Eine Methode zur optimalen Redundanzallokation im Vorentwurf fehlertoleranter Flugzeugsysteme

Vom Promotionsausschuss der
Technischen Universität Hamburg-Harburg
zur Erlangung des akademischen Grades
Doktor-Ingenieur
genehmigte Dissertation

von

Dipl.-Ing.
Christian Raksch

aus Neumünster

2013

1. Gutachter:	Prof. Dr.-Ing. Frank Thielecke Institut für Flugzeug-Systemtechnik Technische Universität Hamburg-Harburg
2. Gutachter:	Prof. Dr.-Ing. Robert Luckner Fachgebiet Flugmechanik, Flugregelung und Aeroelastizität Institut für Luft- und Raumfahrt Technische Universität Berlin
Tag der mündlichen Prüfung:	18. Februar 2013

Safety is built in, not added on.
Duane Kritzinger [61]

Danksagung

Die vorliegende Arbeit ist während meiner Zeit als wissenschaftlicher Mitarbeiter am Institut für Flugzeug-Systemtechnik der Technischen Universität Hamburg-Harburg entstanden.

Für die Betreuung und die Freiheiten bei der Erstellung dieser Dissertation, die Erfahrungen in der Forschung, Lehre und im Projektmanagement, die unzähligen Dialoge und die Zeit am Institut für Flugzeug-Systemtechnik danke ich Prof. Dr.-Ing. Frank Thielecke. Für das Interesse an meiner Arbeit und die Erstellung des zweiten Gutachtens danke ich Prof. Dr.-Ing. Robert Luckner. Meine andauernde Faszination an der Systemtechnik bleibt Prof. Dr.-Ing. Udo B. Carl geschuldet, vielen Dank.

Für die Impulse und Dialoge zu dieser Arbeit danke ich allen ehemaligen Projektpartnern aus den Forschungsprojekten MOET, SIMKAB und BRINKS. Besonders für die Erfahrungen im Bereich der Sicherheits- und Zuverlässigkeitsanalyse der Airbus Operations GmbH danke ich Dr.-Ing. Michael Oppermann und Dr.-Ing. Uwe Wiezcorek. Für Ihre Anteile an dieser Dissertation möchte ich zudem allen meinen ehemaligen Studenten danken, die Ihre Arbeit bei mir geschrieben haben. Zudem gilt mein Dank jenen, die durch kritische Nachfragen und Anregungen auf Konferenzen oder während der Doktorandendialoge meine Arbeit unterstützt und vorangebracht haben.

Die Zeit am Institut für Flugzeug-Systemtechnik war nicht nur von herausfordernden und interessanten Projekten geprägt, sondern auch von einzigartigen Kollegen. Ich bedanke mich herzlich bei allen ehemaligen Kollegen während meiner Zeit dort und besonders bei Dipl.-Ing. Martin Halle, Dipl.-Ing. Torben Pielburg, Dr.-Ing. Malte Pfennig und Dr.-Ing. Dominick Rehage.

Diese Arbeit und die vorherige Ausbildung wären ohne die Unterstützung meiner Eltern und meiner Familie so nicht möglich gewesen, hierfür werde ich Ihnen stets dankbar sein. Ganz besonders bin ich meiner Frau Franca zu Dank verpflichtet: sie hat mich seit der Konstruktionsaufgabe im dritten Semester immer unterstützt. Diese Arbeit ist ihr, unserem Sohn Lasse und unserer wachsenden Familie gewidmet.

Boostedt im Juli 2013 Christian Raksch

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Nomenklatur

Formelzeichen

Lateinische Formelzeichen

Zeichen	Einheit	Bedeutung
c	$[-]$	normierte Federsteifigkeit im RadViz-Verfahren
d	$[-]$	Degradationsstufe
e	$[-]$	Entscheidungsebene
g	$[-]$	Nebenbedingung
k	$[-]$	Kosten, allg. summative Zielgröße
n	$[-]$	allg. Anzahl
p	$[-]$	Pfad
s	$[-]$	summative Zielfunktion
t	$[-]$	Zeit
u	$[-]$	Projektionierte Koordinate
x	$[-]$	Architekturvariable
$\mathbf{x}$	$[-]$	Architekturvektor
A	$[-]$	Ankerpunkt im RadViz-Verfahren
$\mathbf{A}$	$[-]$	Menge der Ankerpunkte im RadViz-Verfahren
CFSM	$[-]$	Modell eines nebenläufigen, endlichen Zustandsautomaten
E	$[-]$	Elementarzustand
F	$[-]$	Ausfallwahrscheinlichkeit, Sicherheits- und Zuverlässigkeitsfunktion
$\mathbf{F}$	$[-]$	Eingabealphabet
K	$[-]$	Indikatorvariable eines Ereignisses

Zeichen	Einheit	Bedeutung
K	[−]	Ereignismenge
KA	[−]	Ereignismenge aktiver Komponenten
KG	[−]	Menge der Ereignisse einer Nebenbedingung
KM	[−]	Kandidatenmenge serieller Ereignisse
KW	[−]	Ereignismenge passiv-warmer Zustände
LB	[−]	untere Grenze
MP	[−]	Minimalpfad
MP	[−]	Menge der Minimalpfade
N	[−]	Matrix der neutralen Elemente
P	[−]	Eintrittswahrscheinlichkeit, Zustandswahrscheinlichkeit
P	[−]	Zustandswahrscheinlichkeitsmatrix
PF	[−]	PARETO-Front, Menge der nichtdominierten Architekturen
PST	[−]	Wahrscheinlichkeit einer Zustandstransition
R	[−]	Zuverlässigkeit, Überlebenswahrscheinlichkeit, Sicherheit- und Zuverlässigkeitsfunktion
T	[−]	Zustandstransition
UB	[−]	obere Grenze
VA	[−]	Vorgängerargument
VT	[−]	Vorgängerterm
X	[−]	Architekturmatrix, -raum
Y	[−]	Ausgabealphabet
Z	[−]	Zustand
Z	[−]	Zustandsmenge
$\mathcal{A}$	[−]	Menge variabler Zielfunktionen
$\mathcal{B}$	[−]	Zielwertmenge
$\mathcal{E}$	[−]	Erwartungswert
$\mathcal{F}$	[−]	Eingabealphabet
$\mathcal{G}$	[−]	Menge der Nebenbedingungen g
$\mathcal{O}$	[−]	O-Notation, Komplexitätsordnung einer Aufgabe

Zeichen	Einheit	Bedeutung
$\mathcal{R}$	$[-]$	Menge variabler Zuverlässigkeitsfunktionen
$\mathcal{S}$	$[-]$	Menge variabler summativer Zielfunktionen
$\mathcal{Y}$	$[-]$	Ausgabealphabet

Griechische Formelzeichen

Zeichen	Einheit	Bedeutung
α	$[-]$	WEIBULL-Faktor, Skalierungsparameter
β	$[-]$	WEIBULL-Faktor, Ausfallsteilheit
γ	$[-]$	Übergangsfunktion
δ	$[-]$	Ausgabefunktion
ε	$[-]$	Toleranzparameter einer PARETO-Menge
θ	$[-]$	RADVIZ-WINKEL
λ	$[\mathrm{s}^{-1}]$	Ausfallrate
μ	$[-]$	PAES, eingehender Archivfaktor
ν	$[-]$	PAES, ausgehender Archivfaktor
Φ	$[-]$	Systemstrukturfunktion

Indizes

Index	Bedeutung
100	reale PARETO-Menge
a	Ausgang
a	aktiv
akt	aktuell bekannte PARETO-Menge
approx	approximiert
c	Kosten, monetär
c	passiv-kalt
d	degradiert

Index	**Bedeutung**
dom	dominiert
e	Eingang
f	festes Ereignis
ges	gesamt
h	aktiv-heiss
i	isoliert
k	Kosten, allgemein Aufwand
m	Masse
max	maximal, Maximum
min	minimal, Minimum
p	degradierte Ereignismenge
real	vollständige PARETO-Menge
s	seriell
sys	System
th	theoretisch
v	variables Ereignis
var	variable Größe
w	passiv-warm
A	Archiv
G	Generationen
MP	Minimalpfade
P	Population
R	Anforderung
S	System, gesamt
SP	seriell-parallel
$\mathcal{A}$	Menge der Zielfunktionen
$\mathcal{B}$	Zielwertmenge

Abkürzungen

Abk.	Bedeutung
ACARE	Luftfahrtforschungsgremium, engl. *Advisory Council for Aeronautical Research in Europe*
APU	Hilfsgasturbine, engl. *Auxiliary Power Unit*
ARP	Leitfaden der Luftfahrtindustrie, engl. *Aerospace Recommended Practice*
ASG	Hilfsstarter/-generator, engl. *Auxiliary Starter/Generator*
BB	Optimierungsverfahren mittels Verzweigung und Beschränkung, engl. *Branch & Bound*
BDD	Binärer Entscheidungsbaum, engl. *Binary Decision Diagram*
CAT	katastrophale Fehlerfolge, engl. *catastrophic*
CCA	Gemeinsame Ursachenanalyse, engl. *Common Cause Analysis*
CCF	Fehler gemeinsamer Ursache, engl. *Common Cause Failure*
CFSM	Nebenläufiger, endlicher Zustandsautomat, engl. *Concurrent Finite State Machine*
CMPC	Gruppiertes Kreuzungsverfahren, engl. *Clustered Multi-Point-Crossover*
CoSyIMA	SyRelAn-Modul für die Sicherheitsanalyse rekonfigurierbarer Rechnersysteme, engl. *Complex System Analysis of Integrated Modular Avionics based Systems*
CoSyOp	SyRelAn-Modul für die Optimierung komplexer Flugzeugsysteme, engl. *Complex System Safety and Reliability Optimization*
CoSyRA	SyRelAn-Modul für die Sicherheitsanalyse komplexer Flugzeugsysteme, engl. *Complex System Reliability Analysis*
CS	Zulassungsspezifikation der EASA, engl. *Certification Specification*
CU	elektrischer Wandler, engl. *Converter Unit*
DAL	Gewährleistung eines erforderlichen Entwicklungsprozesses, engl. *Design Assurance Level*
DC	Gleichstrom, engl. *Direct Current*

Abk.	Bedeutung
EASA	Europäische Agentur für Flugsicherheit, engl. *European Aviation Safety Agency*
ENG	Triebwerk, engl. *Engine*
ESS	notwendige Systemfunktion, engl. *Essential*
FC	Brennstoffzelle, engl. *Fuel Cell*
FH	Flugstunde, engl. *Flight Hour*
FHA	Funktionale Gefahrenbewertung, engl. *Functional Hazard Assessment*
FMEA	Fehlermoden- und Effektanalyse, engl. *Failure Mode and Effect Analysis*
FTA	Fehlerbaumanalyse, engl. *Fault Tree Analysis*
GA	Genetischer Algorithmus
GUI	grafische Benutzerschnittstelle, engl. *Graphical User Interface*
HAZ	gefährliche Fehlerfolge, engl. *Hazardous*
HVDC	Hochspannungsnetz mit Gleichstrom, engl. *High Voltage Direct Current*
INCOSE	Internationale Gesellschaft für Systems Engineering, engl. *International Council on Systems Engineering*
KPI	Leistungskennzahl, engl. *Key Performance Indicator*
LMES	Verlust der normalen elektrischen Energieversorgung, engl. *Loss of Main Electrical System*
Lsg	Lösung
MAJ	bedeutende Fehlerfolge, engl. *Major*
MEA	Flugzeug mit teilelektrischer Systemversorgung, engl. *More Electric Aircraft*
MIN	geringe Fehlerfolge, engl. *Minor*
MMEL	Vorlage für betreiberspezifische minimale Ausrüstungslisten, engl. *Master Minimum Equipment List*
MOPS	Mehrkriterielle Parametersynthese, engl. *Multi-Objective Parameter Synthesis*
MP	Minimalpfad
MPC	Kreuzungsverfahren, engl. *Multi-Point-Crossover*
MRS	Mehrfach-redundantes Systemmodell

Abk.	**Bedeutung**
NFL	Theorem zum Verhalten von Optimierungsalgorithmen, engl. *No Free Lunch Theorem*
NP	Komplexitätsklasse mit nicht-deterministischer polynomialer Zeitentwicklung, engl. *Nondeterministic Polynomial Time*
NSE	keine sicherheitskritische Fehlerfolge, engl. *No Safety Effect*
NSGA-II	spezifischer Genetischer Algorithmus, engl. *Non-dominated Sorting Genetic Algorithm II*
PAX	Passagier(anzahl), engl. *Passenger*
PFHA	Vorläufige Funktionale Gefahrenanalyse, engl. *Preliminary Functional Hazard Assessment*
PSSA	Vorläufige Systemsicherheitsanalyse, engl. *Preliminary System Safety Assessment*
PST	Zustandsübergangswahrscheinlichkeit, engl. *Probability of State Transition*
RadViz	Methode zur radialen Visualisierung, engl. *Radial Coordinate Visualization*
RAP	Verfahren der Redundanz- oder Zuverlässigkeitsallokation, engl. *Redundany/Reliability Allocation Problem*
RAT	Staulufttturbine, engl. *Ram Air Turbine*
RBD	Zuverlässigkeitsblockdiagramm, engl. *Reliability Block Diagram*
RU	elektrischer Umrichter, engl. *Rectifier Unit*
SAE	Gesellschaft für Transportwesen, engl. *Society of Automotive Engineers*
SPEA	spezifischer Genetischer Algorithmus, engl. *Strength Pareto Evolutionary Algorithm*
SG	Starter/Generator, engl. *Starter/Generator*
SQP	Optimierungsverfahren, engl. *Sequential Quadratic Programming*
SSA	Systemsicherheitsanalyse, engl. *System Safety Assessment*
SyRelAn	Software zur Sicherheitsanalyse von Flugzeugsystemen des Insituts für Flugzeug-Systemtechnik der Technischen Universität Hamburg-Harburg, engl. *System Reliability Analysis*
TEFO	Ausfall sämtlicher Triebwerke, engl. *Total Engine Flame Out*

Abk.	**Bedeutung**
VA	Vorgängerargument
VE	Vollständige Enumeration
VT	Vorgängerterm

1 Einleitung

Die Entwicklung moderner Flugzeugsysteme erfordert neben verfügbaren Technologien die Entwicklung und Verfügbarkeit unterstützender Werkzeuge, um die inner- und intersystemische Komplexität des Systementwurfs und der Systemtechnologien handhaben zu können [16, 114]. Die Anforderungen an die Systementwicklung werden dabei durch die umzusetzende Funktion gestellt und neben den Kundenanforderungen durch die globalen Themen Umweltverträglichkeit und Sicherheitsbestimmungen ergänzt. Die grundlegenden Einflüsse auf die Systementwicklung sind in Abbildung 1.1 dargestellt, wobei die hauptsächlichen Aspekte für die vorliegende Arbeit hervorgehoben sind [2, 59, 60].

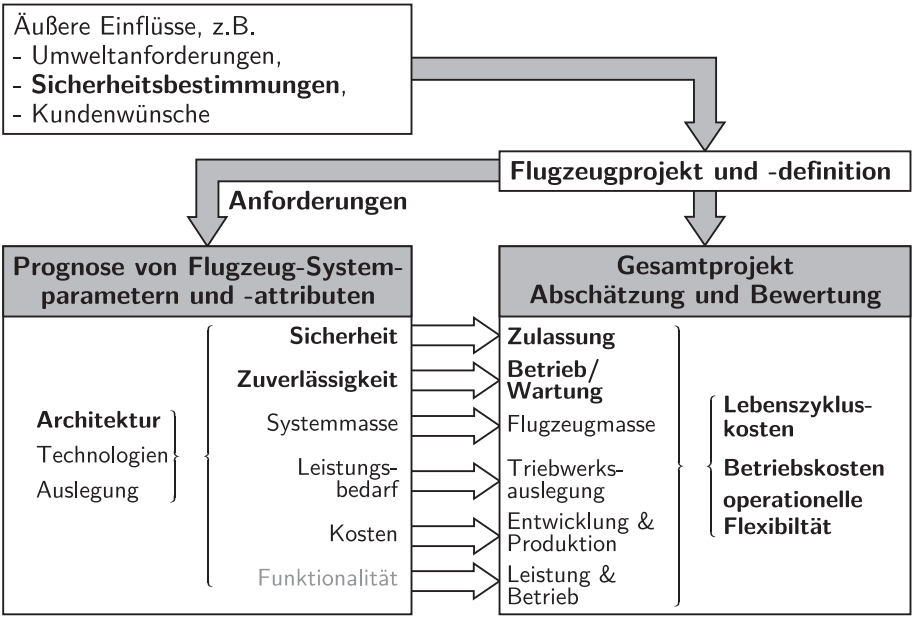


Abb. 1.1: Anforderungen und Einflüsse auf die Entwicklung von Flugzeugsystemen und auf den Flugzeugentwurf, nach [60]

Die Abbildung deutlich wie die äußerlichen Einflüsse der Sicherheitsbestimmungen, in Europa durch die Zulassungsvorschrift EASA CS 25 für Verkehrsflugzeuge spezifiziert, über die Projektdefinition die Systementwicklung und auch den Betrieb beeinflussen. Diese und die weiteren Einflüsse sind sowohl bei der System- und somit auch bei der Methodenentwicklung zu berücksichtigen, um unterstützende Werkzeuge bereitstellen zu können.

Den dominierenden Trend bei der Entwicklung neuer Flugzeug-Systemtechnologien, vor allem für die Aktuatorik und somit auch Energieversorgung, stellt seit einigen Jahren die teilweise bis vollständige Elektrifizierung der Systeme dar, der Weg zum *More Electric Aircraft* [29, 32, 52, 78]. Das elektrische Energieversorgungssystem rückt dabei in den Mittelpunkt der Sekundärenergieversorgung und bedient in allen Flugphasen die Verbraucher der Grund- und Kabinensysteme.

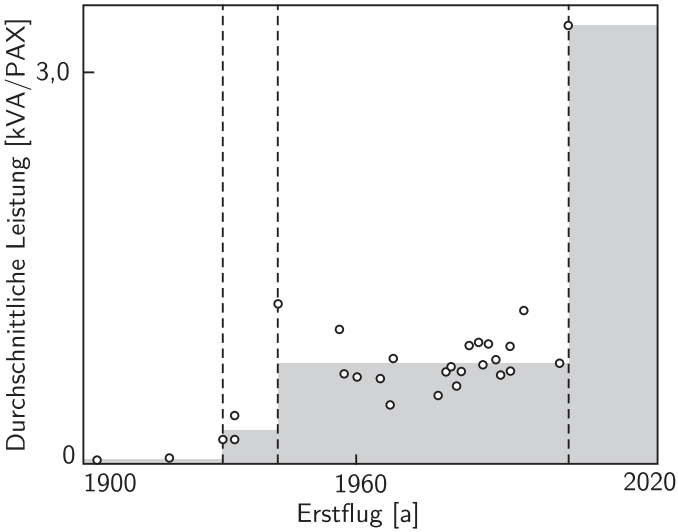


Abb. 1.2: Entwicklung der durchschnittlichen installierten elektrischen Leistung der Hauptgeneratoren pro Passagier in Verkehrsflugzeugen

Abbildung 1.2 verdeutlicht diesen Trend zur Elektrifizierung der Flugzeugsysteme. In der ersten Phase der Luftfahrt versorgten in wenigen Flugzeugen Batterien die elektrischen Verbraucher. In der nächsten Phase folgten kleine Gleichstromgeneratoren zur zusätzlichen Versorgung der Bordinstrumente ne-

ben der Beleuchtung. Da dieses auch die Kabinenbeleuchtung umfasste, zeigt sich bereits ein erster Einfluss der Passagieranzahl. Bereits in den 1930er Jahren wurden die Anfänge der konventionellen elektrischen Systemarchitektur geprägt. Die Versorgung der elektrischen Bordküchen stellt aktuell in den meisten betriebenen Verkehrsflugzeugen den größten elektrischen Verbraucher dar und somit eine starke Abhängigkeit von der Passagierzahl [79]. Der letzte Leistungssprung in der elektrischen Energieversorgung wurde durch die Elektrifizierung der Klimaanlagen und des Enteisungssystems geprägt und somit dem Schritt zum *More Electric Aircraft*. Neben der Zunahme der elektrischen Leistung pro Passagier zeigt sich jedoch auch eine kontinuierliche Steigerung der Kritikalität der elektrischen Energieversorgung. Mittlerweile sind alle modernen Verkehrsflugzeuge von einer sicheren elektrischen Energieversorgung abhängig, zum Beispiel zur Versorgung der *Fly-by-Wire* Flugsteuerung. Der Fokus auf das elektrische Energieversorgungssystem bezieht sich somit auf der einen Seite auf die Leistungsverbraucher, auf der anderen Seite auf die Signalübertragung. Diese Zentralisierung der Systemfunktionen lässt sich auch bei anderen Flugzeugsystemen feststellen, beispielsweise im Bereich der Avionik seit der Einführung der Integrierten Modularen Avionik (IMA) [13, 102].

Die möglichen Vorteile einer solchen Zentralisierung des elektrischen Energieversorgungssystems können in der Reduktion der installierten Leistung, des Leistungsbedarfs und somit in der flugzeugweiten Reduktion der Systemmasse liegen, die sich wiederum auf den Treibstoffverbrauch auswirkt [32, 52, 78]. Weitere Vorteile wären die Modularisierung der Komponenten und somit die Vereinfachung von Wartung und Ersatzteilbereithaltung und die Einbringung systemübergreifender statt systemspezifischer Redundanzen [32, 52]. Diesen Vorteilen steht jedoch eine erhöhte Komplexität der Systeme und somit auch des Systementwurfs gegenüber [29, 78]. Die Komplexität des Architekturentwurfs wird dabei durch die steigenden Varianten der Funktionsallokation geprägt. Dabei kann die systemübergreifende Nutzung von Komponenten zu einer sinkenden Diversität der Energieversorgung führen und somit zu singulären Punkten in der Sicherheits- und Zuverlässigkeitsbewertung der Systeme. Aus diesem Grund sind für die Realisierung moderner, vernetzter Flugzeugsysteme neue Systemarchitekturen notwendig [80].

Der Architekturentwurf von Flugzeugsystemen wird dabei neben den funktionalen Anforderungen vor allem durch die Anforderungen an die Systemsicherheit und -zuverlässigkeit getrieben [9, 13, 40, 48, 61, 119]. Seit der Einführung des risikobasierten Nachweisprozesses für Flugzeugsysteme in den 1970er Jahren haben sich die Flugzeug-Systemarchitekturen evolutionär zu den heuti-

gen konventionellen Lösungen entwickelt. Diese evolutionäre Entwicklung ist im heutigen kostensensitiven Geschäftsmodell der Flugzeughersteller und -betreiber nicht mehr möglich. Die Erkenntnisse zu neuen Technologien müssen daher verstärkt vor einer möglichen Serieneinführung ermittelt und bewertet werden. Die Einführung neuer Systemtechnologien erfordert dabei auch stets eine umfassende Analyse der Systemarchitekturen, unterstützt durch geeignete Werkzeuge und Methoden.

Die vorliegende Arbeit begegnet der steigenden Komplexität im Architekturentwurf daher mit einer integrierten Entwurfsmethode zur Verteilung von Redundanzen komplexer Flugzeug-Systemarchitekturen und berücksichtigt hierbei vor allem die Aspekte der Systemsicherheit und -zuverlässigkeit.

1.1 Vorentwurfsmethoden für Flugzeugsysteme

Aufgrund der steigenden Komplexität und Interaktion von modernen Flugzeugsystemen ist eine Anwendung unterstützender Werkzeuge bereits in der frühen Konzeptphase unerlässlich [16]. Im Folgenden werden ausgewählte wissenschaftliche Entwurfsmethoden zur Unterstützung der Konzeptphase vorgestellt, wie sie in den letzten Jahren zunehmend entwickelt wurden. Die hierbei genutzten Analyse- und Optimierungsansätze und die hinterlegten Anforderungen an die Methoden werden untersucht, um für die Redundanzallokation komplexer Systemarchitekturen Anforderungen abzuleiten. Aufgrund der weltweit zahlreichen Arbeiten auf dem Gebiet der Konzeptoptimierung kann die folgende Auflistung keinen Anspruch auf Vollständigkeit erheben. Es werden jedoch Verfahren mit ähnlichen Problemstellungen bzw. Verfahren mit potentiell geeigneten Lösungsmethoden vorgestellt.

Antriebskinematik
Zur Kinematiksynthese und Unterstützung des Entwurfprozesses von Antriebsmechanismen für Hochauftriebssysteme stellt Holert ein Verfahren vor, dass mit Hilfe eines Genetischen Algorithmus den zulässigen, kontinuierlichen Lösungsraum durchsucht [46]. Als Zielwerte werden dabei unterschiedliche Kostenfunktionale ψ_i betrachtet, beispielsweise die Größe der *Fairing*, die Kinematikmasse und die Antriebslastcharakteristik. Mit Hilfe eines Genetischen Algorithmus wird entsprechend der verwendeten Heuristik ein Teil des Lösungsraumes untersucht und daraus die Pareto-optimale Menge bestimmt, anhand derer die optimalen Lösungen für den detaillierten Entwurf ausgewählt werden.

Dabei wird die nicht-dominierte Menge für den Anwender als Übersicht über den Lösungsraum dargestellt und das Verhalten der Lösungsmenge ermittelt. In einem anschließenden Prozessschritt kann der Anwender angeleitet Lösungen auswählen.

Antriebssysteme
Als Gegenstück zur Antriebskinematik hat Pfennig ein Verfahren für den Vorentwurf von Landeklappenantriebssystemen entwickelt. Kernstück dieser Methodik ist ein *Interval Constraint Satisfaction Problem (ICSP)*, das auf Grundlage von Intervallparametern einer Komponentendatenbank mögliche Datensätze für eine gültige Parametrisierung der verwendeten Komponenten findet [87]. Neben dieser Analysefähigkeit ist auch ein Optimierungsmodul integriert, das innerhalb des Parameterraums, der durch die Parameterintervalle aufgespannt wird, massenoptimale Lösungen sucht. Als Optimierungsverfahren wird hierbei aufgrund der benötigten ungerichteten Zielfunktionen eine Evolutionsstrategie verwendet, die das mehrkriterielle Optimierungsproblem löst. Der mehrkriterielle Parameterraum wird mittels der Definition des *ICSP* beschränkt, so dass die Entwicklung technisch sinnvoller Antriebssysteme sichergestellt ist [88].

Energieversorgung
Liscouet-Hanke betrachtet ebenfalls die Abschätzung der Systemmasse auf Grundlage physikalischer Simulationsmodelle. Hierbei berücksichtigt sie neben nominellen Systemzuständen auch degradierte Systemzustände von Energieversorgungssystemen im Rahmen der Lastanalyse und behandelt somit auch Aspekte der Systemsicherheit. Diese werden jedoch nicht probabilistisch quantifiziert, sondern nur hinsichtlich der verfügbaren Leistung berücksichtigt. Eine Erweiterung um die Zielgrößen Sicherheit und Zuverlässigkeit wird von Liscouet-Hanke für eine umfassende Konzeptbewertung jedoch als sinnvoll eingeschätzt [66].

Einen Ansatz zur Kombination der optimalen Dimensionierung von Komponenten und der Sicherheitsbewertung von elektrischen Energieversorgungssystemen liefert Schallert [105]. Auf Grundlage eines physikalisch motivierten Simulationsmodelles können unterschiedliche Fehlermodi einer Komponente abgebildet werden. Anhand von Lastprofilen für unterschiedliche Flugphasen und -zustände des elektrischen Netzwerkes ist somit eine Dimensionierung der Komponenten möglich. Dabei berücksichtigt Schallert nicht nur die nominellen Systemzustände, sondern betrachtet auch definierte degradierte Zustände. Hierfür werden über das Simulationsmodell verteilt die verfügbaren simulierten Leistungswerte abgegriffen. Sofern eine Kombination von Komponentenfehlern

zu einem Ausfall der elektrischen Energieversorgung an einer beobachteten Stelle im Modell führt, wird die Kombination als Minimalschnitt gespeichert. Die Iteration über alle möglichen Systemzustände gestattet folglich die Ermittlung der vollständigen Menge der Minimalschnitte. Somit kann die physikalische Dimensionierung der Komponenten an eine Sicherheitsanalyse des Systems gekoppelt werden. Die wiederholte Dimensionierung der Komponentenparameter in Verbindung mit der Sicherheitsanalyse könnte zudem zur Optimierung der Systemarchitektur unter Berücksichtigung der sicherheitstechnischen Eigenschaften genutzt werden.

Lüdders et al. haben für die Optimierung einer Brennstoffzellenarchitektur für Transportflugzeuge einen Entwurfsprozess mit fünf Schritten definiert, der systematisch von den Systemanforderungen zur Validation führt [72]. Dabei werden im ersten Schritt die Anforderungen auf Flugzeugebene definiert, aus denen im zweiten Schritt die Systemanforderungen abgeleitet werden. Die nachfolgende Architekturdefinition basiert auf den vorherigen Anforderungen, Sicherheits- und Zuverlässigkeitsaspekten und Vorkenntnissen der Systemingenieure, wendet jedoch keine Verfahren zur quantitativen Bewertung unterschiedlicher Systemarchitekturen an. Die Systemarchitektur ist im vierten Schritt Ausgangspunkt für eine parametrische Systemoptimierung, hierbei wird mit Hilfe eines *Non-Dominated Sorting Genetic Algoritm-II* die Pareto-Front des kontiniuerlichen Zielwertraumes ermittelt. Im letzten Schritt wird ein ausgewähltes Konzept validiert, hierfür werden detaillierte Modelle der Komponenten genutzt, mit deren Hilfe auch die Ermittlung globaler Kennzahlen möglich ist, wie eine Zunahme des Flugzeugwiderstands.

Systeme im Flugzeugvorentwurf

Während die vorherigen Verfahren die Massen der Komponenten und Systeme auf Grundlage detaillierter System- und Simulationsmodelle abschätzen, ermittelt Koeppen mit Hilfe statistischer Datensätze und systemspezifischer Funktionale die Massen einzelner Flugzeugsysteme und deren Einfluss auf den Flugzeugentwurf. Neben der Masse wird zudem der Leistungsbedarf des betrachteten Systems ermittelt. Beide Zielgrößen unterstützen dabei das Entwurfswerkzeug PrADO (*Preliminary Aircraft Design and Optimization*) und die Methode von Koeppen bietet eine direkte Schnittstelle an [44, 59]. Auch wenn diese Methodik keine Optimierungsalgorithmen nutzt, kann die gesamte Analyse zur Optimierung der Flugzeug- und Systemmassen genutzt werden. Das Flugzeugentwurfsprogramm PrADO zielt dabei auf die Optimierung des vollständigen Flugzeugentwurfs und betrachtet neben den Systemen weitere Aspekte wie die Geometrie des Flugzeugs, die Antriebe, strukturelle Ausle-

gung und eine Abschätzung der Betriebskosten. Als Ergebnis wird dabei eine singuläre, optimale Lösung mittels eines Gradientenverfahrens ausgegeben [69]. Einen ähnlichen Ansatz verfolgt auch das Flugzeugentwurfswerkzeug PrESTo (*Aircraft Preliminary Sizing Tool*) von Seeckt et al. [109].

Flugsteuerung
Haitao et al. untersuchen im Kontext der Entwicklung eines *More Eletric Aircraft* die parametrische Architekturoptimierung eines hybriden primären Flugsteuerungssystems [42]. Hierfür variieren sie mit Hilfe eines Genetischen Algorithmus die Aktuatoren, Steuerungsrechner und Energieversorgungssysteme der unterschiedlichen Stellflächen einer vorab definierten und festen Systemarchitektur. Als Zielfunktionen werden sowohl die Systemmasse durch Summation der Komponentenmassen als auch der Wirkungsgrad durch Multiplikation der individuellen Wirkungsgrade berechnet. Als Nebenbedingung wird die quantitative Sicherheit des Systems, klassifiziert als *catastrophic*, berücksichtigt und durch Summation der Fehlerraten der variablen Fehlerraten der einzelnen Stellflächen abgeschätzt.

Einen weiteren Ansatz zur Optimierung von Flugsteuerungsarchitekturen haben Bauer et al. entwickelt. Aufgrund von ca. 10^{13} möglichen Architekturen für eine konventionelle Airbus A340 Architektur, nutzen sie einen *Branch & Bound* Algorithmus, der systematisch den Architekturraum durchsucht und die Massen für die einzelnen Lösungen abschätzt. Dabei wird der Architekturraum sowohl durch funktionale als auch sicherheitstechnische Anforderungen beschränkt. Die Sicherheitsanforderungen beziehen sich dabei auf die Funktionswahrscheinlichkeit der benötigen Stellflächenfunktion, hierfür wird jedoch nicht die Eintrittwahrscheinlichkeit berechnet, sondern jeder Klassifizierung einer Fehlerbedingung wird eine Rollrate zugeordnet. Somit ergibt sich aus dem Quotienten aus der Rollrate des aktuellen Konzepts und der Rollrate der Klassifizierung ein Wert, der die aktuelle Architektur bewertet [9].

Sensorik
Für die optimale Anordnung von Sensoren und Aktuatoren in unterschiedlichen Anwendungen bieten Padula und Kincaid eine umfassende Übersicht bisheriger Arbeiten und nutzen für ihre Arbeit eine Formulierung als kombinatorisches Optimierungsproblem [86]. Sie stellen drei Ansätze für die Optimierung von Flugzeugsystemen vor: zunächst die Anwendung eines *Tabu Search* Verfahrens auf die optimale Anordnung von Aktuatoren für eine optimierte aktive akustische Strukturdämpfung. Dieser Anwendung folgt eine Optimierung der Anordnung von Sensoren für eine Systemidentifikation der aerolastischen

Struktur ebenfalls mittels *Tabu Search*. Abschließend betrachten Padula und Kincaid die Anordnung hochintegrierter Aktuatoren für eine Formänderung von Flugzeugstrukturen mit Hilfe eines Genetischen Algorithmus. Als Zielgrößen werden dabei beispielsweise die Minimierung der Aktuator- und Sensoranzahl betrachtet, beschränkt durch Nebenbedingungen bezüglich Leistungsanforderungen an die Aktuatorfunktion oder die Sensitivität der Sensoren. Die wichtigsten Schlüsse ihrer Arbeit sind die sorgfältige Formulierung des Optimierungsproblem, so dass zahlreiche Architekturen bereits ohne Evaluierung ausgeschlossen werden können, die Approximation von Zielfunktionen zur zeitoptimierten Auswertung und die Berücksichtigung technischer Limitierungen der betrachteten Komponenten, beispielsweise die maximalen Stellkräfte von Aktuatoren.

Klimaanlagen
Für die Optimierung von Klimaanlagen im Flugzeug, engl. *Environmental Control System*, haben Giese et al. ein mehrstufiges Konzept vorgestellt. Als globale Zielwerte werden der Kraftstoffverbrauch und die Systemmasse abgeschätzt [38]. Der Optimierungsprozess ist in einen globalen Optimierungsprozess der Flugzeugentwicklung eingebettet und beginnt mit der Definition erfolgsversprechender Systemarchitekturen, die in folgenden Prozessschritten optimal parametrisiert werden. In einer ersten Optimierungsschleife wird das Optimierungsproblem als *Mixed Integer Nonlinear Programming* Problem klassifiziert und mit Hilfe eines passenden heuristischen Optimierungsverfahrens gelöst, hierbei werden ausschließlich globale Parameter wie der Treibstoffverbrauch betrachtet. In einer inneren Schleife wird die Dimensionierung der Komponenten mittels eines *Sequential Quadratic Programming (SQP)* Verfahrens gelöst. Die Rechenzeit für eine Architektur beträgt dabei mehrere Wochen, unter Verwendung von parallelen Rechensystemen. Als Ergebnis wird die anteilige ermittelte Pareto-Front der untersuchten Systemarchitektur ausgegeben.

Avionik
Schulz et al. betrachten in Ihrer Arbeit die Optimierung von Avioniktopologien nicht nur hinsichtlich der prinzipiellen Verschaltung, sondern auch auf Grundlage von dreidimensionalen, räumlichen Konstruktionsmodellen, die beispielsweise eine möglichst genaue Ermittlung von benötigten Kabellängen gestatten. Die betrachteten Zielgrößen der Topologien sind dabei die Kosten, die Systemmasse und die maximal mögliche Datenrate. Für die Optimierung wurden zwei Topologien betrachtet, eine ringartige Struktur und zudem eine baumartige Topologie. Die Freiheitsgrade in dem Systementwurf sind beispielsweise die Verwendung unterschiedlicher Kabeltechnologien sowie die Variation

der Übertragungswege. Auch wenn Schulz et al. keine Optimierungsalgorithmen verwenden, gestatten das Analysemodell und die implementierten Methoden eine iterative, manuelle diskrete Optimierung der Systemtopologie und -architektur [107].

Die Optimierung zukünftiger Avioniksysteme verfolgt ebenso Annighöfer in seiner Arbeit. Auf Grundlage eines komplexen Systemmodells, abgebildet in einer eigenen Modellierungsumgebung basierend auf dem Eclipse Modeling Framework, werden neben der Systemarchitektur auch Integrationsaspekte der Topologie betrachtet und hinsichtlich Massen und avionikspezifischen Zielwerten iterativ optimiert. Die Berücksichtigung von Sicherheits- und Zuverlässigkeitsaspekten wird dabei durch qualitative Beschränkungen berücksichtigt, die unter anderem eine Segregation der untersuchten Funktionen vorschreiben. In dem aktuellen Stand der Entwurfsumgebung eignet sich das Verfahren für eine Variantenanalyse und eine schnelle Reaktion auf veränderliche Systemanforderungen und -parameter. In den nächsten Schritten ist eine Erweiterung der Methode zu einer diskreten Optimierungsumgebung vorgesehen [6].

Eine weitere spezifische Lösung für den Vorentwurf von IMA-Systemen hat Salomon entwickelt [104]. Der Fokus liegt hierbei in einer automatisierten Sicherheitsanalyse variabler Systemkonfigurationen und Funktionsallokationen, eingebunden in den Entwicklungsprozess. Basierend auf einem generischen, formalen Systemmodell werden hierfür automatisch Fehlerbäume abgeleitet. Dabei ist es möglich durch dedizierte Modellkomponenten die Erstellung von einem Fehlerbaum auszulösen und somit anhand des formalen Modells unterschiedliche Fehlerbedingungen abzuprüfen. Die Erstellung der Fehlerbäume basiert auf physikalischen Komponentenmodellen mit unterschiedlichen Fehlermodi, die auch die Fehlerpropagierung beschreiben. Dieser automatisierte Analyseprozess ist in eine Optimierungsumgebung eingebettet, der die Systemstruktur beeinflussen und somit unterschiedliche Varianten hinsichtlich der Sicherheitsanforderungen und der minimalen Systemmassen und -kosten auswerten kann.

Zusammenfassender Vergleich und Diskussion
Die untersuchten Verfahren und Studien zeigen, dass für die Vorentwicklung von Flugzeugsystemen eine Begrenzung auf einen Zielwert nicht zielführend ist. Es ist vielmehr die Analyse von system- und disziplinspezifischen Zielwerten mit flugzeugweiten Zielgrößen, den so genannten *Key Performance Parameters*, erforderlich. Koeppen betrachtet für den Flugzeugentwurf sogar ausschließlich globale Zielgrößen wie die Systemmasse, unter Berücksichtigung von Auslegungsapekten des betrachteten Systems und basierend auf bestehenden

Systemenrealisierungen. Optimierungsansätze wie von Holert, Pfennig und Giese verfolgen im Gegensatz dazu mit unterschiedlichen Optimierungsverfahren systemspezifische Anforderungen, wie dynamische Eigenschaften und Stellzeiten - jedoch unter Berücksichtigung der globalen Flugzeugparameter. Für die Optimierung werden dabei häufig Heuristiken verwendet.

Die Zielgrößen *Systemsicherheit und -zuverlässigkeit* werden nur von wenigen Analyse- und Optimierungsmethoden betrachtet. Wobei Koeppen, Liscouet-Hanke und Lüdders et al. explizit auf die Bedeutung der Sicherheit und Zuverlässigkeit für die Architekturauswahl hinweisen [59, 66, 72]. Als problematisch stellt sich hierbei wiederholt die Integration der Sicherheits- und Zuverlässigkeitsanalysen in den physikalisch basierten Systementwurf heraus.

Zur Integration der Sicherheitsanalysen bestehen in den untersuchten Methoden zwei unterschiedliche Ansätze. Im ersten Ansatz wird für die Systemauslegung die Prämisse definiert, dass die Sicherheitsanforderungen erfüllt werden, eine methodische Analyse erfolgt nicht [5]. Diesem Ansatz folgen beispielsweise Lüdders et al., implizit jedoch jede Analyse- und Optimierungsmethode, die quantitative Sicherheits- und Zuverlässigkeitsanalysen nicht als Zielgröße berücksichtigt. Vor allem bei neuen Systemtechnologien und geänderten Anforderungen an eine Systemarchitektur, kann jedoch häufig nicht auf Wissen aus vorherigen Flugzeugprojekten zurückgegriffen werden, so dass eine ausschließliche Betrachtung der Komponentenparameter zum Entwurf und zur Dimensionierung nicht ausreichend ist. Eine Erweiterung von dem ersten Ansatz zur Berücksichtigung der Sicherheits- und Zuverlässigkeitsanforderungen stellt die Methode von Annighöfer dar. Die Sicherheitsaspekte werden durch Nebenbedingungen der Optimierung abgebildet und somit wird sichergestellt, dass das Wissen aus vorherigen Systementwürfen aber auch explizite Redundanzanforderungen durch die Zulassungsvorschriften einfließen; auch wenn dieses nicht die Erfüllung der quantitativen Sicherheitsanforderungen gewährleistet. Den zweiten Ansatz bildet die Kopplung von Sicherheits- und Zuverlässigkeitsanalysen und dynamischen Simulationsmodellen mittels Monte-Carlo Simulation oder rechenintensiven, strukturierten Ansätzen, die alle Fehlerkombinationen eines Systems analysieren. Diesen Ansatz verfolgen beispielsweise Schallert und Saloman, wobei Letzterer die physikalischen Zusammenhänge zwischen den Komponenten vor allem zur Fehlerpropagation nutzt.

Die Verfahren zur sicherheitstechnischen Bewertung in Verbindung mit dynamischen Simulationen unterscheiden sich weitergehend in architekturvariable und -invariable Methoden. Schallert ermöglicht eine sicherheitstechnische

Analyse für bestimmte Fehlerfälle einer invariablen Architektur eines elektrischen Energieversorgungssystems, gekoppelt mit der Dimensionierung der enthaltenen Komponenten. Da die Minimalschnitte über die Kombinationen von injizierten Komponentenfehlern ermittelt werden, fließt die betrachtete Fehlertiefe und die Anzahl betrachteter Fehlerbedingungen stark in die Analysedauer ein. Zudem werden die Anforderungen Sicherheit und Zuverlässigkeit vor allem durch die Architektur getrieben.

Salomon, Bauer et al., Haitao et al. und weitere variieren daher im Gegensatz zu Schallert die Architektur und führen weitere Systemanalysen, z.B. Massenberechnungen, durch. Hierbei verfolgt Salomon eine Systemtopologie mit Fehlermodi der Komponenten, aus der Minimalschnitte für eine Sicherheitsfunktion ausgelesen werden. Die verwendeten Simulationsmodelle eigenen sich aufgrund der implementierten Fehlerpropagtion vor allem für avionische Systeme, ist jedoch mit angepassten Komponentenmodellen auch für mechatronische Systeme anwendbar. Als problematisch stellen sich bei der Fehlerpropagation mögliche Doppelfehler im System dar. Neben der logischen Abhängigkeit der Fehlermodi besteht durch das Modell auch eine zeitliche Abhängigkeit, die eine Berücksichtigung von Doppelfehlern nicht gestattet. Zudem ist eine Berücksichtigung externer Ereignisse bei dem Ansatz nicht möglich. Die Analysemodelle von Bauer et al. nutzen für die Architekturvariation übliche Boolesche Ansätze, berücksichtigen jedoch nur singuläre Sicherheitsanforderungen und gestatten nicht die Analyse komplexer Logiken.

