die Greedy-Lösung nach Kapitel 4.2.4 und die nicht-approximative Lösung untereinander dargestellt:

Redundanz	0	$\frac{1}{5}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$	$1 - \frac{1}{2}$	$1 - \frac{1}{3}$	$1 - \frac{1}{4}$	$1 - \frac{1}{5}$
greedy, nelx=2 4 Kn., 6 St.									
		700	700	700	700	700			
nelx=2									
	300	700	700	700	700	700	0	0	0
greedy, nelx=3 9 Kn., 28 St.									
		600	750	600	600	600	1083		
nelx=3									
	300	512	512	525	600	600	1083	0	

Redundanz	0	$\frac{1}{5}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$	$1 - \frac{1}{2}$	$1 - \frac{1}{3}$	$1 - \frac{1}{4}$	$1 - \frac{1}{5}$
greedy, nelx=4 16 Kn., 86 St.									
		526	503	554	572	635	936	1563	2233
nelx=4									
	283	446	468	572	572	936	1356	1952	
greedy, nelx=5 25 Kn., 200 St.									
		414	443	448	558	559	894		
nelx=5									
	275	409	444	558	558	888	1228	1609	
greedy, nelx=6 36 Kn., 418 St.									
		396	434	553	553	872			
nelx=6									
	273	392	429	553	553	859	1176		

Redundanz	0	$\frac{1}{5}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$	$1 - \frac{1}{2}$	$1 - \frac{1}{3}$	$1 - \frac{1}{4}$	$1 - \frac{1}{5}$
greedy, nelx=7 49 Kn., 748 St.									
				421	552	552	842		
nelx=7									
	274			421	551	551	842		
greedy, nelx=8 64 Kn., 1282 St.									
					549	567	839		
nelx=8									
	274			417	549	549	834		
greedy, nelx=9 81 Kn., 2040 St.									
				420	549	549	835		
nelx=9									
	273			416	548	548	832		

## F.2 3-dim. Strukturen

In den folgenden 3-dimensionalen Beispielen ist jeweils der Boden gelagert.

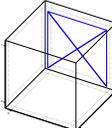
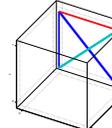
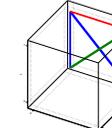
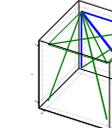
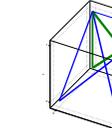
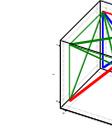
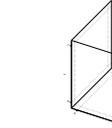
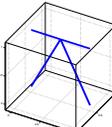
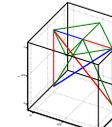
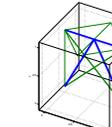
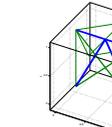
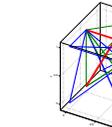
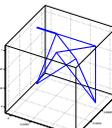
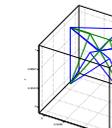
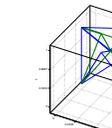
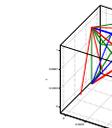
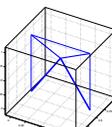
### F.2.1 Redundanz $\frac{1}{n}$ vgl. $1 - \frac{1}{n}$ : kran1x1 lastfall1

In diesen Darstellungen oben links wirkt eine horizontale Kraft.

Redundanz	0	$\frac{1}{5}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$	$1 - \frac{1}{2}$	$1 - \frac{1}{3}$	$1 - \frac{1}{4}$	$1 - \frac{1}{5}$
nelx=2 8 Kn., 22 St.									
	300	600	600	600	700	700	1200	2200	0
nelx=3 27 Kn., 274 St.									
	300	425	433	475	600	600	950	1300	
nelx=4 64 Kn., 1630 St.									
	283			439	572	572	878		
nelx=5 125 Kn., 6212 St.									
	275				558	558	855		

## F.2.2 Redundanz $\frac{1}{n}$ vgl. $1 - \frac{1}{n}$ : kran1x1 lastfall2

In diesen Darstellungen wirken oben links oder rechts horizontale Kräfte abwechselnd.

Redundanz	0	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$	$1 - \frac{1}{2}$	$1 - \frac{1}{3}$	$1 - \frac{1}{4}$	
nelx=2 8 Kn., 22 St.								
	350	700	700	1150	1150	2300	0	
nelx=3 27 Kn., 274 St.								
	350		625	800	800	1350		
nelx=4 64 Kn., 1630 St.								
	333			722	722	1128		
nelx=5 125 Kn., 6212 St.								
	325							

### F.2.3 Redundanz $\frac{1}{n}$ vgl. $1 - \frac{1}{n}$ : kran1x1 lastfall3

In diesen Darstellungen wirken oben in der Mitte abwechselnd Kräfte in alle 3 Raumrichtungen.