Es mangelt daher an einer integrierten Entwurfsmethode von komplexen Flugzeug-Systemarchitekturen, die die Aspekte der Sicherheit- und Zuverlässigkeit berücksichtigt und hierbei auch weitere Entwurfsparameter, wie die Systemmasse analysiert. Dieses ermöglicht die Verdeutlichung der *Kosten* für unterschiedliche Redundanzkonzepte, unabhängig von der Technologie. Zudem ist eine Methode erforderlich die alle dimensionierenden Fehlerbedingungen aus der Gefahrenanalyse des betrachteten Systems berücksichtigt und nicht ausschließlich die Funktionsfähigkeit der Hauptfunktion.

1.2 Ziele der Arbeit

Diese Arbeit verfolgt eine interdisziplinäre Redundanzallokation und baut auf dem hybriden Analysemodell zur Sicherheits- und Zuverlässigkeitsanalyse von Vahl und Rehage auf [102, 126]. Im Gegensatz zu dem dort verfolgten Ansatz

zum quantitativen Nachweis von Systemanforderungen und der Identifikation von Schwachstellen im Systementwurf, verfolgt diese Arbeit den Vergleich unterschiedlicher Systementwürfe mit Hilfe eines variablen Architekturmodells [76]. Hierfür wurden die folgenden vier Kernfragen formuliert, die mit der entwickelten Methode zu beantworten sind und in den folgenden Abschnitten aufgegriffen werden:

1. Welches Redundanzkonzept ist systemweit und systemübergreifend zielführend bezüglich Sicherheit und Zuverlässigkeit?
2. Welchen Einfluss haben unterschiedliche Redundanzkonzepte auf das Degradationsverhalten eines Systems?
3. Wie wirken sich lokale Architekturvariationen auf die Systemsicherheit und -zuverlässigkeit aus?
4. Wie wirken sich Anforderungen und Klassifizierungen von Fehlerbedingungen auf die weiteren Systemparameter, z.B. die Systemmasse, aus?

Die Methode fokussiert somit den Architekturentwurf komplexer Systeme, wobei der Begriff „komplex“ nach EASA AMC25.1309 derart definiert ist, dass sowohl Betrieb, Fehlermodi und Fehlereffekte ohne analytische Methoden nur schwer zu handhaben sind[1] [30]. Als Zielwerte des Architekturentwurfs stehen die probabilistischen Gleichungen zur Berechnung der Sicherheit und Zuverlässigkeit im Vordergrund, die die notwendigen Redundanzen im System beschreiben. Desweiteren werden hierzu konträre Entwurfsparameter berücksichtigt. Der Architekturentwurf wird in dieser Arbeit als kombinatorisches Optimierungsproblem im Vorentwurf betrachtet. Es werden somit im vollständigen Optimierungs- und Auswahlprozess Komponenten und notwendige Redundanzen ausgewählt und deren Verschaltung definiert; dieses wird durch den Begriff der Redundanzallokation zusammengefasst. Die weitere Parametrisierung der ausgewählten Komponenten ist Gegenstand des nachfolgenden Systementwurfs, geeignete Verfahren hierfür wurden bereits im vorherigen Abschnitt betrachtet. Die Umsetzung des Entwurfsprozesses als Optimierungsproblem ermöglicht dabei nicht nur die Suche nach global optimalen Systemarchitekturen, sondern auch den Vergleich unterschiedlicher Systemvarianten und durch geeignete Modelle eine Reaktion auf veränderte Randbedingungen im Systementwurf und somit eine Überprüfung der Robustheit einer Lösung [70].

[1] Im Original: „*A system is Complex when its operation, failure modes, or failure effects are difficult to comprehend without the aid of analytical methods.*“

1.3 Gliederung der Arbeit

Die vorliegende Arbeit gliedert sich in sechs thematische Abschnitte. In Kapitel 2 wird der *Entwicklungsprozess* heutiger Flugzeugsysteme erläutert, wobei der Schwerpunkt des Kapitels auf dem Aspekt der *Systemsicherheit* und deren Nachweis liegt. Hierfür werden neben dem allgemeinen Entwicklungsprozess nach SAE ARP 4754 auch spezifische Umsetzungen unterschiedlicher Flugzeughersteller betrachtet. Das Kapitel dient zur Eingliederung der vorliegenden Arbeit in den Entwicklungsprozess und zeigt die Notwendigkeit eines automatisierten Verfahrens für den Vorentwurf komplexer Flugzeugsysteme.

Das Kapitel 3 leitet in den Stand der Technik der *Sicherheits- und Zuverlässigkeitsbewertung* komplexer Flugzeugsysteme und der Methoden der *Redundanzallokation* im Vorentwurf ein. Als Grundlage für den Architekturentwurf wird zur Zielwertberechnung zunächst das *hybride Systemmodell* von Vahl und Rehage vorgestellt. Dieses basiert auf Zuverlässigkeitsblockdiagrammen zur Abbildung der Systemstruktur in einer ersten Modellierungsebene und in einer zweiten Ebene hinterlegten nebenläufigen, endlichen Zustandsautomaten zur Abbildung von Rekonfigurationslogiken. Nachfolgend werden Verfahren zur *Sicherheits- und Zuverlässigkeitsoptimierung* allgemein industrieller Technologien in der frühen Entwicklungsphase untersucht. Hierbei werden die Begriffe zur *Zuverlässigkeits- und Redundanzallokation* eingeführt und die Möglichkeiten bisheriger Optimierungsverfahren untersucht. Das Kapitel schließt mit einem Konzept für ein *integriertes Verfahren zur optimalen Redundanzallokation* komplexer Flugzeug-Systemarchitekturen.

Das hybride Systemmodell von Vahl und Rehage wird nachfolgend für den *Architekturentwurf* komplexer Systemstrukturen zum *mehrfach-redundanten Systemmodell* erweitert. Dieses umfasst die Möglichkeit zur Einbringung von Freiheitsgraden in das Systemmodell, die Aufstellung des möglichen Architekturraumes anhand variabler Ereignisse und dessen Beschränkung durch *Nebenbedingungen*. Zudem werden Algorithmen zur automatisierten Ableitung degradierter Systemzustände und zur Berücksichtigung variabler, serieller Strukturen vorgestellt. Dieses ermöglicht die Untersuchung beliebiger, komplexer Strukturen und somit auch eine gegensätzliche Berücksichtigung von identischen Ereignissen für unterschiedliche Fehlerbedingungen und Zuverlässigkeitsanforderungen.

Auf Grundlage der vorgestellten Bewertungsverfahren mit Hilfe des erweiterten hybriden Systemmodells wird in Kapitel 5 eine Methode zur mehrkriteriel-

len Redundanzallokation komplexer Flugzeug-Systemarchitekturen vorgestellt. Dieses Verfahren gibt nicht nur eine einzelne Lösung aus, sondern verschafft dem Anwender einen Überblick über den optimalen Architekturraum: die so genannte Pareto-*Menge*. Die *Architekturfindung* wird hiermit quantifiziert und methodisch unterstützt. Diese Methode nutzt entsprechend der Charakteristika des behandelten Optimierungsproblems drei *problemspezifische Verfahren*: eine vollständige Enumeration, einen *Branch & Bound* Algorithmus und einen Genetischen Algorithmus. Die Auswahl und Konditionierung der Verfahren wird anhand von Beispielanwendungen erläutert. Der Ansatz zur Verwendung von drei unterschiedlichen Verfahren und die Auswahl der Verfahren wird zum Ende des Kapitels mit einem generischen Testszenario validiert.

Der Entwurfsprozess komplexer Systemarchitekturen umfasst neben der Berücksichtigung mehrkriterieller Bewertungsparameter auch die Unterstützung zur *Auswahl geeigneter Architekturen*. In Kapitel 6 wird daher die Integration in den zuvor erläuterten Entwicklungsprozess betrachtet. Dieses umfasst eine entwickelte Methode zur *Visualisierung mehrkriterieller Datensätze* und ein *hierarchisches Verfahren* zur Unsicherheitsbehandlung und Zielwertraumreduktion. Beide Verfahren unterstützen die Systementwicklung mit einem angeleiteten Entscheidungsprozess im Vorentwurf. Zur Veranschaulichung wird zudem ein illustratives Anwendungsbeispiel vorgestellt. Das Kapitel schließt mit einem Überblick über die Implementierung der entwickelten Methode.

Die Anwendbarkeit und Leistungsfähigkeit des entwickelten Verfahrens wird in Kapitel 7 anhand eines *industriellen Beispiels* nachgewiesen. Auf den Entwurf eines neuartigen *elektrischen Energieversorgungssystems* mit unterschiedlichen sekundären Generatoren und mehrfachen Sicherheits- und Zuverlässigkeitszielwerten wird die integrierte Entwurfsmethodik angewendet. Neben der Diskussion des Optimierungsverfahrens erfolgt hierfür auch eine Betrachtung der Systemarchitekturen anhand der Optimierungsergebnisse und eine angeleitete Reduktion des Zielwertraums, hierbei werden die vier formulierten Kernfragen der Arbeit aus dem vorherigen Abschnitt aufgegriffen.

Der Anhang der Arbeit enthält weiterführende Informationen zum illustrativen und industriellen Beispiel. Vor allem Ersteres wird anhand der orthogonalisierten Minimalpfade vollständig offen gelegt, so dass es für weitere Arbeiten im Bereich der Redundanzallokation von Flugzeug-Systemarchitekturen verfügbar ist. Das industrielle Beispiel wird um weitere Darstellungen der Zielwertmenge ergänzt.

2 Entwicklung von Flugzeug-Systemarchitekturen

In dem vorherigen Kapitel wurden unterschiedliche Methoden und Verfahren für den Vorentwurf komplexer Flugzeugsysteme betrachtet, um hieraus Anforderungen an ein Werkzeug zur Redundanzallokation abzuleiten. Das folgende Kapitel stellt ergänzend hierzu den Entwicklungsprozess und die sicherheitstechnischen Anforderungen an Flugzeugsysteme vor sowie die Einordnung dieser Arbeit in den bestehenden Prozess. Der vorgestellte Ablauf in Abschnitt 2.1 orientiert sich maßgeblich an der Richtlinie SAE ARP 4754 und einer möglichen industriellen Umsetzung des Entwicklungsprozesses. Der Fokus liegt dabei auf den Aspekten Systemsicherheit und -zuverlässigkeit; weitere Anforderungen werden jedoch ebenfalls berücksichtigt. Der Abschnitt 2.2 vertieft speziell den aktuellen Stand der Entwicklung von Flugzeug-Systemarchitekturen.

Zu Beginn jeder systematischen Systementwicklung steht dabei die Konzeptphase; der Ablauf ist in Bild 2.1 dargestellt. Das Ziel dieser Phase ist die Auswahl weniger Konzepte, deren Realisierungen die Anforderungen an das betrachtete System erfüllen können. Der erste Schritt ist hierfür auf Grundlage der Systemanforderungen die Entwicklung möglicher Systemkonzepte. Hierbei sind Kenntnisse und Erfahrungen aus ähnlichen Projekten, verfügbare Technologien und Ergebnisse aus abgeschlossenen und laufenden Forschungs- und Entwicklungsprojekten zu berücksichtigen [80]. Die Menge der entwickelten Systemkonzepte und deren vorläufige Systemarchitekturen sind anschließend mit Hilfe geeigneter Metriken gegen die Systemanforderungen zu validieren. Aufgrund der frühen Entwicklungsphase sind die dabei verwendeten Daten geprägt von hohen Parameterunsicherheiten, dieses ist bei der Bewertung von Konzepten zu berücksichtigen [88]. Entsprechend sollten bei der Konzeptbewertung eher wenige, validierte Daten verwendet werden, anstatt zahlreicher unsicherer Schätzwerte, so dass die Aussagekraft der Konzeptbewertung transparent ist. Die verwendeten Werkzeuge und Methoden sollten die Konzeptbewertung, Datenaufbereitung und Entscheidungsfindung unterstützen sowie die Kreativität und die Konzeptionsfreiheiten des Systemingenieurs nicht einschränken [106]. Die anschließende Reduktion der vollständigen Konzeptmenge auf ei-

ne überschaubare Untermenge ermöglicht in der folgenden Entwicklungsphase, der Systemdefinition, eine detaillierte Betrachtung der Lösungskandidaten und vorläufigen Systemarchitekturen.

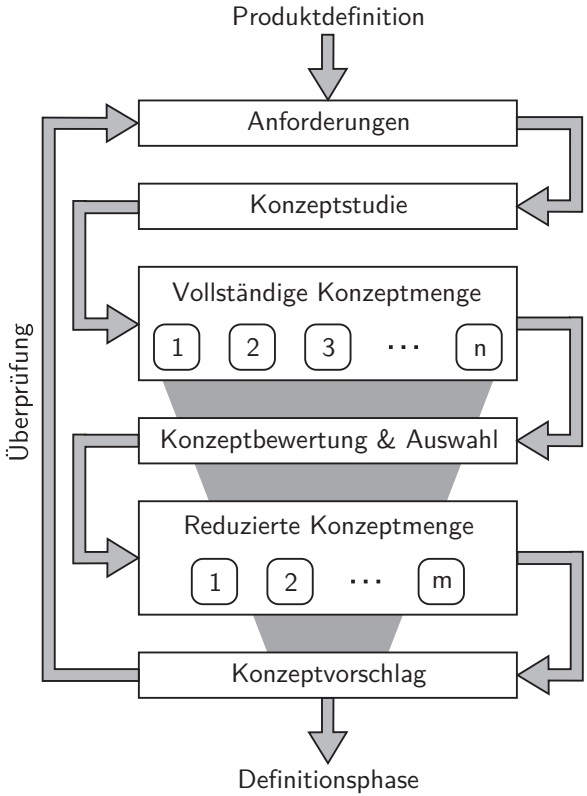


Abb. 2.1: Arbeitsschritte der Konzeptphase, nach [80]

Die Entwicklung der Systemarchitektur gehört laut INCOSE[1] zu einer der wichtigsten Aufgabe im Systementwurf, da sie die Grundlage für die weitere Entwicklung schafft [51]. Der Begriff „Architektur“ ist dabei nach RTCA DO178B und DO254 definiert als [91, 92]:

[1]Internationale Gesellschaft für Systems Engineering, engl. *International Council on Systems Engineering*

Definition Architektur nach RTCA: Die Struktur der Hardware und der Software, die ausgewählt wurde, um die Systemanforderungen umzusetzen[2].

Analog zu der vorherigen Definition ist der Begriff „Architektur“ gemäß IEEE wie folgt definiert [50]:

Definition Architektur nach IEEE: Die Architektur ist die Organisationsstruktur eines Systems oder einer Komponente[3].

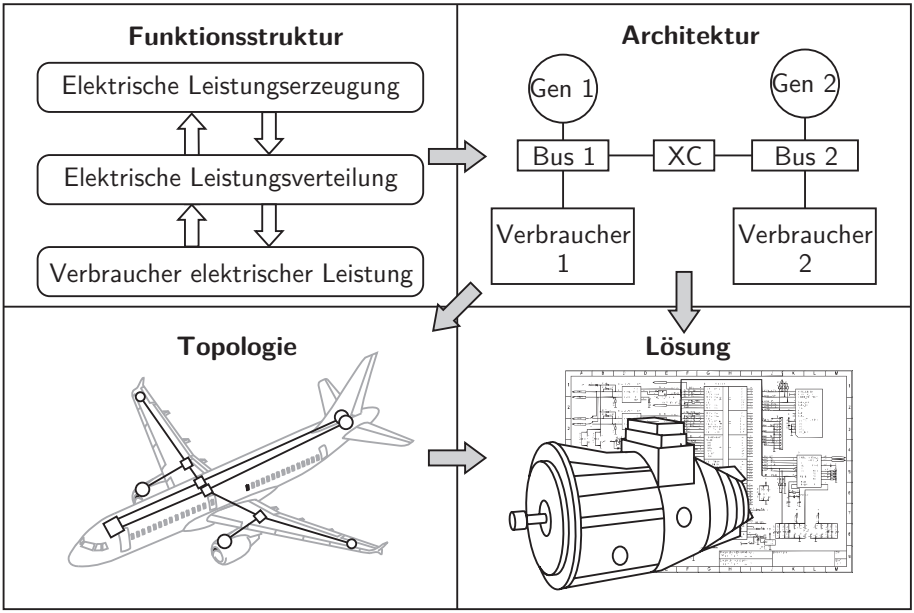


Abb. 2.2: Vergleich der vier definierten Beschreibungsformen im Rahmen der Systementwicklung

Die Systemarchitektur enthält dabei noch keine Informationen zur Installation oder räumlichen Lage der verwendeten Komponenten, diese sind in der späteren Topologie enthalten. Aufgrund der unterschiedlichen Definitionen und Verständnisse der verschiedenen Systembeschreibungen in der Literatur wer-

[2]Im Original: „*The structure of the hardware and the software selected to implement the system requirements.*“

[3]Im Original: „*The organizational structure of a system or component*“

den die folgenden Begriffe zur Vergleichbarkeit definiert, die auch im Laufe der weiteren Arbeit genutzt werden.

Definition Funktionsstruktur: Die *Funktionsstruktur* stellt auf höchster Ebene die Verbindung der prinzipiellen funktionalen Elemente einer Architektur dar. Sie ist somit der erste Entwurf einer Systembeschreibung und dient der weiteren Konkretisierung der funktionalen Elemente.

Definition Architektur: Die *Architektur* stellt die Verbindung dedizierter funktionaler Elemente und Technologien einer Funktionsstruktur dar. Sie dient zur Festlegung der Elemente und deren Verbindungen und somit zur Definition der Systemredundanzen. Sie kann neben der Verbindungen der Komponenten auch zustandsdiskrete Informationen wie Rekonfigurationslogiken oder weitere Parameter der Komponenten enthalten.

Definition Topologie: Die *Topologie* umfasst die weitere Konkretisierung einer Architektur um deren räumliche Anordnung im übergeordneten System. Sie enthält somit Aspekte der Integration, wie Bauraum und Leitungslängen und ist nicht Gegenstand dieser Arbeit, sondern einzuordnen in die Definitionsphase.

Definition Lösung: Das technologisch und parametrisch vollständig definierte System wird als *Lösung* bezeichnet. Sie ist nicht Gegenstand dieser Arbeit und einzuordnen in den Detailentwurf, somit umfasst sie erste Konstruktionszeichnungen und Bauunterlagen.

In Abbildung 2.2 ist der Zusammenhang zwischen den vier Systembeschreibungen exemplarisch für den Entwurf eines elektrischen Energieversorgungssystems dargestellt. Die Schnittstellen zwischen den einzelnen Systembeschreibungen umfassen dabei Anforderungen, die sich aus den einzelnen Bewertungsphasen ergeben. So können anhand der Analyse der Systemarchitektur Anforderungen bezüglich der physikalischen Segregation redundanter Komponenten an die Systemtopologie gestellt werden. Zudem beeinflusst die Auswahl einer Systemarchitektur auf Grundlage der Fehlerklassifizierungen die Systemlösung hinsichtlich der notwendigen Dokumente und Entwurfsrichtlinien.

Im Folgenden wird der industrielle Entwicklungsprozess von Flugzeugsystemen näher betrachtet. Der Fokus liegt hierbei vor allem auf dem Aspekt der Systemsicherheit.

2.1 Entwicklungsprozess von Flugzeugsystemen

Die Entwicklung von Flugzeugsystemen wird maßgeblich durch das Geschäfts-, das Projekt- sowie das Produktumfeld, die Betriebsbedingungen und Schnittstellendefinitionen beeinflusst, z.B. zu weiteren Systemen oder Subsystemen [80]. Hieraus resultiert die inner- und intersystemische Komplexität bei der Entwicklung von Flugzeugsystemen. Aufgrund der zahlreichen Einflüsse ist für eine strukturierte Systementwicklung ein industrieller Entwicklungsprozess notwendig, der das gesamte Systemumfeld berücksichtigt, alle Anforderungen an das System erfasst und deren Erfüllung sicherstellt. Dieser Ansatz des *Systems Engineering* ermöglicht unter Berücksichtigung der luftfahrtspezifischen Richtlinie SAE ARP 4754 zur Entwicklung komplexer Systeme eine Zertifizierbarkeit von Flugzeugsystemen und somit eine zielgerichtete Systementwicklung [117]. Desweiteren verlangen auch die allgemeinen industriellen Normen ISO9001 zum Qualitätsmanagement und ISO14001 zum Umweltmanagement den Nachweis eines struktierten Entwicklungsprozesses [115]. Die hierfür notwendigen Schritte sind in Abbildung 2.3 für Flugzeugsysteme dargestellt.

Der dargestellte, allgemeine Prozess bedarf zur Realisierung einer Umsetzung in einen unternehmensspezifischen Prozess, wie er exemplarisch bei Airbus durch den Prozess AP2288 *Requirements for Systems and Equipment Development* umgesetzt wurde [115]. Anhand der Direktiven ABD0100 zur Komponentenentwicklung und ABD0200 zur Systementwicklung kann dieser Prozess weiter unterteilt werden. Dabei ist der definierte Prozess der Richtlinie ABD0200, *Requirements for the System Designer*, für die vorliegende Arbeit maßgebend. Ähnliche Prozesse haben weitere Flugzeughersteller sowie die NASA definiert[4].

Zu Beginn der Systementwicklung stehen die Anforderungen auf Flugzeugebene, die von den Verantwortlichen des Flugzeugprogramms definiert werden, siehe Abbildung Abbildung 2.3. Verglichen mit der gängigen Darstellung als V-Prozess werden die Flugzeuganforderungen im oberen linken Teil aufgestellt. Mit Hilfe der so genannten *Top-Level*-Anforderungen können die Flugzeugfunktionen auf die definierten Systeme allokiert werden. Die Umsetzung dieser Funktionen erfordert in Abhängigkeit der Systemkritikalität und funktionalen Anforderungen eine Systemarchitektur, die den Entwurfsanforderungen gerecht wird.

[4] Boeing definiert programmspezifisch die *Design Requirements and Objectives*, die NASA hat das *System Engineering Handbook* zur Definition des Entwicklungsprozesses komplexer Systeme herausgegeben [82, 128]

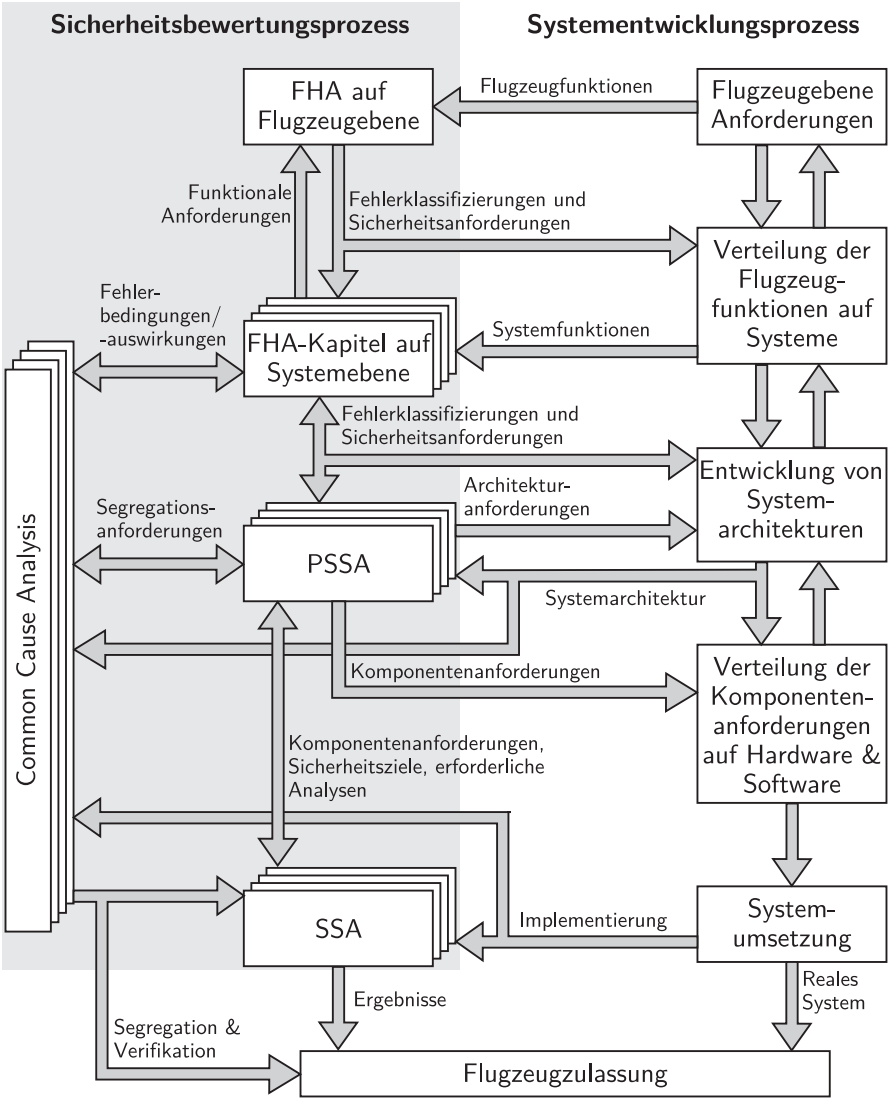


Abb. 2.3: Sicherheitsbewertungsprozess im Rahmen der Entwicklung von Flugzeugsystemen, nach [117]

Nachdem die Systemarchitektur definiert ist und die Funktionen allokiert sind, kann auf der Komponentenebene die Umsetzung der Funktionen entworfen werden. Dieses umfasst in modernen mechatronischen Flugzeugsystemen nicht nur die konstruktive Umsetzung, sondern vorab auch die Untersuchung inwieweit sich Komponenten- aber auch Systemfunktionen durch Hard- oder Software umsetzen lassen. Die nachfolgende Komponentenentwicklung bildet gemäß des konventionellen V-Prozesses den Scheitelpunkt der Systementwicklung [80]. Den Abschluss des allgemeinen Entwurfsprozesses bildet die Systemimplementierung und -integration und Überprüfung der gestellten Anforderungen an das System; dieses entspricht dem aufsteigenden Ast des V-Prozesses und Verbindungen zur gegenüberliegenden Anforderungs- und Entwurfsphase.

Als integraler Bestandteil des Systementwicklungsprozesses von Flugzeugen verläuft der Sicherheitsbewertunsgprozess parallel zu diesem und liefert in den einzelnen Phasen der Systementwicklung wichtige Bewertungen, die für eine Zulassung notwendig sind. Abbildung 2.3 veranschaulicht die starke Interaktion der Systementwicklung mit dem Sicherheitsbewertungsprozess nach SAE ARP 4761 [116]. Für die Definition der Systemarchitektur ist dabei der Prozessschritt „*Entwicklung von Systemarchitekturen*“ entscheidend und wird in Abschnitt 2.2 näher betrachtet. Aufgrund der Bedeutung des gesamten Sicherheitsbewertungsprozesses für die Architekturdefinition wird dieser im Folgenden näher erläutert und detailliert.

Zu Beginn der Sicherheitsanalysen steht die funktionale Gefahrenbewertung (engl. *Functional Hazard Assessment, FHA*) auf Systemebene. Neben einer vollständigen Systembeschreibung enthält diese Bewertung alle Fehlerszenarien des Systems und dessen Komponenten, z.B. „*Funktionsverlust der elektrischen Notenergieversorgung*“. Jedes Fehlerszenario wird dabei entsprechend Tabelle 2.1 gemäß der Zulassungsvorschriften klassifiziert [31]. Die Klassifizierung berücksichtigt dabei die Folgen des Fehlerszenarios und verknüpft diese mit einer zulässigen Eintrittswahrscheinlichkeit pro Flugstunde, so dass ein konstantes Risiko, dem Produkt aus Gefährdung und Eintrittwahrscheinlichkeit, verfolgt wird. Neben den innersystemischen Fehlerszenarien werden zudem intersystemische Fehler klassifiziert, die das analysierte System betreffen, z.B. die Leistungsversorgung oder die Datenübertragung. Die Klassifizierung dieser abhängigen Systemfunktionen wird an die ursprüngliche Systemanalyse weitergeleitet, um die entsprechende Klassifizierung sicherzustellen. Desweiteren werden externe Ereignisse in Kombination mit inner- und intersystemischen Fehlerszenarien berücksichtigt, entsprechend des oben genannten Beispiels exemplarisch

das Szenario „*Funktionsverlust der elektrischen Notenergieversorgung im Falle eines vollständigen Triebwerksverlustes*“.

Tab. 2.1: Gefahrenklassen und korrelierte zulässige Eintrittswahrscheinlichkeiten nach EASA CS 25

R_S [1/FH]	Deskriptiv	Gefahrenklasse	Fehlereffekt	*DAL* [117]
$\leq 10^{-0}$	Keine Anforderung	*No Safety Effect*	Keine Auswirkungen auf den Flugbetrieb	E
$\leq 10^{-3}$	*Probable*	*Minor*	leichte Erhöhung der Arbeitslast für die Flugzeugbesatzung, leichte Reduktion der Sicherheitsreserven	D
$\leq 10^{-5}$	*Remote*	*Major*	Erhöhung der Arbeitslast für die Flugzeugbesatzung, Reduktion der Sicherheitsreserven, Unbehagen der Passagiere	C
$\leq 10^{-7}$	*Improbable*	*Hazardous*	wesentliche Erhöhung der Arbeitslast für die Flugzeugbesatzung, wesentliche Reduktion der Sicherheitsreserven, Verletzungen der Passagiere	B
$\leq 10^{-9}$	*Extremely Improbable*	*Catastrophic*	sicherer Flug nicht mehr möglich	A

Aufgrund der Auswirkungen der einzelnen Fehlerszenarien können diese zu Fehlerbedingungen (engl. *Failure Conditions*) auf Systemebene zusammengefasst werden, die höchste Fehlerklassifizierung eines Szenarios klassifiziert dabei die Fehlerbedingung. An die betroffenen Komponenten, ebenso die verwendete Software, der einzelnen Fehlerbedingungen werden anschließend aufgrund der Klassifizierung Anforderungen gestellt. Die Klassifizierung gibt dabei nicht

nur die Kritikalität der Komponente an, sondern definiert auch den weiteren Entwicklungsprozess. Das Entwurfsniveau (engl. *Design Assurance Level, DAL*) gilt dabei sowohl für Komponenten als auch für Systeme. Wobei sich dieses Entwurfsniveau aufgrund der Anzahl der Ereignisse zur Erfüllung der Fehlerbedingungen degradieren lässt. Tabelle 2.2 veranschaulicht die zulässige Degradation des Komponenten-DALs [117]. Die Fehlerklassifizierung und somit die Zuordnung des *Design Assurance Levels* für die verwendeten Komponenten und die genutzte Software sind ein hauptsächliches Ergebnis der FHA für die Gestaltung des weiteren Entwicklungsprozesses [61].

Tab. 2.2: Zulässige Degradation des *Design Assurance Levels* [117]

		Redundanzgrad		
		0	1	2
Klassifizierung	*Catastophic*	A	B	C
	Hazardous	B	C	D
	Major	C	D	D
	Minor	D	D	D
	No Safety Effect	E	E	E
		Degradierter DAL		

Basierend auf den Ergebnissen der FHA können in den nächsten Schritten zunächst vereinfachte und nachfolgend detailliertere Systemarchitekturen entworfen und bezüglich der Sicherheit in der vorläufigen Systemsicherheitsanalyse (engl. *Preliminary System Safety Analyis, PSSA*) untersucht werden. Der Beschreibung des Entwurfsprozesses hierfür folgt in Abschnitt 2.2. Für die PSSA werden für die zuvor klassifizierten Fehlerbedingungen die Nachweise entsprechend EASA CS 25.1309 erbracht, die mit den Zielen in Tabelle 2.1 übereinstimmen [31]. Gängige und akzeptierte Nachweismethoden werden in der Richtlinie SAE ARP 4761 vorgestellt. Diese enthält Fehlerbäume, Zuverlässigkeitsblockdiagramme (engl. *Reliability Block Diagrams/Dependency Diagrams*) und Markov-Ketten zur Abbildung reaktiver und reparierbarer Systeme. Die Grundlagen dieser Berechnungsmethoden und äquivalenter Verfahren werden in Kapitel 3 vorgestellt. Das Ergebnis der PSSA besteht zum einen in einer Bewertung der Systemarchitektur, zum anderen in der Ableitung von Komponenten- und Systemanforderungen, z.B. bezüglich Fehlerraten oder Überwachungsfunktionen. Somit ist es beispielsweise möglich für offene Punkte im Systementwurf

technisch sinnvolle Annahmen zu treffen und deren Erfüllung für die Implementierung zu fordern. Der Prozess zur Erstellung der PSSA und Entwicklung von Systemarchitekturen, wie er in Abbildung 2.3 veranschaulicht wird, wird im nachfolgenden Abschnitt näher erläutert, um die vorliegende Arbeit in diesem Kontext einzuordnen.

Das implementierte System wird im Rahmen der Systemsicherheitsanalyse (engl. *System Safety Assessment, SSA*) untersucht und ist daher im Entwicklungsprozess in Form des V-Modells auf der rechten, aufsteigenden Seite einzuordnen [61, 80]. Neben der Detaillierung der vorherigen quantitativen Analysen, enthält die SSA die Festlegung sicherheitsrelevanter Wartungschecks sowie die Ergebnisse der qualitativen Analysen zur zonalen Sicherheit, zu gemeinsamen Fehlermodi und speziellen Risiken, wie Blitzschlag. Die SSA dient somit nicht nur als Dokumentation der Systementwicklung, sondern ist auch ein zertifizierungsrelevantes Dokument für die Zulassungsbehörden [67].

Die drei zuvor genannten qualitativen Analysen werden zusammengefasst als gemeinsame Ursachenanalyse (engl. *Common Cause Analysis, CCA*) und untersuchen alle als katastrophal (*CAT*) klassifizierten Fehlerbedingungen auf eventuell redundanzüberbrückende Ereignisse [57, 61]. Dabei werden im Rahmen der *Zonal Safety Analysis* mögliche Interaktionen von installierten Systemen innerhalb einer Flugzeugsektion untersucht. Dieses kann somit Aspekte wie austretende Flüssigkeiten, Entflammbarkeit von Fluiden oder auch elektromagnetische Verträglichkeit betreffen. Die *Particular Risk Analysis* hingegen untersucht inwieweit durch einzelne zonenübergreifende Ereignisse oder auch externe Ereignisse vermeintliche Redundanzen im Systementwurf überbrückt werden. Beispiele hierfür sind die Untersuchung möglicher Trajektorien von Reifen- oder Triebswerkteilen, so dass durch solch ein Einzelereignis alle redundanten Energieversorgungssysteme zerstört werden. Über den gesamten Entwicklungsprozess wird desweiteren die *Common Mode Analysis* fortgeführt, die nach singulären Fehlermodi von Redundanzen sucht. Dieses kann sowohl innersystemische Fehlermodi wie kaskadierte Fehlerfolgen betreffen, jedoch auch Aspekte wie Komponentenentwurf, Programmimplementierung und Anforderungen an die Wartung betreffen. Die Ergebnisse der *CCA* stellen maßgeblich Anforderungen an die Installation von Flugzeugsystemen, jedoch auch an den Betrieb, so dass sich aktive aber auch passive Redundanzstrategien etabliert haben, zum Beispiel durch die Isolation redundanter Generatoren vom elektrischen Netz zum Schutz gegen Überspannungen [116, 124].

Neben den nominellen Systemzuständen erfordert ein wirtschaftlich erfolgreicher Flottenbetrieb einer Fluggesellschaft zudem häufig die Analyse degradierter Systemzustände [66]. Somit wird sichergestellt, dass auch mit definierten Komponentenausfällen ein sicherer Flugbetrieb möglich ist und die beiden häufig konträren Aspekte Sicherheit des Flugbetriebs und Zuverlässigkeit des Flugzeugbetriebs vereint werden können. Die Systemzuverlässigkeit ist dabei für die Zertifizierung durch die Zulassungsbehörden nicht relevant, sondern nur gegenüber den späteren Flugzeugbetreibern, denen für einen Flottenbetrieb eine operationelle Zuverlässigkeit garantiert wird. Die operationelle Zuverlässigkeit (*Operational Reliability, OR*) gibt dabei an, mit welcher Wahrscheinlichkeit ein Flug planmäßig begonnen und auch beendet werden kann und ist somit im Verbund mit der *Dispatch Reliability*, die nur den planmäßigen Start berücksichtigt, die signifikante Größe für den Systementwurf [80]. Die Berechnungsmethoden dabei sind ähnlich denen der Sicherheitsbewertung, die Fehlereignisse der integrierten Komponenten werden jedoch entsprechend des untersuchten Ereignisses in vielen Fällen konträr berücksichtigt. Eine Redundanz und somit eine parallele Anordnung von zwei Komponenten verringert zwar die Ausfallwahrscheinlichkeit bezüglich der Systemsicherheit, für die Zuverlässigkeitsberechnung würde diese Redundanz jedoch als serielle Logik betrachtet werden und somit die Ausfallwahrscheinlichkeit erhöhen. Dieses einfache Beispiel zeigt, dass die Systemzuverlässigkeit ebenfalls einen Parameter darstellt, der maßgeblich durch die Systemarchitektur bestimmt wird und dadurch ebenfalls frühzeitig im Vorentwurf zu berücksichtigen ist. Würden bei der Komplexität von modernen Flugzeugsystemen bereits Einfachfehler zum Ausfall des betrachteten Gesamtsystems führen, wären die herausfordernden und stetig steigenden Zuverlässigkeitsanforderungen der Flugzeugbetreiber aufgrund der Menge der beteiligten Komponenten nicht zu erfüllen. Aus diesem Grund werden die bereits zuvor erwähnten degradierten Systemzustände untersucht und zulässige Systemdegradationen in einer minimalen Betriebsliste (engl. *Master Minimum Equipment Liste, MMEL*) festgehalten [4]. Anhand dieser MMEL können sich Flugzeugbetreiber eine MEL (engl. *Minimum Equipment List*) ableiten. Diese enthält Informationen über die Anzahl installierter Komponenten und wie viele der Komponenten für einen Start erforderlich sind. Die letzte Entscheidungsinstanz bleibt jedoch beim Piloten, zudem gelten für erweiterte Flugstrecken (engl. *Extended Operations, ETOPS*) in vielen Fällen gesonderte Bedingungen [4].

Neben den Aspekten der Systemsicherheit und -zuverlässigkeit müssen weitere System- und Flugzeugparameter in den unterschiedlichen Entwicklungsstufen

auf Grundlage der verfügbaren Daten berechnet werden, Abbildung 1.1 hat bereits exemplarisch einige Schnittstellen zwischen System- und Flugzeugentwicklung aufgezeigt.

2.2 Integrierte Entwicklung von Systemarchitekturen

Die Entwicklung einer fehlertoleranten Flugzeug-Systemarchitektur ist der einzige Weg die anspruchsvollen Sicherheits- und Zuverlässigkeitsziele zu erreichen. Fehlertolerant bedeutet in diesem Fall, dass die betroffene Systemfunktion nach einem Fehler noch vollständig oder nur bedingt einschränkt ausgeführt werden kann. Hierfür werden unterschiedliche Redundanzstrategien verwendet [27]. Im Folgenden wird daher separat der Prozess zur Entwicklung von Systemarchitekturen aus Abbildung 2.3 betrachtet und weiter aufgegliedert.

Entsprechend Abbildung 2.3 wird die Entwicklung und Auswahl von möglichen Systemarchitekturen neben den funktionalen Anforderungen maßgeblich durch die Sicherheitsanforderungen an die Systemfunktionen bestimmt. In Abbildung 2.4 ist der Entwurfs- und Entscheidungsprozess für die Architekturauswahl detaillierter dargestellt. Basierend auf den Systemfunktionen kann eine vorläufige Gefahrenbewertung (*Preliminary Functional Harzard Assessment, PFHA*) durchgeführt werden, die die grundlegenden Funktionen und Systembestandteile klassifiziert. Die Ergebnisse der *PFHA* werden als Ergebnisse bezüglich der notwendigen Redundanzen und Funktionssegrationen für die Entwicklung der Systemarchitekturen genutzt.

Bei der Entwicklung von Flugzeug-Systemarchitekturen kann generell zwischen drei Redundanzkonzepten unterschieden werden: die primäre/integrale Redundanz, die sekundäre Redundanz und Redundanzen zur Vermeidung weiterer Schäden [48]. Bei der primären Redundanz handelt es sich um eine offensichtliche Redundanz in Form einer weiteren funktionalen Komponente, die auch im normalen Betrieb genutzt wird, beispielsweise die Verwendung von zwei unabhängigen elektrischen Energieversorgungssystemen. Im Bereich der Sicherheits- und Zuverlässigkeitstheorie wird hierbei auch von einer heißen Redundanz gesprochen [10]. Die sekundäre Redundanz hingegen nutzt als weiteren Pfad entweder eine kalte Redundanz baugleich zum primären Lastpfad oder einen unabhängigen Lastpfad, beispielsweise eine *Ram Air Turbine* nach einem Ausfall der primären elektrischen Triebwerksgeneratoren. Daneben besteht zudem das

Konzept der schadenstoleranten Redundanz, die größtenteils die Integration der Komponenten berücksichtigt und z.B. aufgrund eines strukturellen Schutzes die Ausbreitung kaskadierter Fehler verhindert. Ein Beispiel für letztere Redundanz ist die Integration einer Feuerschutzwand zwischen den Leistungsversorgungszentren im elektrischen Energieversorgungssystem, jedoch auch Sicherungen gegen Überlast. Die Konzepte der schadenstoleranten Redundanz gewährleisten somit bezüglich der *Common Cause Analysis* auch eine Unabhängigkeit der weiteren Redundanzen. Die genannte Schutzwand entkoppelt somit ein Feuer auf einer Systemseite von der Funktionsfähigkeit der verbleibenden Seite und somit ist ein Aspekt zur unabhängigen Berücksichtigung der Redundanzen in den Zuverlässigkeitsblockdiagrammen erfüllt.

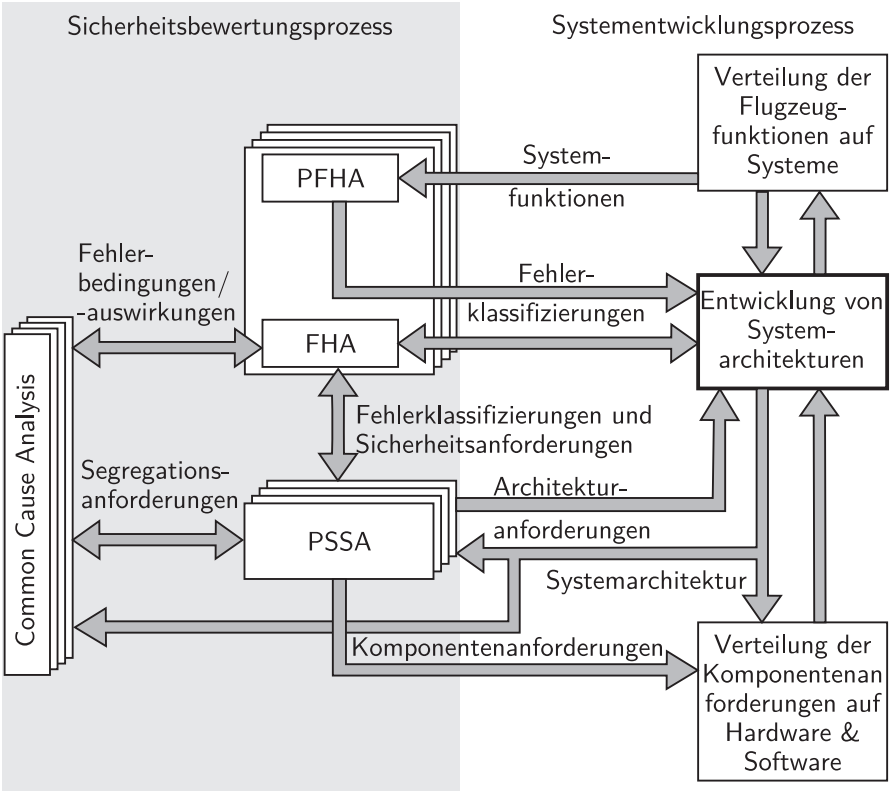


Abb. 2.4: Entwicklungsprozess von Systemarchitekturen

Für die vorliegende Arbeit sind vor allem die ersten beiden Konzepte zur primären und sekundären Redundanz relevant. Wobei zu berücksichtigen ist, dass komplexe, fehlertolerante Systemarchitekturen nur für Systeme mit einem System-DAL A oder B relevant sind, dieses betrifft normalerweise die Grundsysteme von Flugzeugen wie beispielsweise die Energieversorgungssysteme und die Flugsteuerung. Doch selbst sicherheitskritische Systeme müssen nicht zwangsläufig über komplexe Systemstrukturen verfügen, sofern Ereignisse die zu katastrophalen Ereignisfolgen führen, nicht innerhalb des Systems liegen. Solche Systeme gelten als intrinsisch sicher. Ein Beispiel hierfür ist das Brandschutzsystem von Flugzeugen, bei denen der Ausfall des Systems solange unkritisch ist, bis als externes Ereignis ein Feuer hinzukommt. Wesentlich für die intrinsische Sicherheit ist hierbei jedoch, dass die Ursache des Feuers nicht in dem Brandschutzsystem selbst liegt, was entsprechend konstruktiv umzusetzen ist.

Die Ergebnisse der PFHA basieren auf der Funktionsstruktur des betrachteten Systems und den Konzepten zur Realisierung der Systemfunktionen, sie können somit auch nur entsprechend detaillierte Fehlerszenarien betrachten. In die Klassifizierung der Fehlerszenarien fließen dabei, soweit verfügbar, die Auswirkungen auf die Flugzeugsicherheit, die Detektierbarkeit, der Arbeitsaufwand der Flugzeugbesatzung sowie der Zustand und die Funktionsfähigkeit nach einem Fehler ein [67]. Die Ergebnisse werden direkt als Eingabe für die Architekturauswahl genutzt, wobei vor allem die dimensionierenden Szenarien und die resultierenden Fehlerbedingungen zu berücksichtigen sind. Diese beinhalten neben den katastrophalen Fehlerbedingungen auch jene Fälle, die beispielsweise die Integration einer neuen Technologie betreffen oder hinsichtlich möglicher Systemarchitekturen als kritisch gelten. Überlicherweise werden dabei nur die Klassifizierungen *catastrophic* bis *major* berücksichtigt, die Klassifizierung *minor* wird mit Hilfe gängiger Luftfahrtkomponenten durch die Zulassungsbehörden als erfüllt angesehen. Das Erreichen von Eintrittswahrscheinlichkeiten der Kategorie *major* kann hingegen bereits redundante Pfade erfordern, da eine serielle Logik mit den Fehlerraten üblicher Luftfahrtkomponenten je nach System nicht die Erfüllung garantiert. Entsprechend sind auch diese Fehlerbedingungen für den Architekturentwurf zu bedenken.

Auf Grundlage der Fehlerbedingungen und deren Klassifizierungen ist es möglich, unterschiedliche Systemarchitekturen zu untersuchen und so eine Architektur für den weiteren Entwicklungsprozess auszuwählen. Neben der Sicherheit und Zuverlässigkeit einer Architektur wird die Auswahl auch durch Aspekte wie die Systemmasse und zu erwartende Lebenszyklenkosten bestimmt [80].

Die Entwicklung von Systemarchitekturen stellt im weiteren Verlauf einen interativen Prozess dar, an dessen Anfang eine möglichst große Lösungsvielfalt bestehen sollte, da die Detailanalysen auf der Auswahl durch die vorherigen Bewertungen basieren [64, 80]. Die Variation der Architekturen besteht dabei neben den bereits erläuterten Redundanzkonzepten in der verwendeten Technologie zur Erfüllung von Systemfunktionen und Fragestellungen zur Anbindung dieser Funktionen. Beschränkt wird der mögliche Architekturraum in dieser frühen Phase durch die Erkenntnisse vorheriger Programme, Normen und Richtlinien, verfügbare Kenntnisse und Technologien und technologisch und ökonomisch sinnvolle Variationen, die der Systemingenieur vorgibt [80].

Die abschließend ausgewählte Architektur durchläuft den weiteren Entwicklungprozess gemäß Abbildung 2.4, wobei zu berücksichtigen ist, dass die Ergebnisse der *FHA*, der *PSSA* und der weiteren Bewertungen weiterhin die Systemarchitektur beeinflußen können. Eine Adaption der Architektur ist immer dann erforderlich, wenn die detaillierten Analysen ergeben, dass die vorherigen oder auch neue Anforderungen nicht erfüllt werden können. Um die Veränderungen der Architektur im weiteren Verlauf gering zu halten, ist die Architektur zu Beginn möglichst robust gegenüber den Systemanforderungen auszuwählen.

3 Sicherheits- und zuverlässigkeitstechnische Bewertungsverfahren

Das folgende Kapitel stellt zunächst das von Vahl und Rehage entwickelte hybride Systemmodell unter Verwendung von Zuverlässigkeitsblockdiagrammen und nebenläufigen, endlichen Zustandsautomaten vor. Entsprechend der Leitfragen aus Abschnitt 1.2 bildet dieses Systemmodell die Grundlage für die spätere Architekturbewertung. Neben der Formulierung der Zielfunktionen werden in diesem Kapitel zudem Methoden der Sicherheits- und Zuverlässigkeitsallokation vorgestellt. Auf Grundlage der Methoden und Prozesse der vorherigen Abschnitte und den Verfahren aus dem folgenden Abschnitt wird zum Schluss dieses Kapitels ein Konzept zur Redundanzallokation mit Systemmodellen variabler Struktur abgeleitet.

3.1 Hybride Sicherheits- und Zuverlässigkeitsanalyse

Die Architekturoptimierung von komplexen Flugzeugsystemen hinsichtlich der quantitativen Sicherheit und Zuverlässigkeit erfordert eine Abbildung von architekturellen Freiheitsgraden. Zur Differenzierung der Architekturen ist zudem eine exakte Berechnung der Eintrittswahrscheinlichkeiten auf Basis der individuellen Fehlerereignisse notwendig. Nachdem in den vorherigen Abschnitten die Inhalte und Entwurfsmethoden der Konzeptphase vorgestellt wurden, erläutert dieser Abschnitt die Zielfunktionen auf Grundlage das hybriden Systemmodells von Vahl und Rehage. Dieses Modell umfasst die Analyse mit Hilfe von Zuverlässigkeitsblockdiagrammen und nebenläufigen, endlichen Zustandsautomaten und ist in dem Werkzeug SyRelAn implementiert [102, 126]. Die Eignung des hybriden Systemmodells für die Redundanzallokation wird im Folgenden nachgewiesen.

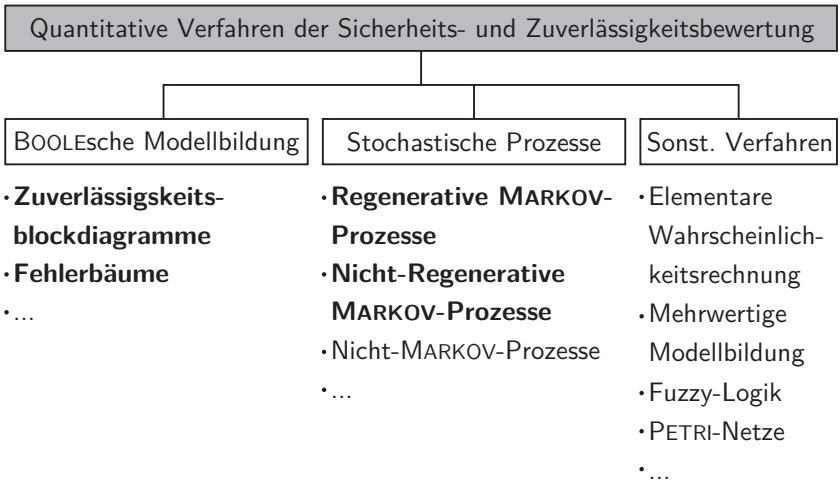


Abb. 3.1: Übersicht quantitativer Sicherheits- und Zuverlässigkeitsbewertungsverfahren

Abbildung 3.1 zeigt einen Überblick über vorhandene Verfahren zur Sicherheits- und Zuverlässigkeitsbewertung [77, 93]. Die empfohlenen Methoden zur Nachweisführung und Bewertung von Flugzeugsystemen gemäß SAE ARP 4761 sind gesondert hervorgehoben. Diese stellen einen etablierten Standard in der Luftfahrtindustrie dar und unterstützen somit das Analyseverständnis von Systemingenieuren [116].

ABELE und RAKOWSKY bieten eine umfassende Diskussion der wichtigsten Verfahren und Methoden der Sicherheits- und Zuverlässigkeitstechnik [1, 93]. Die verfügbaren Methoden und deren Eigenschaften lassen sich mit den Kernfragen zur Redundanzallokation aus Abschnitt 1.2 abgleichen. Dabei zeigt sich, dass sich vor allem die klassischen Ansätze der Fehlerbaumanalyse und Zuverlässigkeitsblockdiagramme für eine variable Strukturmodellierung eignen. Neuere Ansätze wie die Fuzzy-Methoden und Petri-Netze hingegen bieten nicht die notwendige Transparenz im komplexen Systementwurf. Zudem sind diese Verfahren rechenintensiv, was gegen eine Nutzung zur wiederholten Zielwertberechnung in einer Optimierung spricht [93].

Die Eigenschaften der weiteren Verfahrensgruppen aus Abbildung 3.1 lassen sich ebenfalls den Kernfragen gegenüberstellen. Verfahren der elementa-

ren Wahrscheinlichkeitsrechnung, zu denen auch die Zusammenfassung seriell-paralleler Strukturen gehört, ermöglichen nicht die exakte Berechnung komplexer Strukturen, wie sie bei Flugzeugsystemen vorkommen. Eine konservative Abschätzung der Systemsicherheit würde eine Berechnung komplexer Strukturen mittels einfacher Zielfunktionen ermöglichen. Aufgrund der Varianz des Lösungsraumes würden pessimistische Abschätzungen jedoch die Unterschiede der Architekturen verschwimmen lassen. Daher kommen diese Verfahren nicht für die Analyse beliebiger, komplexer Flugzeugsysteme in Frage [93]. Eine weitere Gruppe von rechenzeitoptimierten jedoch approximativen Verfahren, beispielsweise Varianten der elementaren Wahrscheinlichkeitsrechnungen oder das Vernachlässigen von Zustandswahrscheinlichkeiten ab einer zu definierenden Degradationsstufe, eigenen sich aus diesem Grund ebenfalls nicht für eine automatische Bewertung des Architekturraumes [96].

Die Berücksichtigung regenerativer Prozesse, beispielsweise mit Hilfe von Markov-Prozessen, ist in der Vorentwurfsphase aufgrund der zur Verfügung stehenden Informationen nicht sinnvoll. Zur Abbildung der Systemregeneration wären Informationen zur Verfügbarkeit von Ersatzteilen und Reparaturzeiten notwendig, die in der Vorentwurfsphase üblicherweise jedoch nicht vorliegen und für die Architekturdefinition auch nicht notwendig sind. Aus diesem Grund kommt die relativ komplexe Modellierung und Analyse der Modelle für die Optimierung nicht in Frage [88, 102]. Nicht-Regenerative Prozesse, beispielsweise Markov-Prozesse ohne Reparatur oder die Modellierung mit Hilfe nebenläufiger, endlicher Zustandsautomaten, bieten die Möglichkeit zustandsdiskretes Verhalten von Systemen abzubilden. Hierbei sind jedoch noch keine Wartungszyklen und -vorgänge im Vorentwurf zu berücksichtigen [102].

Aufgrund der Möglichkeiten der bestehenden Verfahren zur Zielwertberechnung innerhalb einer Architekturbewertung komplexer Flugzeugsysteme und des zur Verfügung stehenden hybriden Systemmodells wird dieses für den weiteren Entwurfsprozess genutzt. Die Verwendung von Zuverlässigkeitsblockdiagrammen entspricht dabei dem intuitiven Nutzen eines Systemingenieurs zur Modellierung von Fehlerbedingungen und ist äquivalent zur Modellierung mittels Fehlerbäumen [126]. Während die nebenläufigen, endlichen Zustandsautomaten im Gegensatz zu Markov-Prozessen die Möglichkeit zur lokalen Definition von Zustandsübergängen auf Ereignisebene bieten, aus denen im Anschluss der vollständige Zustandsraum ermittelt wird [102]. In den nachfolgenden Abschnitten wird daher das bestehende hybride Systemmodell erläutert. Die Erweiterung zur Einbringung architektureller Freiheitsgrade und somit die Entwicklung eines Systemmodells variabler Struktur wird in Kapitel 4 betrachtet.

3.1.1 Zuverlässigkeitsblockdiagramme

Die erste Ebene des hybriden Systemmodells bilden Zuverlässigkeitsblockdiagramme (engl. *Reliability Block Diagrams*, *RBD*) der zu untersuchenden Fehlerbedingungen. Im Rahmen der Entwicklung von Flugzeugsystemen bildet das RBD-Modell die Systemstruktur hinsichtlich der untersuchten Systemfunktion bzw. -fehlfunktion ab. Die enthaltenen Ereignisse entsprechen somit den Fehlermodi der Komponenten und werden daher häufig mit diesen gleichgesetzt [95, 126]. Im Folgenden wird jedoch weiterhin die Trennung zwischen dem Ereignis und der Komponente verfolgt, da mehrfache Fehlermodi technischer Komponenten keine Reduktion der Komponenten auf einen Ereignisblock zulassen, sondern unterschiedlicher Blöcke bedürfen, um alle Fehlermodi abzudecken [10]. Je nach betrachteter Fehlerbedingung kann eine Komponente somit mit unterschiedlichen Fehlermodi und somit auch unterschiedlichen Fehlerraten berücksichtigt werden.

Neben den systeminternen Ereignissen beeinflussen in der Entwicklung von Flugzeugsystemen jedoch auch zahlreiche externe Ereignisse, beispielsweise der umweltbedingte Ausfall aller Triebwerke durch Vogelschlag oder vulkanische Asche, das Ausfallverhalten und die Ausfallwahrscheinlichkeit eines Systems. Externe Ereignisse lassen sich analog zu den Komponenten als Ereignisblock darstellen. Somit werden externe sowie interne Ereignisse identisch behandelt und zur Modellierung mit Hilfe einer binären Indikatorvariable $K_i \in \mathbf{K}$ beschrieben, die wie folgt definiert ist [126]:

$$K_i = 1 \qquad \text{für ein } \textbf{nicht} \text{ eingetretenes Ereignis,} \tag{3.1}$$

und

$$K_i = 0 \qquad \text{für ein eingetretenes Ereignis.} \tag{3.2}$$

Mit Hilfe der Indikatorvariable K_i kann der Erwartungswert $\mathcal{E}$ für das jeweilige externe oder interne Ereignis bestimmt werden,

$$\begin{aligned} \mathcal{E}_i[K_i] &= 0 \cdot P[K_i = 0] + 1 \cdot P[K_i = 1] \\ &= P[K_i = 1] \quad . \end{aligned} \tag{3.3}$$

Der Erwartungswert entspricht somit der Wahrscheinlichkeit, dass sich ein Ereignis im Zustand $K_i = 1$ befindet und stimmt mit der Überlebenswahrscheinlichkeit bzw. Zuverlässigkeit R_i überein. Im Fall von Fehlermodi technischer

Komponenten entspricht dieses der Überlebenswahrscheinlichkeit des Fehlermodus. Für Flugzeugsysteme kann von einem altersunabhängigen und zufälligen Ausfallverhalten gemäß des mittleren Abschnitts der Badewannenkurve in Abbildung 3.2 ausgegangen werden [126]. Somit kann die Exponentialverteilung der Komponentenzuverlässigkeit R_i mit Hilfe der konstanten Fehlerrate λ_i und der Zusammenhang mit der Ausfallwahrscheinlichkeit F_i beschrieben werden:

$$R_i = e^{-\lambda_i \cdot t} = 1 - F_i(t) \tag{3.4}$$

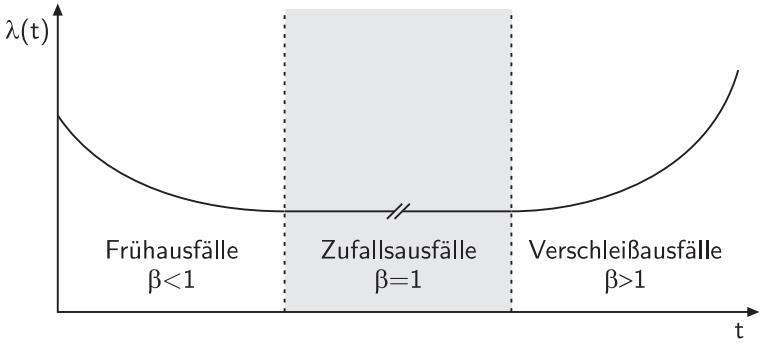


Abb. 3.2: Typischer Verlauf der Fehlerrate für mechatronische Systemkomponenten

Die Komponentenzuverlässigkeit ist dabei allgemein gültig für Sicherheits- und Zuverlässigkeitsanalysen auf Systemebene. Der angenommene zeitinvariante Bereich der Fehlerrate λ_i ist in Abbildung 3.2 hervorgehoben und wird durch die generelle Beschreibung der Weibull-Verteilung

$$\lambda_i = \alpha \cdot \beta \cdot t^{\beta-1} \qquad \text{für } t > 0 \tag{3.5}$$

mit $\beta = 1$ beschrieben, d.h. $\lambda_i = \alpha$. Für externe Ereignisse wird ebenfalls ein zufälliges Eintrittsverhalten mit konstanter Eintrittsrate vorausgesetzt [30].

Durch Gleichung 3.4 und der konstanten Ausfallrate lässt sich somit die Eintrittswahrscheinlichkeit beliebiger Fehlerbedingungen berechnen. Auf Systemebene lässt sich dieses mit Hilfe der Struktur der funktionslogischen Abhängig-

keiten der Ereignisse und der BOOLEschen Algebra erreichen. Die Berechnung der Ausfallwahrscheinlichkeit auf Systemebene unterliegt dabei den folgenden Monotoniebedingungen [77, 126]:

- Das System ist funktionsfähig, sofern kein Ereignis eingetreten ist.
- Das System ist ausgefallen, sofern alle Ereignisse eingetreten sind.
- Ein ausgefallenes System wird durch Eintreten weiterer Ereignisse nicht funktionsfähig.

Neben diesen Monotoniebedingungen ist vor allem die stochastische Unabhängigkeit der Ereignisse bei der Modellierung zu beachten [10, 126]. Sofern die Unabhängigkeit der Ereignisse nicht gegeben ist, muss zum einen das System entsprechend modelliert werden, siehe Abschnitt 3.1.2, und im Rahmen von Flugzeug-Systementwicklungen ist eine Untersuchung der gemeinsamen Ursachen für katastrophale Ereignisse (engl. *Common Cause Analysis*) notwendig, vgl. Abschnitt 2.1.

Analog zu der Indikatorvariable eines Ereignisses wird das Systemverhalten mit Hilfe der Strukturfunktion $\Phi(\mathbf{K})$ beschrieben, die den kausalen Zusammenhang zwischen Komponenten- und Systemverhalten beschreibt und wie folgt definiert ist [126]:

$$\Phi(\mathbf{K}) = 1 \qquad \text{für ein funktionsfähiges System,} \tag{3.6}$$

und

$$\Phi(\mathbf{K}) = 0 \qquad \text{für ein } \textbf{nicht} \text{ funktionsfähiges System.} \tag{3.7}$$

Für die Erstellung der Strukturfunktion eigenen sich Minimalpfade, die aus der RBD-Struktur ausgelesen werden können [77, 126].

Definition Minimalpfad
Es sei $\mathbf{K} = \{K_1, \ldots, K_n\}$ die Menge der Ereignisse und p von $\mathbf{K}$ eine Teilmenge der funktionsfähigen Komponenten und nicht eingetretenen Ereignisse, so dass gilt: $\Phi(\mathbf{K}) = 1$. In diesem Fall heißt p von $\mathbf{K}$ Pfad des Systems. Ein Pfad p heißt Minimalpfad *MP*, wenn er keine anderen Pfade als echte Teilmenge enthält.

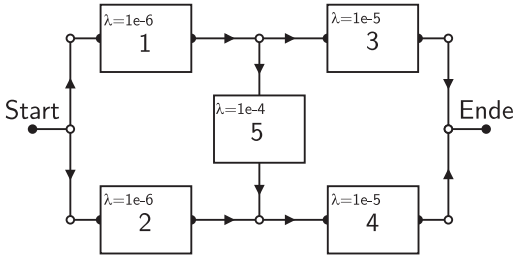


Abb. 3.3: Beispiel eines Zuverlässigkeitsblockdiagrammes einer unidirektionalen Brückenstruktur

Abbildung 3.3 zeigt eine unidirektionale Brückenstruktur eines exemplarischen RBD-Modells. Das dargestellte System ist funktionsfähig, wenn eine der folgenden Ereignisfolgen nicht eingetreten ist: K_1K_3, K_2K_4, $K_1K_4K_5$.

Mit Hilfe der Elementarzustände E_i des Systemmodells kann die Menge der Minimalpfade **MP** ausgelesen werden. Die Tabelle 3.1 zeigt die binären Elementarzustände der unidirektionalen Brückenstruktur und die Überschneidungen der Minimalpfade. Durch eine anschließende Orthogonalisierung der Minimalpfade werden sich gegenseitig ausschließende Terme erreicht, die im Anschluss für die Erstellung der Strukturfunktion disjunktiv verknüpft werden.

In der aktuellen Version des Werkzeugs SyRelAn erfolgt die Orthogonalisierung der Minimalpfade mit Hilfe des Heidtmann-Algorithmus *KDH88* [75, 102]. Dabei erzeugt der Algorithmus *KDH88* im Vergleich zum Abraham Algorithmus durch die gemeinsame Invertierung von Indikatorvariablen eine geringere Anzahl zusätzlicher orthogonaler Minimalpfade [43, 126]. Da durch die disjunktive Verknüpfung aller orthogonalen Minimalpfade einer Fehlerbedingung die Zielfunktion für die spätere Optimierung generiert wird, wirkt sich die geringere Pfadanzahl bei jeder Zielwertberechnung auf die Rechenzeit aus.

Die orthogonalen Minimalpfade für die unidirektionale Brückenstruktur aus Abbildung 3.3 lauten in diesem Fall:

$$MP_1 = K_1 \wedge K_3, \tag{3.8}$$

$$MP_2 = K_2 \wedge K_4 \wedge (\overline{K_1 K_3}) \text{ und} \tag{3.9}$$

$$MP_3 = K_1 \wedge K_5 \wedge K_4 \wedge \overline{K_2} \wedge \overline{K_3}\ . \tag{3.10}$$

Tab. 3.1: Elementarzustände der unidirektionalen Brückenstruktur

Elementarzustand	K_1	K_2	K_3	K_4	K_5	$\Phi(\mathbf{K})$	MP_i
E_1	1	1	1	1	1	1	MP_1, MP_2, MP_3
E_2	1	1	1	1	0	1	MP_1, MP_2
E_3	1	1	1	0	1	1	MP_1
E_4	1	1	1	0	0	1	MP_1
E_5	1	1	0	1	1	1	MP_2
E_6	1	1	0	1	0	1	MP_2
E_7	1	1	0	0	1	0	
E_8	1	1	0	0	0	0	
E_9	1	0	1	1	1	1	MP_1, MP_3
E_{10}	1	0	1	1	0	1	MP_1
E_{11}	1	0	1	0	1	1	MP_1
E_{12}	1	0	1	0	0	1	MP_1
E_{13}	1	0	0	1	1	1	MP_3
E_{14}	1	0	0	1	0	0	
E_{15}	1	0	0	0	1	0	
E_{16}	1	0	0	0	0	0	
E_{17}	0	1	1	1	1	1	MP_2
E_{18}	0	1	1	1	0	1	MP_2
E_{19}	0	1	1	0	1	0	
E_{20}	0	1	1	0	0	0	
E_{21}	0	1	0	1	1	1	MP_2
E_{22}	0	1	0	1	0	1	MP_2
E_{23}	0	1	0	0	1	0	
E_{24}	0	1	0	0	0	0	
E_{25}	0	0	1	1	1	0	
E_{26}	0	0	1	1	0	0	
E_{27}	0	0	1	0	1	0	
E_{28}	0	0	1	0	0	0	
E_{29}	0	0	0	1	1	0	
E_{30}	0	0	0	1	0	0	
E_{31}	0	0	0	0	1	0	
E_{32}	0	0	0	0	0	0	

Die BOOLEsche Funktion aus der vorherigen Gleichung lässt sich somit in die real-algebraische Systemstrukturfunktion für die Systemzuverlässigkeit $R_S(t)$ überführen [126]:

$$\begin{aligned} R_S(t) &= R_1(t)R_3(t) + R_2(t)R_4(t)\,(1 - R_1(t)R_3(t)) \\ &\quad + R_1(t)R_4(t)R_5(t)\,(1 - R_2(t))\,(1 - R_3(t)) \ . \end{aligned} \tag{3.11}$$

3.1.2 Zustandsdiskretes Systemmodell

Mit Hilfe des BOOLEschen RBD-Modells aus dem vorherigen Abschnitt können große, komplexe Modelle rechnergestützt übersichtlich erstellt und analysiert werden. Das binäre Ausfall- und Betriebsmuster stellt jedoch in fehlertoleranten Systemen eine starke, bei der Untersuchung unterschiedlicher Redundanzstrategien unzulässige, Vereinfachung dar. Zudem ist die Anforderung unabhängiger Ereignisse bei regenerativen Flugzeugsystemen nicht immer gegeben. In vielen Fällen führt hier der Ausfall einer aktiven Komponente zur Erhöhung der Last der redundanten Komponente [77, 93, 102]. Beispiele hierfür sind duplex angesteuerte Flugsteuerungsflächen, elektrische Energieversorgungssysteme in *Split-* oder *Parallel-Bus*-Bauweise oder hydraulische Energieversorgungssysteme mit hydromechanischen Leistungstransfereinheiten.

Aus den genannten Gründen ist es in einigen Fällen sinnvoll, das binäre Ausfallmuster um ein zustandsdiskretes Verhalten zu erweitern. Das gängige Modell hierfür ist die Modellierung durch das MARKOVsche Modell [77, 102]. Dabei wird jeder Systemzustand durch die einzelnen Komponentenzustände beschrieben und die stochastischen Übergänge zwischen diesen Zuständen anhand der Wahrscheinlichkeit einer Zustandsänderung der betrachteten Komponenten. Je nach Verfügbarkeit der Parameter und Ziel der Modellbildung ist eine Modellierung unterschiedlicher Abhängigkeiten zwischen den Komponenten und die Berücksichtigung von Reparaturinformationen möglich [77]. Anhand des aufgestellten Zustandsraums und der Transitionen wird ein Differentialgleichungssystem erster Ordnung aufgestellt, welches anschließend mit entsprechenden Verfahren gelöst werden kann [102]. REHAGE hat im Gegensatz zur MARKOV-Kette für die Abbildung eines zustandsdiskreten Verhaltens eine Methode entwickelt, die anstatt des impliziten Systemverhaltens des MARKOVschen Modells, explizit die Komponentenzustände analysiert und aus diesen das zustandsdiskrete, dynamische Systemverhalten ableitet. Die Vorteile des Modells von REHA-

GE im Vergleich zu den MARKOV-Ketten liegen in der Verwendung rekursiver Terme anstatt der Lösung eines Differentialgleichungssystems und der lokalen Modellierung der Komponentenabhängigkeiten anstatt des globalen Systemverhaltens [102]. Im Folgenden wird das zustandsdiskrete Systemmodell unter Verwendung nebenläufiger, endlicher Zustandsautomaten (engl. *Concurrent Finite State Machines, CFSM*) vorgestellt, das in dem nachfolgenden Abschnitt zum hybriden Modell zur Sicherheits- und Zuverlässigkeitsanalyse erweitert wird.

Das zustandsdiskrete Systemmodell wird hierfür formell durch den 6-Tupel $\mathbf{CFSM} =< \mathbf{Z}, \mathbf{Z_0}, \mathcal{F}, \mathcal{Y}, f, g >$ beschrieben, die einzelnen Bestandteile sind nachfolgend definiert [102]:

- $\mathbf{Z}$, endliche, nichtleere Menge interner Ereigniszustände,
- $\mathbf{Z_0} \subset \mathbf{Z}$, Menge der Initialzustände,
- $\mathcal{F}$, Eingabealphabet,
- $\mathcal{Y}$, Ausgabealphabet,
- $\gamma : \mathcal{F} \times Z \rightarrow Z$, Übergangsfunktion,
- $\delta : Z \rightarrow \mathcal{Y}$, Ausgabefunktion.

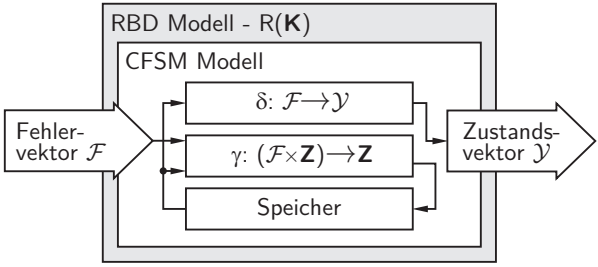


Abb. 3.4: Einbettung der nebenläufigen, endlichen Zustandsautomaten in die Zuverlässigkeitsblockdiagramme

Die Ausgänge dieses Zustandsautomaten hängen somit von den internen Zuständen $\mathbf{Z}$ ab und sind unabhängig von den Eingangsgrößen $\mathcal{F}$. Dabei verfügt jedes Ereignis mindestens über die Zustände *„aktiv“* und *„isoliert“*, die dem binären Muster *funktionsfähig* und *ausgefallen* gemäß der Gleichungen 3.1 und 3.2 entsprechen [102]. In Abbildung 3.4 ist der Ablauf für die Zustandsänderun-

gen aufgrund eines Fehlervektors dargestellt. Dabei wird die erste Transition global immer durch eine externe Fehlerinjektion ausgelöst und durchläuft danach sämtliche Zustandsautomaten, bis wieder ein stabiler Zustand erreicht ist, d.h. es ist keine weitere Transition schaltbar.

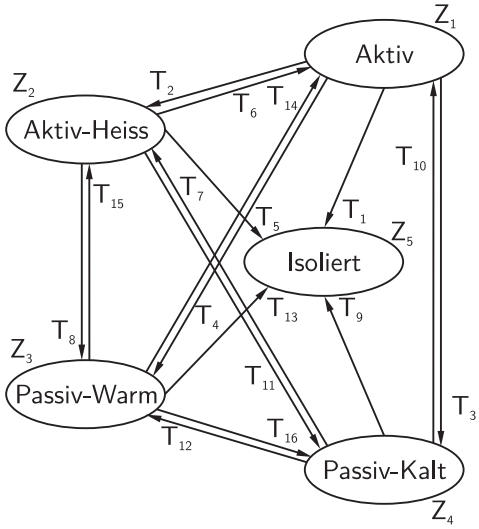


Abb. 3.5: Mögliche Zustände und Transitionen eines nebenläufigen, endlichen Zustandsautomaten

Die folgenden Ereigniszustände sind für die Menge der Anfangszustände $\mathbf{Z_0}$ und der aktuellen Zustände $\hat{\mathbf{Z}}$ möglich. Dabei ist zu berücksichtigen, dass für zustandsdiskrete Modelle mit mehr als den beiden Minimalzuständen nicht mehr von beliebigen Ereignissen analog zum Zuverlässigkeitsblockdiagramm auszugehen ist. Es handelt sich vielmehr um konkrete Komponenten mit einem zustandsdiskreten Verhalten [102]:

Definition aktiv: die Komponente a ist mit Beginn der Flugmission der vollen Belastung ausgesetzt. Die Fehlerrate wird durch λ_a beschrieben.

Definition aktiv–heiss: mit Beginn der Flugmission ist die redundante Komponente h der gleichen Belastung wie der Arbeitskomponente a ausgesetzt. Für die Fehlerrate der Komponente gilt: $\lambda_h = \lambda_a$.

Definiton passiv–warm: die redundante Komponente w ist einer geringe-

ren Belastung ausgesetzt, sofern die Arbeitskomponente a funktionsfähig ist oder bis die Komponente selbst ausfällt. Die Fehlerrate liegt in den Intervall $0 < \lambda_w < \lambda_a$.

Definition passiv–kalt: sofern kein erster Fehler im System vorliegt, ist die redundante Komponente c keiner Belastung ausgesetzt. Daraus folgt für die Fehlerrate: $\lambda_c = 0$.

Definition isoliert: Ausgefallener oder logisch isolierter terminierender Zustand einer Komponente. Die Zustandsendung ist „i“.

Die Transitionen T_i zwischen den definierten Zuständen sind dabei in Abbildung 3.5 definiert. Dabei wird jede Transition mit Hilfe einer speziellen Syntax angesprochen, die die weiteren verwendeten Ereignisse im Systemmodell und deren Zustände adressiert, so dass eine Zustandstransition dann ausgeführt wird, wenn die Transitionsbedingung eingetreten ist [102].

3.1.3 Hybrides Systemmodell

Die Kopplung der beiden zuvor erläuterten Modellierungsebenen bildet das hybride Systemmodell von Vahl und Rehage. In der ersten Ebene wird mit Hilfe der Zuverlässigkeitsblockdiagramme die Logik einer Fehlerbedingung abgebildet. Auf der zweiten Ebene bieten die nebenläufigen, endlichen Zustandsautomaten die Möglichkeit zur Modellierung logischer Abhängigkeiten zwischen unterschiedlichen Ereignissen und Komponenten. Nachfolgend wird der Aufbau des hybriden Systemmodells und die Zustandsraumermittlung erläutert. Das vorhandene Systemmodell wird anschließend in Kapitel 4 zum mehrfachredundanten Systemmodell erweitert und bietet somit die Möglichkeit zur Einbringung architektureller Freiheitsgrade.

Abbildung 3.6 verdeutlich den Ansatz des hybriden Systemmodells und die Interaktion der Modellierungsebenen. Das Modell erlaubt es somit nicht nur unterschiedliche Komponentenzustände, sondern auch zustandsspezifische Fehlerraten zu berücksichtigen und somit die zu Beginn von Abschnitt 3.1.1 genannten Defizite des reinen Booleschen Modells zu beseitigen [102].

Mit Hilfe eines Tiefe-Zuerst-Algorithmus kann das hybride Systemmodell genutzt werden, um einen vollständigen Zustandsraum des Systemmodells aufzustellen. Hierfür werden systematisch Fehler in das System injiziert, so dass die Zustandstransitionen des zustandsdiskreten Modells angesprochen werden.

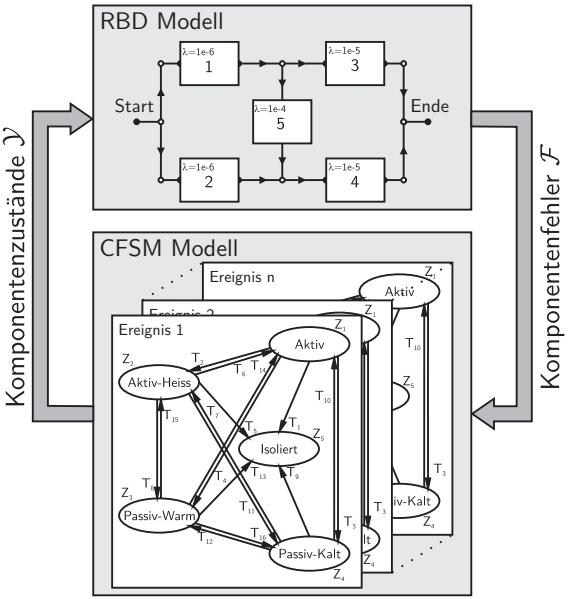


Abb. 3.6: Verbindung der beiden Modellierungsebenen des hybriden Systemmodells

Nachdem wieder ein stabiler Zustand erreicht wurde, kann mit Hilfe der Minimalpfade des Zuverlässigkeitsblockdiagramms und der Menge der funktionsfähigen Komponenten überprüft werden, ob das System in dem aktuellen Zustand funktionsfähig ist. Sofern es nicht funktionsfähig ist, wird mit der Fehlerinjektion auf der gleichen Ebene fortgefahren. Ist das System jedoch auch im degradierten Zustand funktionsfähig, wird die Wahrscheinlichkeit zu dem aktuellen Zustand berechnet und die Systemdegradation weitergeführt.

Der ermittelte Zustandsraum entspricht hierbei dem Zustandsraum von Markov-Ketten [102]. Die weitere Nutzung bedingt jedoch keine Lösung von linearen Differentialgleichungen zur Lösung der Zustandstransitionen, stattdessen kann eine rekursive Formel genutzt werden, die auf Faltungsintegralen basiert. Gleichung 3.13 verdeutlicht die Nutzung der rekursiven Formel zur Bestimmung der Zustandswahrscheinlichkeiten. Dabei werden die Vorgängerargumente VA und Vorgängerterme VT in den unterschiedlichen Degradationsstufen wieder verwendet. Neben den rekursiven Anteilen werden zudem

Zustandsübergangswahrscheinlichkeiten berücksichtigt, die den Zustandswechsel von abhängigen Komponenten probabilistisch charakterisieren, die entsprechenden Terme sind in Tabelle 3.2 aufgeführt [102].

Tab. 3.2: Zustandsübergangswahrscheinlichkeiten des hybriden Systemmodells

Zustandsübergang der Komponente	**PST**
aktiv oder *aktiv-heiss* zu *isoliert*	(R_i^{a})
passiv-warm zu *isoliert*	(R_i^{w})
aktiv oder *aktiv-heiss* zu *passiv-warm*	$\left(\frac{R_i^{\mathrm{a}}}{R_i^{\mathrm{w}}}\right)$
aktiv oder *aktiv-heiss* zu *passiv-kalt*	(R_i^{a})
passiv-warm zu *aktiv* oder *aktiv-heiss*	$\left(\frac{R_i^{\mathrm{w}}}{R_i^{\mathrm{a}}}\right)$
passiv-warm zu *passiv-kalt*	(R_i^{w})
passiv-kalt zu *aktiv* oder *aktiv-heiss*	$\left(\frac{1}{R_i^{\mathrm{a}}}\right)$
passiv-kalt zu *passiv-warm*	$\left(\frac{1}{R_i^{\mathrm{w}}}\right)$

Die folgenden Formeln zur Berechnung der Zustandswahrscheinlichkeit basieren dabei auf einem hybriden Systemmodell mit aktiven und passiven Komponenten und lassen sich für die ausschließliche Verwendung aktiver Komponenten vereinfachen [102]. Die resultierenden Ergebnisse würden in diesem Fall ohne weitere definierte Transitionen den Ergebnissen eines entsprechenden Zuverlässigkeitblockdiagramms entsprechen, jedoch mit der Möglichkeit redundanzübergreifende Logiken mit Hilfe der Zustandsautomaten abbilden zu können.

$$P_d^{\mathrm{p}}[\phi(\mathbf{K})](t) = \begin{cases} \underbrace{\frac{\lambda_d^{\{\mathrm{a,w}\}}}{\lambda[VA_1^1]}}_{VT_1^1} \cdot (1 - \underbrace{R_d^{\{\mathrm{a,w}\}} \cdot PST_1}_{VA_1^1}) \cdot \\ \cdot E[\phi(\mathbf{KA}_1)] \cdot E[\phi(\mathbf{KW}_1)] \end{cases} \quad \forall \quad d = 1 \quad (3.12)$$

$$
P_d^{\mathrm{p}}[\phi(\mathbf{K})](t) = \begin{cases} \Bigg[\sum_{j=1}^{2^{(d-2)}} \underbrace{VT_j^{(d-1)} \cdot \frac{\lambda_d^{\{\mathrm{a,w}\}}}{\lambda[VA_j^d]}}_{VT_j^d} \cdot \\ \cdot (1 - \underbrace{R_d^{\{\mathrm{a,w}\}} \cdot PST_d}_{VA_j^d}) + \\ + \sum_{j=1}^{2^{d-2}} \underbrace{VT_j^{(d-1)} \cdot \frac{-\lambda_d^{\{\mathrm{a,w}\}}}{\lambda[VA_{j+2^{(d-2)}}^d]}}_{VT_{j+2^{(d-2)}}^d} \cdot \\ \cdot (1 - \underbrace{R_d^{\{\mathrm{a,w}\}} \cdot PST_d \cdot VA_j^{(d-1)}}_{VA_{j+2^{(d-2)}}^d}) \Bigg] \cdot \\ \cdot E[\phi(\mathbf{KA}_d)] \cdot E[\phi(\mathbf{KW}_d)] \end{cases} \quad \forall \quad d \geq 2 \tag{3.13}
$$

$$
\text{mit} \quad \mathbf{KA}_d, \mathbf{KW}_d \subseteq \mathbf{K} \quad \text{und} \tag{3.14}
$$

$$
\phi(\mathbf{KA}_d) = \bigwedge_{g}^{g_{\max}} K_g \quad , \tag{3.15}
$$

g ist der Zähler der *aktiven* und *aktiv-heißen* Komponenten nach dem Funktionsverlust der Komponente i zu der Degradationsstufe d, die Degradationsstufe beschreibt dabei die Anzahl der ausgefallenen Komponenten,

$$
\phi(\mathbf{KW}_d) = \bigwedge_{h}^{h_{\max}} K_h \quad , \tag{3.16}
$$

h ist der Zähler der *passiv-warmen* Komponenten nach dem Funktionsverlust der Komponente i zu der Degradationsstufe d.

Durch Summation aller Zustandsgleichungen gemäß Gleichung 3.13, für die gilt $\phi(\mathbf{K}) = 1$, kann die quantitative Sicherheit und Zuverlässigkeit eines Systemmodells berechnet werden [102]. Für die spätere Optimierung werden die Zu-

standsgleichungen in der folgenden Zustandsmatrix $\mathbf{P}(t)$ abgelegt, dabei beschreibt die jeweilige Spalte die Degradationsstufe des Systems [96]:

$$\mathbf{P}(t) = \left[\begin{array}{c|cccc} P_0 & P_{x_1} & P_{x_1+1} & \cdots & P_{x_1+n_{max}-1} \\ 0 & 0 & 0 & \cdots & P_{x_1+n_{max}-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & P_{x_2} & P_{x_2+1} & \cdots & P_{x_2+n_{max}-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{array}\right] . \tag{3.17}$$

Mit Hilfe des vorgestellten hybriden Systemmodells lassen sich rekonfigurierbare Flugzeugsysteme derart abbilden, dass die Fehlermodi der verwendeten Komponenten mit einer zustandsdiskreten Fehlerrate beschrieben werden können. Die Systemmodellierung ist somit genauer im Vergleich zur Modellierung mit Zuverlässigkeitsblockdiagrammen und bildet vor allem bei der Untersuchung unterschiedlicher Redundanzkonzepte das reale Ausfall- und Rekonfigurationsverhalten besser ab [96]. Die relevanten Redundanzkonzepte für Flugzeugsysteme sind *aktiv-aktiv* und *aktiv-passiv* Redundanzen, wie sie beispielsweise für elektrische Generatoren verwendet werden.