Redundanz	0	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$	$1 - \frac{1}{2}$	$1 - \frac{1}{3}$	$1 - \frac{1}{4}$
nelx=2 8 Kn., 22 St.							
	400	750	800	1100	1100	0	0
nelx=3 27 Kn., 274 St.							
	300		546	700	700	1150	
nelx=4 64 Kn., 1630 St.							
	311						
nelx=5 125 Kn., 6212 St.							
	300						

F Vergleich der Redundanz  $\frac{1}{n}$  und  $1 - \frac{1}{n}$  beim Stabwerk

---

---

## G Trivialitäten

In diesem Anhang werden ein paar wissenswerte Kleinigkeiten gesammelt.

**Definition G.1 (Größter gemeinsamer Teiler ggT)** Seien  $a_i \in \mathbb{N}$ ,  $i = 1, 2, \dots, N$  natürliche Zahlen mit der Primfaktorzerlegung

$$a_i = \prod_{j=1}^M p_j^{e_{i,j}}, e_{i,j} \in \mathbb{N}_0, p_j \in \mathbb{P} \text{ für } i = 1, 2, \dots, N.$$

Dann heißt

$$\text{ggT}(a_1, a_2, \dots, a_N) := \prod_{j=1}^M p_j^{\min_i e_{i,j}}$$

der größte gemeinsame Teiler der Zahlen  $a_i$ . □

**Satz G.1 (ggT von mehreren Zahlen durch ggT von 2 Zahlen)** Seien  $a, b, c \in \mathbb{N}$  natürliche Zahlen mit den Primfaktorzerlegungen

$$a = \prod_{j=1}^M p_j^{e_{a,j}}; \quad b = \prod_{j=1}^M p_j^{e_{b,j}}; \quad c = \prod_{j=1}^M p_j^{e_{c,j}}; \quad p_j \in \mathbb{P}.$$

Dann gilt:

$$\begin{aligned} \text{ggT}(a, b, c) &= \text{ggT}(\text{ggT}(a, b), \text{ggT}(a, c)) \\ &= \text{ggT}(\text{ggT}(a, b), \text{ggT}(b, c)) \\ &= \text{ggT}(\text{ggT}(a, c), \text{ggT}(b, c)) \\ &= \text{ggT}(\text{ggT}(a, b), c) \end{aligned}$$

□

BEWEIS (VON SATZ [G.1](#))

$$\begin{aligned} \text{ggT}(a, b, c) &= \prod_{j=1}^M p_j^{\min\{e_{a,j}, e_{c,j}, e_{b,j}\}} \\ &= \text{ggT}(\text{ggT}(a, b), \text{ggT}(a, c)) \\ &= \text{ggT}(\text{ggT}(a, b), \text{ggT}(b, c)) \\ &= \text{ggT}(\text{ggT}(a, c), \text{ggT}(b, c)) \\ &= \text{ggT}(\text{ggT}(a, b), c) \end{aligned}$$

■

**Definition G.2 (Kronecker-Delta  $\delta_{a,b}$ )** Für Elemente  $a, b$  einer beliebigen Indexmenge  $I$  heißt  $\delta_{a,b}$  mit

$$\delta_{a,b} := \begin{cases} 1 & \text{falls } a = b \\ 0 & \text{sonst} \end{cases}$$

Kronecker-Delta. □

Mehrere Kronecker-Deltas zusammen können nicht ohne weiteres zusammengefasst werden. Beispielsweise ist in

$$\delta_{a,b}\delta_{a,c} = \begin{cases} 1 & \text{falls } a = b = c \\ 0 & \text{sonst} \end{cases}$$

die Gleichheit aller 3 Indizes für den Wert 1 notwendig und daher gilt im Allgemeinen:

$$\delta_{a,b}\delta_{a,c} = \delta_{a,b}\delta_{b,c} \neq \delta_{b,c}$$

Für eine beliebige Indexmenge  $I$  gilt bei einer Summierung  $\sum_{a \in I} \delta_{a,b}\delta_{a,c}$ ,  $b, c \in I$  genau bei einem Summanden  $a = b$  und für einen evtl. anderen Summanden  $a = c$ . Somit kann dies zusammengefasst werden:

$$\sum_{a \in I} \delta_{a,b}\delta_{a,c} = \delta_{b,c}$$

---

## Literatur

- [Ach93] ACHTZIGER, Wolfgang: *Optimierung von einfach und mehrfach belasteten Stabwerken*. Universität Bayreuth, 1993 (Bayreuther Mathematische Schriften 46). ISSN 0172–1062. – Doktorarbeit
- [AH01] ALLAIRE, Grégoire ; HENROT, Antoine: On some recent advances in shape optimization. In: *Comptes Rendus de l'Académie des Sciences. Série Iib. Mécanique* 329 (2001), Nr. 5, S. 383–396. [http://dx.doi.org/10.1016/S1620-7742\(01\)01349-6](http://dx.doi.org/10.1016/S1620-7742(01)01349-6). – DOI 10.1016/S1620-7742(01)01349-6
- [All01] ALLAIRE, G.: *Shape Optimization by the Homogenization Method*. Springer, 2001 (Applied Mathematical Sciences 146). – ISBN 9781441929426
- [BK88] BENDSØE, Martin P. ; KIKUCHI, Noboru: Generating optimal topologies in structural design using a homogenization method. In: *Computer Methods in Applied Mechanics and Engineering* 71 (1988), Nr. 2, 197 - 224. [http://dx.doi.org/10.1016/0045-7825\(88\)90086-2](http://dx.doi.org/10.1016/0045-7825(88)90086-2). – DOI 10.1016/0045-7825(88)90086-2. – ISSN 0045–7825
- [BLZ09] BHATTACHARYA, Baidurya ; LU, Qiang ; ZHONG, Jinquan: Reliability of redundant ductile structures with uncertain system failure criteria. In: *Sadhana* 34 (2009), 903-921. <http://dx.doi.org/10.1007/s12046-009-0054-5>. – DOI 10.1007/s12046-009-0054-5. – ISSN 0256–2499
- [BMPR11] BELZUNCE, Félix ; MARTÍNEZ-PUERTAS, Helena ; RUIZ, José M.: On optimal allocation of redundant components for series and parallel systems of two dependent components. In: *Journal of Statistical Planning and Inference* 141 (2011), Nr. 9, 3094 - 3104. <http://dx.doi.org/10.1016/j.jspi.2011.03.031>. – DOI 10.1016/j.jspi.2011.03.031. – ISSN 0378–3758
- [Bro98] BROY, Manfred: *Informatik – Eine grundlegende Einführung, Band 2: Systemstrukturen und Theoretische Informatik*. Heidelberg : Springer-Verlag, 1998. – ISBN 3–540–64392–3
- [BS04] BENDSØE, Martin P. ; SIGMUND, Ole: *Topology Optimization*. Berlin, Heidelberg : Springer, 2004. – ISBN 3–540–42992–1

## LITERATUR

---

- [BS07] BEYER, Hans-Georg ; SENDHOFF, Bernhard: Robust optimization — A comprehensive survey. In: *Comput. Meth. Appl. Mech. Eng.* 196 (2007), Nr. 33-34, S. 3190–3218. <http://dx.doi.org/10.1016/j.cma.2007.03.003>. – DOI 10.1016/j.cma.2007.03.003
- [BtKNZ97] BEN-TAL, Aharon ; KOČVARA, Michal ; NEMIROVSKI, Arkadi ; ZOWE, Jochem: Free Material Design via semidefinite programming, Multi-Load Case with Contact. In: *Conditions, Optimization Laboratory, Faculty of Industrial Engineering and Management, Technion - Israel Institute of Technology* 9 (1997), S. 813–832
- [BTN98] BEN-TAL, Aharon ; NEMIROVSKI, Arkadi: Robust Convex Optimization. In: *Mathematics of Operations Research* 23 (1998), November, Nr. 4, S. 769–805. <http://dx.doi.org/10.1287/moor.23.4.769>. – DOI 10.1287/moor.23.4.769
- [CCL10] CHEN, Shikui ; CHEN, Wei ; LEE, Sanghoon: Level set based robust shape and topology optimization under random field uncertainties. In: *Structural and Multidisciplinary Optimization* 41 (2010), 507-524. <http://dx.doi.org/10.1007/s00158-009-0449-2>. – DOI 10.1007/s00158-009-0449-2. – ISSN 1615-147X
- [CD08] CALAFIORE, Giuseppe ; DABBENE, Fabrizio: Optimization under uncertainty with applications to design of truss structures. In: *Struct. Multidisc. Optim.* 35 (2008), S. 189–200. <http://dx.doi.org/10.1007/s00158-007-0145-z>. – DOI 10.1007/s00158-007-0145-z. – ISSN 1615-147X
- [CGM73] CEA, Jean ; GIOAN, A. ; MICHEL, J.: Quelques résultats sur l'identification de domaines. In: *Calcolo* 10 (1973), S. 133–145
- [Che11] CHERKAEV, Andrej: Optimal Three-Material Wheel Assemblage of Conducting and Elastic Composites. In: *arXiv – Mathematical Physics* (2011). <http://arxiv.org/abs/1105.4302v1>. – arXiv:1105.4302v1 [math-ph]
- [CKR06] CAMPS, Thorsten ; KÜHLING, Stefan ; ROSENBERGER, Gerhard: *Berliner Studienreihe zur Mathematik*. Bd. 15: *Einführung in die mengentheoretische und die algebraische Topologie*. Lemgo : Heldermann, 2006. – ISBN 3-88538-115-X