Basierend auf der unidirektionalen Brückenschaltung aus Abbildung 3.3 sind nachfolgend die Zustandsgleichungen $P_d(t)$ dargestellt. Die dargestellten Terme gelten für den Fall ausschließlich aktiver Redundanzen und somit ist die Modellierung äquivalent zur Nutzung von Zuverlässigkeitsblockdiagrammen. Die nachfolgenden Gleichungen verdeutlichen die Komplexität der Systemanalyse mittels des hybriden Systemmodells im Vergleich zu Gleichung 3.11, die das identische Ergebnis liefert. Die Verwendung des hybriden Systemmodells für eine Untersuchung eines variablen Strukturmodells bleibt somit auf kleine Systeme beschränkt, die sich jedoch durch ein stark zustandsdiskretes Verhalten auszeichnen [96]. Die Größe des Zustandsraumes wird hierbei nicht nur durch die Anzahl der Ereignisse im Systementwurf beschränkt, sondern auch durch die Anzahl der Abhängigkeiten, beschrieben durch die Transitionslogiken. Hierbei zeigt sich, dass eine möglichst vollständige Modellierung der Transitionen einen größeren Zustandsraum ermöglicht, da aufgrund der Abhängigkeiten zwischen den Ereignissen nach wenigen Degradationen bereits terminierende Zustände erreicht werden und eine weitere Verzweigung somit nicht verfolgt wird.

$$
\begin{aligned}
P_0 &= R_1R_2R_3R_4R_5 \\
P_1 &= \frac{\lambda_1}{\lambda_1+\lambda_3+\lambda_5}(1-R_1R_3R_5)\cdot R_2R_4 \\
P_2 &= \frac{\lambda_2}{\lambda_2}(1-R_2)\cdot R_1R_3R_4R_5 \\
P_{2,3} &= \left(\frac{\lambda_2}{\lambda_2}\cdot\frac{\lambda_3}{\lambda_3}(1-R_3)+\frac{\lambda_2}{\lambda_2}\cdot\frac{-\lambda_3}{\lambda_2+\lambda_3}\cdot(1-R_2R_3)\right)\cdot\ldots \\
&\quad \ldots\cdot R_1R_4R_5 \\
P_{2,4} &= \left(\frac{\lambda_2}{\lambda_2}\cdot\frac{\lambda_4}{\lambda_4+\lambda_5}(1-R_4R_5)+\frac{\lambda_2}{\lambda_2}\cdot\frac{-\lambda_4}{\lambda_2+\lambda_4+\lambda_5}(1-R_2R_4R_5)\right)\cdot\ldots \\
&\quad \ldots\cdot R_1R_3 \\
P_{2,5} &= \left(\frac{\lambda_2}{\lambda_2}\cdot\frac{\lambda_5}{\lambda_5+\lambda_4}(1-R_5R_4)+\frac{\lambda_2}{\lambda_2}\cdot\frac{-\lambda_5}{\lambda_2+\lambda_5+\lambda_4}(1-R_2R_5R_4)\right)\cdot\ldots \\
&\quad \ldots\cdot R_1R_3 \\
P_3 &= \frac{\lambda_3}{\lambda_3}(1-R_3)\cdot R_1R_2R_4R_5 \\
P_{3,1} &= \left(\frac{\lambda_3}{\lambda_3}\cdot\frac{\lambda_1}{\lambda_1+\lambda_5}(1-R_1R_5)+\frac{\lambda_3}{\lambda_3}\cdot\frac{-\lambda_1}{\lambda_3+\lambda_1+\lambda_5}(1-R_3R_1R_5)\right)\cdot\ldots \\
&\quad \ldots\cdot R_2R_4 \\
P_{3,2} &= \left(\frac{\lambda_3}{\lambda_3}\cdot\frac{\lambda_2}{\lambda_2}(1-R_2)+\frac{\lambda_3}{\lambda_3}\cdot\frac{-\lambda_2}{\lambda_3+\lambda_2}(1-R_3R_2)\right)\cdot\ldots \\
&\quad \ldots\cdot R_1R_4R_5 \\
P_{3,5} &= \left(\frac{\lambda_3}{\lambda_3}\cdot\frac{\lambda_5}{\lambda_5+\lambda_1}(1-R_5R_1)+\frac{\lambda_3}{\lambda_3}\cdot\frac{-\lambda_5}{\lambda_3+\lambda_5+\lambda_1}(1-R_3R_5R_1)\right)\cdot\ldots \\
&\quad \ldots\cdot R_2R_4 \\
P_4 &= \frac{\lambda_4}{\lambda_4+\lambda_2+\lambda_5}(1-R_4R_2R_5)\cdot R_1R_3 \\
P_5 &= \frac{\lambda_5}{\lambda_5}(1-R_5)\cdot R_1R_2R_3R_4 \\
P_{5,1} &= \left(\frac{\lambda_5}{\lambda_5}\cdot\frac{\lambda_1}{\lambda_1+\lambda_3}(1-R_1R_3)+\frac{\lambda_5}{\lambda_5}\cdot\frac{-\lambda_1}{\lambda_5+\lambda_1+\lambda_3}(1-R_5R_1R_3)\right)\cdot\ldots \\
&\quad \ldots\cdot R_2R_4
\end{aligned}
$$

$$
\begin{aligned}
P_{5,2} &= \left(\frac{\lambda_5}{\lambda_5}\cdot\frac{\lambda_2}{\lambda_2+\lambda_4}(1-R_2R_4)+\frac{\lambda_5}{\lambda_5}\cdot\frac{-\lambda_2}{\lambda_5+\lambda_2+\lambda_4}(1-R_5R_2R_4)\right)\cdot\ldots \\
&\quad \ldots\cdot R_1R_3 \\
P_{5,3} &= \left(\frac{\lambda_5}{\lambda_5}\cdot\frac{\lambda_3}{\lambda_3+\lambda_1}(1-R_3R_1)+\frac{\lambda_5}{\lambda_5}\cdot\frac{-\lambda_3}{\lambda_5+\lambda_3+\lambda_1}(1-R_5R_3R_1)\right)\cdot\ldots \\
&\quad \ldots\cdot R_2R_4 \\
P_{5,4} &= \left(\frac{\lambda_5}{\lambda_5}\cdot\frac{\lambda_4}{\lambda_4+\lambda_2}(1-R_4R_2)+\frac{\lambda_5}{\lambda_5}\cdot\frac{-\lambda_4}{\lambda_5+\lambda_4+\lambda_2}(1-R_5R_4R_2)\right)\cdot\ldots \\
&\quad \ldots\cdot R_1R_3
\end{aligned}
\tag{3.18}
$$

Somit ergibt sich anhand der ermittelten Zustandsgleichungen die folgende Zustandsmatrix für das Beispielsystem. Anhand der Matrix können in der nachfolgenden Variation der Systemstruktur unterschiedliche Operationen durchgeführt werden, die in Kapitel 4 vorgestellt werden.

$$
\mathbf{P}(t)=\left[\begin{array}{c|cccc} P_0 & P_1 & 0 & 0 & 0 & 0 \\ 0 & P_2 & P_{2,3} & P_{2,4} & P_{2,5} & 0 \\ 0 & P_3 & P_{3,1} & P_{3,2} & P_{3,5} & 0 \\ 0 & P_4 & 0 & 0 & 0 & 0 \\ 0 & P_5 & P_{5,1} & P_{5,2} & P_{5,3} & P_{5,4} \end{array}\right] . \tag{3.19}
$$

3.2 Methoden der Redundanzallokation

Nachdem in den vorherigen Abschnitten das hybride Systemmodell zur Sicherheits- und Zuverlässigkeitsanalyse vorgestellt wurden, folgt in diesem Abschnitt eine Übersicht über den Stand der Technik zu Methoden der Sicherheits- und Zuverlässigkeitsoptimierung. Die hierfür verfügbaren Methoden und Verfahren lassen sich in die Problemstellungen der Redundanz- und Zuverlässigkeitsallokation unterteilen. Abbildung 3.7 verdeutlicht die Unterscheidung der beiden Ansätze und die weitere Kategorisierung.

Bei den Verfahren zur Zuverlässigkeitsallokation (engl. *Reliability Allocation Problem*) handelt es sich um parametrische Optimierungen, die für eine feste Architektur optimale Zuverlässigkeitswerte der verwendeten Komponenten suchen. Dabei stehen die Zuverlässigkeitswerte im Gegensatz zu weiteren Eigenschaften der Komponenten, wie den Kosten k_k und der Masse k_m. Das Ziel

der Optimierung ist somit eine Balance der Zielwerte zwischen den verwendeten Komponenten. Die zugewiesenen Werte können nachfolgend zur Spezifikation der Komponenten genutzt werden. Die Eigenschaften der Komponenten werden in diesem Fall häufig als Kennlinie, z.B. $R = f(k_k, k_m)$ hinterlegt [54]. Eine Lösung der Zuverlässigkeitsallokation ist in Abhängigkeit der Klassifizierung der Zielfunktionen mit Hilfe gängiger Optimierungsverfahren möglich [122].

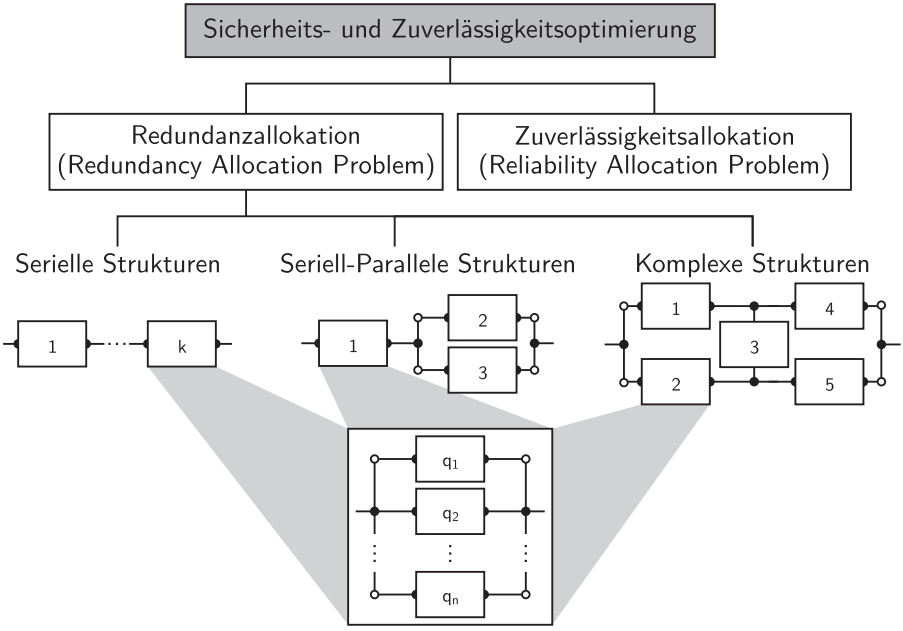


Abb. 3.7: Unterscheidung der prinzipiellen Verfahren zur Sicherheits- und Zuverlässigkeitsoptimierung

Im Gegensatz zur Zuverlässigkeitsallokation suchen die Verfahren zur Redundanzallokation (engl. *Redundancy Allocation Problem*)[1] nach optimalen Systemarchitekturen, wobei die Eigenschaften der Komponenten nicht variiert werden oder nur eine sehr eingeschränkte, diskrete Variation zugelassen wird. Die

[1] Aufgrund der englischen Bezeichnungen der beiden Optimierungskategorien werden beide Verfahren mit *RAP* abgekürzt, was häufig zu Missverständissen führt, da es sich hierbei um stark unterschiedliche Optimierungsprobleme handelt. Aus diesem Grund werden im Folgenden weiterhin nicht die gängigen englischen Bezeichnungen genutzt.

vorliegende Arbeit ist der zweiten Gruppe der Optimierungsverfahren zuzuordnen, daher werden die wesentlichen bestehenden Verfahren der Redundanzallokation im Folgenden näher betrachtet. Basierend auf der Terminologie der Zuverlässigkeitsblockdiagramme ergeben sich drei Untergruppen für die Optimierung: serielle Strukturen, seriell-parallele Strukturen und komplexe Strukturen. Dabei wird, wie in Abbildung 3.7 dargestellt, für die einzelnen Subsysteme die optimale Anzahl an redundanten Komponenten gesucht [89].

3.2.1 Übersicht bestehender Methoden

Im Folgenden werden die wesentlichen bestehenden Methoden der Redundanzallokation vorgestellt. Diese unterteilen sich in Methoden zur Optimierung serieller, seriell-paralleler und komplexer Strukturen.

Serielle Strukturen
Seit Beginn der 1960er Jahre werden unterschiedliche Verfahren und Algorithmen zur Redundanzallokation serieller Strukturen untersucht [89]. Die Systemgleichungen lassen sich hierbei allgemeingültig aufstellen und ergeben sich zu: $R_{SP}(t) = \prod_{i=1}^{n} \left(1 - \prod_{j=1}^{k} (1 - R_{i,j}(t))\right)$. Während zu Beginn der Methodenentwicklungen eine maximale Systemsicherheit bei der Beschränkung einer weiteren Kostenfunktion bzw. eine minimale Masse bei Anforderungen an die Sicherheit verfolgt wurden, folgten später mehrkriterielle Optimierungsprobleme. Misra und Tillman haben mit unterschiedlichen Ansätzen unter Verwendung von *Integer Programming* und *Mixed Integer Nonlinear Programming* Verfahren, einzelne mehrkriterielle, serielle Probleme gelöst, wobei die Zielwertfunktionen entsprechend der behandelten Problemklasse approximiert wurden [89].

Seriell-parallele Strukturen
Als Erweiterung der seriellen Struktur wurden ab den 1970er Jahren seriell-parallele Strukturen untersucht. Auch wenn sich in diesem Fall keine allgemeingültige Systemstrukturfunktion aufstellen lässt, können seriell-parallele Strukturen ohne eine mehrfache Verwendung von Komponenten immer durch einfache Grundoperationen zusammengefasst werden [10, 77]. Erstmalig haben Burton et al. diese Redundanzallokation für einen Einzelfall gelöst [89]. Das allgemeine Redundanzallokationsproblem gilt dabei als NP-vollständig [127]. Im Vergleich zu weiteren Entscheidungsproblemen wie dem *Travelling Salesman Problem* hat sich somit herausgestellt, dass es vermutlich keinen Algo-

rithmus gibt, der in Lage ist das Problem allgemein und vollständig und zudem für beliebige Systeme zeiteffizient zu lösen [36]. Der Nachweis der NP-Vollständigkeit hat als Konsequenz die Forschung auf dem Gebiet der Redundanzallokation maßgeblich beeinflusst, da nicht mehr die exakte Lösung für beliebige Probleme gesucht wurde, sondern der Fokus auf angepasste Verfahren für Einzelfälle gelenkt wurde. Während die ersten Arbeiten zur Untersuchung seriell-paralleler Systeme deterministische Optimierungsverfahren und Linearisierungen der Zielwertfunktionen genutzt haben, folgten daher aufgrund des Nachweises der NP-Vollständigkeit auch vermehrt heuristische Lösungsverfahren [89, 123]. Im Folgenden werden einige Ansätze näher betrachtet, die entweder gute Ansätze hinsichtlich der Optimierung seriell-paralleler Strukturen bieten oder Einzelfälle von komplexen Systemstrukturen untersuchen.

Die meisten Arbeiten der Redundanzallokation konzentrieren sich auf seriell-parallele Strukturen [89]. Für die Redundanzoptimierung dieser Systeme haben erstmals Busacca et al. einen Genetischen Algorithmus genutzt [12]. Neben der Zuverlässigkeit des Systems haben sie die Systemkosten, bestehend aus Installations- und Reparaturkosten berücksichtigt. Das Ergebnis des Optimierungsprozesses ist die Menge der nicht-dominierten Lösungen, aus derer der Anwender eine endgültige Lösung auswählen muss. Dabei untersuchen Busacca et al. auch Aspekte der Modellvalidierung und nutzen hier ein Konzept mittels einer einkriteriellen Optimierung. Hierbei werden nur die Systemkosten betrachtet, die sich im Gegensatz zu der Zuverlässigkeit aus unterschiedlichen Komponentenfaktoren ergeben.

Einen Ansatz zur Optimierung elektrischer Energieversorgungssysteme betrachten Levitin et al. [65]. Den einzelnen Komponenten werden Leistungsniveaus zugeordnet, so dass die Wahrscheinlichkeit der Leistungsversorgung und Leistungsdegradation gegenüber der benötigten elektrischen Leistung betrachtet wird. Die Komponenten werden dabei durch unterschiedliche Fehlermodi abgebildet, wobei jeder Fehlerzustand durch die Eintrittswahrscheinlichkeit und die verbleibende Leistung charakterisiert wird. Die Zielgrößen der Optimierung sind die Wahrscheinlichkeit eines Leistungsdefizits gegenüber zu den erwartenden Kosten; gelöst wird dieses Problem mittels eines Genetischen Algorithmus.

Zur Optimierung der Struktur elektrischer und weiterer Netzwerke haben ebenfalls Zio et al. unterschiedliche Ansätze zur Redundanzallokation verfolgt [15]. Neben ähnlichen Ansätzen zur Netzwerkoptimierung wie zuvor, konzentrieren sich die Arbeiten dabei auch auf Aspekte der Parameterunsicherheiten [73]. Hierfür werden ein Genetischer Algorithmus mit einer *Monte Carlo* Simu-

lation zur Zielwertauswertung gekoppelt. Wobei die *Monte Carlo* Simulation für unterschiedliche Kombinationen des Parameterraumes Zielwerte berechnet und somit einen Eindruck über die Verteilung des Zielwertraumes vermittelt. Neben neuen Optimierungsansätzen werden zudem spezielle Aspekte der bestehenden Verfahren analysiert, beispielsweise die Konzentration auf bestimmte gewünschte Zielwertbereiche mittels Genetischer Algorithmen [15].

Coit et al. haben weitere Problemstellung zur Redundanzoptimierung untersucht und entsprechende Methoden und Verfahren entwickelt. Die Systemmodelle beschränken sich dabei auf seriell-parallele Logiken mit unterschiedlichen zusätzlichen Attributen. Die Arbeit von Taboada et al. berücksichtigt dabei verschiedene Leistungsniveaus von Komponenten in Kombination mit der Zuverlässigkeit der Komponente, den monetären Kosten und der Masse. Analog zur Arbeit von Levitin handelt es sich dabei um eine Optimierung mit diskreten Komponentenzuständen. Das mehrkriterielle Optimierungsproblem wird mit Hilfe eines Genetischen Algorithmus gelöst [120]. Neben der Lösung mittels Genetischem Algorithmus wird zudem die Optimierung mit weiteren Heuristiken, zum Beispiel *Tabu Search*, untersucht [98].

Komplexe Strukturen

Neben den seriellen und seriell-parallelen Strukturen erfordern eine steigende Vernetzung und Fehlertoleranz beliebiger industrieller Systeme auch immer häufiger die Verwendung von komplexen Strukturen [89]. Die einfachste komplexe Struktur stellt dabei die bidirektionale Brückenstruktur entsprechend Abbildung 3.7 dar. Da sich komplexe Strukturen nicht durch Boolesche Grundoperationen zusammenfassen lassen, ist eine Lösung des komplexen Redundanzallokationsproblems nur mit Hilfe von Minimalpfaden oder analogen Methoden möglich [10]. Aggarwal hat ein deterministisches Verfahren zur Optimierung einer einfachen Brückenschaltung vorgestellt [3]. Die Methode basiert auf der systematischen Durchsuchung des Architekturraums mit Hilfe der marginalen Importanz $\frac{\delta R_i}{\delta R_S}$. Entsprechend Abbildung 3.7 wird mit einem Mehrschrittverfahren für jede Funktion die Anzahl der optimalen Redundanzen untersucht, als konträre Zielgrößen können nichtlineare Nebenbedingungen berücksichtigt werden. Das Verfahren beginnt dabei mit der Funktion mit der höchsten Zuverlässigkeit und schlägt eine identische redundante Komponente vor, nach einer Überprüfung der Nebenbedingungen wird entweder eine weitere redundante Komponente hinzugefügt oder die vorherige Komponente entfernt und mit der nächsten Systemfunktion fortgefahren. Das Verfahren liefert dabei nicht garantiert das globale Optimum, jedoch gute Ergebnisse für unterschiedliche Parameterkombinationen [3]. Als Erweiterung des Verfahrens von Aggarwal

betrachtet Shi nicht die Komponente mit der höchsten Zuverlässigkeit, sondern den Minimalpfad des betrachteten Systems mit der höchsten Sensitivität und aus diesem erst die Komponente mit der höchsten Zuverlässigkeit. Das weitere Verfahren basiert auf ähnlichen Schritten wie bei dem ursprünglichen Verfahren, findet durch die Betrachtung der Minimalpfade und der Sensitivität jedoch bessere Lösungen als Aggarwal [111]. Eine Heuristik für die Redundanzoptimierung fester, komplexer Struktur haben erstmals Ravi et al. genutzt. Mit Hilfe eines *Simulated Annealing* Algorithmus wurde das globale Optimum einer Brückenstruktur gesucht; wobei das Problem wie zuvor durch Massenparameter beschränkt wurde [100].

3.2.2 Zusammenfassender Vergleich und Diskussion

Die Methoden und Werkzeuge zur Sicherheits- und Zuverlässigkeitsoptimierung zeigen, dass das Gebiet der Redundanzallokation seriell-paralleler Systeme ingenieurwissenschaftlich ausreichend untersucht ist und zahlreiche Verfahren zur Optimierung diverser Probleme vorliegen. Wobei sich über die Menge der behandelten Probleme zeigt, dass kein Lösungsverfahren den weiteren Algorithmen überlegen ist [89]. Für die Optimierung komplexer, fehlertoleranter Flugzeug-Systemarchitekturen lassen sich jedoch die folgenden Defizite feststellen.

Die vorhandenen Verfahren sind nicht in der Lage beliebige, komplexe Strukturen zu optimieren [5, 89]. Dabei ist zu berücksichtigen, dass die vorhandenen Verfahren zur Optimierung komplexer Strukturen auf festen Systemstrukturfunktionen basieren und somit nur singuläre Problemstellungen darstellen, anstatt einer allgemeingültigen Methode. Dabei gilt für alle betrachteten Redundanzallokationen, dass eine Systemstruktur derart variiert wird, dass für die einzelnen Funktionen die parallele Anordnung redundanter Komponenten untersucht wird. Es besteht nicht die Möglichkeit weitere Strukturen, zum Beispiel die serielle Anordnung variabler Komponenten, für eine Systemfunktion zu berücksichtigen. Ein Beispiel hierfür ist die Untersuchung unterschiedlicher Sensorkonzepte, während eine Variante aus der seriellen Anordnung von Sensor, Datenübertragung und einer übergeordneten Datenverarbeitung besteht, liefert eine alternative, redundante Komponente selbst ein digitales Signal. Zur quantitativen Bewertung der Architekturen ist dabei eine exakte Berechnung notwendig, da approximative Verfahren zur Zielwertberechnung nicht die nötige Granularität besitzen. Desweiteren wird der Entwurf von Systemen nur in we-

nigen Fällen durch einfache Sicherheitsziele dominiert, vielmehr bestimmen vor allem bei dem Entwurf von Flugzeugsystemen unterschiedliche Sicherheitsanforderungen die Redundanzkonzepte für die betrachteten Systemfunktionen. Es fehlt daher an einer Methode mehrfache Systemanforderungen in der Optimierung abzubilden. Zudem erfordert die Optimierung fehlertoleranter Flugzeugsysteme auch die Untersuchung degradierter Systemzustände, um beispielsweise Fehlertoleranz und Aspekte der Systemzuverlässigkeit abzudecken [66].

Zum Abschluss dieses Kapitels wird daher auf Grundlage der vorherigen Betrachtungen zu Flugzeugsystemen, dem Entwurfsprozess und verfügbaren Methoden und deren Defiziten ein Konzept zur Optimierung von Flugzeug-Systemarchitekturen auf Basis eines variablen Strukturmodells vorgestellt.

3.3 Konzept zur Redundanzallokation komplexer Flugzeug-Systemarchitekturen

Sowohl die Betrachtung der bestehenden Entwurfsmethoden als auch des Entwurfsprozesses haben gezeigt, dass die Auswahl einer Systemarchitektur einen mehrkriteriellen Entscheidungsprozess darstellt. Auch wenn die Untersuchung von Architekturvarianten im Entwicklungsprozess von Flugzeugsystemen vorgesehen ist, wird häufig nur eine geringe Menge der möglichen Varianten analysiert, was zu suboptimalen Architekturen führen kann und somit entweder zu zahlreichen Nachbesserungen im weiteren Entwicklungsprozess oder zu überredundanten Architekturen mit entsprechend großer Systemmasse.

Die Optimierungsumgebungen beschränken sich dabei größtenteils auf Parameteroptimierungen detaillierter Systemmodelle, die anhand einer festen Architektur den parametrischen Lösungsraum untersuchen und auf diese Weise Parameter für die Systembewertung berechnen oder den möglichen Lösungsraum ermitteln [46, 88]. Es zeigt sich jedoch auch, dass für den Entwurfs- und Entscheidungsprozess nicht nur eine global optimale Lösung von Interesse ist, sondern vor allem bei mehrkriteriellen Optimierungsproblemen, gleich ob diskreter oder parametrischer Art, auch Kenntnisse über den ermittelten Zielwertraum und die Verteilung der Lösungsmenge. Aspekte der Architekturoptimierung von Flugzeugsystemen greifen vor allem Annighöfer, Bauer, Haitao, Salomon und Schulz auf, wobei die quantitative Bewertung der Sicherheits- und Zuverlässigkeit hierbei nur in drei Arbeiten aufgegriffen wird [6, 9, 42, 104, 107]. Hierbei ist zu beachten, dass diese Aspekte nach dem Entwurfsprozess nach

SAE ARP 4754 einen wesentlichen Teil zur Architekturdefinition beitragen. Ein Grund hierfür liegt in der schwierigen Integration von Sicherheits- und Zuverlässigkeitsmodellen in die funktionalen Systemmodelle. Diese werden von Systementwicklern für die funktionale Bewertung, wie zur Abschätzung des dynamischem Verhaltens oder des Leistungsbedarfs, aber auch zur Abschätzung der Systemmasse und Betriebs- sowie Lebenszykluskosten genutzt. Dabei wird auch in diesen Arbeiten explizit die Bedeutung der Sicherheit und Zuverlässigkeit für die Systemarchitektur hervorgehoben [66, 59]. Problematisch bei einer Integration der Sicherheitsanalysen in die funktionalen Systemmodelle ist zudem die notwendige Unabhängigkeit der Sicherheitsanalysen von der weiteren Systementwicklung. Diese Segregation innerhalb der Entwicklung ist vor allem bei Systemen mit hohen DAL notwendig, um den Aspekten der *Common Mode Analyse* gerecht zu werden.

Die Betrachtung der bestehenden Optimierungsverfahren für die Redundanzallokation hat gezeigt, dass es vor allem an einer allgemeingültigen Methodik zur Modellierung und Optimierung komplexer Systemmodelle fehlt. Für eine Anwendung im Systementwicklungsprozess ist zudem die Berücksichtigung mehrfacher Fehlerbedingungen wichtig, um die dimensionierenden Anforderungen der *PFHA* zu untersuchen. Dabei fehlt es den bisherigen Methoden jedoch an einem Ansatz zur Modellierung und Auswertung mehrfacher sicherheits- und zuverlässigkeitsrelevanter Zielgrößen. Die Kopplung von Sicherheit und Zuverlässigkeit erfordert für Flugzeugsysteme zudem eine Untersuchung degradierter Systemzustände.

Sowohl die Untersuchung der spezifischen Verfahren zur Analyse und Optimierung von Flugzeugsystemen als auch der Methoden der Redundanzallokation haben gezeigt, dass die Ermittlung der Pareto-Front als Lösung des mehrkriteriellen Optimierungsproblems sinnvoll ist, da diese Lösung die Transparenz nicht einschränkt. Dadurch ist jedoch auch ein weiterhin unübersichtlicher Zielwertraum möglich, wodurch der Anwender durch die zu entwickelnde Methode keinen Vorteil hätte. Für die weitere Reduktion auf einen überschaubaren Architekturraum ist daher ein technologiespezifisches Verfahren notwendig, das die Anforderungen an Flugzeugsysteme und deren Entwicklungsprozess berücksichtigt.

Aufgrund der Bedeutung von Sicherheits- und Zuverlässigkeitsanforderungen für den Architekturentwurf wurde für diese Arbeit ein Bewertungsprozess definiert, der die Bewertung dieser Kriterien in den Mittelpunkt des Systemmodells und somit auch der Optimierung stellt. Um jedoch die Möglichkeit

zur Abschätzung ergänzender Systemcharakteristika zu geben, wurde zudem die Abschätzung summativer Komponentenparameter, wie der Systemmasse, berücksichtigt. Das vollständige Konzept zur Architekturoptimierung ist in Abbildung 3.8 dargestellt.

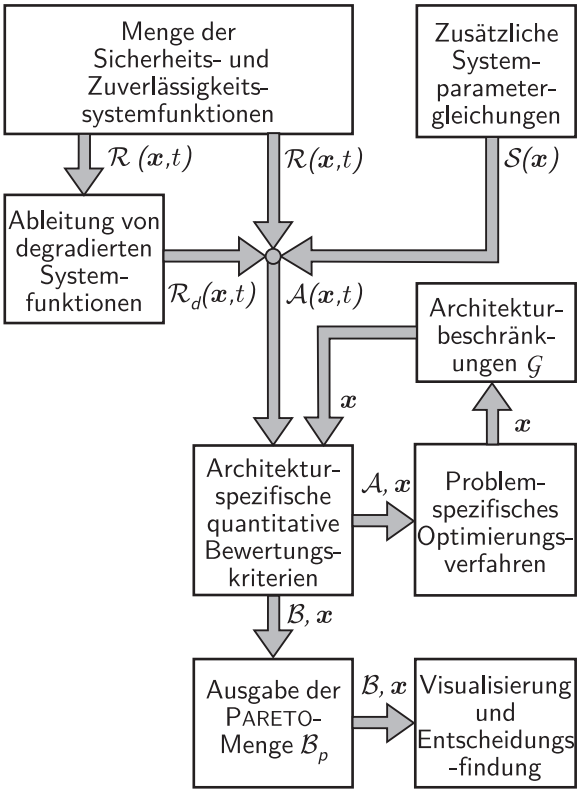


Abb. 3.8: Darstellung des Konzepts zur Redundanzoptimierung komplexer Systemarchitekturen

Basierend auf den Ergebnissen der *PFHA* werden für die identifizierten dimensionierenden Fehlerbedingungen des untersuchten Systems zeit- sowie architekturvariable Systemfunktionen aufgestellt. Das Analysemodell beruht dabei auf dem hybriden Systemmodell von REHAGE und VAHL, die Einbringung der Freiheitsgrade wird im nachfolgenden Kapitel untersucht. Die Nutzung dieses

erweiterten Systemmodells mit variabler Struktur gestattet dabei eine Modellierung beliebiger, komplexer Strukturen mit mehrfachen Fehlerbedingungen. Anhand der nominellen Systemgleichungen lassen sich degradierte Systemfunktionen ableiten, die somit auch die Untersuchung der Fehlertoleranz einer Systemarchitektur ermöglichen. Neben diesen Sicherheits- und Zuverlässigkeitsmodellen werden zudem weitere Systemparameter summativ abgeschätzt, beispielsweise die Systemmasse auf Grundlage der verwendeten Komponenten. Die Menge der aufgestellten Systemfunktionen $\mathcal{A}(\mathbf{x}, t)$ und der Architekturraum $\mathbf{X}$ beschreiben somit den erreichbaren Zielwertraum. Hierbei wird der theoretische Architekturraum $\mathbf{X}_{th}$ mittels der Nebenbedingungen $\mathcal{G}$ bereits derart reduziert, dass nur noch technisch sinnvolle Architekturen vorhanden sind. Zudem lassen sich mit den Nebenbedingungen Präferenzen und Vorwissen in den Architekturentwurf einbringen, was für die Optimierung von essentieller Bedeutung ist [18]. Aufgrund der angestrebten Allgemeingültigkeit des Optimierungsverfahrens ist die Auswahl eines Optimierungsverfahrens für die Menge der möglichen Probleme nicht denkbar, da sich hierfür Problemgröße und Komplexität zu stark unterscheiden und es sich nach den vorherigen Betrachtungen um ein NP-vollständigen Problem handelt. Aus diesem Grund wird in Kapitel 5 die Auswahl geeigneter Verfahren untersucht. Das Ergebnis der Architekturoptimierung ist in jedem Fall jedoch die Menge der ermittelten nicht-dominierten Lösungen, die so genannte PARETO-Menge $\mathcal{B}_p$. Die Ausgabe der optimalen Lösungsmenge unterstützt somit die Transparenz der Optimierung und einen Aspekt des *Systems Engineering* von komplexen Systemen, dass eine Lösung solcher Systeme niemals ein Optimum aller Zielgrößen darstellen kann [101]. Die Ausgabe der Lösungsmenge erfordert im anschließenden Prozess jedoch eine angeleitete Auswahl der Architekturen, die für den weiteren Entwicklungsprozess lohnenswert erscheinen. Solange die Sicherheitsanforderungen erfüllt werden, sind jedoch nicht zwangsläufig die absoluten Zielwerte entscheidend, sondern die Darstellung welche Architekturen in welchen Zielgrößen Stärken und Schwächen besitzen [93]. Aus diesem Grund ist eine Visualisierung der mehrkriteriellen Zielwerte notwendig, die den Entscheidungsprozess unterstützt und anhand derer ein Vergleich der Architekturen möglich ist.

Für den Entwurf einer Methode zur Redundanzallokation komplexer Flugzeugsysteme unter Berücksichtigung der Systemsicherheit und -zuverlässigkeit wurden die folgenden Anforderungen an die Methode und auch an eine programmtechnische Umsetzung definiert:

Systemstrukturen: die neue Methode muss komplexe Systemstrukturen bewerten können. Dieses umfasst für die Sicherheits- und Zuverlässigkeitbewer-

tung der Systemarchitekturen somit nicht nur serielle und parallele Strukturen, sondern vor allem Brückenstrukturen und ähnliche Logiken, die sich nicht mit Hilfe BOOLEscher Grundoperationen vereinfachen lassen.

Transparenz: der Optimierungsprozess soll möglichst transparent ablaufen und den Anwender an dem gesamten Optimierungsprozess beteiligen, der aus Optimierung und Lösungsauswahl besteht. Anhand der Optimierungsergebnisse und der Visualisierung soll der anwendende Systemingenieur die Grenzen des Systementwurfs sowie Auswirkungen der Randbedingungen intuitiv nachvollziehen können.

Allgemeingültigkeit: die zu entwickelnde Methode muss allgemeingültig auf alle Flugzeugsysteme anwendbar sein, die Reduktion der Analysemodelle auf feste Strukturen mit hinterlegten Zielfunktionen ist somit nicht zulässig. Die Beschränkung der Beispiele auf elektrische Systeme in den Kapiteln 6 und 7 ist der Motivation dieser Arbeit geschuldet.

Zielfunktionen: als Zielfunktionen sollen die Analysegleichungen des bereits bestehenden hybriden Systemmodells nach VAHL und REHAGE, vgl. Abschnitt 3.1, und dessen Analysefähigkeiten genutzt werden. Weitere Zielgrößen, die den Entscheidungsprozess in der frühen Entwicklungsphase sinnvoll unterstützen und den Aufwand unterschiedlicher Redundanzkonzepte abbilden, sind zu berücksichtigen.

Assistenzfunktion: das Ziel der zu entwickelnden Methode ist die Unterstützung des anwendenden Systemingenieurs im Vorentwurf komplexer, sicherheitskritischer Flugzeugsysteme. Die Art der Problembeschreibung und die Ergebnisdarstellung sind daher auf die Kenntnisse des Anwenders anzupassen. Die Methode und deren Implementierung dürfen nicht die Entwurfsfreiheiten und die Kreativität des Anwenders einschränken oder ihm die Verantwortung für den Systementwurf abnehmen [106].

Rechenleistung: die Assistenzfunktion soll auf einem üblichen Arbeitsplatzrechner lauffähig sein und explizit nicht auf die Verfügbarkeit von Parallelrechnern angewiesen sein.

Die genannten Anforderungen definieren den Rahmen für die weitere Methodenentwicklung und werden im weiteren Verlauf der Arbeit als Referenz herangezogen. Das entwickelte Konzept soll dabei keine automatische Optimierungsfunktion zur Generierung fehlertoleranter Systemarchitekturen darstellen, sondern dem Ingenieur im Zentrum des Entwicklungsprozesses als Assistenzfunktion unterstützen und ihm aufbereitete Informationen für den Entscheidungs-

prozess liefern. Die Entscheidungen inwieweit der Architekturraum anhand der Nebenbedingungen oder im Anschluss an die Redundanzallokation durch den Auswahlprozess reduziert wird, obliegt daher dem anwendenden Systemingenieur. Somit ist auch ein Vergleich der Optimierungsergebnisse mit Ergebnissen weiterer Optimierungsumgebungen mit anderen, funktionalen Zielwerten möglich und dadurch ein ganzheitlicher Auswahlprozess der optimalen Systemarchitektur.

Für die Umsetzung des vorgestellten Konzepts folgt in dem nächsten Abschnitt die Entwicklung eines Systemmodells, dass die Berücksichtigung von Freiheitsgraden im hybriden Systemmodell ermöglicht und dessen Architekturraum durch Nebenbedingungen auf technisch sinnvolle Lösungen reduziert wird. Zur Untersuchung des möglichen Zielwertraumes folgt in Kapitel 5 die Auswahl und Adaption geeigneter Optimierungsverfahren. Die Integration der entwickelten Methode in den zuvor vorgestellten Prozess wird in Kapitel 6 untersucht. Abschließend folgt die exemplarische Anwendung des Verfahrens auf den Architekturentwurf eines elektrischen Energieversorgungssystems.

4 Hybride Systemmodellierung variabler Strukturen

Die Verwendung des vorgestellten hybriden Systemmodells, mit den Bestandteilen von Vahl und Rehage, erfordert für eine automatische Architekturbewertung die Einbringung architektureller Freiheitsgrade. Hierfür wurde das mehrfach-redundante Systemmodell (MRS) entwickelt, ein hybrides Systemmodell mit variabler Struktur. Dieses besteht aus den Bestandteilen *hybrides Systemmodell* zur Modellierung der benötigten Ausfalllogiken und Freiheitsgrade, *binärer Entscheidungsbaum* mit beschränkenden Nebenbedingungen, der *Ableitung degradierter Systemzustände*, der *Optimierung serieller Systemstrukturen* sowie der Einbringung weiterer, konträrer *summativer Zielgrößen*. Die genannten Bestandteile werden nachfolgend erläutert und verifiziert.

Das mehrfach-redundante Systemmodell basiert auf den lokalen Entscheidungspunkten einer Systemarchitektur, den Redundanzkonzepten, den Technologieentscheidungen und Anbindungsfragen für unterschiedliche Systemfunktionen. Das Prinzip ist in Abbildung 4.1 dargestellt. Anhand der Systemfunktionen lässt sich ein Systemmodell erstellen, wobei jeder Block einer Komponentenfunktion bzw. einem externen Ereignis entspricht. Für jede betrachtete Funktion können die möglichen Umsetzungen mit Hilfe der Ereignisblöcke abgebildet werden. Im Normalfall wird hierfür eine parallele Modellierung der Ereignisse verwendet, so dass das entsprechende MRS auf Ereignisebene mehr Redundanzen enthält, als in einer möglichen Umsetzung. Diese Redundanzen stellen die Freiheitsgrade der Systemoptimierung dar und lassen sich mittels technisch- und präferenzbedingter Nebenbedingungen und der in Abschnitt 4.2 beschriebenen Methode zunächst auf eine gültige Lösung reduzieren. Anschließend können die Systemparameter mit Hilfe der variablen Strukturgleichungen ausgewertet werden. Für die Bewertung von Zuverlässigkeitsgrößen und weiteren seriellen Strukturen ist zudem die Berücksichtigung serieller variabler Logiken möglich, die durch einen gesonderten Algorithmus berücksichtigt werden.

Für das mehrfach-redundante Systemmodell wird die Ereignismenge $\mathbf{K}$ in die Mengen der variablen Ereignisse $\mathbf{K}_v$ und festen Ereignisse $\mathbf{K}_f$ unterteilt. Eine explizite Architektur wird dabei mit Hilfe des Architekturvektors $\mathbf{x}$ beschrie-

ben, der die Ereignisse der Menge $\mathbf{K}_v$ adressiert, somit gilt bezüglich der Kardinalität $\mathbf{x} \in \{0,1\}^{|\mathbf{K}_v|}$ und

$$|\mathbf{K}| = |\mathbf{K}_v| \cup |\mathbf{K}_f| \; . \tag{4.1}$$

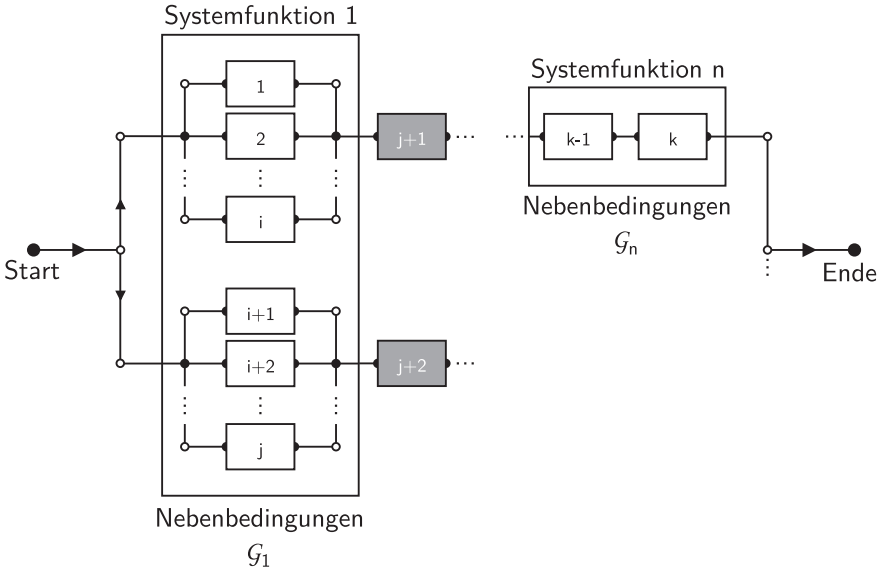


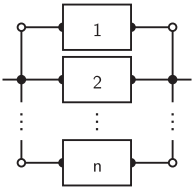
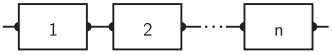
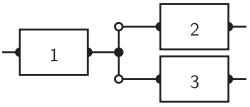
Abb. 4.1: Konzept des mehrfach-redundanten Systemmodells

4.1 Beschränkung binärer Entscheidungsbäume durch Nebenbedingungen

Um technisch sinnvolle und gültige Lösungen zu erreichen, ist es notwendig den theoretischen, diskreten Architekturraum der Größe $2^{|\mathbf{K}_v|}$ durch Nebenbedingungen zu beschränken. Jede Nebenbedingung besteht dabei aus einer Gleichung oder Ungleichung, die die Architekturvariablen von Ereignissen der Menge $\mathbf{K}_v$ adressiert und der Verknüpfung der Ereignisse einen Wert zuweist. Die folgende Tabelle enthält unterschiedliche variable Strukturen, die in Sy-

stemmodellen auftreten können und zeigt exemplarisch die entsprechenden Nebenbedingungen.