- [Dak65] DAKIN, Robert J.: A tree-search algorithm for mixed integer programming problems. In: *The Computer Journal* 8 (1965), 250–255. <http://dx.doi.org/10.1093/comjnl/8.3.250>. – DOI 10.1093/comjnl/8.3.250
- [DGG64] DORN, William S. ; GOMORY, Ralph E. ; GREENBERG, H. J.: Automatic design of optimal structures. In: *J. Mecanique* 3 (1964), Nr. 1, S. 25–52
- [DK04] DOLTSINIS, Ioannis ; KANG, Zhan: Robust design of structures using optimization methods. In: *Comput. Meth. Appl. Mech. Eng.* 193 (2004), S. 2221–2237. <http://dx.doi.org/10.1016/j.cma.2003.12.055>. – DOI 10.1016/j.cma.2003.12.055
- [Dud08] DUDDECK, Fabian: Multidisciplinary optimization of car bodies. In: *Struct Multidisc Optim* 35 (2008), S. 375–389. <http://dx.doi.org/10.1007/s00158-007-0130-6>. – DOI 10.1007/s00158-007-0130-6
- [EKS94] ESCHENAUER, Hans A. ; KOBELÉV, Vladimir V. ; SCHUHMACHER, A.: Bubble method for topology and shape optimization of structures. In: *Structural Optimization* 8 (1994), Nr. 1, S. 42–51. <http://dx.doi.org/10.1007/BF01742933>. – DOI 10.1007/BF01742933
- [EO01] ESCHENAUER, Hans A. ; OLHOFF, Niels: Topology optimization of continuum structures: A review. In: *Applied Mechanics Reviews* 54 (2001), Nr. 4, S. 331–390. <http://dx.doi.org/10.1115/1.1388075>. – DOI 10.1115/1.1388075
- [GF07] GROTE, Karl-Heinrich (Hrsg.) ; FELDHUSEN, Jörg (Hrsg.): *DUBBEL – Taschenbuch für den Maschinenbau*. 22. Berlin : Springer, 2007 <http://dx.doi.org/10.1007/978-3-540-68191-5>. – ISBN 978-3-540-49714-1
- [Gom58] GOMORY, Ralph E.: Outline of an algorithm for integer solutions to linear programs. In: *Bulletin of the American Mathematical Society* 64 (1958), S. 275–278. <http://dx.doi.org/10.1090/S0002-9904-1958-10224-4>. – DOI 10.1090/S0002-9904-1958-10224-4
- [Haf83] HAFTKA, Raphael T.: Damage Tolerant Design Using Collapse Techniques. In: *AIAA Journal* 21 (1983), Nr. 10, S. 1462–1466. <http://dx.doi.org/10.2514/3.8268>. – DOI 10.2514/3.8268

## LITERATUR

---

- [Ham50] HAMMING, Richard W.: Error detecting and error correcting codes. 1950 (2). – Forschungsbericht. – 147–160 S. – The Bell System Technical Journal
- [Hem73] HEMP, W. S.: *Optimum structures*. Oxford : Clarendon Press, 1973
- [HQ95] HEISE, Werner ; QUATTROCCHI, Pasquale: *Informations- und Codierungstheorie*. Berlin, Heidelberg : Springer, 1995. – ISBN 3–540–57477–8
- [HY02] HOSHIYA, Masaru ; YAMAMOTO, Kinya: Redundancy Index of Lifeline Systems. In: *Journal of Engineering Mechanics* 128 (2002), Nr. 9, 961-968. [http://dx.doi.org/10.1061/\(ASCE\)0733-9399\(2002\)128:9\(961\)](http://dx.doi.org/10.1061/(ASCE)0733-9399(2002)128:9(961)). – DOI 10.1061/(ASCE)0733-9399(2002)128:9(961). – ISSN 07339399
- [Kar72] KARP, Richard M.: Reducibility Among Combinatorial Problems. In: MILLER, Raymond E. (Hrsg.) ; THATCHER, James W. (Hrsg.): *Complexity of Computer Computations*. New York : Plenum Press, 1972 (The IBM Research Symposia Series). – ISBN 0–306–30707–3, S. 85–103
- [KB97] KRÄTZIG, Wilfried B. ; BAŞAR, Yuvez: *Tragwerke 3. Theorie und Anwendung der Methode der finiten Elemente*. Berlin, Heidelberg, New York : Springer, 1997. – ISBN 3–540–62440–6
- [KBH11] KANNO, Yoshihiro ; BEN-HAIM, Yakov: Redundancy and Robustness, or When Is Redundancy Redundant? In: *Journal of Structural Engineering* 137 (2011), Nr. 9, 935-945. [http://dx.doi.org/10.1061/\(ASCE\)ST.1943-541X.0000416](http://dx.doi.org/10.1061/(ASCE)ST.1943-541X.0000416). – DOI 10.1061/(ASCE)ST.1943-541X.0000416. – ISSN 07339445
- [KKÖ10] KAYA, Necmettin ; KAREN, Idris ; ÖZTÜRK, Ferruh: Re-design of a failed clutch fork using topology and shape optimisation by the response surface method. In: *Materials and Design* 31 (2010), Nr. 6, 3008 - 3014. <http://dx.doi.org/10.1016/j.matdes.2010.01.002>. – DOI 10.1016/j.matdes.2010.01.002. – ISSN 0261–3069
- [KV08] KORTE, Bernhard ; VYGEN, Jens: *Kombinatorische Optimierung*. Berlin : Springer, 2008. – ISBN 978–3–540–76918–7

- [LD60] LAND, Ailsa H. ; DOIG, Alison G.: An automatic method of solving discrete programming problems. In: *Econometrica* 28 (1960), Juli, Nr. 3, 467-520. <http://www.jstor.org/stable/1910129>
- [Mak10] MAKHORIN, Andrew ; DEPARTMENT FOR APPLIED INFORMATICS, MOSCOW AVIATION (Hrsg.): *GNU Linear Programming Kit*. Version 4.45. Moscow: Department for Applied Informatics, Moscow Aviation, Dezember 2010. <http://www.gnu.org/software/glpk/glpk.html>
- [Mar03] MARTI, Kurt: Stochastic optimization methods in optimal engineering design under stochastic uncertainty. In: *ZAMM* 83 (2003), Nr. 12, S. 795–811. <http://dx.doi.org/10.1002/zamm.200310072>. – DOI 10.1002/zamm.200310072
- [MF03] MAUTE, Kurt ; FRANGOPOL, Dan M.: Reliability-based design of MEMS mechanisms by topology optimization. In: *Computers and Structures* 81 (2003), Nr. 8-11, 813 - 824. [http://dx.doi.org/10.1016/S0045-7949\(03\)00008-7](http://dx.doi.org/10.1016/S0045-7949(03)00008-7). – DOI 10.1016/S0045-7949(03)00008-7. – ISSN 0045-7949. – K.J. Bathe 60th Anniversary Issue
- [MG06] MARTI, Kurt ; GRÖGER, Detlef: *Stochastische Strukturoptimierung von Stab- und Balkentragwerken*. Berlin, Heidelberg : Springer, 2006. – ISBN 3-540-26038-2
- [Mic04] MICHELL, Anthony George M.: The limits of economy of material in frame structures. In: *Phil. Mag.* 8 (1904), Nr. 47, S. 589–597. <http://dx.doi.org/10.1080/14786440409463229>. – DOI 10.1080/14786440409463229
- [MSMK11] MOHR, Daniel P. ; STEIN, Ina ; MATZIES, Thomas ; KNAPEK, Christina A.: Robust Topology Optimization of Truss with regard to Volume. In: *arXiv – Mathematics, Optimization and Control* (2011). <http://arxiv.org/abs/1109.3782v1>. – arXiv:1109.3782v1 [math.OC]
- [OAH96] OBERNDORFER, J. M. ; ACHTZIGER, Wolfgang ; HÖRNLEIN, Herbert R. E. M.: Two approaches for truss topology optimization: a comparison for practical use. In: *Structural and Multidisciplinary Optimization* 11 (1996), S. 137–144. <http://dx.doi.org/10.1007/BF01934231>