Tab. 4.1: Typische Nebenbedingungen fehlertoleranter Flugzeugsystem-Architekturen

Bezeichnung	Mögliche Struktur	Nebenbedingung g
k-aus-n		$g_1 = \mathbf{x}(1) + \mathbf{x}(2) + \ldots + \mathbf{x}(n) \leq k$
identisch		$g_2 = \mathbf{x}(1) + \mathbf{x}(2) + \ldots + \mathbf{x}(n) = k$
abhängig		$g_3 = \mathbf{x}(1) - \mathbf{x}(2) - \mathbf{x}(3) \leq 0$

Bei der *k-aus-n*-Struktur werden maximal k Ereignisse aus der Ereignismenge ausgewählt, dieses ermöglicht zum Beispiel die Berücksichtigung unterschiedlicher Technologien für eine Komponentenfunktion und somit unterschiedliche Ausfallraten und weitere technologiespezifische Parameter. Sofern in einem Pfad eines Zuverlässigkeitsblockdiagramms mehrere Komponentenfunktionen für die Funktionsfähigkeit des Pfades notwendig sind, kann dieses durch die allgemeine Nebenbedingung g_2 berücksichtigt werden. Die Nebenbedingung diktiert hierbei identische Werte für die Architekturvariablen der Ereignisse. Diese Form der Nebenbedingung wird zudem für die Berücksichtigung serieller Strukturen in Abschnitt 4.3 genutzt. Sofern keine Nebenbedingung vorliegt, die die gemeinsame Verwendung der Ereignisse diktiert, handelt es sich um eine serielle, variable Struktur, für die das neutrale Element der variablen Ereignisse geändert werden muss. Eine weitere Möglichkeit zur Abbildung von Abhängigkeiten zwischen einzelnen Komponenten bietet die Nebenbedingung

g_3. Sofern eine Komponentenfunktion $\mathbf{x}(1)$ auf die Funktionsfähigkeit weiterer Komponenten, diesem Fall $\mathbf{x}(2)$ und $\mathbf{x}(3)$ angewiesen ist, diese jedoch auch unabhängig von dieser Funktion genutzt werden können, kann dieses ebenfalls durch die Nebenbedingungen abgebildet werden. Ein Beispiel hierfür ist ein fakultativer Sensor, der ein Signal an eine lokale Recheneinheit überträgt. Die variable Recheneinheit wird dabei wahlweise auch ohne diesen Sensor für weitere Funktionen verwendet. Sollte jedoch der Sensor ausgewählt werden, ist auch die Recheneinheit erforderlich. Bei den dargestellten Nebenbedingungen handelt es sich um exemplarische Abhängigkeiten zur Beschränkung des möglichen Architekturraums auf zulässige Lösungen, prinzipiell ist jede lineare Abhängigkeit zwischen den einzelnen Ereignissen möglich.

Die Ereignisse der variablen Menge können weiterhin mit Hilfe der Menge der Nebenbedingungen $\mathcal{G}$ in lokale Entscheidungspunkte im komplexen Systemmodell eingeteilt werden. Jede Nebenbedingung g_i adressiert dabei eine Menge $\mathbf{KG}_i$ an Ereignissen. Sofern mehrere Nebenbedingungen sich überschneidende Ereignismengen addressieren

$$\mathbf{KG}_n \cap \mathbf{KG}_m = \{\mathbf{x}(p) | \mathbf{x}(p) \in \mathbf{KG}_n \wedge \mathbf{x}(p) \in \mathbf{KG}_m\} \tag{4.2}$$

werden die betroffenen Ereignismengen vollständig zu einer Ereignismenge $\mathbf{KG}_i \in \mathbf{K}_v$ zusammengefasst. Somit beschreiben die Mengen $\mathbf{KG}_i$ die Abhängigkeiten der variablen Ereignisse und somit die lokalen Entscheidungspunkte im Architekturentwurf. Für die spätere Nutzung durch die Optimierungsalgorithmen wurde daher ein Algorithmus implementiert, der diese Kombinationen automatisch ausliest.

Zur Veranschaulichung des möglichen Architekturraums kann dieser mit Hilfe eines binären Entscheidungsbaumes (engl. *Binary Decision Diagram, BDD*) dargestellt werden [26]. In Abbildung 4.2 ist der vollständige Entscheidungsbaum für die zuvor betrachtete unidirektionale Brückenstruktur aus Abbildung 3.3 dargestellt. Zur besseren Veranschaulichung wurden dabei keine Ordnungsreduktionen durch Zusammenfassung von Ereignismengen vorgenommen. Hierbei wurden alle Ereignisse als variabel definiert, es ergeben sich somit $2^5 = 32$ theoretisch mögliche Lösungen. Bei jeder Verzweigung des Entscheidungsbaumes wird der betrachteten Architekturvariablen ein Wert $x_i \in \{0, 1\}$ zugewiesen. Der theoretische Lösungsraum wurde mit Hilfe der nachfolgenden Nebenbedingungen derart reduziert, dass ausschließlich technisch sinnvolle Lösungen

betrachtet werden, beispielsweise keine Architekturen ohne die Ereignisse 1 und 2, was anderweitig zu trivialen Ergebnissen mit $R(t) = 0 \quad \forall \quad \{\mathbf{x}|\mathbf{x}(1) = 0 \wedge \mathbf{x}(2) = 0\}$ führen würde. Die gültigen Lösungen entsprechend der Nebenbedingungen sind in der folgenden Abbildung hervorgehoben, wobei für die ungültigen Lösungen jeweils ein Verstoß gegen eine Nebenbedingung angegeben ist.

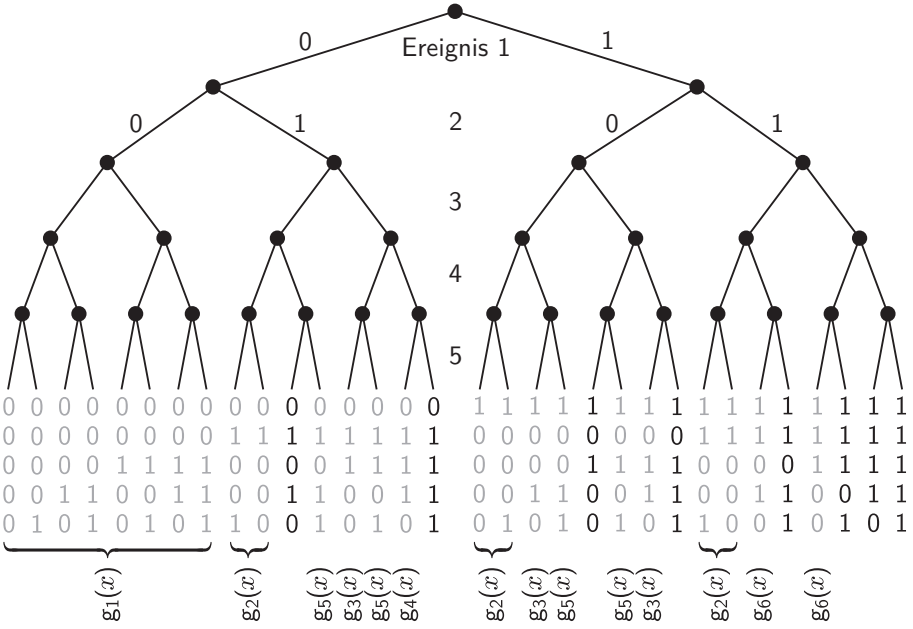


Abb. 4.2: Binärer Entscheidungsbaum der unidirektionalen Brückenstruktur

Die folgenden konjunktiven Nebenbedingungen im Gleichungssystem 4.3 schränken den theoretischen Architekturraum derart ein, dass alle Nebenbedingungen erfüllt sein müssen [83]. Entscheidend für den weiteren Optimierungsprozess ist, dass mit Hilfe der Nebenbedingungen nur der Architekturraum beschränkt wird, jedoch nicht der Zielwertraum, was die Transparenz im Entwurfsverfahren unterstützt. Eine Beschränkung der Zielwerte, beispielsweise

der Sicherheitsfunktionen aufgrund der Fehlerklassifizierung, würde den Entscheidungsprozess beeinflussen und ist somit entsprechend der Anforderungen aus Abschnitt 3.3 an die zu entwickelnde Methode weder zulässig noch sinnvoll bezüglich einer transparenten Architekturauswahl.

$$\mathcal{G} = \begin{cases} g(1) & : & \mathbf{x}(1) + \mathbf{x}(2) \geq 1 \\ g(2) & : & \mathbf{x}(3) + \mathbf{x}(4) \geq 1 \\ g(3) & : & \mathbf{x}(4) - \mathbf{x}(2) - \mathbf{x}(5) \leq 0 \\ g(4) & : & \mathbf{x}(3) - \mathbf{x}(1) - \mathbf{x}(5) \leq 0 \\ g(5) & : & \mathbf{x}(3) + \mathbf{x}(4) - \mathbf{x}(5) \geq 1 \\ g(6) & : & \mathbf{x}(1) + \mathbf{x}(2) - \mathbf{x}(3) - \mathbf{x}(4) - \mathbf{x}(5) \leq 0 \end{cases} \tag{4.3}$$

Aufgrund der komplexen Systemcharakeristika und der Fülle der Nebenbedingungen für große Optimierungsprobleme sollten die Architekturbeschränkungen im Rahmen der Problemmodellierung validiert werden. Die Nebenbedingungen $\mathcal{G}$ lassen sich mittels der enthaltenen Ereignisse in die Gruppen von Nebenbedingungen $\mathcal{G}_1, \ldots, \mathcal{G}_n$ unterteilen und separat validieren. Mit Hilfe der fiktiven Nebenbedingungen für die Ereignismenge q

$$x_i = 1 \; \forall \; x_i \notin \mathbf{KG}_q \tag{4.4}$$

lassen sich die Einträge eines Architekturvektors als invariabel deklarieren, wodurch nur die betrachtete Menge von Nebenbedingungen $\mathcal{G}_q$ berücksichtigt und auch nur der lokale Entscheidungsraum KG_q variiert wird. Somit lässt sich die vollständige Überprüfung aller Systemarchitekturen der Anzahl $n_{ges} \leq 2^{|KG|}$ der variablen Ereignissen aufteilen in die Überprüfung der m lokalen Entscheidungspunkte, die durch die Überschneidung der entsprechenden Nebenbedingungen beschrieben werden. Für die Variantenanzahl jedes Entscheidungspunktes gilt dabei:

$$\begin{aligned} n_i \quad &\leq \quad 2^{|\mathbf{KG}_i|} \\ &\text{mit} \quad n_{ges} = \prod_{1}^{m} n_i \end{aligned} \tag{4.5}$$

Somit wird durch die Gruppierung der Ereignisse mit Hilfe der Nebenbedingungen erreicht, dass das kombinatorische Problem nicht mehr durch die binären

Ereigniszustände beschrieben wird, sondern anhand technisch sinnvoller Subsystemarchitekturen. Diese Reduktion der Entscheidungsebenen wird in den später folgenden Optimierungsverfahren zur Laufzeitoptimierung und Abschätzung der erreichbaren Zielwerte genutzt.

4.2 Ermittlung der Systemfunktionen variabler Strukturen

Für die Bewertung von Systemarchitekturen hinsichtlich der quantitativen Sicherheit und Zuverlässigkeit wurde das mehrfach-redundante Systemmodell aus Abbildung 4.1 entwickelt. Anhand dieses erweiterten hybriden Systemmodells wird ein Architekturraum aufgespannt, der durch die Nebenbedingungen begrenzt wird. Im Nachfolgenden wird die Abbildung der unterschiedlichen Freiheitsgrade durch die Systemgleichungen des hybriden Systemmodells untersucht. Dieses betrifft zum einen die Abbildung durch die disjunktive Verknüpfung orthogonaler Minimalpfade. Zum anderen die erreichbaren, funktionsfähigen Systemzustände und deren Eintrittswahrscheinlichkeit auf Basis der nebenläufigen, endlichen Zustandsautomaten. In beiden Fällen ist eine Reduktion der ermittelten Systemgleichungen entsprechend der betrachteten Architektur erforderlich.

Die folgende Grafik zeigt hierfür die Reduktion des Beispielsystems aus Abbildung 3.3, hierfür wird das Ereignis 2 nicht mehr berücksichtigt. Das Ziel der nachfolgenden Betrachtungen ist die Reduktion der Systemmodelle auf Grundlage der ausgelesenen Minimalpfade und Zustandsgleichungen des mehrfach-redundanten Systemmodells gemäß eines aktuellen Architekturvektors $\mathbf{x}$.

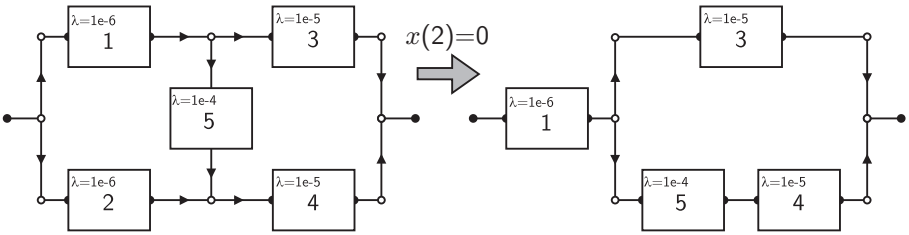


Abb. 4.3: Nachträgliche Reduktion einer unidirektionalen Brückenstruktur

4.2.1 Zuverlässigkeitsblockdiagramme variabler Strukturen

Die nachträgliche Reduktion eines Zuverlässigkeitsblockdiagramms lässt sich mit Hilfe der Elementarzustände erläutern. Die Elementarzustände werden zur Bestimmung der Minimalpfade genutzt, deren anschließende Orthogonalisierung entsprechend Abschnitt 3.1.1 zur Systemstrukturfunktion $\Phi(\mathbf{K})$ führt. Soll ein Ereignis im Nachhinein in der Wahrscheinlichkeitsberechnung nicht berücksichtigt werden, entspricht dieses dem Entfernen aller Zustände, in denen das Fehlerereignis nicht eingetreten bzw. die Komponente funktionsfähig ist [77]. Für die unidirektionale Brückenschaltung aus Abbildung 4.3 ergeben sich die in Tabelle 3.1 enthaltenen Elementarzustände. Anhand dieser Zustände ist es möglich, das betrachtete System auf eine seriell-parallele Struktur zu reduzieren, indem das Beispiel ohne das Ereignis K_2 betrachtet wird. Somit ergeben sich, in Anlehnung an die vorherigen Elementarzustände, die in Tabelle 4.2 enthaltenen reduzierten Zustände.

Tab. 4.2: Elementarzustände der unidirektionalen Brückenstruktur ohne Ereignis 2

Elementarzustand	K_1	K_2	K_3	K_4	K_5	$\Phi(\mathbf{K})$	MP_i
E_9	1	0	1	1	1	1	MP_1, MP_3
E_{10}	1	0	1	1	0	1	MP_1
E_{11}	1	0	1	0	1	1	MP_1
E_{12}	1	0	1	0	0	1	MP_1
E_{13}	1	0	0	1	1	1	MP_3
E_{14}	1	0	0	1	0	0	
E_{15}	1	0	0	0	1	0	
E_{16}	1	0	0	0	0	0	
E_{25}	0	0	1	1	1	0	
E_{26}	0	0	1	1	0	0	
E_{27}	0	0	1	0	1	0	
E_{28}	0	0	1	0	0	0	
E_{29}	0	0	0	1	1	0	
E_{30}	0	0	0	1	0	0	
E_{31}	0	0	0	0	1	0	
E_{32}	0	0	0	0	0	0	

Der Minimalpfad MP_2 wurde vollkommen aus der Menge der Minimalpfade gestrichen, so dass nur noch die Minimalpfade MP_1 und MP_3 bestehen bleiben, die jeweils einen Zweig der Parallelschaltung repräsentieren. Die Systemstrukturfunktion ergibt sich nach Gleichung 3.11 für das reduzierte System somit mit Hilfe der Elementarzustände zu:

$$R_S(t) \quad = \quad R_1(t)R_3(t) + R_1(t)R_4(t)R_5(t)\,(1 - R_3(t)) \ . \tag{4.6}$$

Somit ist eine Variation der Zuverlässigkeitsblockdiagramme in Abhängigkeit der betrachteten Architektur im Nachhinein möglich. Die Variation über die Elementarzustände würde jedoch eine wiederholte Aufstellung der Systemstrukturfunktionen $\Phi(\mathbf{K})$ für jede untersuchte Fehlerbedingung erfordern und somit auch eine wiederholte Orthogonalisierung. Dieses würde vor allem unter Berücksichtigung der anschließenden Optimierung zu einer stark erhöhten Rechenzeit führen. Aus diesem Grund werden die Systemstrukturfunktionen nur einmalig aufgestellt und erst anschließend in Abhängigkeit der Architektur variiert. Analog zu der Betrachtung der Elementarzustände ist es somit notwendig, die orthogonalen Minimalpfade auf die Ereignisse zu beschränken, die entsprechend des Architekturvektors in der betrachteten Architektur berücksichtigt werden sollen. Die weiteren Ereignisse sollen hinsichtlich der Eintrittswahrscheinlichkeit ein neutrales Element darstellen. Dieses wird für parallele variable Ereignisse durch die folgende diskrete Parametervariation der Ereigniswahrscheinlichkeiten erreicht [96]:

$$R_i = \begin{cases} e^{-\lambda_i \cdot t} & \forall \quad x_i = 1 \\ 0 & \forall \quad x_i = 0 \end{cases} \ . \tag{4.7}$$

Somit ergibt sich für das oben betrachtete Beispiel der unidirektionalen Brückenschaltung durch die Variation der Zuverlässigkeitswerte für das Ereignis K_2 die folgende Systemstrukturfunktion:

$$\begin{aligned} R_S(t) \quad &= \quad R_1(t) \cdot R_3(t) + 0 \cdot R_4(t)\,(1 - R_1(t) \cdot R_3(t)) \\ &\quad\ \ + R_1(t)R_4(t)R_5(t)(1-0)\,(1 - R_3(t)) \qquad (4.8) \\ &= \quad R_1(t)R_3(t) + R_1(t)R_4(t)R_5(t)\,(1 - R_3(t)) \ . \qquad (4.9) \end{aligned}$$

Die Streichung der verschwindenen Terme führt somit zur identischen Systemstrukturfunktion wie mittels Ermittlung über die Elementarzustände. Dabei gilt die hier dargestellte Variation der Systemstrukturfunktion über ein neutrales Element nur für parallele variable Ereignisse. Die Variation serieller Strukturen wird aufgrund der Notwendigkeit eines gesonderten Algorithmus in Abschnitt 4.3 behandelt.

4.2.2 Hybride Systemmodelle variabler Strukturen

Der Ansatz zur Streichung von Minimalpfaden lässt sich ebenfalls auf die Analyse anhand des hybriden Systemmodells übertragen, da hier die Minimalpfade zur Überprüfung der Funktionsfähigkeit der Systemzustände genutzt werden. Somit werden nur Zustandswahrscheinlichkeiten übernommen, wenn noch mindestens ein funktionsfähiger Minimalpfad besteht und somit auch die betrachtete Fehlerbedingung noch nicht eingetreten ist. Die Berücksichtigung der Minimalpfade würde jedoch, analog der vorherigen variablen Systemstrukturfunktion, zu einer wiederholten Aufstellung aller Gleichungen für die Eintrittswahrscheinlichkeiten der Systemzustände führen. Aus diesem Grund wurde ein Ansatz entwickelt, der auf der Basis der vollständigen Menge der Zustandsgleichungen spezifisch für jede Architektur gültige Gleichungen generiert. In diesem Fall wird der ermittelte Zustandsraum derart verschoben, dass nur noch ein geringer Teil, definiert durch den Architekturvektor $\mathbf{x}$, erreichbar ist. Die weiteren Zustände können aufgrund der variierten Fehlerraten der Ereignisse und somit der Zustandswahrscheinlichkeiten nicht mehr erreicht werden. In Abbildung 4.4 ist die Verschiebung des Zustandsraumes für das vorherige Beispiel der Brückenstruktur dargestellt.

Im linken Teil der Abbildung ist der vollständige, unbeschränkte Zustandsraum des mehrfach-redundanten Systemmodells enthalten. Der Initialzustand Z_0 ist dabei der Zustand, in dem das System modelliert wurde. Der beschränkte Architekturraum im rechten Teil der Abbildung enthält den reduzierten Zustandsraum, wobei in diesem Beispiel das Ereignis K_2 nicht berücksichtigt werden soll. Somit wird der Initialzustand in den Zustand Z_2 verschoben, der nun als Zustand Z_2^0 bezeichnet wird.

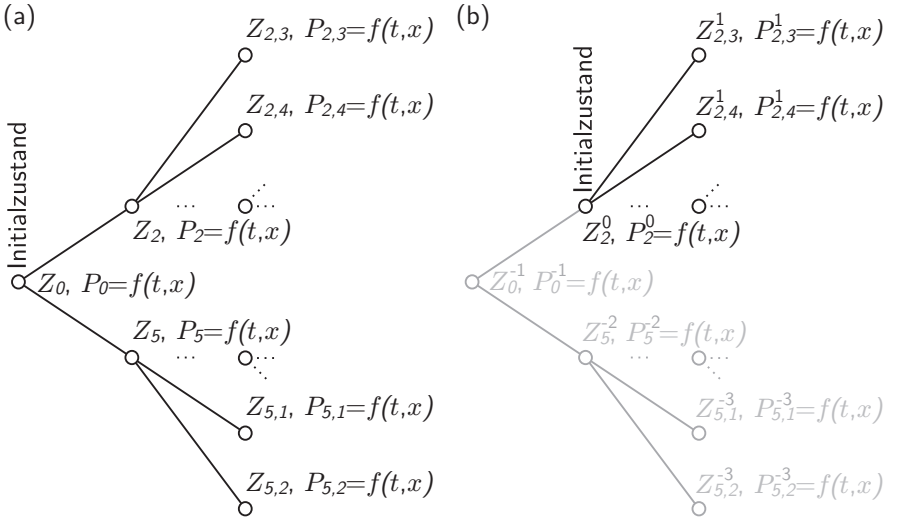


Abb. 4.4: Verschiebung des Initialzustandes eines hybriden Systemmodells

Durch die Verschiebung des Initialzustandes gilt für alle weiteren Zustandswahrscheinlichkeiten somit für die Degradationsstufe d und die degradierten Ereignisse p:

$$P_p^d = \begin{cases} P_i^j(t,x) & \forall \quad d \geq 0 \\ 0 & \forall \quad d < 0 \end{cases} . \tag{4.10}$$

Die Verschiebung des Zustandsraumes kann dabei durch die folgende Parametervariation der Fehlerrate λ eines variablen Ereignisses erreicht werden:

$$\lambda_i \begin{cases} = \lambda_i & \forall \quad x_i = 1 \\ \to \infty & \forall \quad x_i = 0 \end{cases} . \tag{4.11}$$

Somit ergibt sich aus der Parametervariation der Fehlerrate die folgende Variation der Eintrittswahrscheinlichkeit und somit eine Variation der Zustandsgleichungen:

$$R_i = \begin{cases} e^{-\lambda_i \cdot t} & \forall \quad x_i = 1 \\ 0 & \forall \quad x_i = 0 \end{cases} . \tag{4.12}$$

Neben der Verschiebung des Zustandsraums ist zudem eine Betrachtung der Zustandstransitionen notwendig. Eine allgemeine Aussage zur Kürzung der Transitionen ist aufgrund der spezifischen Formulierung durch den Anwender nicht möglich. Bei der Systemmodellierung ist jedoch darauf zu achten, dass die Transitionen derart modelliert werden, dass sich durch die Verschiebung des Zustandsraums keine ungültigen Transitionsbedingungen ergeben. Generell ist dieses durch disjunktive Transitionsbedingungen möglich. Somit werden die Bedingungen auch unabhängig vom Zustand der weiteren Ereignisse geschaltet.

Mit dem oben genannten Ansatz ergeben sich für die unidirektionale Brückenstruktur aus dem vorherigen Beispiel für eine reduzierte, zustandsdiskrete Bewertung ohne Ereignis K_2 die folgenden Zustandsgleichungen:

$$\begin{aligned} P_2^0 &= R_1 R_3 R_4 R_5 \\ P_{2,3}^1 &= \frac{\lambda_3}{\lambda_3} \cdot (1 - R_3) \cdot R_1 R_4 R_5 \\ P_{2,4}^1 &= \frac{\lambda_4}{\lambda_4 + \lambda_5} \cdot (1 - R_4 R_5) \cdot R_1 R_3 \\ P_{2,5}^1 &= \frac{\lambda_5}{\lambda_5 + \lambda_4} \cdot (1 - R_5 R_4) \cdot R_1 R_3 \end{aligned} \tag{4.13}$$

Alle weiteren Zustandswahrscheinlichkeiten laufen gegen null durch die Variation der Fehlerrate $\lambda_2 \to \infty$ und somit der Zuverlässigkeit $R_2 \to 0$. Für die Zustandswahrscheinlichkeit $P_{2,5}$ ergibt sich exemplarisch nach Gleichung 3.18:

$$\begin{aligned} P_{2,5} &= \Bigg(\underbrace{\overbrace{\frac{\lambda_5}{\lambda_5}}^{=1} \cdot \overbrace{\frac{\lambda_2}{\lambda_2 + \lambda_4}}^{\to 1} \cdot \overbrace{(1 - R_2 R_4)}^{\to 1}}_{\to 1} + \dots \\ &\quad \dots + \underbrace{\overbrace{\frac{\lambda_5}{\lambda_5}}^{=1} \cdot \overbrace{\frac{-\lambda_2}{\lambda_5 + \lambda_2 + \lambda_4}}^{\to -1} \cdot \overbrace{(1 - R_5 R_2 R_4)}^{\to 1}}_{\to -1} \Bigg) \cdot \dots \\ &\quad \dots \cdot R_1 R_3 \\ &\to 0 \,. \end{aligned} \tag{4.14}$$

Bei der Berücksichtigung *passiv-warmer* und *passiv-kalter* Automatenzustände mit reduzierten Fehlerraten entsprechend Abschnitt 3.1.2 gilt ebenfalls die

zuvor dargestellte Variation der Fehlerrate und Zuverlässigkeit. Wie bei der Aufstellung der variablen Systemstrukturfunktion ist dieses Verfahren jedoch nur für parallele Ereignisse in der Ausfalllogik anwendbar, die Berücksichtigung variabler, serieller Ereignisse folgt als Sonderfall im nächsten Abschnitt.

4.3 Variation serieller Strukturen

Die Bewertung der Systemzuverlässigkeit erfordert in vielen Fällen die serielle Anordnung von zwei Ereignissen, beispielsweise bei der Zuverlässigkeitsanalyse von redundanten Komponenten, die jedoch aus Sicherheitsgründen beide für einen sicheren Flug notwendig sind. Während die quantitative Sicherheit in diesem Fall durch die parallele Anordnung steigt, sinkt die operationelle Zuverlässigkeit durch die notwendige serielle Anordnung der Ereignisse. Somit ist es möglich, dass zwei Ereignisse unterschiedlich in der Menge der Systemstrukturfunktionen berücksichtigt werden. Da jedoch der oben genutzte Ansatz zur Streichung von Minimalpfaden bei seriellen Strukturen zur Streichung der vollständigen Systemstrukturfunktion führen kann, ist ein erweiterter Ansatz notwendig. Im Folgenden wird daher für das hybride Systemmodell ein Verfahren vorgestellt, dass auch für serielle Strukturen eine Variation zulässt. Dabei bleibt zu beachten, dass die parallele Anordnung von variablen Ereignissen der Normalfall bleibt, da er der typischen Modellierung von Flugzeugsystemen entspricht [126]. Durch das entwickelte Verfahren lässt sich zudem die Komplexität des mehrfach-redundanten Systemmodells reduzieren, da *1-aus-n* Entscheidungen entsprechend Tabelle 4.1 durch eine serielle Struktur abgebildet werden können und somit die Anzahl der Minimalpfade stark reduziert wird, was die Optimierung sehr großer Probleme ermöglicht.

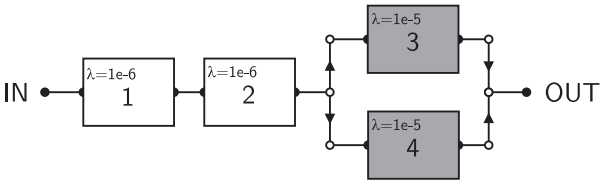


Abb. 4.5: Exemplarisches Zuverlässigkeitsblockdiagramm zur Herleitung serieller Variationslogiken

Als Beispiel wird für die Herleitung der Verfahren zur Berücksichtigung serieller Strukturen bei Zuverlässigkeitsblockdiagrammen und hybriden Systemmodellen das in Abbildung 4.5 dargestellte seriell-parallele Zuverlässigkeitsblockdiagramm genutzt. Die Ereignisse 1 und 2 wurden hierbei als variabel deklariert und durch keine Nebenbedingungen beschränkt. Zur Vereinfachung wurden für das hybride Systemmodell keine Rekonfigurationslogiken implementiert.

4.3.1 Serielle Strukturen von Zuverlässigkeitsblockdiagrammen

Aufgrund der seriellen Anordnung von variablen Ereignissen würde das neutrale Element gemäß der oben genannten Variation zu einem Wegfall aller betroffenen Minimalpfade führen. Im Fall einer reinen seriellen Struktur entsprechend Abbildung 4.5 ergibt sich somit das triviale Ergebnis $R(t, \mathbf{x}) = 0 \quad \forall \quad t, \mathbf{x} \neq \{1, 1\}$. Entsprechend muss bei der Verwendung serieller Strukturen das neutrale Element in der folgenden Form variiert werden:

$$R_i = \begin{cases} e^{-\lambda_i \cdot t} & \forall \quad x_i = 1 \\ 1 & \forall \quad x_i = 0 \end{cases} . \tag{4.15}$$

Da hierbei jedoch zwei Szenarien serieller Ereignisse auftreten können, wurde der in Abbildung 4.6 und 4.7 folgende Algorithmus entwickelt, um die betroffenen seriellen Ereignisse automatisch zu identifizieren. Die Unterscheidung ist notwendig, um automatisiert zwischen einer beliebigen Anzahl serieller, unabhängiger Ereignisse und serieller, abhängiger Ereignisse unterscheiden zu können. Seriell, unabhängige Ereignisse besitzen dabei keine gemeinsamen Nebenbedingungen. Im Gegensatz dazu werden seriell, abhängige Ereignisse entsprechend Tabelle 4.1 durch gemeinsame Nebenbedingungen beschränkt. Letztere Ereignisse könnten mit dem Ansatz für parallele Ereignisse variiert werden, da somit alle betroffenen Minimalpfade gestrichen würden. Seriell, unabhängige Ereignisse erfordern hingegen die Variation der Systemstrukturfunktion zur Erzeugung eines neutralen Elementes entsprechend Gleichung 4.15. Aus diesem Grund wird zunächst untersucht, welche Ereignisse in welchen orthogonalen Minimalpfaden auftreten, wobei hierfür nur die rein multiplikativen Terme berücksichtigt werden müssen. Sofern eine Menge an Ereignissen in den identischen Minimalpfaden auftritt, sind diese Kanidaten für serielle Ereignisse. Zur Unterscheidung zwischen abhängigen und unabhängigen Ereignissen werden anschließend die Nebenbedingungen überprüft. Der erste Schritt ist hierbei

zu prüfen, ob sich die berücksichtigten Ereignisse der Nebenbedingungen für die Kandidatenmenge überschneiden. Ist dieses nicht der Fall, stehen die Ereignisse in keinem Zusammenhang zueinander, es handelt sich somit um serielle, unabhängige Ereignisse. Liegen hingegen Nebenbedingungen vor, die die vollständige Kandidatenmenge in einen Zusammenhang stellt, müssen diese vollständig ausgewertet werden. Schränken die Nebenbedingungen dabei den Architekturraum derart ein, dass die Ereignisse der Kandidatenmenge nur zusammen auftreten, handelt es sich um serielle, abhängige Ereignisse, die entsprechend Gleichung 4.7 variiert werden können.

Neben der Berücksichtigung serieller variabler Strukturen ermöglicht die Modellierung zudem eine alternative Darstellung von *1-aus-n* Entscheidungen. Entsprechend des im vorherigen Abschnitt dargestellten Ansatzes würde diese lokale Entscheidung im Zuverlässigkeitsblockdiagramm durch eine parallele Anordnung der Komponenten und eine Nebenbedingung der Form $\sum_i^{n_p} x_j = 1$ modelliert werden. In Abhängigkeit der Anzahl n_p der zur Verfügung stehenden Alternativen ergibt sich somit eine signifikante Erhöhung der Menge der Minimalpfade um $|MP| = n_p \cdot |MP_p|$. Der oben dargestellte Ansatz ermöglicht es jedoch, *1-aus-n*-Entscheidungen durch eine serielle Struktur und die vorherige Nebenbedingung abzubilden und somit eine Reduktion der Minimalpfade im Vergleich zum vorherigen Ansatz. Die reduzierte Anzahl der Minimalpfade führt wiederum zu einer exponentiell vereinfachten Ermittlung der orthogonalisierten Minimalpfade und einer vereinfachten Auswertung der Systemgleichungen im Optimierungsprozess.

Für das vollständige MRS des Beispielsystem ergeben sich die folgenden orthogonalen Minimalpfade:

$$MP_1 = K_1 \wedge K_2 \wedge K_3 \tag{4.16}$$
$$MP_2 = K_1 \wedge K_2 \wedge K_4 \wedge (\overline{K_3}) \; . \tag{4.17}$$

Wird das Ereignis 2 bei der Architekturvariation nicht berücksichtigt, ergeben sich die folgenden orthogonalen Minimalpfade mit dem vorgestellten Ansatz:

$$MP_1 = K_1 \wedge K_3 \tag{4.18}$$
$$MP_2 = K_1 \wedge K_4 \wedge (\overline{K_3}) \; . \tag{4.19}$$

Ohne eine Berücksichtigung der seriellen Struktur des Ereignisses K_2 in den Minimalpfaden hätte die Variation der Zuverlässigkeit nach Gleichung 4.7 zu einer Streichung sämtlicher Minimalpfade geführt.

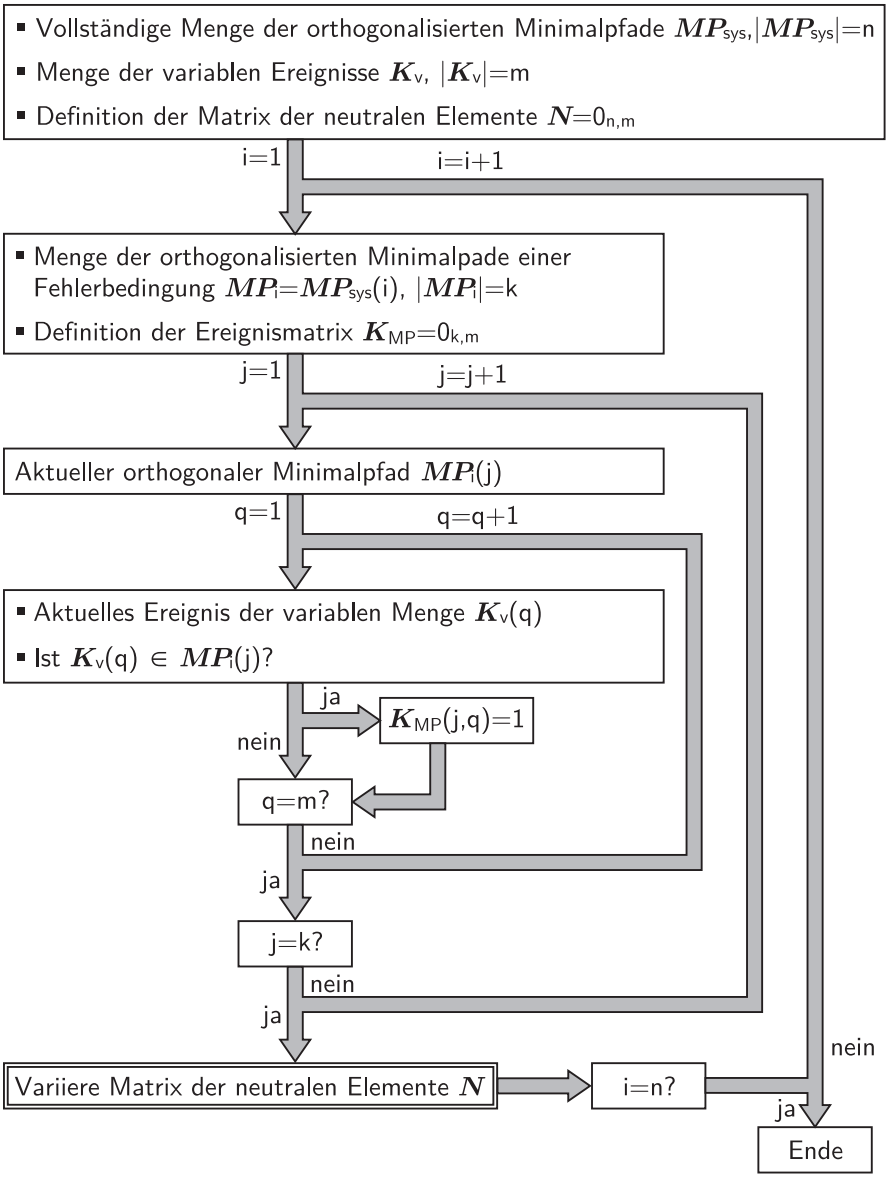


Abb. 4.6: Algorithmus zur Identifikation variabler serieller Logiken

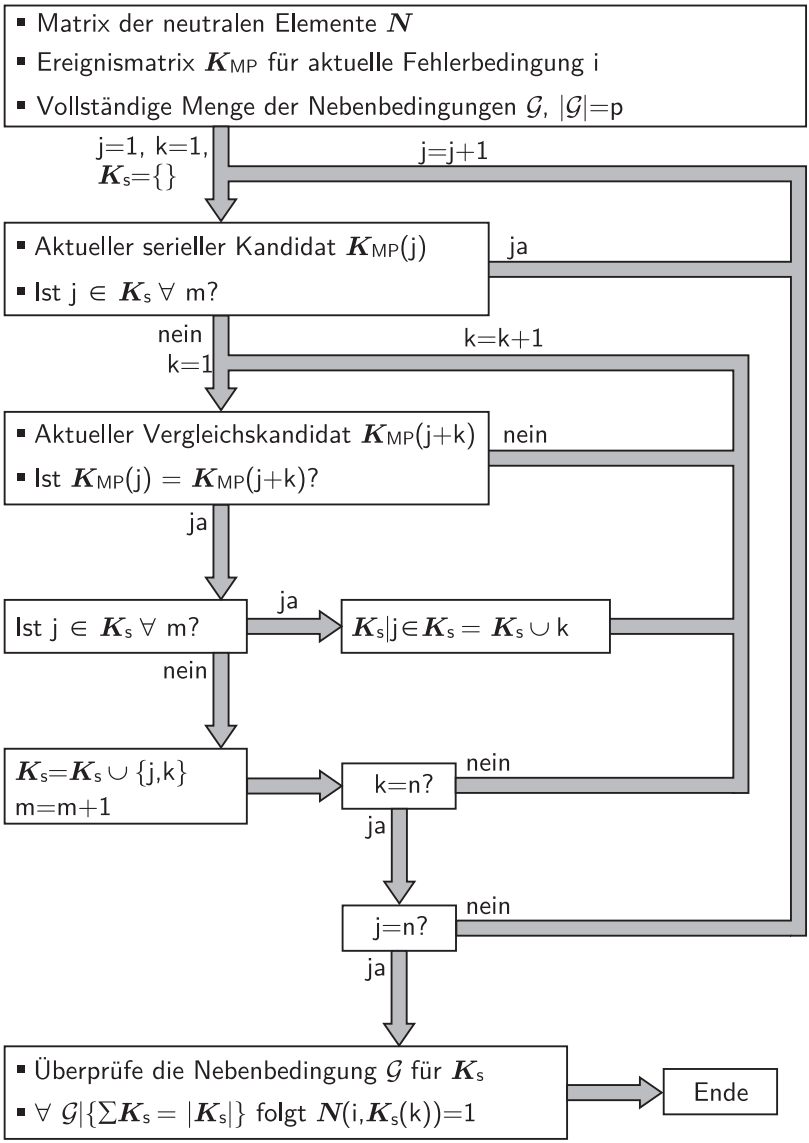


Abb. 4.7: Subalgorithmus zur Variation der Matrix neutraler Elemente

4.3.2 Serielle Strukturen von hybriden Systemmodellen

Entsprechend der vorherigen Betrachtungen zur Optimierung serieller Strukturen von Zuverlässigkeitsblockdiagrammen muss auch die Optimierung serieller Strukturen bei der Verwendung des hybriden Systemmodells variiert werden. Serielle Ereignisse zeichnen sich in den Zustandsgleichungen dadurch aus, dass diese im Term der verbleibenden, funktionsfähigen Komponenten stets zusammen auftreten. Verglichen mit dem vorherigen Verfahren für parallele variable Ereignisse ist in diesem Fall keine Verschiebung des Zustandsraumes notwendig, sondern eine Begrenzung des Zustandsraums auf die gültigen Zustände. Das nachträgliche Entfernen von Systemzuständen kann dabei wie folgt über die Parametervariation der Zuverlässigkeitswerte gesteuert werden:

$$R_i = \begin{cases} e^{-\lambda_i \cdot t} & \forall \quad x_i = 1 \\ 1 & \forall \quad x_i = 0 \end{cases} , \tag{4.20}$$

$$\lambda_i = \begin{cases} \lambda_i & \forall \quad x_i = 1 \\ 0 & \forall \quad x_i = 0 \end{cases} . \tag{4.21}$$

Somit ergibt sich die Zustandswahrscheinlichkeit der eliminierten Zustände zu null und der Zustandsbaum entsprechend Abbildung 4.4 wird auf die zulässigen Systemzustände beschränkt. Für das zuvor betrachtete seriell-parallele Beispiel ergeben sich die folgenden Zustandswahrscheinlichkeiten:

$$\begin{aligned} P_0 &= R_1 \cdot R_2 \cdot R_3 \cdot R_4 \\ P_3 &= \frac{\lambda_3}{\lambda_3} \cdot (1 - R_3) \cdot R_1 R_2 R_4 \\ P_4 &= \frac{\lambda_4}{\lambda_4} \cdot (1 - R_4) \cdot R_1 R_2 R_3 \, . \end{aligned} \tag{4.22}$$

Analog zu der vorherigen Analyse der Zuverlässigkeitsblockdiagramme wird anhand des Terms $R_1 R_2$ in allen Zustandsgleichungen automatisch die serielle Struktur erkannt. Da keine Nebenbedingung vorliegt, die die gemeinsame Verwendung der beiden variablen Ereignisse diktiert, handelt es sich um eine serielle variable Struktur, wobei die variablen Ereignisse gemäß Gleichung

4.21 zu variieren sind. Somit ergeben sich für den beschränkten Zustandsbaum durch Streichung des Ereignisses 2 die folgenden Zustandswahrscheinlichkeiten:

$$\begin{aligned} P_0 &= R_1 \cdot R_3 \cdot R_4 \\ P_3 &= \frac{\lambda_3}{\lambda_3} \cdot (1 - R_3) \cdot R_1 R_4 \\ P_4 &= \frac{\lambda_4}{\lambda_4} \cdot (1 - R_4) \cdot R_1 R_3 \; . \end{aligned} \tag{4.23}$$

4.4 Ableitung degradierter Systemzustände

Die vorgestellten Modellierungsansätze mit Hilfe von Zuverlässigkeitsblockdiagrammen und dem erweiterten hybriden Systemmodell basieren auf einer vorab definierten Fehlerbedingung, die gemäß der Systemanforderungen zu analysieren ist. Der Systementwurf fehlertoleranter Flugzeugsysteme wird dabei nicht ausschließlich durch den nominalen, fehlerfreien Systemzustand getrieben, sondern teilweise auch durch degradierte Systemzustände, für die ebenfalls die quantitativen Sicherheitsziele erfüllt werden müssen [4, 66]. Unter Beachtung der konträren Ziele einer hohen Sicherheit und einer operationellen Zuverlässigkeit bietet die Definition der *Master Minimum Equipment List* die Möglichkeit mit zulässigen Komponentenausfällen den Betrieb mindestens zeitlich begrenzt fortzusetzen, sofern die Zertifierungsziele auch mit dem degradierten System erfüllt werden [4]. Die *MMEL* bietet somit die Möglichkeit zur Erhöhung der operationellen Zuverlässigkeit unter Gewährleistung der quantitativen Sicherheitsvorschriften. Erste Betrachtungen hierfür müssen somit auch in die frühe Systementwicklung und somit in die Systemarchitektur einfließen [4, 9, 66].