## LITERATUR

---

- [org/10.1007/BF01197027](http://dx.doi.org/10.1007/BF01197027). – DOI 10.1007/BF01197027. – ISSN 1615–147X
- [Ozt11] OZTURK, Umud: Efficient method for fatigue based shape optimization of the oil sump carrying a differential case in four wheel drive vehicles. In: *Structural and Multidisciplinary Optimization* (2011), 1-8. <http://dx.doi.org/10.1007/s00158-011-0678-z>. – DOI 10.1007/s00158–011–0678–z. – ISSN 1615–147X
- [PGK87] PATTERSON, David A. ; GIBSON, Garth A. ; KATZ, Randy H.: A Case for Redundant Arrays of Inexpensive Disks (RAID). Version: Dec 1987. <http://www.eecs.berkeley.edu/Pubs/TechRpts/1987/5853.html>. 1987 (UCB/CSD-87-391). – Forschungsbericht. – EECS Department, University of California, Berkeley
- [Prz68] PRZEMIENIECKI, Janusz S.: *Theory of matrix structural analysis*. New York : McGraw-Hill, Inc., 1968
- [Rez61] REZA, Fazlollah M.: *An Introduction to Information Theory*. New York, Toronto, London : McGraw-Hill, Inc., 1961
- [Roz96] ROZVANY, G. I. N.: Some shortcomings in Michell’s truss theory. In: *Structural and Multidisciplinary Optimization* 12 (1996), 244-250. <http://dx.doi.org/10.1007/BF01197364>. – DOI 10.1007/BF01197364. – ISSN 1615–147X
- [Roz97] ROZVANY, G. I. N.: Some shortcomings in Michell’s truss theory. In: *Structural and Multidisciplinary Optimization* 13 (1997), 203-204. <http://dx.doi.org/10.1007/BF01199243>. – DOI 10.1007/BF01199243. – ISSN 1615–147X
- [Roz09] ROZVANY, G.: A critical review of established methods of structural topology optimization. In: *Structural and Multidisciplinary Optimization* 37 (2009), 217-237. <http://dx.doi.org/10.1007/s00158-007-0217-0>. – DOI 10.1007/s00158–007–0217–0. – ISSN 1615–147X
- [RT73] ROSSOW, M. P. ; TAYLOR, J. E.: A finite element method for the optimal design of variable thickness sheets. In: *AIAA* 11 (1973), S. 1566–1569. <http://dx.doi.org/10.2514/3.50631>. – DOI 10.2514/3.50631

- [Rüe94] RÜEGG, Alan: *Wahrscheinlichkeitsrechnung und Statistik*. München : R. Oldenbourg Verlag GmbH, 1994. – ISBN 3–486–23038–7
- [Sha48] SHANNON, Claude E.: A Mathematical Theory of Communication. Version: 1948. <http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html>. 1948 (1). – Forschungsbericht. – 379–423, 623–656 S. – The Bell System Technical Journal
- [Sig01] SIGMUND, Ole: A 99 line topology optimization code written in Matlab. In: *Structural and Multidisciplinary Optimization* 21 (2001), Nr. 2, 120 - 127. [http://www.topopt.dtu.dk/cgi-bin/matlab/matlab\\_counter.cgi](http://www.topopt.dtu.dk/cgi-bin/matlab/matlab_counter.cgi)
- [Sok11] SOKÓŁ, Tomasz: A 99 line code for discretized Michell truss optimization written in Mathematica. In: *Structural and Multidisciplinary Optimization* 43 (2011), 181-190. <http://dx.doi.org/10.1007/s00158-010-0557-z>. – DOI 10.1007/s00158-010-0557-z. – ISSN 1615–147X
- [Soy73] SOYSTER, Allen L.: Convex Programming with Set-Inclusive Constraints and Applications to Inexact Linear Programming. In: *Operations Research* 21 (1973), Sep. - Oct., Nr. 5, S. 1154–1157. <http://dx.doi.org/10.1287/opre.21.5.1154>. – DOI 10.1287/opre.21.5.1154
- [Sto89] STOER, Josef: *Numerische Mathematik 1*. Berlin, Heidelberg, New York : Springer-Verlag, 1989. – ISBN 3–540–51481–3
- [Stö03] STÖCKL, Gerald: Optimaler Entwurf elastoplastischer mechanischer Strukturen unter stochastischer Unsicherheit. In: *Fortschritt-Berichte VDI* 18 (2003), Nr. 278
- [Sve54] SVED, G.: The minimum weight of certain redundant structures. In: *Australian Journal Of Applied Science* 5 (1954), S. 1–8
- [Tod84] TODOROV, Kalcho: Measure of the information redundancy for some semigroups of transformations of a finite set. In: *God. Vissh. Uchebn. Zaved., Prilozhna Mat.* 20 (1984), Nr. 2, S. 9–25
- [VDI07] *Kapitel VDI 4003: Zuverlässigkeitsmanagement / Reliability management*. In: VDI-GESELLSCHAFT SYSTEMENTWICKLUNG UND PROJEKTGESTALTUNG (Hrsg.): *VDI-Handbuch Zuverlässigkeit / VDI manual Reliability*. Berlin : Beuth Verlag GmbH, 2007

## LITERATUR

---

- [Ž00] ŽIHA, K.: Redundancy and robustness of systems of events. In: *Probabilistic Engineering Mechanics* 15 (2000), Nr. 4, 347-357. [http://dx.doi.org/10.1016/S0266-8920\(99\)00036-3](http://dx.doi.org/10.1016/S0266-8920(99)00036-3). – DOI 10.1016/S0266-8920(99)00036-3
- [Was60] WASIUTYNSKI, Zbigniew: On the congruency of the forming according to the minimum potential energy with that according to equal strength. In: *Bull. de L'Academie Polonaise des Sciences, Serie des Sciences Techniques* 8 (1960), S. 259-268
- [ZKB97] ZOWE, Jochem ; KOČVARA, Michal ; BENDSØE, Martin P.: Free material optimization via mathematical programming. In: *Mathematical Programming* 79 (1997), Nr. 1-3, S. 445-466. <http://dx.doi.org/10.1007/BF02614328>. – DOI 10.1007/BF02614328

## Tabellenverzeichnis

3.1	Vergleich der Funktions- und Ausfallwahrscheinlichkeiten verschiedener Möglichkeiten die Datenkapazität von Festplatten zu nutzen. . . . .	14
3.2	Wertetabelle der Codierung bei einem RAID 5 aus 3 Festplatten mit den zugehörigen Wahrscheinlichkeiten . . . . .	25
3.3	Vergleich der Funktions- und Ausfallwahrscheinlichkeiten verschiedener Möglichkeiten die Datenkapazität von Festplatten zu nutzen – inkl. Redundanz und Effizienz. . . . .	27
4.4	Die Probleme zu unterschiedlichen Diskretisierungen und unterschiedlichen Graden der Redundanz wurden sowohl mit einem Greedy-Algorithmus, als auch mit einer vollständigen Optimumssuche gelöst. Die Zielfunktionswerte (Volumen) der Ergebnisse sind hier in der Tabelle aufgelistet. Man erkennt deutlich die geringe Abweichung der Greedy-Lösung von der Optimallösung. . . . .	76
A.1	Beispiel einer Übergangstabelle bestehend aus den Zuständen $s_i \in S$ und den Eingangszeichen $t_j \in T$ . . . . .	90

## TABELLENVERZEICHNIS

---

## Abbildungsverzeichnis

2.1	In diesem Bild wird eine mit dem SIMP-Ansatz (vgl. Anhang C.1) berechnete 2-dimensionale Struktur dargestellt. Jedes erkennbare Pixel repräsentiert ein finites Element. Die linke Seite ist gelagert und rechts unten wirkt eine Kraft nach unten. Was passiert, wenn die Scheren die Struktur an der einen oder der anderen Stelle zerschneiden? . . . . .	7
3.1	Automat zu RAID 1 als Graph . . . . .	17
3.2	Automat zu Fahrrad soll bremsen . . . . .	19
4.1	Beispiel eines Zuverlässigkeitsschaltbild: $\Sigma_1$ und $\Sigma_2$ bilden ein Parallelsystem $P_1$ . $P_1$ bildet mit $\Sigma_3$ ein Seriensystem. . . .	31
4.2	Nichtlineare Abhängigkeit der Compliance vom Volumen . . . . .	34
4.3	Mögliche Werte für die Redundanz . . . . .	35
4.4	Verschiedene Strukturen mit zugehörigen Compliance-Werten mit unterschiedlichen Redundanzgraden von $1 - \frac{1}{n}$ für die angegebenen $n$ . Unter den Grafiken stehen jeweils 2 Zahlenwerte. Der obere gibt das Volumen $\sum_{e=1}^N \rho(e)$ an und der untere den Compliance-Wert $f^T u_\rho$ . Die Bilder der ersten Zeile haben einen einfachen Lastfall unten rechts mit einer Kraft nach unten. In der zweiten Bildzeile liegt ein 2-facher Lastfall unten rechts mit einer Kraft nach unten und oben rechts mit einer Kraft nach oben. Die letzte Bildzeile ist ebenfalls ein 2-facher Lastfall mit Kräften nach unten die unten rechts und unten in der Mitte wirken. In allen Beispielen ist die linke Seite gelagert. . . . .	44
4.5	Verschiedene Strukturen mit unterschiedlichen Redundanzgraden von $1 - \frac{1}{n}$ für die angegebenen $n$ ; wegen der feinen Auflösung wurde nicht fertig ausiteriert. Die Bilder der ersten Zeile stellen einen einfachen Lastfall dar. Die beiden anderen Bildzeilen sind jeweils ein 2-facher Lastfall. In Abb. 4.4 werden die Lastfälle genau beschrieben. . . . .	45
4.6	Verschiedene als Stabwerk interpretierte Strukturen mit einem unterschiedlichen Redundanzgrad . . . . .	46