Im Rahmen der quantitativen Zuverlässigkeits- und Sicherheitsbewertung ist somit eine Überprüfung von Konzepten hinsichtlich degradierter Systemzustände unerlässlich. Die benötigten Analysemodelle hierfür können aus den nominellen Modellen unter Berücksichtigung vorheriger Komponentenausfälle und Ereignisse abgeleitet werden. Die Ableitung der degradierten Modelle anhand des erstellten MRS unterteilt sich in die Derivation der Zuverlässigkeitsblockdiagramme und der hybriden Systemmodelle. In beiden Fällen wird die Degradation der Ereignisse $\mathbf{K}_d$ einer Zielfunktion i durch die folgende Syntax adressiert:

$$\text{Zielfunktion: } i, \; \mathbf{K}_d \in \mathbf{K} \tag{4.24}$$

Die Ableitung von Zielfunktionen degradierter Systemzustände ist somit nicht ausschließlich für die variablen Ereignisse möglich, sondern auch für die festen Ereignisse des Systemmodells. Dieses ermöglicht beispielsweise bei einer festen Generatorstruktur eines elektrischen Energieversorgungssystems, die Überprüfung inwieweit das variable Energieverteilungssystem den Ausfall eines Generators kompensieren kann.

Für die Modellierung unter Verwendung von Zuverlässigkeitsblockdiagrammen bedeutet die automatische Ableitung eine Streichung derjenigen orthogonalisierten Minimalpfade, die das betroffene Ereignis und somit die ausgefallene Komponente zu Beginn der Flugmission enthalten. Entsprechend des vorherigen Ansatzes zur Variation der Systemstrukturfunktionen in Abhängigkeit der betrachteten Systemarchitektur und der parallelen bzw. seriellen Verschaltung der Ereignisse kann die neue Zielfunktion durch eine Variation der Ausfallwahrscheinlichkeit abgeleitet werden.

Im Falle einer hybriden Modellierung des Systemverhaltens ist wie zuvor eine Transformation der ermittelten Zustandsgleichungen notwendig. Das nominelle Gleichungssystem basiert auf einem Initialzustand, der vom degradierten Systemzustand bei einem monotonen Fehlerverhalten nicht erreicht werden kann. Somit stellt ein degradierter Systemzustand des Gleichungssystems für das nominelle Systemverhalten den Initialzustand des neuen abgeleiteten Systemmodells dar. Die Verbindung von Gleichung 3.19 ergibt mit der Berechnung des degradierten Systemzustands für das Beispielsystem die folgende degradierte Zustandsmatrix, dabei wird das Ereignis K_2 als degradiert angenommen und über die Syntax $\{1, 2\}$ adressiert [96]:

$$\mathbf{P} = \left[\begin{array}{c|c} P_0 & P_3 \\ 0 & P_4 \\ 0 & P_5 \end{array}\right] \quad . \tag{4.25}$$

Mit Hilfe dieser Transformationen ist eine Verschiebung des Initialzustands des Analysemodells möglich und somit eine Bewertung von sicherheitsrelevanten degradierten Systemzuständen anhand des hybriden Systemmodells.

Die ermittelten strukturvariablen Funktionen der Zuverlässigkeitsblockdiagramme und des hybriden Systemmodells werden nachfolgend als Zielfunktionen in den Optimierungsprozess übernommen. Sie stellen somit im weiteren Prozess entsprechend der Abschnitte 4.2 und 4.3 eigenständige Zielfunktionen

dar, die ausgehend von ihrem Initialzustand in Abhängigkeit der betrachteten Architektur, beschrieben durch den Vektor $\mathbf{x}$, variiert werden.

4.5 Berücksichtigung summativer Zielgrößen

In den Kapiteln 1 und 3 wurde anhand unterschiedlicher Analyse- und Optimierungsumgebungen für die Konzeptionierung von Flugzeugsystemen und weiterer allgemein industrieller Systeme der mehrkriterielle Charakter der Konzeptbewertung aufgezeigt. Für den Vorentwurf komplexer Systeme ist neben der Bewertung der Architekturen auch die Einschätzung der Auswirkungen von Änderungen einiger Systemcharakteristika auf den weiteren Parameterraum wichtig, vergleiche Abbildung 1.1 [59, 80]. Im Rahmen der Redundanzallokation werden aus diesem Grund für viele Optimierungen ergänzende, häufig konträre Zielwerte mitgeführt, beispielsweise die Abschätzung der Systemmasse zur Quantifizierung der Kosten des Redundanzkonzeptes.

Im Rahmen dieser Arbeit sollen beliebige sich ergänzende und konträre aber auch übereinstimmende Zielgrößen neben der Sicherheits- und Zuverlässigkeitsbewertung berücksichtigt werden. Die Zielgrößen müssen sich hierfür als Systemgröße in der folgenden Form abschätzen lassen:

$$\delta_i = \underbrace{\sum_{j}^{k} x_j \cdot k_j}_{\forall K_j \in \mathbf{K}_v} + \underbrace{\sum_{n}^{m} k_n}_{\forall K_n \in \mathbf{K}_f} \quad \text{mit} \quad k \in \mathbb{R}. \tag{4.26}$$

Dabei beschreibt k allgemein die Kosten für die Berücksichtigung eines Ereignisses. Hiermit sind explizit nicht nur monetäre Kosten gemeint, sondern im Sinne der Optimierungstheorie jede Art des Aufwands zur Berücksichtigung eines Ereignisses.

Die lineare und stetige Struktur zur Abschätzung dieser weiteren Systemparameter ermöglicht neben der Berücksichtigung weiterer Architekturcharakteristika zudem die systematische Durchsuchung des Architekturraumes und somit der Ermittlung der nicht-dominierten Architekturmenge. Die Auswahl und Konditionierung dieser Verfahren folgt in dem nächsten Kapitel.

5 Optimale Redundanzallokation variabler Strukturen

Nachdem in den vorherigen Abschnitten sowohl das Analysemodell als auch das Einbringen von Freiheitsgraden erläutert wurden, folgt in diesem Kapitel die Auswahl geeigneter, problemspezifischer Verfahren zur Auswertung des Systemmodells mit variabler Struktur. Hierfür wird die Redundanzallokation als mehrkriterielles Optimierungsproblem formuliert. Die Auswahl geeigneter Optimierungsverfahren wird zum einen durch die mathematischen Eigenschaften des Optimierungsproblems definiert, zum anderen durch die zur Verfügung stehenden und geeigneten Optimierungsalgorithmen beschränkt [8]. Aus diesem Grund wird im Folgenden zunächst das vorliegende diskrete Optimierungsproblem klassifiziert, im Anschluss folgt die Vorstellung zur Verfügung stehender Algorithmen zur Lösung des Problems. Aus der Menge der verfügbaren Optimierungsalgorithmen werden drei Verfahren ausgewählt: eine vollständige Enumeration, ein mehrkriterielles *Branch & Bound* Verfahren und ein Genetischer Algorithmus. Diese werden für die Redundanzallokation angepasst und die Auswahl anhand eines erweiterbaren Beispielproblems mit Hilfe der Dauer der Optimierungsläufe validiert.

5.1 Formulierung und Klassifizierung des Optimierungsproblems

Zur Auswahl geeigneter Optimierungsverfahren ist es zunächst notwendig, die entscheidenden Charakteristika des vorliegenden Optimierungsproblems zu identifizieren. Bei dem vorliegenden Optimierungsproblem handelt es sich um ein mehrkriterielles, diskretes Optimierungsproblem der Zielwertmenge $\mathcal{A}(\mathbf{x}, t)$, die sich aus den Zuverlässigkeitswerten $\mathcal{R}_i(\mathbf{x}, t)$ und $\mathcal{R}_{d,i}(\mathbf{x}, t)$ sowie den summativen Zielwerten $\mathcal{S}_i(\mathbf{x})$ ergibt [18, 28]:

$$
\begin{aligned}
\text{Optimiere} \quad \mathcal{A}(\mathbf{x},t) &= (\mathcal{R}_1(\mathbf{x},t),\ldots,\mathcal{R}_m(\mathbf{x},t),\ldots \\
&\quad \mathcal{R}_{d,1}(\mathbf{x},t),\ldots,\mathcal{R}_{d,n}(\mathbf{x},t),\ldots \\
&\quad \ldots,\mathcal{S}_1(\mathbf{x}),\ldots,\mathcal{S}_l(\mathbf{x})) && (5.1)\\
\text{mit} \quad \mathbf{x} &= [x_1 \ldots x_p]^T \text{ und } \mathbf{x} \in \{0,1\}^{|\mathbf{K}_v|} && (5.2)\\
\text{unter Berücksichtigung von} & \\
g_i(\mathbf{x}) &\leq 0 \quad \text{mit} \quad i = 1,\ldots,k && (5.3)\\
\text{mit} \quad g_i(\mathbf{x}) &\in \mathcal{G}\,. && (5.4)
\end{aligned}
$$

Wobei es sich bei der Menge der Nebenbedingungen $\mathcal{G}$ um explizite algebraische Funktionen handelt, die den theoretisch möglichen Architekturraum der Größe $2^{|\mathbf{K}_v|}$ beschränken. Die Definition der Nebenbedingungen wurde bereits in Abschnitt 4.1 zur Beschränkung des binären Entscheidungsbaums vorgestellt. Bei dem Entscheidungsraum handelt es sich aufgrund der diskreten Variablen ebenfalls um einen diskreten Architekturraum, analog zum resultierenden Zielwertraum [18]. Die Menge der Zielfunktionen $\mathcal{A}(\mathbf{x},t)$ ergibt sich dabei aus den Systemfunktionen des mehrfach-redundanten Systemmodells gemäß Abschnitt 4.2 und 4.3, den abgeleiteten Systemfunktionen gemäß Abschnitt 4.4 und der Abschätzung weiterer Systemparameter. Entsprechend Abschnitt 4.5 handelt es sich bei den Parametergleichungen zur Abschätzung der konträren Zielwerte um lineare Zielfunktionen. Im Gegensatz dazu handelt es sich bei den Zielfunktionen, die aus dem hybriden Systemmodell ermittelt werden, um nichtlineare, unstetige und nicht-monoton steigende Funktionen, bezogen auf den Architekturvektor $\mathbf{x}$.

Die unterschiedlichen Zielfunktionen der Menge $\mathcal{A}(\mathbf{x},t)$ verhalten sich dabei zum Teil konträr zueinander. So führen beispielsweise mehr Redundanzen in einem System zu einer erhöhten Funktionswahrscheinlichkeit, jedoch auch zu einer höheren Systemmasse [96, 97]. Ebenfalls kann eine erhöhte Sicherheit zu einer geringeren operationellen Zuverlässigkeit führen, da für den Betrieb des Flugzeuges mehr funktionsfähige Komponenten notwendig sind. Die weitere Korrelation zwischen der Systemsicherheit und der operationellen Zuverlässigkeit wurde bereits in Abschnitt 2.1 erläutert. Aufgrund des gegenläufigen Verhaltens der Zielgrößen wird bei der Klassifizierung des vorliegenden Problems allgemein die Optimierung anstatt der Minimierung oder Maximierung betrachtet, zwischen diesen beiden Zielwertbetrachtungen ist eine Variation jedoch durch Änderung der Vorzeichen der Zielwerte möglich. Der nächste Ab-

schnitt betrachtet zur Lösung dieses kombinatorischen, mehrkriteriellen Optimierungsproblems mit gegensätzlichen Zielgrößen unterschiedliche Verfahren und wählt geeignete Verfahren anhand der Anforderungen an die Optimierung aus.

5.2 Diskussion der Optimierungsverfahren

Nach dem *No-Free-Lunch* Theorem von Wolpert et al. ist die Leistungsfähigkeit aller Optimierungsverfahren über die Menge sämtlicher bestehender Optimierungsprobleme gleich [129]. Das *No-Free-Lunch* Theorem drückt sich dabei auch in den konkurrierenden Kriterien *Universalität*, *Robustheit*, *Konvergenzsicherheit* und *Konvergenzgeschwindigkeit* für die Auswahl von Optimierungsverfahren aus [8]. Somit ist für eine leistungsfähige Optimierung eine problemspezifische Auswahl und eventuelle Anpassung eines Optimierungsverfahrens unabdingbar. In Abbildung 5.1 ist eine Klassifizierung gängiger globaler Optimierungsmethoden für diskrete Optimierungsprobleme dargestellt [8].

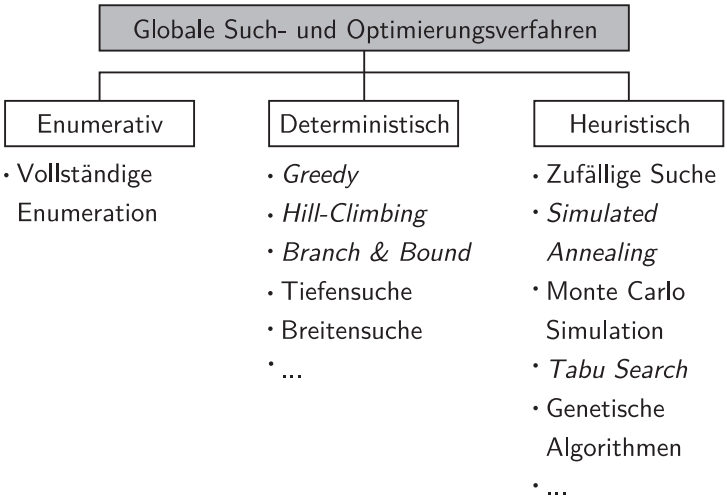


Abb. 5.1: Klassifizierung gängiger Such- und Optimierungsverfahren, nach [8]

Mit der vorherigen Klassifizierung des Optimierungsproblems und den Anforderungen an die Optimierung entsprechend Abschnitt 3.3 lassen sich geeignete Verfahren zur Ermittlung optimaler Lösungen auswählen. Die Menge der verfügbaren Optimierungsverfahren lässt sich dabei neben der Charakteristik der hinterlegten Lösungsverfahren entsprechend Abbildung 5.1 auch durch die ermittelte Lösungsmenge einteilen. Die enumerativen und Teile der deterministischen Verfahren ermitteln dabei das globale Optimum und werden somit auch als *optimale/exakte Verfahren* bezeichnet. Im Gegensatz dazu finden heuristische Methoden unterschiedlicher Ansätze und einige der Determinismen in vielen Fällen nur eine Annäherung des globalen Optimums oder bleiben in lokalen Optima hängen. Diese Verfahren werden somit auch als *suboptimale Verfahren* bezeichnet. In Abhängigkeit der geforderten Ergebnisgüte sowie der verfügbaren Rechenzeit und -leistung entsprechend der Anforderungen in Abschnitt 3.3 sind geeignete Verfahren auszuwählen [113].

Die Ermittlung der optimalen Lösungsmenge erfordert bei mehrkriteriellen Optimierungsproblemen zunächst eine Definition der gesuchten Lösungsmenge. Während sich einkriterielle Optimierungsprobleme als reine Minimierung oder Maximierung des betrachteten Optimierungskriteriums durchführen lassen, ist die Definition des globalen Optimums bei mehrkriteriellen Problemen mit konkurrierenden Zielwerten nicht ohne weitere Informationen und Entscheidungen des Anwenders möglich. Hierbei wird zwischen drei prinzipiell unterschiedlichen Ansätzen unterschieden [19].

Bei der *a-priori* Definition fließen zu Beginn der Optimierung Kenntnisse oder Präferenzen des Anwenders in die Definition der Zielwertfunktionen ein, beispielsweise durch eine gewichtete Summenfunktion. Für den Fall der betrachteten Redundanzallokation würde sich mit Hilfe der Definition des Optimierungsproblems als Zielfunktion beispielsweise ergeben:

$$\begin{aligned} \mathcal{A}(\mathbf{x},t) \quad &= \quad \sum_i a_i \mathcal{R}_i(\mathbf{x},t) + \sum_j a_j \mathcal{R}_{d,j}(\mathbf{x},t) + \sum_k a_k \mathcal{S}_k(\mathbf{x}) \qquad (5.5) \\ &\text{mit} \qquad \sum_n a_n = 1 \; . \end{aligned}$$

Hierdurch werden vom Optimierungsalgorithmus nicht mehr die einzelnen Zielwerte betrachtet, sondern das Ergebnis der Summe aller Zielwerte, wobei diese zuvor durch einen Parameter gewichtet werden können. Das mehrkriterielle Optimierungsproblem wird durch die a-priori Definition mathematisch auf ein

einkriterielles Problem reduziert. In der Regel wird somit als Optimierungsergebnis auch ausschließlich eine singuläre Architektur ausgegeben. Im Gegensatz wird bei *a-posteriori* Verfahren das Optimierungsergebnis zunächst nicht durch den Anwender beeinflußt. Stattdessen ermittelt der verwendete Algorithmus eine zu spezifizierende Lösungsmenge und der Anwender entscheidet im Nachhinein anhand der ermittelten Zielwerte über die Präferenzen. Zwischen den beiden vorgestellten Verfahren liegen die *progressiven* Optimierungsverfahren, hierbei wird wiederholt eine Lösungsmenge ermittelt und der Anwender entscheidet über die bevorzugten Architekturen. Anhand dieser Architekturen wird eine Initialmenge für die nachfolgende Optimierung gebildet und der Prozess mehrfach durchlaufen, bis der Anwender mit dem Optimierungsergebnis zufrieden ist.

Um eine hohe Transparenz in dem Optimierungsprozess komplexer Systemstrukturen zu gewährleisten, wird für die Auswahl der Verfahren eine *a-posteriori* Strategie verfolgt. Diese ermöglicht entsprechend der gestellten Kernfragen für diese Arbeit und der Anforderungen an die spätere Implementierung eine nachfolgende Entscheidungsfindung anhand der ermittelten Zielwerte und somit eine Diskussion im Entwicklungsprozess auf Grundlage verfügbarer, quantitativer Informationen. Im Gegensatz dazu müssen für die *a-priori* Strategien stark unterschiedliche und konkurriende Zielgrößen gewichtet werden, was aufgrund des interdisziplinären Charakters für die Optimierung im Vorentwurf nur bedingt möglich ist. Zudem betrachtet diese Strategie nicht ohne weitere Nebenbedingungen an die Zielwerte die Erfüllung von Sicherheits- und Zuverlässigkeitsanforderungen, so dass das ausgegebene singuläre Optimum gegen diese Anforderungen verstoßen könnte. Die *progressiven* Verfahren erfordern einen wiederholten Eingriff des Anwenders in den Optimierungsprozess, was aufgrund der zahlreichen Systemgleichungen einen hohen Arbeitsaufwand erfordert, um anhand suboptimaler Lösungen geeignete Konzepte für eine weitere Verbesserung auszuwählen. Sofern die Systementwicklung einen wiederholten Eingriff in die Optimierung erfordert, lässt eine Implementierung eines *a-posteriori* Verfahrens auch stückweise eine *progressive* Optimierung zu. Hierfür wird die ermittelte Lösungsmenge zu einer möglichen Initialmenge einer Heuristik reduziert und somit wird für wiederholte Optimierungsläufe ein Einfluss des Anwenders ermöglicht. Als Kompromiss zwischen der *a-priori* und der *a-posteriori* Optimierung wird daher als Lösungsmenge die Pareto-optimale Menge nicht dominierter Architekturen ermittelt, die nachfolgend definiert ist [18].

Definition PARETO**-optimal**
Eine Lösung $\mathbf{x} \in \mathbf{X}$ heißt PARETO-optimal bezüglich des Lösungsraums $\mathcal{B}$ und für $t = konst.$, sofern kein $\mathbf{x}^\star$ existiert für das gilt: $\mathcal{B}(\mathbf{x}, t) \preceq \mathcal{B}(\mathbf{x}^\star, t)$, so dass die Lösung $\mathbf{x}^\star$ in keiner Zielgröße besser als $\mathbf{x}$ ist, ohne in mindestens einem weiteren Kriterium schlechter zu sein, d.h. die Lösung $\mathbf{x}$ wird nicht von der Lösung $\mathbf{x}^\star$ dominiert.

Abbildung 5.2 veranschaulicht das Konzept der Dominanz und PARETO-Optimalität für einen zweidimensionalen Zielwertraum eines diskreten Optimierungsproblems. Die hervorgehobenen Lösungen gehören zu der nicht-dominierten Menge und dominieren jeweils selbst die Lösungen in dem Dominanzbereich. Für die Ermittlung der PARETO-Menge werden gemäß der oben genannten Definition identische Werte aller Zielfunktionen innerhalb der nicht-dominierten Menge zugelassen. Im Gegensatz hierzu steht die strenge PARETO-Optimalität, die identische Zielwerte ausschließt, jedoch zu einem schlechteren Konvergenzverhalten der Optimierung führen kann [18].

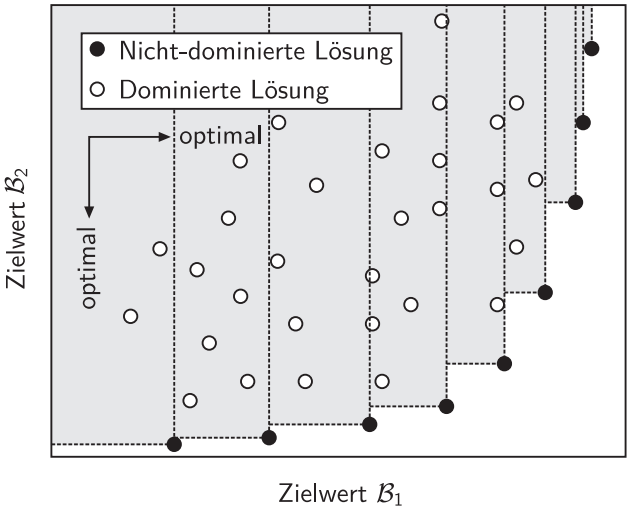


Abb. 5.2: Veranschaulichung der nicht-dominierten Menge an einem zweidimensionalen Lösungsraum

Aufgrund der Problemcharakteristik und der NP-Vollständigkeit der Redundanzallokation wurden für die Ermittlung der PARETO-optimalen Menge drei

Verfahren ausgewählt, die für unterschiedliche große Optimierungsprobleme geeignet sind: eine vollständige Enumeration mittels Breitensuche, ein *Branch & Bound* Verfahren und ein Genetischer Algorithmus. Dabei finden die beiden ersten Verfahren die exakte nichtdominierte Menge, während der heuristische Genetische Algorithmus in Abhängigkeit von Konditionierung und Laufzeit nur einen Teil der globalen Lösungsmenge finden kann. In den folgenden Abschnitten werden die Entscheidungen begründet, die Verfahren vorgestellt und die Auswahl anhand eines erweiterbaren Beispielsystems validiert. Die Unterstützung zur Auswahl geeigneter Lösungen und somit die Integration der Verfahren in den Entwicklungs- und Entscheidungsprozess erfolgt in Kapitel 6.

5.2.1 Vollständige Enumeration

Die Anwendung eines Optimierungsverfahrens auf ein kombinatorisches Optimierungsproblem erfordert aufgrund des unstetigen Lösungsraums und fehlender analytischer Verfahren stets verfahrensspezifische Schritte zur gezielten Durchsuchung des Architekturraumes. Für kleine bis mittlere variable Ereignismengen $\mathbf{K}_v$ liegt daher die Überlegung nahe, sämtliche gültigen Architekturen zu untersuchen und anschließend die PARETO-optimale Menge zu bestimmen. Somit wird die verfügbare Rechenleistung und -zeit für eine vollständige Untersuchung des Architektur- und Zielwertraumes genutzt, die dem Anwender einen vollständigen Überblick verschafft, anstatt sie für Nebenrechnungen eines Algorithmus zu nutzen.

Entsprechend wird als erstes Verfahren eine vollständige Enumeration des gültigen Architekturraumes mit anschließender Ermittlung der tatsächlichen PARETO-Menge PF_{real} betrachtet. Die vollständige Enumeration ist nur bedingt ein Optimierungsverfahren, sondern eher ein triviales, explizites Lösungsverfahren, das alle Lösungen zunächst ohne Berücksichtigung der Zielwerte berechnet [41]. In diesem Fall wird als vollständige Enumeration die Ermittlung des vollständigen gültigen und somit beschränkten Architekturraumes verstanden.

Zunächst wird der gültige Architekturraum aufgestellt, hierfür kann der vollständige binäre Entscheidungsbaum mit $2^{|\mathbf{K}_v|}$ Lösungen entweder zunächst in der Tiefe oder Breite durchsucht und überprüft werden [85]. In Abbildung 5.3 ist der Vergleich der beiden Suchverfahren für unterschiedliche Probleme mit variablen Ereignismengen $\mathbf{K}_v$ anhand der notwendigen Auswertezeit t dargestellt. Bei der Tiefensuche wird eine Architektur zunächst durch seine Bit-

Repräsentation vollständig definiert und anschließend mit Hilfe der Nebenbedingungen überprüft. Sollte die Architektur ungültig sein, wird die Verzweigung des Entscheidungsbaumes verworfen und es wird mit der nächsten Architektur fortgefahren. Gültige Architekturen werden hingegen mit Hilfe der Zielfunktionen ausgewertet. Die Breitensuche durchsucht den binären Entscheidungsbaum hingegen erst in der Breite, so dass auf jeder Entscheidungsebene die Nebenbedingungen überprüft werden. Sofern eine Aussage zu einem Bit im Architekturvektor noch nicht möglich ist oder alle Nebenbedingungen erfüllt sind, wird die Verzweigung weiter geführt. Ansonsten wird die weitere Verzweigung abgebrochen und alle nachfolgenden Architekturen werden verworfen [20]. Aufgrund der

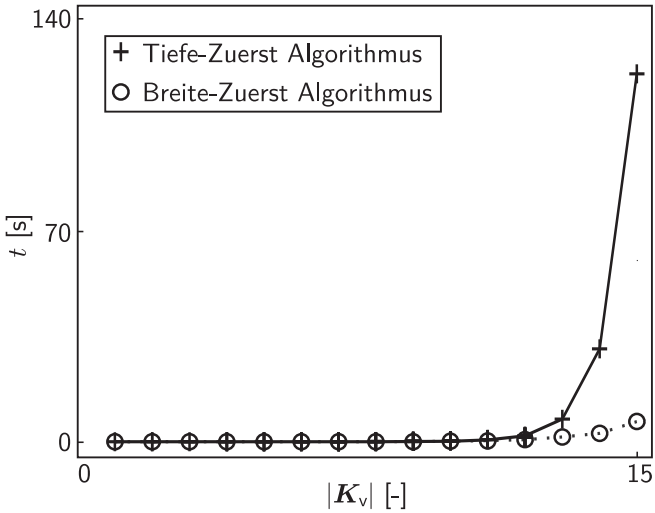


Abb. 5.3: Vergleich zwischen Tiefen- und Breitensuche bei der vollständigen Enumeration

frühzeitigen Begrenzung des Entscheidungsbaumes und der dadurch bedingten partiellen Auswertung des Entscheidungsbaumes zeigt sich deutlich, dass die Breitensuche gegenüber der Tiefensuche weit überlegen ist. Gerade bei stark beschränkten Optimierungsproblemen bricht die Breitensuche die Verzweigung zahlreicher Äste des Entscheidungsdiagramms frühzeitig ab, während die Tiefensuche alle 2^n Lösungen auf deren Gültigkeit überprüft. Dabei wirkt sich die Auswahl des Verfahrens nicht nur auf die reine Überprüfung möglicher Architekturen aus, sondern auch auf alle nachfolgenden Matrixoperationen. Die

Tiefensuche wird in diesem Fall somit nicht nur durch die benötigte Zeit zur Ermittlung des gültigen Architekturraums beschränkt, sondern in der Implementierung auch durch die Verfügbarkeit von Speicherressourcen. Für beide Verfahren lässt sich jedoch feststellen, dass mit steigender Anzahl der Nebenbedingungen die Zeit zur Ermittlung des gültigen Architekturraums abnimmt. Da die Nebenbedingungen den Lösungsraum nicht nur auf technisch sinnvolle Lösungen beschränken, sondern auch die Vorkenntnisse des Systemingenieurs in den Optimierungsprozess einfließen lassen, zeigt sich hier auch, dass nicht nur die Auswahl des Verfahrens für die Optimierungszeit entscheidend ist, sondern auch wesentlich die Berücksichtigung der Systemkenntnisse [18].

Die vollständige Analyse des zulässigen Zielwertraumes wird im Folgenden als Referenz für die weiteren Optimierungsverfahren und die Bewertung der Güte beziehungsweise Validität der Ergebnisse genutzt.

5.2.2 Branch & Bound Verfahren

Das vorgestellte Verfahren zur vollständigen Enumeration ermöglicht die Untersuchung des vollständigen Architekturraums für beliebige Zielfunktionen des mehrfach-redundanten Systemmodells. Die Aufstellung des gültigen Zustandsraums, die Evaluation aller Systemarchitekturen und die Ermittlung der nichtdominierten Architekturen hängen jedoch stark von der Größe des Architekturraums und der Beschränkung ab. Somit wird das vorgestellte Verfahren für große Probleme ineffizient. Aus diesem Grund wird die Ermittlung des gültigen Architekturraums mit Hilfe des vorgestellten Konzepts der Breitensuche um eine Funktion zur Beschränkung des Architekturraums anhand bereits ermittelter Zielwerte erweitert.

Das Ziel des nachfolgenden Verfahrens ist es, nicht den vollständigen Architekturraum zu untersuchen, jedoch trotzdem die tatsächliche und vollständige Pareto-Front PF_{real} zu ermitteln. Bei einkriteriellen kombinatorischen Optimierungsproblemen eignet sich hierfür das *Branch & Bound*[1] Verfahren. Das vorgestellte Verfahren basiert dabei auf den Grundlagen von Land und Doig, Erweiterungen von Nakagawa und McLeavy sowie spezifischen Erweiterungen zur Optimierung mehrkriterieller Probleme mit nicht-monoton steigenden Zielfunktionen [11, 63, 74, 81].

[1] Der Name leitet sich von den englischen Bezeichnungen der beiden Hauptschritte des Verfahrens ab, *Branch* für das Verzweigen des binären Entscheidungsbaums und *Bound* für das Begrenzen des Lösungsraumes anhand aktueller Zielwerte.

Das *Branch & Bound* Verfahren von LAND und DOIG wurde zur Untersuchung einkriterieller Optimierungsprobleme entwickelt. Die Überführung zu mehrkriteriellen Problemen wurde u.a. von SOURD ET AL. untersucht, daneben hat NAKAGAWA die Redundanzoptimierung seriell-paralleler Strukturen untersucht [62, 81, 118]. Im Folgenden wird auf Grundlage dieser Ergebnisse ein Verfahren entwickelt, um das vorliegende mehrkriterielle Optimierungsproblem zur Untersuchung komplexer Redundanzstrukturen zu lösen und wie zuvor die PARETO-Menge zu ermitteln.

Anhand des in Abbildung 5.4 dargestellten binären Entscheidungsbaums kann der allgemeine Ablauf des *Branch & Bound* Verfahrens verdeutlicht werden. Mit Hilfe des Architekturvektors $\mathbf{x}$ wird sukzessive eine gültige Architektur aufgebaut, es handelt sich hierbei um die Verzweigung (*Branching*). Die weitere Verzweigung des Entscheidungsbaumes hängt dabei von den erreichbaren Zielwerten der aktuellen, partiellen Lösung ab, wobei die Abschätzung mit Hilfe einer spezifischen Heuristik erfolgt. Sofern die abgeschätzten Zielwerte des Teilproblems von der aktuell besten Lösung dominiert werden, wird die weitere Verzweigung verworfen und ein neues Teilproblem wird aufgebaut, dieser Schritt wird *Bounding* genannt. Auf diese Weise kann eine vollständige Untersuchung des Architekturraums vermieden und trotzdem die Ermittlung der realen PARETO-Menge garantiert werden [23].

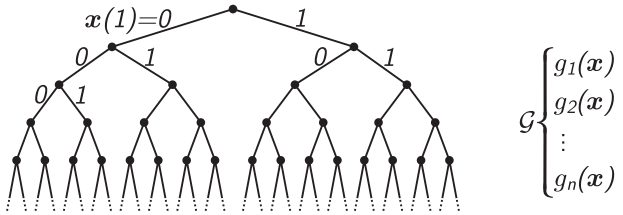


Abb. 5.4: Variation des binären Entscheidungsbaumes für das *Branch & Bound* Verfahren

Der binäre Entscheidungsbaum des üblichen *Branch & Bound* Verfahrens ist vor allem für unbeschränkte Optimierungsprobleme anwendbar. Die Redundanzallokation komplexer Flugzeugsysteme wird jedoch durch physikalische Korrelationen der Komponenten und gemeinsame Nebenbedingungen beschrieben, die zum Beispiel die Nutzung unterschiedlicher Generatoren beschreiben. In Abschnitt 4.1 wurde bereits die Abhängigkeit der lokalen Entscheidungsvaria-

blen x_i betrachtet. Aufgrund der lokalen Entscheidungspunkte im Systementwurf wird neben der Reduktion des Architekturraumes auch eine Gruppierung der Entscheidungsvariablen durch die Nebenbedingungen erreicht. Diese Gruppierung kann genutzt werden, um in jeder Ebene des binären Entscheidungsbaumes nicht nur über eine Variable zu entscheiden, sondern aus der Menge der möglichen Kombinationen von Variablen ein Subsystem für eine Entscheidungsaufgabe auszuwählen. In Abbildung 5.5 ist der entsprechende ganzzahlige Entscheidungsbaum dargestellt.

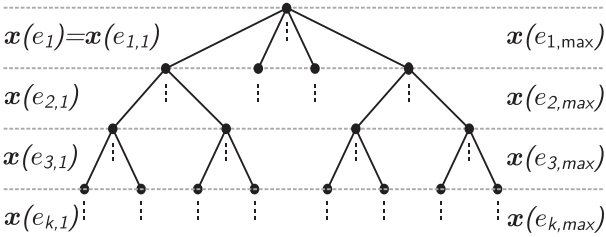


Abb. 5.5: Reduktion der Entscheidungsebenen für das *Branch & Bound* Verfahren

Die Anzahl der möglichen Subsystemvarianten $e_{i,max}$ einer Entscheidungsebene i ergibt sich dabei aus den Beschränkungen durch die relevanten Nebenbedingungen $\mathbf{g}(i) \in \mathcal{G}$. Somit wird in jeder Ebene mindestens über eine Variable x_i entschieden, maximal über die vollständige verbleibende variable Ereignismenge $\mathbf{K}_v$, sofern alle Variablen voneinander abhängig sind. Der Architekturvektor einer dedizierten Architektur kann mit Hilfe der einzelnen Entscheidungsebenen e_i, $i \in 1, k$ wie folgt beschrieben werden:

$$\mathbf{x} = [\mathbf{x}(e_1) \quad \mathbf{x}(e_2) \quad \dots \quad \mathbf{x}(e_k)] \quad . \tag{5.6}$$

Um anhand des Architekturvektors $\mathbf{x}$ sukzessive eine Architektur bewerten und zeitgleich weitere Architekturen ausschließen zu können, ist eine monoton steigende oder sinkende Zielfunktion notwendig [63]. Entsprechend der Definition des Optimierungsproblems wird für die Redundanzallokation eine Menge $\mathcal{R}_{(d)}(\mathbf{x}, t)$ nichtlinearer, nicht-monotoner Sicherheits- und Zuverlässigkeitsfunktionen sowie eine Menge $\mathcal{S}(\mathbf{x})$ monoton-steigender, summativer Zielfunktionen betrachtet. Als Suchraumvariable wird daher die erste summative Zielfunktion genutzt. Sollte keine summative Zielfunktion definiert sein, wird eine fiktive

Funktion mit der Eigenschaft $\mathcal{S}(\mathbf{x}) = 0 \ \forall \ \mathbf{x} \in \mathbf{X}$ definiert und anschließend aus dem Lösungsraum entfernt, wobei die Monotoniebedingung durch den konstanten Wert nicht verletzt wird. Die verbleibenden Zielfunktionen aus den Funktionsmengen $\mathcal{R}_{(d)}(\mathbf{x}, t)$ und $\mathcal{S}(\mathbf{x})$ werden im weiteren Prozess nur bei Bedarf betrachtet, auf der einen Seite zur Beschränkung des Zielwertraumes, auf der anderen Seite zur Architekturbewertung. Der vollständige Prozess zur Beschränkung und Bewertung ist in Abbildung 5.7 dargestellt und wird im Folgenden erläutert.

Branching: Zu Beginn des Verfahrens werden für die einzelnen Subsysteme aller Entscheidungsebenen e_i die erreichbaren Werte der Suchraumvariable $\mathcal{S}_s \in \mathcal{S}(\mathbf{x})$ berechnet. Anhand der minimal und maximal erreichbaren Zielwerte durch Summation der Werte jeder Ebene können eine globale obere Grenze UB und eine globale untere Grenze LB für die Suchraumvariable des Architekturraumes abgeschätzt werden. Die entsprechenden Architekturen werden abschließend in die Menge der Kandidaten nicht-dominierter Lösungen übernommen. Wobei die Architektur der unteren Grenze gemäß der Definition der Pareto-Menge eine nicht-dominierte Lösung unabhängig der weiteren Zielwerte darstellt, da sie die gültige Architektur mit der minimalen Masse besitzt und somit das Optimum bezüglich der Zielfunktion $\mathcal{S}(\mathbf{x})$. Generell ist jede Kombination der Subsysteme entsprechend der Nebenbedingungen zulässig und somit technisch sinnvoll, da diese bereits zur Erstellung des Entscheidungsbaumes herangezogen wurden.

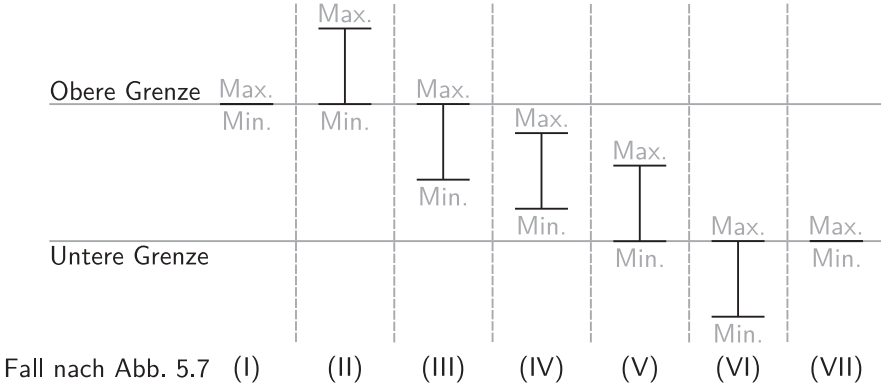


Abb. 5.6: Lage möglicher Zielwertintervalle zu der aktuellen oberen und unteren Grenze

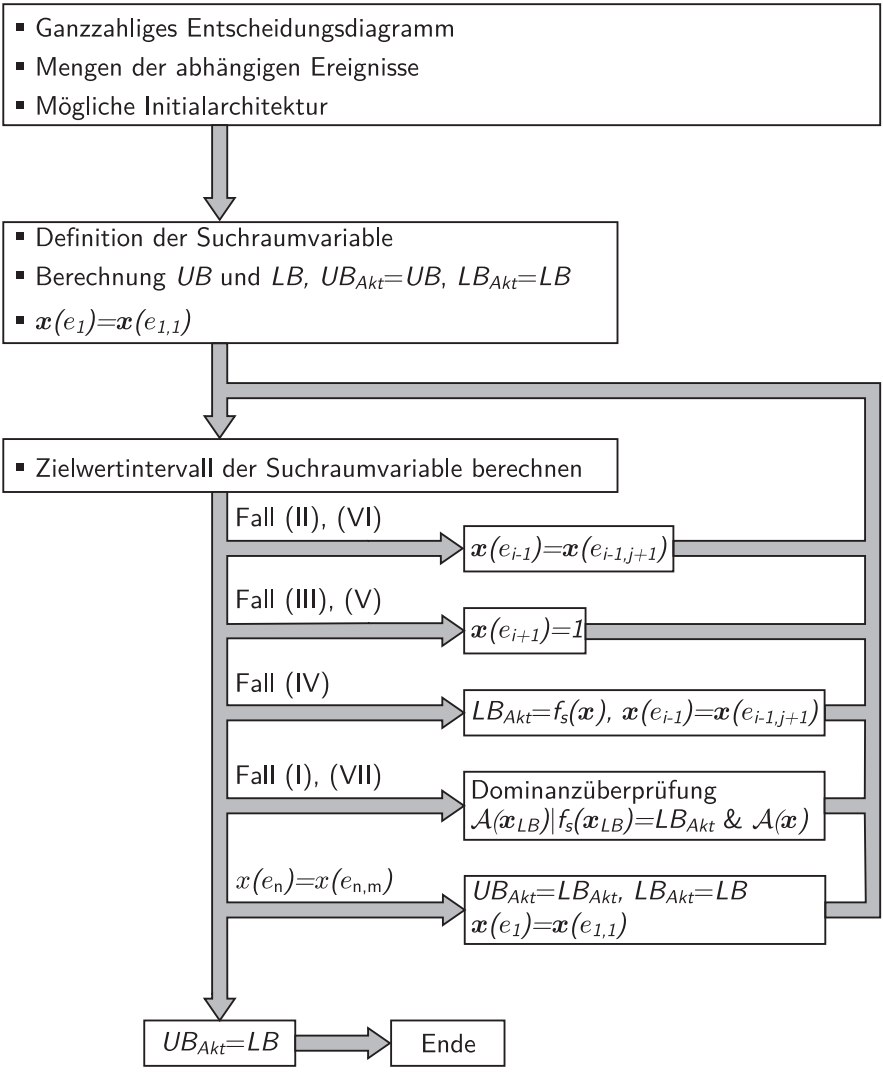


Abb. 5.7: Vollständig angepasster *Branch* & *Bound* Algorithmus für mehrkriterielle Redundanzallokationen

Begonnen bei der ersten Ebene des Entscheidungsbaumes werden sukzessive Teilarchitekturen ausgewählt, so dass eine neue aktuelle untere Schranke LB_{Akt} gefunden wird und somit der Architekturraum eingegrenzt wird. Das Verfahren durchsucht somit von der kleinsten Suchraumzielgröße kommend, den vollständigen Architekturraum. Architekturen, die nicht innerhalb der aktuellen Grenzen liegen, werden für die aktuelle Schleife verworfen. Die unterschiedlichen Szenarien bei der Abschätzung sind in Abbildung 5.6 dargestellt. Sofern die kleinstmögliche Distanz zwischen UB_{Akt} und LB_{Akt} gefunden wurde, wird die aktuelle Obergrenze der aktuellen Untergrenze gleichgesetzt und die aktuelle Untergrenze wird der globalen Untergrenze gleichgesetzt, die entsprechenden Grenzen sind in Abbildung 5.8 dargestellt. Der Algorithmus durchläuft mit diesen Grenzen wieder den Begrenzungsprozess (*Branching*) und terminiert, sofern die aktuelle Unter- und Obergrenze identische Werte annehmen.

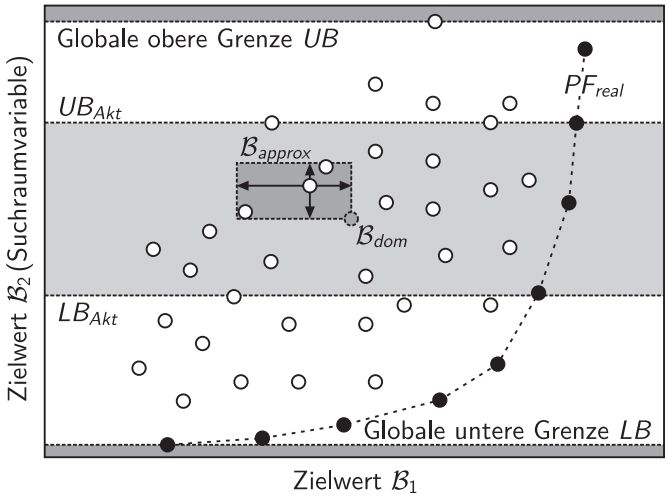


Abb. 5.8: Begrenzung des Zustandsraums und Relaxierung durch das *Branch & Bound* Verfahren

Bounding: Für den einkriteriellen Fall von Land und Doig reicht für die Abschätzung der erreichbaren Zielwerte ein einmaliges Überschreiten der aktuellen Obergrenze aus, um eine Lösung auszuschließen. Aufgrund der dort betrachteten monoton-steigenden Zielfunktionen ist diese Aussage zulässig [63]. Die Ermittlung der Pareto-Front anhand komplexer, nicht-monotoner Funktionen

lässt diesen Schluss nicht zu, weshalb im Folgenden ein alternatives Verfahren hergeleitet wird.

In Abbildung 5.8 ist neben der Begrenzung des Architekturraums die Abschätzung der Zielwerte der möglichen Architekturen der Entscheidungsebene e_i dargestellt, wobei es sich bei Zielwert 1 um eine unstetige und nicht-monotone Zielfunktion handelt. Die aktuelle PARETO-Front ist hervorgehoben und veranschaulicht, dass unabhängig von den nachfolgenden Architekturvariablen, jede mögliche Lösung dominiert wird und eine weitere Verzweigung des Entscheidungsbaumes somit nicht sinnvoll ist. Die Abschätzung der maximalen und minimalen Zielwerte für die diskreten, nicht-monotonen Zielfunktionen ist mit Hilfe der Erreichbarkeitsanalyse nichtlinearer, unstetiger Systeme möglich. Um die Ermittlung der vollständigen und tatsächlichen PARETO-Front zu garantieren, ist das Ziel der Erreichbarkeitsanalyse eine Überapproximation der erreichbaren Zielwerte [37]. Hierzu werden die minimal und maximal möglichen Zielwerte der Architektur mittels Relaxierung berechnet. Es werden somit die konjungtiven Nebenbedingungen und die resultierenden Architekturbeschränkungen vernachlässigt. Somit gilt für die beiden aktuellen Architekturvektoren $\mathbf{x}_{min}$ und $\mathbf{x}_{max}$ zwecks Relaxierung für die Entscheidungsebene e_k:

$$\mathbf{x}_{min} = \begin{cases} \mathbf{x}_{akt,i} & \forall \quad \{\mathbf{x}_i | \mathbf{x}_i \in \mathbf{x}_{akt}\} \\ 0 & \forall \quad \{\mathbf{x}_i | \mathbf{x}_i \notin \mathbf{x}_{akt}\} \end{cases} , \tag{5.7}$$

$$\mathbf{x}_{max} = \begin{cases} \mathbf{x}_{akt,i} & \forall \quad \{\mathbf{x}_i | \mathbf{x}_i \in \mathbf{x}_{akt}\} \\ 1 & \forall \quad \{\mathbf{x}_i | \mathbf{x}_i \notin \mathbf{x}_{akt}\} \end{cases} , \tag{5.8}$$

$$\text{mit} \qquad \mathbf{x}_{akt} = [\mathbf{x}(e_1) \quad \dots \quad \mathbf{x}(e_k)] \ .$$

Somit ergibt sich für die relaxierten Zielwerte anhand der Architekturvektoren $\mathbf{x}_{min}$ und $\mathbf{x}_{max}$:

$$\mathcal{B}_{min} = \{\mathcal{R}(\mathbf{x}_{min}), \mathcal{R}_d(\mathbf{x}_{min}), \mathcal{S}(\mathbf{x}_{min})\} , \tag{5.9}$$

$$\mathcal{B}_{max} = \{\mathcal{R}(\mathbf{x}_{max}), \mathcal{R}_d(\mathbf{x}_{max}), \mathcal{S}(\mathbf{x}_{max})\} . \tag{5.10}$$

Die beiden ermittelten Zielwertmengen und die weiteren $2^{|\mathcal{B}|} - 2$ Eckpunkte durch die möglichen Zielwertkombinationen mit $|\mathcal{B}|$ Zielfunktionen spannen einen $|\mathcal{B}|$-dimensionalen Würfel $\mathcal{B}_{approx}$ auf, der den tatsächlich erreichbaren Zielwertraum des betrachteten Astes überapproximiert. Aufgrund der orthogonalen Kanten des Würfels gilt:

$$\exists \mathcal{B}_{dom} \in \mathcal{B}_{approx} \tag{5.11}$$
$$\text{mit } \mathcal{B}_i \preceq \mathcal{B}_{dom} \; \forall \left\{ \mathcal{B}_i | \mathcal{B}_i \in \mathcal{B}_{approx} \vee \mathcal{B}_i \neq \mathcal{B}_{dom} \right\} ,$$
$$\text{daraus folgt } \mathcal{B}_{dom} = \left\{ \mathcal{R}(\mathbf{x}_{max}), \mathcal{R}_d(\mathbf{x}_{max}), \mathcal{S}(\mathbf{x}_{min}) \right\} . \tag{5.12}$$

Somit dominieren die Zielwerte $\mathcal{B}_{dom}$ den vollständigen erreichbaren Zielwertraum des aktuellen Astes. Um über die weitere Entwicklung des aktuelles Astes zu entscheiden, wird überprüft, ob die Zielwerte dieser nicht-dominierten Architektur der aktuellen Überapproximation von den Zielwerten der Architekturen der aktuellen globalen PARETO-Front dominiert werden. Trifft dieses zu, wird die weitere Verzweigung verworfen. Werden die Zielwerte nicht dominiert, wird die Verzweigung fortgesetzt.

Das vorgestellte angepasste *Branch* & *Bound* Verfahren ermöglicht die Ermittlung der vollständigen und tatsächlichen PARETO-Front. Aufgrund der schrittweisen Beschränkung des Zielwertraumes durch die Ober- und Untergrenze, wird die nicht-dominierte Menge sukzessive ermittelt, was effizienter ist als die Betrachtung aller Architekturen wie bei der vollständigen Enumeration. Zudem ist es im Vergleich zum *Branch* & *Bound* Verfahren nach LAND und DOIG möglich auch für mehrfache und nicht-monotone Zielfunktionen den Zielwertraum zu durchsuchen.

Neben der Ermittlung der oberen und unteren Grenze der Suchraumvariable ist es zudem möglich, eine Initialarchitektur zu Beginn des Verfahren vorzugeben. Diese Architektur wird als erstes Element der PARETO-Menge genutzt, so dass eine Referenzarchitektur berücksichtigt werden kann. Diese Referenzarchitektur wird zum Ende der Optimierung ausgegeben, auch wenn sie aufgrund der weiteren ermittelten Zielwerte zum Ende der Optimierung nicht mehr Element der nicht-dominierten Menge sein sollte. Der Anwender kann somit bestehendes Vorwissen und Minimalanforderungen einfließen lassen und die Optimierung beschleunigen. Unabhängig von einer Referenzlösung wird jedoch aufgrund des deterministischen Verhaltens des *Branch* & *Bound* Verfahrens stets die gleiche nicht-dominierte Menge ausgegeben.

5.2.3 Genetischer Algorithmus

Die beiden zuvor vorgestellten Optimierungsverfahren ermöglichen eine deterministische Ermittlung der vollständigen PARETO-Front. Für Optimierungsprobleme mit einer großen variablen Ereignismenge $\mathbf{K}_v$ wächst für die Verfahren jedoch der Aufwand zum Aufstellen des gültigen Architekturraums, zur Auswertung des Zielwertraumes und zur Bestimmung der nicht-dominierten Menge. Das nachfolgende Verfahren wurde daher für große Optimierungsprobleme konditioniert, deren vollständige Untersuchung deterministisch nicht in einer akzeptablen Rechenzeit zu bewältigen ist. Aufgrund der NP-Vollständigkeit der Redundanzallokation existiert kein Algorithmus, der das Optimierungsproblem für beliebig große Systeme zeiteffektiv lösen kann [23, 125]. Heuristische Lösungsverfahren bieten hier jedoch eine Lösung an, indem sie beispielsweise nicht garantiert das globale Optimum finden, sondern nur eine möglichst gute Näherung. Inwieweit diese Näherung die Anforderungen an das Optimierungsproblem erfüllt, muss spezifisch für das Optimierungsproblem betrachtet werden [113]. Im Vergleich zu den beiden vorherigen Verfahren zeigt sich jedoch eine weitere Reduktion der berechneten Architekturen, um die Komplexität des Optimierungsproblems zu beherrschen. Für die Untersuchung des Architekturraumes mit Hilfe heuristischer Verfahren stehen unterschiedliche Verfahren zur Verfügung, wie sie bereits in Abbildung 5.1 vorgestellt wurden. Da die Verwendung einer Heuristik zu Lasten der garantierten Ermittlung der vollständigen und tatsächlichen PARETO-Front geht, wurden die folgenden Anforderungen an die Auswahl und Konditionierung des heuristischen Optimierungsverfahrens gestellt [49]:

1. **Breitensuche**: zur Untersuchung des Architekturraums ist eine hohe Streuung und Divergenz der ermittelten Zielwerte anzustreben. Während eine Konvergenz des Lösungsraums einen singulären Punkt als Optimierungsergebnis anstrebt, bedeutet eine geringe Streuung der Ergebnisse eine schlechte Verteilung der berechneten Architekturen im Zielwertraum, so dass beispielsweise nur die Extrema der einzelnen Zielwerte angestrebt werden. Anhand einer guten Streuung und hohen Divergenz des Lösungsraumes ist es möglich, den zulässigen Architekturraum in weiteren Optimierungsläufen einzuschränken und mit einem deterministischen Verfahren das Optimierungsproblem exakt lösen zu können.
2. **Qualität der ermittelten** PARETO**-Front**: der tatsächliche nichtdominierte Lösungsraum soll möglichst gut angenähert werden, so dass

die Verwendung eines weiteren, deterministischen Verfahrens nicht notwendig ist.

3. **Erhalt der Qualität der** Pareto**-Front**: stochastisch gefundene, gültige Lösungen sollen möglichst in der Architekturmenge erhalten bleiben und so für den nachfolgenden Entscheidungsprozess einen Eindruck vom vollständigen Zielwertraum geben.

Die zur Zeit gängigsten heuristischen Verfahren zur Lösung beliebiger kombinatorischer Optimierunsprobleme sind *Tabu-Search* nach Glover, *Simulated Annealing* entwickelt von Kirkpatrick et al., *Sintflut*-Algorithmen basierend auf den Arbeiten von Dueck und die von Holland entwickelten *Genetischen Algorithmen*[2] [24, 39, 47, 55, 113].

Untersuchungen welches Verfahren für welche Optimierungsprobleme am besten geeignet sind, sind nur schwer möglich, da es stets auf die problemspezifische Implementierung ankommt und die untersuchten Testprobleme nur eine geringe Aussagekraft für reale Optimierungsprobleme besitzen [18]. So lassen sich beispielsweise die Ergebnisse und Laufzeiten eines Genetischen Algorithmus mit unterschiedlichen Populationsgrößen nur bedingt mit den Eigenschaften einer Optimierung mittels Tabu-Search vergleichen. Versuche von Axelsson und Dussa-Zieger unterschiedliche Verfahren zu vergleichen, führen daher teilweise zu widersprüchlichen Ergebnissen und lassen methodisch keine Auswahl eines Verfahrens zu [7, 25, 113]. Die Widersprüche unterstreichen jedoch die Aussagen des *No Free Lunch* Theorems nach Wolpert, das über die Menge der Optimierungsprobleme den Optimierungsverfahren eine identische durchschnittliche Leistungsfähigkeit attestiert [129]. Coello et al. empfehlen jedoch für die kombinatorische Optimierung mehrkriterieller Probleme und die Ermittlung der Pareto-Menge explizit Genetische Algorithmen [18]. Zudem wurden bereits zahlreiche Optimierungen komplexer Systeme erfolgreich mit Genetischen Algorithmen realisiert, vergleiche Abschnitt 1.1 und 3.2. Desweiteren hat Lübke Genetische Algorithmen bereits auf den Entwurf elektrischer Kraftfahrzeugnetze angewendet, jedoch mit anderen Zielfunktionen [70]. Aus diesen Gründen wird die Auswahl eines Verfahrens aus der Gruppe der Genetischen Algorithmen im Folgenden weiter verfolgt.

[2] Genetische Algorithmen gehören neben den parallel entwickelten Evolutionsstrategien zur Klasse der Evolutionären Algorithmen. Unter Ersteren wird mittlerweile aufgrund der verwendeten Bitketten hauptsächlich die Optimierung kombinatorischer Probleme verstanden, Letztere behandeln die Funktionen- und Parameteroptimierung [121].

Genetische Algorithmen bilden den natürlichen evolutionären Prozess nach Darwin ab, wonach sich die Individuen mit der besten Fitness durchsetzen. Die Terminologie lehnt sich daher stark an die biologischen Begriffe an, in Tabelle 5.1 sind die unterschiedlichen Bezeichnungen gegenübergestellt. Anstatt einer einzelnen Lösung wird eine Population mit einer zu definierenden Anzahl von Individuen betrachtet, wobei jedes Individuum häufig durch eine Bitkette abgebildet wird. Anhand der Auswertung der Zielfunktionen für die einzelnen Individuen ist eine Bestimmung der Fitness möglich. Die Lösungen mit der höchsten Fitness werden anschließend in einer Elterngeneration gespeichert und die Bitketten werden für weitere Generationen gekreuzt und mutiert, so dass eine neue Population erstellt wird [14, 47]. Die Qualität der ermittelten Lösung wird dabei bei kleinen Populationen durch den Mutationsschritt bestimmt, bei großen Populationen dominiert die Kreuzung der Individuen den weiteren Verlauf [108].

Tab. 5.1: Gegenüberstellung evolutionärer Begriffe und der Interpretationen für die Redundanzallokation, nach [8]

Evolutionärer Begriff	*Interpretation*
Individuum	Fehlerereignis, Komponente, externes Ereignis
Chromosom	Parametersatz, Architektur
Gen	Entscheidungsvariable
Population	Architekturmenge

Die Fülle an Veröffentlichungen zu Genetischen Algorithmen[3] und den entwickelten Implementierungen bietet zahlreiche Varianten für eine adaptive, stochastische Analyse diskreter Lösungsräume. Die systematische Untersuchung der Verfahren ordnet diese jedoch in eine erste und zweite Generation und reduziert anhand der wesentlichen Algorithmusschritte die verfügbaren Verfahren [17]. Mittlerweile haben sich die Verfahren der zweiten Generation durchgesetzt, die zum großen Teil eine verbesserte Version entsprechender Verfahren erster Generation darstellen. Die Verbesserung der Verfahren liegt vor allem in der Berücksichtigung elitärer Individuen und somit mindestens einem Erhalt der gefundenen nicht-dominierten Lösungen. Drei der bewährtesten Genetischen Algorithmen der zweiten Generation sind der NSGA-II (engl. *Non-Dominated*

[3] Seit Beginn der 1990er Jahre listet alleine das EMOO 2860 Veröffentlichungen und ca. 400 Veröffentlichungen alleine im Jahr 2006 [17, 18]

Sorting Genetic Algorithm-II), der SPEA-II (engl. *Strength Pareto Evolutionary Algorithm-II*) und die PAES (engl. *Pareto Archived Evolution Strategy*) [17, 18].

Der Algorithmus PAES von Knowles und Corne basiert auf einer so genannten (1+1) Strategie zur Bildung der Nachkommen [58]. Wobei aus μ Individuen der Elterngeneration ν Individuen der Nachwuchsgeneration gebildet werden. Die Nachkommen werden mit den Lösungen eines Archivs Pareto-optimaler Lösungen verglichen, wobei das Archiv auch Lösungen vergangener Generationen enthalten kann. Zur Sicherstellung der Diversität im Lösungsraum ist als integraler Bestandteil des Algorithmus ein adapatives Gitter implementiert. Mit Hilfe dieses Gitters wird fortlaufend die Lösungsdichte in allen Bereichen des Lösungsraums bestimmt. Im Selektionsprozess werden somit zur Wahrung der Diversität vor allem Lösungen aus Regionen geringer Dichte ausgewählt. Neben der $(1+1)$ Strategie bestehen zudem allgemeine $(\mu + \nu)$ Strategien, die bei großen Populationen eine bessere Kreuzung ermöglichen [18, 58].

Zitzler und Thiele nutzen in ihrer Weiterentwicklung des bewährten SPEA ebenfalls ein Lösungsarchiv, verwenden als Diversitätsfunktion jedoch kein Gitter, sondern einen gesonderten Fitnesswert, der proportional zur Dominanz einer Lösung ist [130]. Die Dominanz berücksichtigt dabei zum einen wie viele Lösungen die betrachtete Lösung dominiert, zum anderen von wie vielen Lösungen die betrachtete Lösung dominiert wird. Dieses Verfahren bevorzugt somit die Lösungen aus der ersten Pareto-Front. Zur Gewährleistung der Diversität im Lösungsraum werden vor allem Lösungen aus Gegenden geringer Dichte bevorzugt, die mit Hilfe des geringsten Abstands zur nächsten Lösung der aktuellen Pareto-Front berechnet wird. Die Rechenoperationen zur Überprüfung einer Lösung nehmen mit steigender Archivgröße n_A bei einer Populationsgröße n_P und $n_{\mathcal{A}}$ Zielfunktionen mit der Ordnung $\mathcal{O}((n_A + n_P)n_{\mathcal{A}})$ zu, so dass neuere Implementierungen die Größe des Archivs vorab begrenzen, jedoch auf Kosten der Ergebnisqualität. Die Lösungen und vor allem die Randbereiche der ersten Pareto-Front bleiben jedoch bis zum Erreichen der Begrenzung erhalten [18, 130].

Der von Deb et al. vorgestellte NSGA-II nutzt im Gegensatz zu den vorherigen Algorithmen kein Lösungsarchiv, sondern erhält die nicht-dominierten Lösungen über die Generationen, *Elitismus* genannt [22]. Die Zielwerte einer Population werden zunächst ausgewertet und die Lösungen anhand ihres Pareto-Ranks sortiert. Hierfür wird die Pareto-Front der Lösungsmenge ermittelt, gespeichert und aus der Lösungsmenge gelöscht. Dieser Prozess wie-

derholt sich bis die verbleibenden Lösungen die letzte nicht-dominierte Menge bilden. Die Elterngeneration wird nachfolgend aus den Individuen der ersten Pareto-Fronten gebildet. Sollte eine Front aufgrund der Größenbeschränkung der Elterngeneration nicht vollständig übernommen werden können, wird ein distanzbasiertes Verfahren zur Auswahl der Individuen angewendet, das dabei auch die Diversität der Lösungsmenge garantiert. Anhand der nachfolgenden Mutation und Kreuzung wird die Elternpopulation zu einer neuen Generation ergänzt [21, 22].

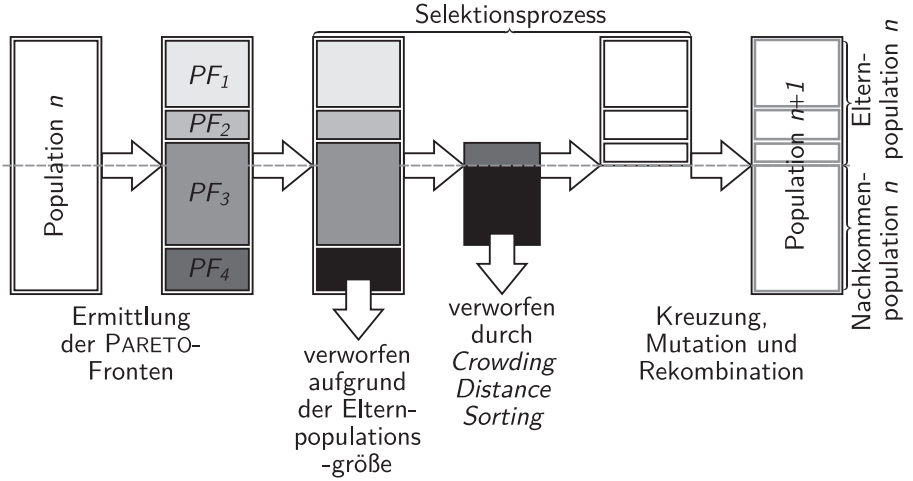


Abb. 5.9: Ablauf des NSGA-2 Algorithmus [22]

Die Auswahl eines geeigneten Genetischen Algorithmus basiert auf den vorherigen Anforderungen an ein heuristisches Verfahren: Breitensuche, Qualität und Erhalt der ermittelten Pareto-Menge. Bezüglich der Komplexität der Verfahren zeigt sich für alle drei Verfahren, dass die Ermittlung der Pareto-optimalen Lösungen die zeitliche Entwicklung dominiert und die Komplexität für alle Verfahren mit $\mathcal{O}(n_{\mathcal{A}} n_P^2)$, mit der Populationsgröße n_P und der Anzahl der Zielfunktionen $n_{\mathcal{A}}$ [22, 68]. Bezüglich der Diversität sowie der Qualität der ermittelten Lösung zeigt Deb anhand mehrer Testbeispiele jedoch, dass der NSGA-II den weiteren Verfahren aufgrund der Elitismus- und Distanzfunktion überlegen ist [21]. Bezüglich des Erhalts von bereits ermittelten Lösungen besitzt der PAES Algorithmus durch das verwendete Lösungsarchiv Vorteile

gegenüber den weiteren Verfahren, hierbei überwiegt jedoch das Kriterium eine möglichst gute Approximation und somit Diversität der nichtdominierten Menge zu erreichen. Aus diesen Gründen wurde für die weitere Konkretisierung und Implementierung der *Non-dominated Sorting Genetic Algorithm-II* ausgewählt.

Abbildung 5.9 veranschaulicht den prinzipiellen Ablauf des NSGA-II und die spezifischen Schritte des verwendeten Algorithmus. Nachfolgend werden die einzelnen Schritte erläutert und die problemspezifischen Änderungen am Algorithmus vorgestellt. Die Änderungen betreffen die Berücksichtigung der Nebenbedingungen, den Selektionsprozess und den Kreuzungsschritt, beeinflussen jedoch nicht die grundlegende Struktur des NSGA-II.

Die Anfangspopulation wird zufällig gebildet, sofern der Anwender keine vollständige oder partielle Anfangspopulation vorgibt. Dabei wird jede Architektur analog zur vollständigen Enumeration mittels einer Bitkette vorgegeben. Bei einer partiellen vorgegebenen Anfangspopulation wird die Population bis zum Erreichen der Populationsgröße mit zufälligen Bitketten aufgefüllt. Vor allem aufgrund der zufälligen Anfangslösungen aber auch durch den Mutationsschritt kann es zu Verstößen gegen die Nebenbedingungen des MRS kommen, die sowohl zu unzulässig guten aber auch schlechten Lösungen führen können. Aus diesem Grund ist eine Berücksichtigung der Nebenbedingungen zwingend notwendig, um im Verlaufe des Optimierungsprozesses die zulässigen Lösungen zu stärken. Zeitgleich soll jedoch eine hohe Diversität der Lösungsmenge erreicht werden. Daher werden unzulässige Lösungen nicht aus der Population gelöscht, sondern durch die Zielwerte entsprechend der Gleichung 5.13 bestraft. Die unzulässigen Lösungen bleiben somit vor allem in den ersten Generationen enthalten und tragen zur Bildung weiterer, auch zulässiger, Lösungen bei.

$$\begin{array}{lcll} \mathcal{R}_i(\mathbf{x},t) & = & 0 & \forall \quad \mathcal{R}_i(\mathbf{x},t) \in \mathcal{R} \\ \mathcal{R}_{d,i}(\mathbf{x},t) & = & 0 & \forall \quad \mathcal{R}_{d,i}(\mathbf{x},t) \in \mathcal{R}_d \\ \mathcal{S}_i(\mathbf{x}) & \to & \infty & \forall \quad \mathcal{S}_i(\mathbf{x}) \in \mathcal{S}\ . \end{array} \tag{5.13}$$

Der ausgewählte NSGA-II vermeidet aufgrund des verwendeten *Crowding Distance Sorting* eine frühzeitige Konvergenz der Architekturraums, da Lösungen bevorzugt werden, die im $n_{\mathcal{A}}$-dimensionalen Lösungsraum eine hohe Distanz zueinander besitzen. Um jedoch eine schnelle Konvergenz des Lösungsraums zu ermöglichen, wird zur Ermittlung der Pareto-Optimalität nicht das Kriterium der *strengen* Pareto-Optimalität angewendet, so dass Lösungen auch

mehrfach in einer Pareto-Menge enthalten sein dürfen. Die Ermittlung der räumlichen Distanz zur nächsten Lösung bevorzugt in diesem Fall periphäre Lösungen, da ihr Distanzwert gegen ∞ strebt. Abbildung 5.10(a) verdeutlicht das Verhalten für den von Deb vorgeschlagenen Algorithmus [21]. Zunächst steigt die Größe der ersten Pareto-Front an, bis sie den Wert der Elternpopulation erreicht hat. Anschließend greift das *Crowding Distance Sorting* Verfahren und wählt entsprechend der Distanzwerte Architekturen für die nächste Elterngeneration aus. In diesem Schritt können wiederholte periphäre Lösungen auf Kosten singulärer Architekturen in der Knieregion der ersten Pareto-Front ausgewählt werden. Durch diese Auswahl verringert sich die Zunahme der Qualität der ermittelten Pareto-Front. Bei Pareto-Fronten mit wenigen stark dominierenden Lösungen kann dieses Verhalten des *Crowding Distance Sorting* dazu führen, dass die ermittelte nicht-dominierte Menge auf vereinzelte, periphäre Architekturen schrumpft und somit die Divergenz des Lösungsraums bewahrt, nicht jedoch eine Streuung zwischen den Extremwerten.

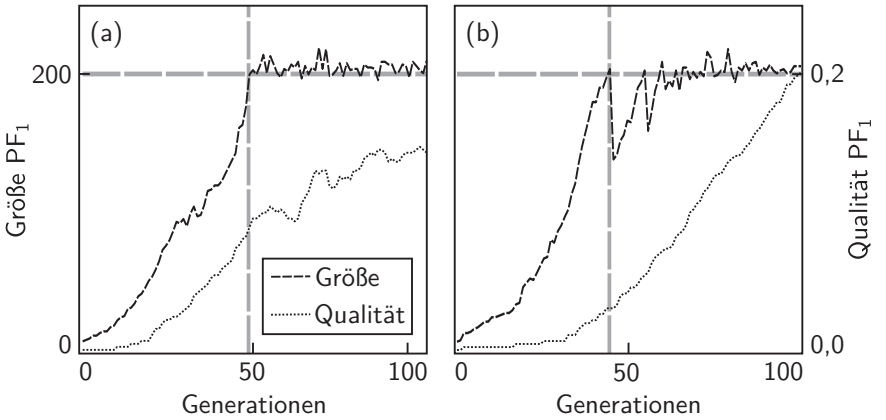


Abb. 5.10: (a) *Crowding Distance Sorting* Verfahren nach Deb, (b) erweitertes Verfahren zum Erhalt der Streuung der Pareto-Front

Das Verfahren von Deb wurde daher um einen Schritt zum Erhalt der Lösungsstreuung erweitert. Abbildung 5.10(b) zeigt das Ergebnis des vorherigen Optimierungsproblems mit einem erweiterten *Crowding Distance Sorting*. Im ersten Schritt werden Lösungen nach ihrer räumlichen Distanz ausgewählt, im zweiten Schritt wird sichergestellt, dass nicht-dominierte Lösungen nur einfach in der nächsten Elterngeneration auftreten. Dieses führt im Verlauf der Größe

der ersten PARETO-Front beim Erreichen der Größe der Elternpopulation zu einem Einbruch, garantiert jedoch den Erhalt aller gefundenen Architekturen der Menge PF_{real}. Im Gegensatz zu dem Verfahren von DEB ergibt sich somit ein monoton-steigender Verlauf der Qualität der ersten PARETO-Front. In den oben dargestellten Verläufen hat, heuristisch bedingt, der linke Optimierungslauf bis zur 50. Generation ungefähr doppelt so viele Lösungen der Menge PF_{real} gefunden wie der rechte Lauf. Aufgrund des zuvor beschriebenen Effekts der Lösungsstreuung erreicht das erweiterte Verfahren nach dem Erreichen der Elternpopulationsgröße jedoch eine höhere Qualität.

Die Elternpopulation wird im nachfolgenden Schritt der Kreuzung und der Mutation unterzogen. Für kleine Populationsgrößen dominiert dabei die Mutation die Qualität der erreichten Lösung, bei großen Population hingegen hat die Kreuzung einen wesentlichen Einfluss auf die Qualität [108]. Die Auswahl und Anpassung eines Kreuzungsverfahrens (engl. *Crossover*) ist vor allem bei stark beschränkten Systementwürfen entscheidet für eine mögliche Evolution der Lösungsmenge. So können *Single-Point-Crossover* Verfahren mit einem zufällig gesetzten Kreuzungspunkt verstärkt zu unzulässigen Architekturen führen, wodurch eine Variation der Bitketten nur noch durch die Mutation bleibt, die jedoch auch zu unzulässigen Lösungen führen kann. *Multi-Point-Crossover* Verfahren sind zwar in der Lage mehrere Teile von Bitketten zu rekombinieren und so die Breitensuche im Architekturraum zu unterstützen, bei zufällig gesetzten Kreuzungspunkten steigt dabei jedoch die Wahrscheinlichkeit im Vergleich zu *Single-Point-Crossover* Verfahren unzulässige Lösungen zu erzeugen [90]. Aus diesem Grund wurde als Kreuzungsschritt für die vorliegenden stark beschränkten Architekturräume ein problemspezifisches Verfahren entwickelt, das für die Kreuzung die problemspezifische Struktur berücksichtigt.

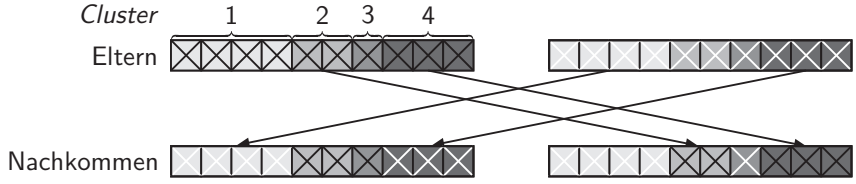


Abb. 5.11: Ablauf der strukturierten mehrfachen Rekombination

Komplexe Flugzeug-Systemarchitekturen weisen, wie bereits beim *Branch & Bound* Verfahren dargelegt, unterschiedliche Entscheidungspunkte auf, die unabhängig voneinander betrachtet werden können. Das vorherige *Branch &*

Bound Verfahren nutzt diese Eigenschaft zur Aufstellung der Systemarchitekturen. Der Genetische Algorithmus verwendet diese Gruppierungen für einen strukturierten Kreuzungsschritt, indem die Bitketten an einer zufälligen Anzahl von Kreuzungspunkten rekombiniert werden. Die Positionen der Kreuzungspunkte werden jedoch nicht zufällig gewählt, sondern anhand der zuvor gebildeten Gruppierungen der Ereignisse durch die Nebenbedingungen. Das Verfahren wird entsprechend als *Clustered Multi-Point-Crossover* bezeichnet und ist in Abbildung 5.11 für zwei Individuen aus der Elterngeneration und die Erzeugung der entsprechenden Nachkommen dargestellt.

In Abbildung 5.12 ist die Entwicklung der ersten PARETO-Front und der Lösungsqualität für ein stochastisches *Multi-Point-Crossover* und das entwickelte *Clustered Multi-Point-Crossover* dargestellt. Es zeigt sich dabei deutlich der Einfluss der strukturierten Kreuzung auf die Qualität der Lösung, aufgrund der mehrfachen Kreuzungspunkte entsteht jedoch trotzdem eine diversitäre Lösungsmenge.

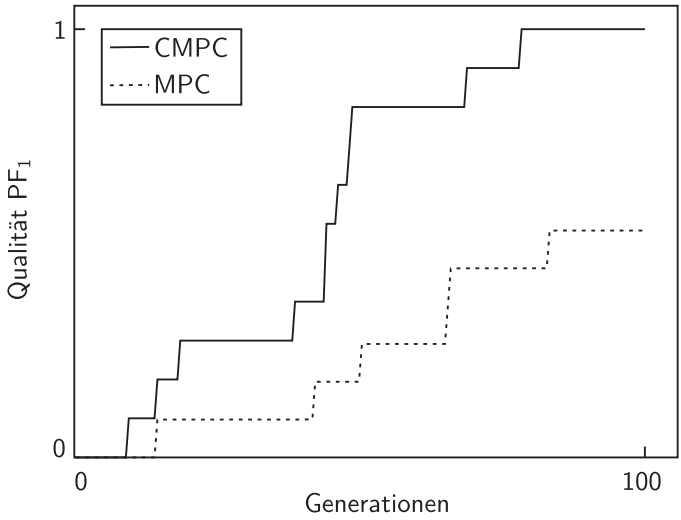


Abb. 5.12: Vergleich des gängigen *Multi-Point-Crossover (MPC)* Verfahrens und des erweiterten *Clustered Multi-Point-Crossover (CMPC)* Verfahrens

Der anschließende Mutationsschritt erfolgt an einer zufälligen Stelle der Bitkette und ermöglicht somit kleine Veränderungen in den Architekturen, die sich über die Anzahl der Generationen zu neuen Lösungsbereichen entwickeln können. Die Wahrscheinlichkeit für eine Mutation wirkt sich dabei auf die Erzeugung gültiger Nachkommen aus. Eine zu hohe Wahrscheinlichkeit kann vor allem bei stark beschränkten Problemen zu einem zu hohen Evolutionsdruck führen und somit verstärkt zu unzulässigen Lösungen. Da unzulässige Lösungen aufgrund der Bestrafung der Zielwerte und der anschließenden PARETO-Rangfolge im fortgeschrittenen Verlauf der Optimierung keinen Einfluss auf die Entwicklung haben, sollten diese vermieden werden und der Wert der Populationsgröße der Beschränkung des MRS angepasst sein.

5.3 Vergleichende Bewertung der Optimierungsverfahren

Zur Verifikation der vorherigen Aussage zum *No Free Lunch* Theorem und der Auswahl der drei Algorithmen für unterschiedliche Problemklassen, werden im Folgenden die drei implementierten Algorithmen mit Hilfe eines generischen Testproblems gegenübergestellt. Abbildung 5.13 zeigt das generische seriell-parallele Zuverlässigkeitsblockdiagramm, das für die Verifikation genutzt wird und entsprechend Abschnitt 3.2 ein serielles Redundanzallokationsproblem darstellt [110]. Zur Abbildung eines realistischen Optimierungsproblems wurden mit jeder Erweiterung des Zuverlässigkeitsblockdiagramms auch beliebige Nebenbedingungen eingebracht.

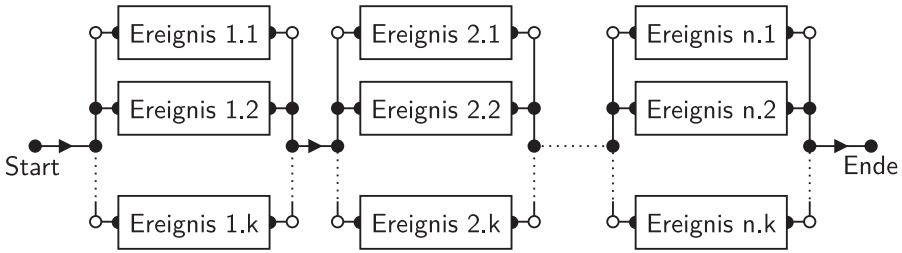


Abb. 5.13: Generisches seriell-paralleles Zuverlässigkeitsblockdiagramm zum Vergleich der Optimierungsverfahren

Die Berechnung der generischen seriell-parallelen Struktur ergibt sich aus [77]:

$$R_{SP}(t) = \prod_{i=1}^{n} \left(1 - \prod_{j=1}^{k} (1 - R_{i,j}(t)) \right) ; . \tag{5.14}$$

Größtenteils unbeschränkte Probleme lassen sich sehr gut mit Hilfe des Genetischen Algorithmus berechnen, da dieser aufgrund der Heuristik in diesem Fall vor allem gültige Lösungen findet und somit schnell zur realen PARETO-Front konvergiert. Für die vollständige Enumeration und das *Branch & Bound* Verfahren erhöht sich hingegen der Rechenaufwand exponentiell mit der Problemgröße $2^{|\mathbf{K}_v|}$, wie es auch die folgende Abbildung zeigt.

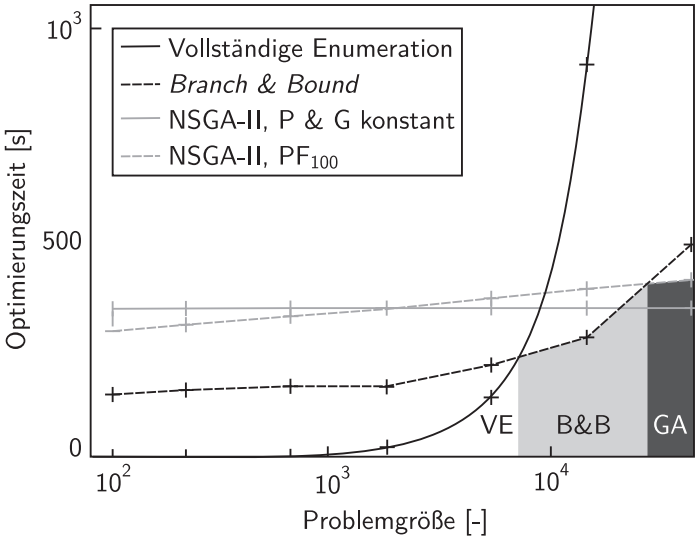


Abb. 5.14: Vergleich der Optimierungszeiten der drei Verfahren in Abhängigkeit der Problemgröße

Für die Verifikation wird die variable Menge des mehrfach-redundanten Systemmodells fortlaufend vergrößert, so dass sich für die unterschiedlichen Algorithmen der in Abbildung 5.14 dargestellte Verlauf ergibt. Für den Genetischen Algorithmus wurden aufgrund der Heuristik zwei Kurven verwendet,

zum einen wurden die Populationsgröße n_P und die Generationenanzahl n_G konstant gehalten, zum anderen wurde eine PF_{100}-Kurve ermittelt. Die Qualität der PARETO-Front letzteren Kurve beträgt 100%, dieses entspricht PF_{real}. Die Ergebnisse des Genetischen Algorithmus werden dabei mit der tatsächlichen nichtdominierten Menge des *Branch & Bound* Verfahrens verglichen. Eine Menge von Initialarchitekturen wurde für den Genetischen Algorithmus nicht vorgegeben, so dass der Algorithmus in Abhängigkeit der prozentualen Beschränkung des Problems zunächst nach gültigen Lösungen suchen muss. Die Interpolation des Verhaltens der vollständigen Enumeration mit Breitensuche entspricht allgemein der Entwicklung der Rechenzeit zur Ermittlung einer nicht-dominierten in Abhängigkeit der Problemgröße.

Die Grafik zeigt deutlich die Eignung der vollständigen Enumeration für kleine bis mittlere Problemgrößen. Mit wachsendem Architekturraum dominiert die zeitliche Entwicklung vor allem die Ermittlung des gültigen Architekturraums und der PARETO-Front sowie die nötigen Matrixoperationen. Generell ist die Komplexität zur Ermittlung der nicht-dominierten Menge aus n_{ges} Lösungen mit $n_{\mathcal{A}}$ Zielfunktionen von der Ordnung $O(n_{\mathcal{A}}(2n_{ges})^2)$ [22]. Da die weiteren Verfahren mit kleineren Lösungsmengen arbeiten, überwiegt ab der dargestellten Problemgröße die Ermittlung der PARETO-Menge die zusätzlichen Rechenschritte der weiteren Algorithmen. Das *Branch & Bound* Verfahren eignet sich für mittlere bis große Optimierungsprobleme mit einer Problemgröße von zu 15000 Varianten, bei einer moderaten Beschränkung des gültigen Zustandsraums auf ca. 8 % des theoretischen Architekturraums. Mit steigender Größe der Menge der variablen Ereignisse $\mathbf{K}_v$ zeigt sich jedoch auch hier eine Steigerung der Rechenzeiten, die vor allem durch die häufige Ermittlung der oberen und unteren Grenze hervorgerufen wird. Bei kleinen Optimierungsproblemen überwiegt die vollständige Enumeration gegenüber dem *Branch & Bound* Verfahren, da dieses zu viele Nebenrechnungen durchführt. Der Genetische Algorithmus zeigt bei kleinen Problemen Schwächen aufgrund der zahlreichen Algorithmusschritte wie dem *Crowding Distance Sorting*. Da die Ermittlung der nicht-dominierten Menge, deren Berechnung quadratisch mit der Problemgröße anwächst, sich bei dem Genetischen Algorithmus nur auf die Populationsgröße bezieht, wächst die Optimierungsdauer nicht entsprechend der vollständigen Enumeration an. Zudem zeigt sich bei gleichbleibender Populationsgröße und Generationenanzahl beinahe eine zeitliche Invarianz gegenüber der Problemgröße, da die Berechnungen von dieser nur gering beeinflusst werden. Der Einfluss der Problemgröße wird bei der Ermittlung der PF_{100}-Kurve deutlich. Bei gleichbleibender Populationsgröße steigt die Anzahl der Generationen bis zur

Ermittlung der benötigten Qualität der ermittelten PARETO-Front. Die Komplexitätsordnung des Genetischen Algorithmus wird dabei analog zur vollständigen Enumeration von der Ermittlung der PARETO-Menge dominiert. Diese bezieht sich bei der Heuristik jedoch nur auf die spezifische Populationsgröße. Abbildung 5.10 verdeutlicht die Konvergenz des Algorithmus gegen die tatsächliche nicht-dominierte Menge, entsprechend verhält sich das zeitliche Verhalten zur Ermittlung der PARETO-Front bei gleichbleibender Qualität.