4.7	Verschiedene Strukturen mit einem Redundanzgrad von $\frac{1}{3}$ . Es wurde nicht nur die Gesamtstruktur $\rho$ mit einer Einfärbung der einzelnen Teilstrukturen $\rho_i$ abgebildet, sondern auch das, was bei einem Ausfall einer Teilstruktur übrig bleibt: $\bar{\rho}_i$ . In allen Beispielen ist die linke Seite gelagert. Die Bilder der ersten Zeile haben einen einfachen Lastfall unten rechts mit einer Kraft nach unten. In der zweiten Bildzeile liegt ein 2-facher Lastfall unten rechts mit einer Kraft nach unten und oben rechts mit einer Kraft nach oben. Die letzte Bildzeile ist ebenfalls ein 2-facher Lastfall mit Kräften nach unten die unten rechts und unten in der Mitte wirken. Blau repräsentiert $\rho_1$ , rot $\rho_2$ und gelb $\rho_3$ . . . . .	60
4.8	Gedankenexperiment: Keine Trennung der Einzelstrukturen	61
4.9	Gleichungsnebenbedingung zu (4.64) . . . . .	68
4.10	Ungleichungsnebenbedingung zu (4.64) . . . . .	68
4.11	Gleichungen zu (4.92) . . . . .	73
4.12	Ungleichungen zu (4.92) . . . . .	74
4.13	Vergleich des Grades der Redundanz und der Diskretisierung im Bezug zum Volumen . . . . .	78
4.14	2-dimensionales Beispiel eines Fachwerks mit verschiedenen Graden der Redundanz bei einer Grundstruktur aus 28 Stäben . . . . .	79
4.15	2-dimensionales Beispiel eines Fachwerks mit verschiedenen Graden der Redundanz bei einer Grundstruktur aus 200 Stäben . . . . .	80
4.16	2-dimensionales Beispiel eines Fachwerks unter mehrfachen Lastfall mit verschiedenen Graden der Redundanz und verschiedenen Diskretisierungen . . . . .	81
4.17	Der Boden ist gelagert und oben links in der Mitte wirkt eine horizontale Last. Gezeigt sind verschiedene Grade der Redundanz . . . . .	82
4.18	Teilstrukturen von Abb. 4.17(e) . . . . .	83

4.19	Der Boden ist gelagert und oben links und rechts wirken abwechselnd Kräfte. In (a) ist die Gesamtstruktur mit einem Grad der Redundanz von $\frac{1}{3}$ abgebildet und in (b), (c), (d) sind die Einzelstrukturen dargestellt. Die Volumen der Strukturen ergeben sich zu: 625, 175, 275 und 175. . . . .	83
5.1	Vergleich von robuster und von redundanter Topologieoptimierung: Die redundante Struktur (a) ist nicht robust und die robuste Struktur (b) ist nicht redundant. . . . .	86
A.1	Beispiel eines Automaten als Graph dargestellt . . . . .	90
B.1	Grundproblem der Informations- und Kodierungstheorie . .	91
C.1	Nummerierung der Elemente . . . . .	106
C.2	Struktur der Steifigkeitsmatrix bei der Nummerierung nach Kapitel C.2.1 in 2 D . . . . .	108
C.3	Nummerierung eines Elementes in der $k$ -ten Ebene, $i$ -ten Spalte und $j$ -ten Zeile in 3 D . . . . .	109
C.4	Struktur der Steifigkeitsmatrix bei der Nummerierung nach Kapitel C.2.2 in 3 D . . . . .	113
C.5	Vergleich der Abschätzung der Anzahl Stäbe bei dieser Vernetzung und der exakten Anzahl . . . . .	132
C.6	Hier ist eine diskretisierte Lösung eines Beispiels von [Mic04] dargestellt: eine typische Michell-Struktur. Sie wurde mit dem Vorgehen aus Kapitel C.3.2 berechnet und besteht aus 346 Stäben. In (b) sind zusätzlich die $40 \cdot 21 = 840$ Knoten der Grundstruktur, die aus 215059 Stäben besteht, dargestellt. Rechts außen wirkt eine Kraft nach unten und links ist ein verteiltes Festlager durch 5 gelagerte Knoten repräsentiert. . . . .	133
D.1	Verschiedene als Stabwerk interpretierte Strukturen in Abb. 4.6 aus Abb. 4.4 wurden hier mit unterschiedlichen Masseverteilungen optimal ausgelegt. . . . .	136
E.1	Beispiel eines vollständigen Binärbaums der Tiefe 2 . . . . .	138