Die Ergebnisse zur Validierung des *No-Free-Lunch* Theorems für das vorliegende Optimierungsergebnis zeigen deutlich, dass die Wahl des Optimierungsverfahrens die Rechenzeit stark beeinflusst. In diesem Fall handelt es sich um ein schwach beschränktes Optimierungsproblem, das heißt die Anzahl der gültigen Lösungen liegt zwischen 5 und 31 % des theoretisch möglichen Lösungsraumes. Bei stärker beschränkten Problemen wächst für die vollständige Enumeration die Zeit zur Ermittlung der gültigen Lösungen aufgrund der notwendigen Matrixoperationen an. Eine stärkere Beschränkung des Entscheidungsbaumes zeigt sich für das *Branch & Bound* Verfahren durch eine Reduktion der Varianten pro Entscheidungsebene bzw. eine Reduktion der Entscheidungsebenen selbst, wenn sich hierdurch Ereignismengen überschneiden. Der NSGA-II erzeugt durch die zufällige Auffüllung der Startpopulation und die Mutation bei stark beschränkten Problemen vermehrt ungültige Lösungen, so dass die Lösungsmenge langsamer zur realen PARETO-Menge konvergiert und somit mehr Generationen für vergleichbare Ergebnisse notwendig sind. Bei stark beschränkten Problemen ist es somit sinnvoll, dem heuristischen Verfahren eine Initialmenge vorzugeben und somit auch das Vorwissen des Anwenders in die Optimierung einfließen zu lassen [18].

6 Unterstützung der Architekturauswahl und Implementierung

Die Entwicklung neuer Assistenzfunktionen und Methoden für den Entwurf komplexer Flugzeugsysteme erfordert neben angepassten Analyse- und Optimierungsverfahren auch eine Aufbereitung und Visualisierung der Ergebnisse, die den nachfolgenden Entscheidungsprozess unterstützen. Entsprechend ergibt sich das Ergebnis eines mehrkriteriellen Optimierungsproblems erst durch die Kombination der *Optimierung* ansich und einer angeleiteten *Lösungsauswahl* [18]. In dem folgenden Kapitel wird die Integration des zuvor vorgestellten Verfahrens in den Entscheidungs- und Entwicklungsprozess vorgestellt. Dieses umfasst die Visualisierung des mehrdimensionalen Zielwertraumes, die Unterstützung bei der Lösungsauswahl und die Übernahme der Ergebnisse für den weiteren Entwicklungsprozess. Anschließend wird der vollständige Lösungsfindungsprozess zusammengefasst und anhand eines illustrativen und offen zugänglichen Beispiels demonstriert. Zum Ende des Kapitels werden Aspekte der Implementierung und die Integration der entwickelten Methode in das Werkzeug SyRelAn erläutert.

6.1 Visualisierung mehrdimensionaler Zielwerte

Die Bewertung mehrdimensionaler Parametersätze erfordert für eine ingenieurwissenschaftliche Betrachtung und Entscheidungsfindung eine aussagekräftige und übersichtliche Darstellung der Daten. Aufgrund der beschränkten Möglichkeiten der zweidimensionalen Visualisierung existieren mehrere Ansätze für eine Darstellung von mehrdimensionalen Parametersätzen. Die gängigsten Methoden stellen die *parallelen Koordinaten* und mit einem ähnlichen Ansatz die *Spinnennetze* dar. Einzelne Parametersätze werden hierbei nach einer Normierung als ein Linienzug dargestellt und ermöglichen somit eine relative Bewertung der unterschiedlichen Parameter zueinander [34]. Die parallelen Koordinaten werden beispielsweise für das Optimierungswerkzeug MOPS (engl. *Multi-Objective Parameter Synthesis*) des Deutschen Zentrums für Luft- und Raumfahrt ge-

nutzt [53]. Neben den Methoden zur relativen Darstellung haben sich im ingenieurwissenschaftlichen Bereich Verfahren zur Visualisierung der absoluten Daten durchgesetzt, beispielsweise durch alle Projektionskombinationen mittels *Scatter Plots*. Ein Nachteil dieser Verfahren liegen darin, dass die Eigenschaften einer Lösung nicht mit einem Blick erfasst werden können. Wobei für die Konzeptphase hierbei nicht zwangsläufig die exakten Zielwerte entscheidend sind, sondern die relative Darstellung der Lösungen zueinander [34, 93].

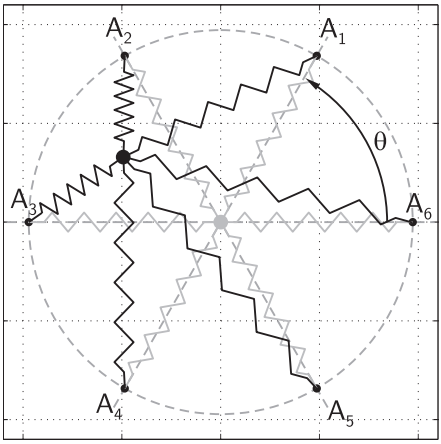


Abb. 6.1: Prinzip der RadViz-Methode auf Grundlage des Federmodells

Für eine relative Bewertung der ermittelten Lösungen zueinander bietet die Visualisierungsmethode RadViz (engl. *Radial Coordinate Visualization*) die Darstellung jeder Lösung aus einem $n_{\mathcal{A}}$-dimensionalen Lösungsraum als Punkt in einem radialen Koordinatensystem [45, 84]. Die Koordinaten der Lösungen werden dabei analog zu einem Federmodell bestimmt, so dass eine neutrale Lösung im Zentrum des Koordinatensystems liegt. Zielfunktionswerte größer null führen zu einer Variation der Federsteifigkeit zwischen dem so genannten Anker A_i der Zielfunktion i und dem Lösungspunkt u_i. Die Federsteifigkeiten c_i werden dabei derart variiert, dass sich für die normierte Lösung wieder eine kraftneutrale Darstellung gemäß

$$\sum_{j=1}^{n_{\mathcal{A}}} (\vec{A_j} - \vec{u_i}) c_i = \vec{0} \tag{6.1}$$

ergibt [84]. Abbildung 6.1 verdeutlicht das radiale Koordinatensystem und die Verschiebung einer Lösung.

Aus Gleichung (6.1) ergibt sich für die Transformation vom $n_{\mathcal{A}}$-dimensionalen Lösungsraum in das radiale Koordinatensystem [84]:

$$u_{i,x} = \frac{\sum_{j=1}^{n_{\mathcal{A}}} c_i \cdot cos(\theta_j)}{\sum_{j=1}^{n_{\mathcal{A}}} c_i} \tag{6.2}$$

$$u_{i,y} = \frac{\sum_{j=1}^{n_{\mathcal{A}}} c_i \cdot sin(\theta_j)}{\sum_{j=1}^{n_{\mathcal{A}}} c_i} \, . \tag{6.3}$$

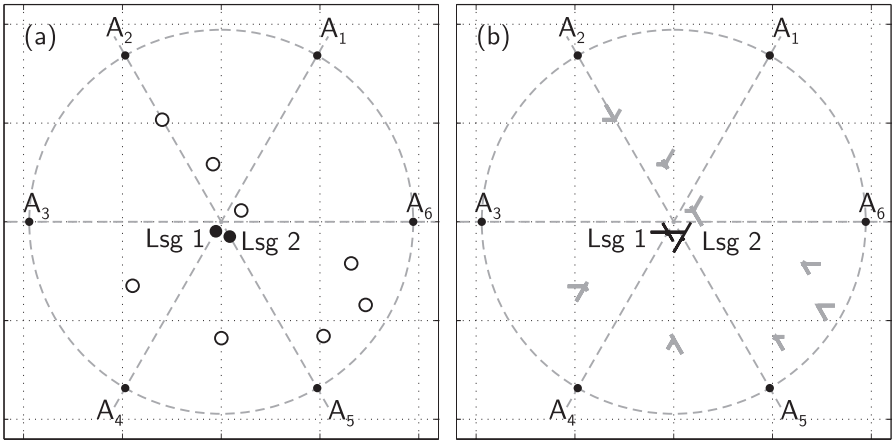


Abb. 6.2: Darstellung einer Pareto-Menge mit Hilfe der RadViz-Methode und Erweiterung zur Darstellung der Federsteifigkeiten

Der Vorteil der vorgestellten Visualisierung liegt in der übersichtlichen Darstellung der Vor- und Nachteile einer Lösung durch das hinterlegte Federmodell. Dieses führt jedoch auch dazu, dass die Darstellung des Lösungsraumes nicht eindeutig ist und von der Anordnung der Anker der Zielfunktionswerte A_i auf dem Einheitskreis abhängt. Zudem lässt sich das Verständnis der euklidischen Darstellung nicht auf die nichtlineare Visualisierung übertragen, da Lösungen, die im Federmodell benachbart sind, nicht zwangsläufig identische Eigenschaften besitzen [34]. Abbildung 6.2 verdeutlicht links diese Eigenschaft

für zwei hervorgehobene Lösungen in der Nähe des Ursprungs, deren Lösungseigenschaften sich stark unterscheiden. Die Unterschiede der Lösungen werden im Folgenden aufgezeigt und das Verfahren entsprechend angepasst.

Für eine Entscheidungsfindung ist es wichtig, neben der Tendenz einer Lösung mit konträren Zielfunktionswerten auch die ermittelten Federsteifigkeiten relativ zum restlichen Lösungsraum darzustellen. Aus diesem Grund wurde die RadViz-Methode wie nachfolgend hergeleitet erweitert. Die Lösungspunkte werden dabei durch Glyphen ersetzt, deren Schenkellänge die Güte eines Zielfunktionswertes angibt, so dass gute Funktionswerte in einer größeren Schenkellänge resultieren. Die Koordinaten der einzelnen Schenkelspitzen der Lösung i ergeben sich somit aufgrund der ermittelten und normierten Zielwerte $\mathcal{B}$ zu:

$$x(i,j) = cos(\theta_j) \cdot \overline{\mathcal{B}}_{i,j} \tag{6.4}$$

$$y(i,j) = sin(\theta_j) \cdot \overline{\mathcal{B}}_{i,j} \; . \tag{6.5}$$

Die ermittelten Glyphen werden zur besseren Darstellung skaliert und in den Punkt u_i verschoben. Die erweiterte Darstellung für das vorherige Beispiel ist rechts in Abbildung 6.2 verdeutlicht. Die unterschiedlichen Eigenschaften der Lösungen im Ursprung können anhand der erweiterten Darstellung visualisiert werden, so dass eine verbesserte Entscheidungsfindung auf Grundlage der Visualisierung besteht. In dem Beispiel besitzt die erste Lösung am Ursprung Stärken in den Zielwerten entsprechend der Ankerpunkte A_1 und A_4. Die benachbarte Lösung verhält sich hierzu konträr und besitzt Stärken in den verbleibenden Zielwerten.

Neben den Vorteilen in der Visualisierung besitzt die erweiterte RadViz-Methode gegenüber Projektionsverfahren den Vorteil, dass nur eine Grafik erzeugt wird, deren Visualisierung von $n_{\mathcal{A}}$ Dimensionen und $n_{\mathcal{B}}$ Datensätzen die Komplexitätsordnung $\mathcal{O}(n_{\mathcal{B}} n_{\mathcal{A}})$ besitzt [34]. Aufgrund der Kombinatorik der erforderlichen Projektionen ergibt sich für die Projektionsverfahren hingegen eine Komplexität von $\mathcal{O}(n_{\mathcal{B}} n_{\mathcal{A}}^2)$.

Aufgrund der relativen Darstellung der Datensätze durch das verwendete RadViz Verfahren ist eine Validierung der Sicherheitsanforderungen nicht ohne weitere Informationen zu den Architekturen möglich. Die Implementierung entsprechend Abschnitt 6.5 sieht daher drei Funktionen zur Überprüfung der Absolutwerte vor. Zunächst werden den dargestellten Glyphen weitere Informationen hinterlegt, die die absoluten Zielwerte enthalten und somit eine manuelle Überprüfung relevanter Architekturen zulassen. Bei großen Lösungsmengen ist

zudem eine automatische Überprüfung der Sicherheitsanforderungen aufgrund der Klassifizierungen gemäß der Zulassungsvorschrift EASA CS §25.1309 möglich. Lösungen, die die Anforderungen nicht erfüllen, werden anschließend aus der PARETO-Menge entfernt. Das entsprechende Verfahren wird im folgenden Abschnitt aufgegriffen. Schlussendlich ist für die Visualisierung von Optimierungsproblemen mit wenigen Zielfunktionen eine kartesische Darstellung aller kombinatorischer Projektionen der Zielwerte möglich. Auch hierbei hinterliegen jeder Lösungen sämtliche Zielwerte und Informationen über die berücksichtigten Ereignisse und die betrachtete Architektur.

6.2 Hierarchische Architekturauswahl

Die Darstellung der Optimierungsergebnisse mittels der vorgestellten, erweiterten RADVIZ Methode ermöglicht den Vergleich der Lösungen relativ zueinander. Da jedoch die Größe der PARETO-Menge mit der Anzahl der betrachteten Zielwerte zunimmt, sind bei großen Optimierungsproblemen mit vielen Zielwerten häufig über einhundert Architekturen in der Ergebnismenge PF_{real} enthalten. Die Nicht-Dominanz der enthalten Architekturen ist somit zur Auswahl geeigneter Architekturen ein zu schwaches Kriterium [46]. Aus diesem Grund wird im Folgenden ein hierarchischer Auswahlprozess betrachtet, der auf Grundlage der ermittelten PARETO-Menge und den Systemanforderungen $\mathcal{B}_R$ den Lösungsraum weiter verkleinert und somit die Entscheidungsfindung systematisch unterstützt. Das Ziel hierbei ist jedoch nicht die Auswahl einer Lösung, sondern gemäß Abbildung 2.1 eine reduzierten Lösungsmenge, die im Anschluss an den Vorentwurf detaillierter betrachtet werden kann. Zudem ist hervorzuheben, dass dem Anwender zu Beginn der Reduzierung der vollständige, ermittelte Zielwertraum dargestellt wird und somit entsprechend der formulierten Kernfragen die Transparenz im Entscheidungsprozess und die Nachvollziehbarkeit von Anforderungen und deren Auswirkungen bewahrt bleibt.

Auswahlschritt I
Vor der Reduktion des Zielwertraumes ist bei der Architekturauswahl zunächst die Robustheit der ermittelten PARETO-Front zu betrachten. Die entwickelte Methode zur optimalen Redundanzallokation fehlertoleranter Systemarchitekturen fokussiert entsprechend Kapitel 3 den Vorentwurf von Flugzeugsystemen. Die Entwicklungsphase zeichnet sich durch eine hohe Parameterunsicherheit auszeichnet [88]. Neben der Unsicherheit der verfügbaren Parameter zeigt sich jedoch auch eine geringe Verfügbarkeit der sicherheitsrelevanten und luftfahrts-

pezifischen Daten, so dass die Robustheit der PARETO-Front nur bedingt auf der Ebene der Ereignisse betrachtet werden kann. Aus diesem Grund ermöglicht die erste Stufe der Architekturauswahl eine Angabe spezifischer prozentualer Abweichungen für die einzelnen Zielgrößen, so dass im Folgenden untersucht wird, welche Lösungen für die anschließende Reduktion in Frage kommen. Die ermittelten dominierten Zielwerte werden hierbei wie folgt variiert:

$$\mathcal{B}_\varepsilon = \{\varepsilon_1 \cdot \mathcal{B}_1, \ldots, \varepsilon_n \cdot \mathcal{B}_n\} \,. \tag{6.6}$$

Dabei ergibt sich analog zu der Relaxierung des *Branch* & *Bound* Verfahrens für jede Architektur eine dominierende Variation, so dass diese gegenüber der ermittelten PARETO-Front zu überprüfen ist. Sofern die variierte Architektur nicht mehr von der ursprünglichen PARETO-Menge dominiert wird, wird diese in die Lösungsmenge übernommen. Die ermittelte ε-PARETO-Front ist in Abbildung 6.3 dargestellt. Auch die dargestellte robuste PARETO-Menge kann dabei nur auf dem hinterlegten Modell basieren. Eine Modellvalidierung wie sie für dynamische Simulationsmodelle vorgenommen werden kann, wäre aufgrund der probabilistischen Werte nur mit Hilfe von Dauerversuchen des Systems oder der Komponenten möglich.

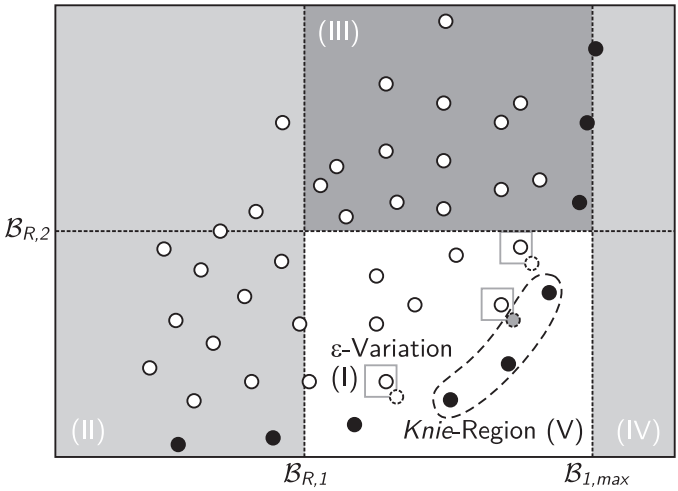


Abb. 6.3: Darstellung der hierarchischen Architekturauswahl für exemplarische Zielwerte

Als Erfahrungswert für den Robustheitsparameter ε zeigt sich, dass maximal ein zehnprozentiges Intervall um die tatsächliche PARETO-Front die bestehenden Unsicherheiten gut abbildet. Dieses beruht vor allem auf der Angabe der Fehlerraten der Komponenten, die überlichweise maximal um eine Größenordnung vom angegebenen Wert abweichen. Aufgrund der Allgemeingültigkeit der entwickelten Methode und den nichtlinearen Zielfunktionen des MRS kann dieses jedoch nur eine Empfehlung sein.

Auswahlschritt II
In der folgenden Stufe der Architekturauswahl wird der Zielwertraum mit Hilfe der Sicherheitsanforderungen gemäß der Zulassungsvorschriften EASA CS 25 verkleinert. Dem mehrfach-redundanten Systemmodell hinterliegen hierfür neben den Ausfalllogiken und Ereignisparametern auch Informationen über die Klassifizierung der betrachteten Fehlerbedingungen. Somit werden Architekturen, deren Systemmodelle die Sicherheitsanforderungen nicht erfüllen, aus der Menge der PARETO-optimalen Lösungen entfernt. Hierbei ist zu beachten, dass es sich bei den Sicherheitsanforderungen um kein Differenzierungsmerkmal der Hersteller handelt, sondern um eine diskrete behördliche Anforderung. Eine Differenzierung ist vielmehr hinsichtlich Robustheit der ermittelten Werte und der Fehlertoleranz der Architekturen möglich. Die reale nicht-dominierte Menge $\overline{PF}$ wird daher im Zielwertraum anhand der Zielwerte $\mathcal{B}$ wie folgt begrenzt:

$$\overline{PF} = \{\mathcal{B} | \mathcal{B}_1, \ldots, \mathcal{B}_m \geq \mathcal{B}_R^1, \ldots, \mathcal{B}_R^m\} \tag{6.7}$$

Auswahlschritt III
Im Gegensatz zu den Sicherheitsanforderungen sind die Zuverlässigkeitsanforderungen nicht durch die Vorschriften der Zulassungsbehörden definiert und stellen daher ein schwächeres Reduktionskriterium dar. Für die Beschränkung des Zielwertraumes ist es daher möglich im Anschluss an die Visualisierung freie Grenzen für die Systemzuverlässigkeit anhand der Flugzeuganforderungen zu definieren und somit den Lösungsraum zu beschränken. Analog zu der Beschränkung durch die Zuverlässigkeitswerte $\mathcal{B}^R$ ist eine Reduktion der PARETO-Menge mit Hilfe der summativen Parameter möglich, so dass sich die folgende *a-posteriori* Nebenbedingung ergibt:

$$\begin{aligned} \overline{PF} \quad = \quad & \{\mathcal{B} | \mathcal{B}_1^R, \ldots, \mathcal{B}_m^R \geq \mathcal{B}_{R,1}^R, \ldots, \mathcal{B}_{R,m}^R \ldots . \\ & \ldots . \wedge \mathcal{B}_{m+1}, \ldots, \mathcal{B}_n \leq \mathcal{B}_{R,m+1}, \ldots, \mathcal{B}_{R,n}\} \, . \end{aligned} \tag{6.8}$$

Auswahlschritt IV
Nachdem der Lösungsraum auf die Architekturen beschränkt wurde, die die adressierten Anforderungen an das zu entwickelnde System erfüllen, können im nächsten Schritt schwächere Kriterien betrachtet werden, die nicht zwangsläufig durch die Systemanforderungen abgedeckt sind. Während für die summativen Zielfunktionen, wie beispielsweise der Systemmasse, sowie der Zuverlässigkeit ein möglichst minimaler bzw. maximaler Zielwert angestrebt wird, gilt diese Anforderung für die Betrachtung der Systemsicherheit nicht zwangsläufig. Die Sicherheitsanforderungen müssen im Rahmen des Entwicklungsprozesses nachgewiesen werden, eine Übererfüllung ist unter Beachtung der Systemmasse für den Flugzeug- und Systemhersteller jedoch wirtschaftlich nicht sinnvoll. Aus diesem Grund kann der Zielwertraum für die Sicherheitswerte auf einen spezifischen maximalen Wert für jede Fehlerbedingung begrenzt werden.

$$\overline{PF} = \{\mathcal{B} | \mathcal{B}_1, \ldots, \mathcal{B}_m \leq \mathcal{B}^1_{R,max}, \ldots, \mathcal{B}^m_{R,max}\} \,. \tag{6.9}$$

Die verbleibenden Zielwerte stellen den endgültigen Entscheidungsraum auf Grundlage der bisherigen Systemanforderungen und -analysen dar. Weitere Beschränkungen aufgrund dieser Ergebnisse sind nur noch auf Grundlage qualitativer Entscheidungen oder Detailanalysen möglich und sind somit Gegenstand des interdisziplinären Entscheidungsprozess unter Berücksichtigung der in Abschnitt 2.1 vorgestellten Einflüsse und nur mit deren Interessenvertretern sinnvoll.

Als bevorzugtes Gebiet der Pareto-Front hat sich allgemein die so genannte *Knie-Region* herausgestellt, da diese die besten Kompromisse hinsichtlich der betrachteten konträren Zielgrößen darstellt [18]. In Abbildung 6.3 ist die Reduktion des Zielwertraumes mittels der vier vorgestellten Reduktionsschritte dargestellt.

Neben diesem *a-posteriori* Prozess zur Auswahl präferierter Lösungen, kann der Systemingenieur den zu ermittelnden Lösungsraum auch *a-priori* durch die Vorgabe von Initiallösungen beeinflußen. Das *Branch & Bound* Verfahren sieht hierfür die Vorgabe einer Initialarchitektur vor, die somit bereits Grenzen der Zielwerte vorgibt und als frühzeitige vollständig ausgewertete Lösung zur Überprüfung der Pareto-Optimalität herangezogen werden kann. Hierdurch ist eine bessere Definition der oberen und unteren Grenze möglich und somit auch eine schnellere Lösung des Optimierungsproblems. Der Genetische Algorithmus lässt sich durch die Vorgabe einer Initialpopulation in einen bestimmten Teil

des vollständigen Architekturraumes steuern [15]. Durch die Rekombination und Mutation mit anschließendem *Crowding-Distance-Sorting* werden trotzdem gestreute und divergierende Architekturen präferiert, die Anwendung des Verfahrens in Kapitel 7 zeigt jedoch dass die Vorgabe präferierter Lösungsbereiche den gewünschten Einfluss auf das heuristische Optimierungsergebnis hat.

6.3 Integration in den Entwicklungsprozess

Die interdisziplinäre Analyse von komplexen FLugzeugsystemen in der Vorentwurfsphase setzt neben der Verfügbarkeit geeigneter Werkzeuge auch eine Interaktion der unterschiedlichen Systemanalysen voraus. Dieses ermöglicht es, eine schnelle Reaktion auf die sich stetig verändernden Systemanforderungen zu gewährleisten und somit die Komplexität des Entwicklungsprozesses zu handhaben. Unterschiedliche Ansätze verfolgen daher eine Integration der Sicherheits- und Zuverlässigkeitsanalysen in den weiteren Entwurfsprozess anhand funktionaler Systemmodelle, entsprechende Ansätze wurden in der Einleitung dieser Arbeit betrachtet [104, 105]. Vor allem der Systementwurf komplexer, sicherheitskritischer Systeme erfordert jedoch auch frühzeitig eine Trennung der funktionalen Analysen und der Sicherheitsbewertung, nur dieses ermöglicht einen unabhängigen Systementwurf [61].

Nicht nur in der Luftfahrtindustrie sondern generell bei sicherheitskritischen Systemen ist die Trennung von Sicherheitsbewertung und funktionalem Design unabdingbar. Der Grad der Trennung hängt bei Flugzeugsystemen vom DAL ab, der sich aus der Klassifizierung der relevanten Fehlerbedingungen ergibt. Aufgrund der notwendigen Segregation wird im Folgenden ein Konzept zur Integration der ermittelten Optimierungsergebnisse in den vollständigen Systementwurf betrachtet, der auf der ermittelten Zielwertmenge des Optimierungsprozesses basiert. Hierbei wird somit nicht ausschließlich die nicht-dominierte Menge betrachtet, sondern der gesamte ermittelte Zielwertraum, der als Archiv möglicher Systemarchitekturen dient.

Abbildung 6.4 veranschaulicht die Interaktion zwischen der Sicherheitsbewertung und dem Systementwicklungsprozess sowie die Übergabe der Ergebnisse. Die Architekturbewertung kann dabei mit Hilfe einer definierten Architektur, repräsentiert durch den Architekturvektor $\mathbf{x}$, auf die Zielwerte der Redundanzallokation zugreifen. Hierbei wird der Prozess weniger zur Optimierung, son-

dern zur automatischen Architekturbewertung genutzt. Die Ergebnisse der Architekturbewertung werden in der Menge der Zielwerte $\mathcal{B}(\mathbf{x})$ abgelegt, das als Archiv genutzt wird, zusätzlich hinterlegt werden Informationen zur Nicht-Dominanz der Architekturen. Die sicherheits- und zuverlässigkeitstechnischen Zielwerte einer Architektur können mit Hilfe des Architekturvektors $\mathbf{x}$ abgefragt werden.

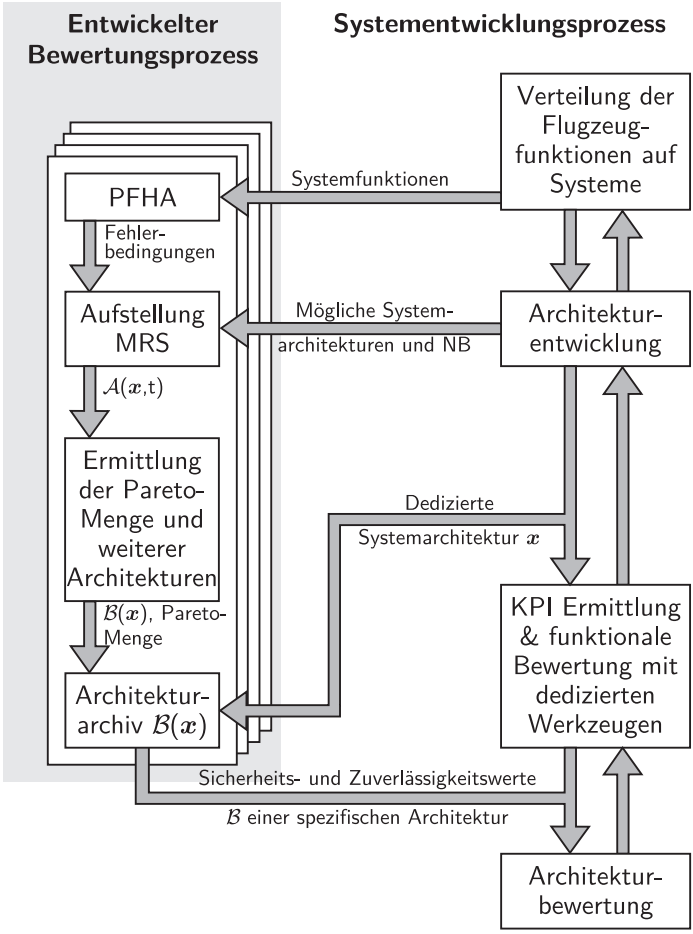


Abb. 6.4: Konzept zur Interaktion zwischen funktionaler und sicherheitstechnischer Bewertung

Statt eines Archives wäre es zudem möglich, nach Bedarf Architekturen zu analysieren und die Ergebnisse der Architekturbewertung zur Verfügung zu stellen. Hierbei wäre zum einen jedoch keine Aussage über die Nicht-Dominanz der Architektur möglich. Zum anderen wäre hierfür eine bedarfsgesteuerte Berechnung der Architekturen notwendig, so dass für die Architekturbewertung nicht nur die Ergebnisse als Archiv vorliegen müssten, sondern das vollständige Analysewerkzeug SyRelAn. Dieses erschwert durch die Anzahl der insgesamt notwendigen Werkzeuge die Integration aufgrund notwendiger Schnittstellen, anstatt eines gemeinsamen Austauschformates der Ergebnisse.

Parallel zu der sicherheitstechnischen Bewertung können mit den gleichen Informationen zu den möglichen und expliziten Systemarchitekturen weitere Analysen zum funktionalen Verhalten oder zur Abschätzung der *Key Performance Parameters* (*KPI*) mit dedizierten Werkzeugen durchgeführt werden. Das Zusammenführen der beiden Ergebnisse erlaubt im Anschluss eine Architekturbewertung unter Verwendung von Analysewerkzeugen unterschiedlicher Disziplinen, die entwickelte Methode stellt hierfür die Sicherheits- und Zuverlässigkeitsbewertung und die notwendigen Schnittstellen bereit.

Der Prozess zur Interaktion zwischen funktionaler und sicherheitstechnischer Bewertung gewährleistet die notwendige Unabhängigkeit zwischen den Disziplinen, definiert zugleich jedoch auch die notwendigen Schnittstellen und ermöglicht somit eine ganzheitliche Bewertung mit unterschiedlichen Werkzeugen.

6.4 Illustratives Anwendungsbeispiel

Zur Demonstration der gesamten Redundanzallokation, d.h. von der Erstellung des mehrfach-redundanten Systemmodells bis zur Visualisierung mit Hilfe der erweiterten RadViz-Methode und der Architekturauswahl, wird im Folgenden ein übersichtliches Beispiel vorgestellt. Der definierte Prozess des Optimierungskonzeptes aus Abschnitt 3.3 ist in Abbildung 6.5 dargestellt, zudem sind die Hinweise auf die entsprechenden Abschnitte in dieser Arbeit enthalten.

Die Ziele der Redundanzallokation fehlertoleranter Flugzeugsysteme wurden in Abschnitt 1.2 anhand von vier Kernfragen für diese Arbeit formuliert:

1. Welches Redundanzkonzept ist systemweit und systemübergreifend zielführend bezüglich Sicherheit und Zuverlässigkeit?

2. Welchen Einfluss haben unterschiedliche Redundanzkonzepte auf das Degradationsverhalten eines Systems?

3. Wie wirken sich lokale Architekturvariationen auf die Systemsicherheit und -zuverlässigkeit aus?

4. Wie wirken sich Anforderungen und Klassifizierungen von Fehlerbedingungen auf die weiteren Systemparameter, z.B. die Systemmasse, aus?

Die Fragen werden für das folgende illustrative Beispiel eines elektrischen Energieversorgungssystems wieder aufgegriffen und exemplarisch beantwortet.

Auf Grundlage der dimensionierenden Fehlerbedingungen der *PFHA* stellt der Systemingenieur die zu untersuchenden mehrfach-redundanten Systemmodelle auf. In dem betrachteten Beispiel handelt es sich um ein elektrisches Energieversorgungssystem eines typischen Verkehrsflugzeuges und die dimensionierende Fehlerbedingung *„Verlust der DC Sammelschienen“*, die im Rahmen der *PFHA* aufgrund der Kritikalität der angeschlossenen Verbraucher mit CAT klassifziert wurde. Zudem wird eine degradierte Systemfunktion basierend auf der ersten Fehlerbedingung betrachtet. Aufgrund des symmetrischen Systemaufbaus werden hierfür die Ereignisse 1 und 2, siehe nachfolgende Tabelle, variiert und die Systemstrukturfunktion abgeleitet, die Fehlerbedingung wurde mit HAZ klassifiziert. Als weiterer Parameter wird zudem die Systemmasse abgeschätzt.

Die Tabelle 6.1 enthält für die betrachteten Komponenten die notwendigen Parameter zur Komponentenmasse, zum betrachteten Fehlermode sowie die Fehlerrate, die für das Aufstellen des mehrfach-redundanten Systemmodells benötigt werden. Die Fehlerraten beziehen dabei auf die genannten Fehlermodi der Komponenten, die entsprechend der betrachteten Fehlerbedingung relevant sind.

Abbildung 6.6 zeigt das variable Strukturmodell für die oben genannte Fehlerbedingung. Neben architekturellen Freiheitsgraden, beispielsweise die Entscheidung über die Verwendung von Querverbindungen, wird für einige Komponentenfunktionen eine Technologie ausgewählt, die sich auch auf die Fehlerrate des betrachteten Fehlermodus auswirkt. Neben den variablen Ereignissen sind zudem zwei feste Fehlerereignisse berücksichtigt, die die beiden zu versorgenden DC-Sammelschienen abbilden. Die vollständigen orthogonalisierten Minimalpfade sind in Anhang A enthalten, diese können zur Wiederholung des Beispiels genutzt werden.

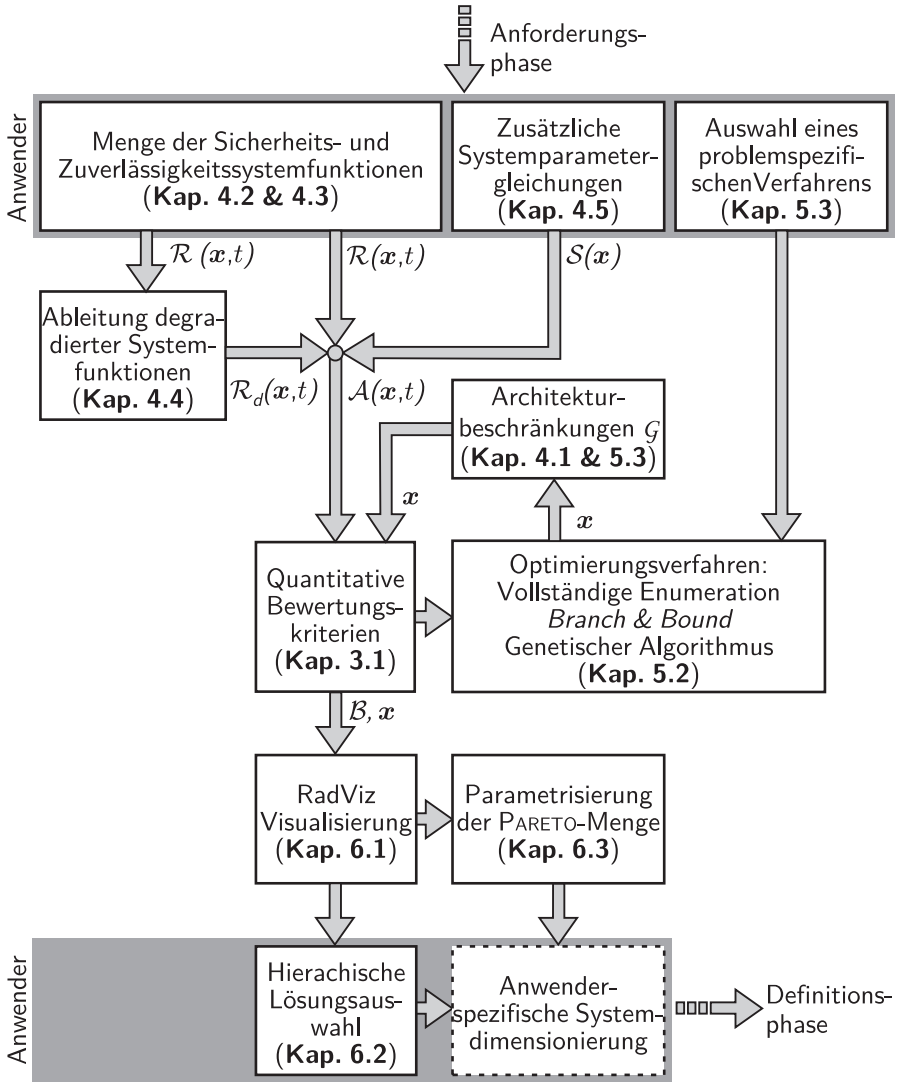


Abb. 6.5: Ablauf der Methode zur optimalen Redundanzallokation

Tab. 6.1: Parameter des illustrativen Beispiels

Nr.	Komponente	Masse [kg]	Fehlermode	Fehlerrate [10^{-5}; 1/FH]
Variable Komponenten				
1	Generatorpfad 1, A	35	Leistungsverlust	20
2	Generatorpfad 1, B	40	Leistungsverlust	10
3	Generatorpfad 2, A	35	Leistungsverlust	20
4	Generatorpfad 2, B	40	Leistungsverlust	10
5	Sekundärer Generatorpfad, A	20	Leistungsverlust	1
6	Sekundärer Generatorpfad, B	25	Leistungsverlust	2
7	Bus Tie Contactor 1	5	Schalter fällt offen aus	2
8	Bus Tie Contactor 2	5	Schalter fällt offen aus	2
9	Gleichrichter 1, A	15	Verlust der Ausgangsleistung	1
10	Gleichrichter 1, B	12	Verlust der Ausgangsleistung	5
11	Gleichrichter 2, A	15	Verlust der Ausgangsleistung	1
12	Gleichrichter 2, B	12	Verlust der Ausgangsleistung	5
13	DC Bus Tie Contactor 1	5	Schalter fällt offen aus	10
14	DC Bus Tie Contactor 1	5	Schalter fällt offen aus	10
Feste Komponenten				
101	DC Sammelschiene 1	4	Verlust der Leistungsverteilung	2
102	DC Sammelschiene 2	4	Verlust der Leistungsverteilung	2

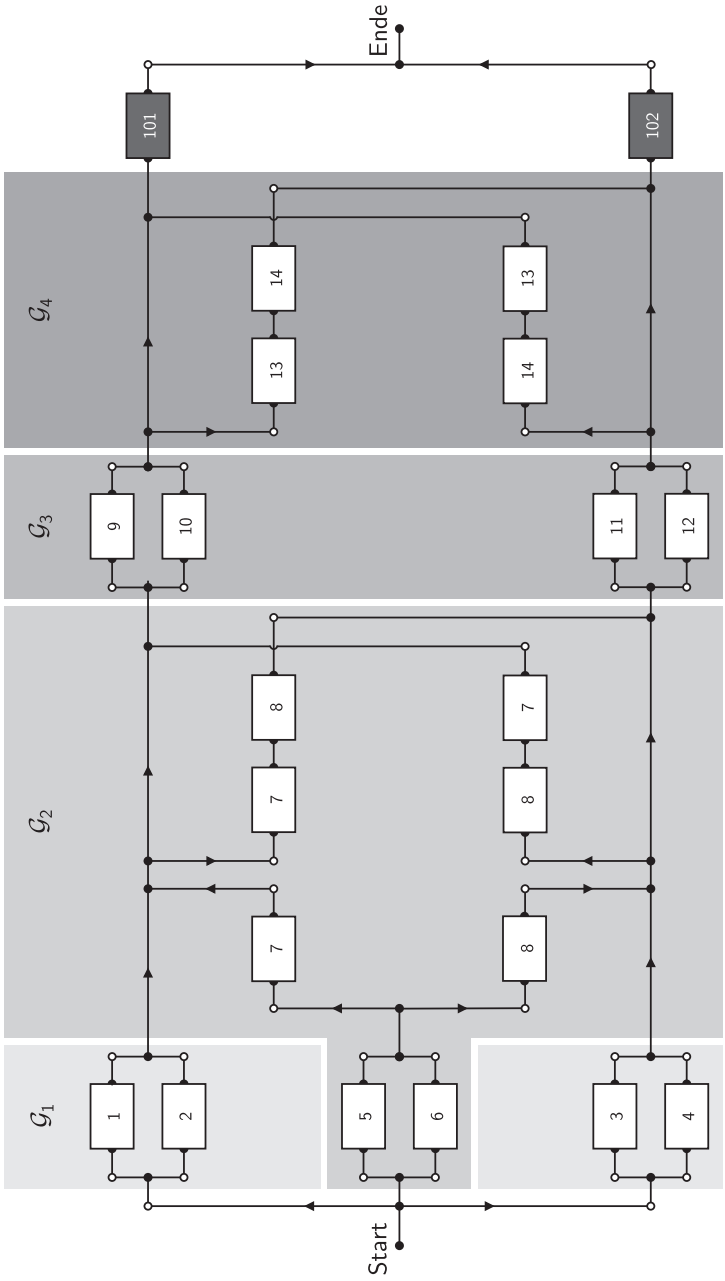


Abb. 6.6: Mehrfach-redundantes Systemmodell des illustrativen Beispiels

Tab. 6.2: Nebenbedingungen des illustrativen Beispiels

	$x_1 + x_2$	$=$	1	:	Generatorpfad 1, A oder B
	$x_3 + x_4$	$=$	1	:	Generatorpfad 2, A oder B
$\mathcal{G}_1$	$x_1 - x_3$	$=$	0	:	identische Generatorpfade 1 und 2
	$x_5 + x_6$	$=$	1	:	sekundärer Generatorpfad A oder B
$\mathcal{G}_2$	$x_5 + x_6 - x_7$	$=$	0	:	sekundärer Generatorpfad nur mit Querverbindungen
	$x_7 - x_8$	$=$	0	:	identische Architekturvariablen der Bus Tie Contactor 1 und 2
$\mathcal{G}_3$	$x_9 + x_{10}$	$=$	1	:	Gleichrichter A oder B
	$x_9 - x_{11}$	$=$	0	:	identische Gleichrichter 1 und 2
$\mathcal{G}_4$	$x_{13} - x_{14}$	$=$	0	:	identische Architekturvariablen der DC Bus Tie Contactor 1 und 2

Aus der Anzahl der variablen Ereignisse ergibt sich theoretisch ein möglicher Architekturraum von $2^{14} = 16384$ Architekturen. Zur Auswahl eines geeigneten Optimierungsverfahrens muss berücksichtigt werden, dass erfahrungsgemäß nur bis zu fünf Prozent der möglichen Architekturen gültig sind. Somit ergibt sich eine Abschätzung von maximal 820 Architekturen und aufgrund der Problemgröße als Optimierungverfahren die vollständige Enumeration.

Die vollständige Enumeration liefert auf einem Arbeitsplatzrechner in $1,3\mathsf{s}$ inklusive der Visualisierung die in Abbildung 6.7 dargestellten Zielwerte der Pareto-Front. Unter Berücksichtigung der Nebenbedingungen in Tabelle 6.2 ergab die vollständige Ermittlung des Architekturraums 32 gültige Architekturen; dieses bestätigt die Verwendung der vollständigen Enumeration als Optimierungsverfahren.

Bei den drei hervorgehobenen Architekturen in Abbildung 6.7 handelt es sich um jeweils unterschiedliche Konzepte, deren Eigenschaften anhand der Lage in dem nichtlinearen Koordindatensystem abgelesen werden können. Die Architektur (a) ist eine hochredundante Architektur mit drei Generatoren und den möglichen Querverbindungen zur Versorgung der betrachteten DC-Sammelschienen. Aus diesem Grund ist die Bewertung der Systemmasse ver-

glichen mit den weiteren Konzepten sehr schlecht. Die Sicherheitsbewertung liefert für dieses Konzept in beiden Zielwerten jedoch sehr gute Ergebnisse. Das Konzept (b) verfügt über zwei Generatoren und eine Querverbindung auf der Ebene der Triebwerksgeneratoren. Dieses Konzept liegt in dem ausgeglichenen Bereich der RadViz-Visualisierung und die absoluten Zielwerte bestätigen die ausgeglichene Architektur. Die Architektur (c) enthält die minimal möglichen Komponenten und ist daher bezüglich der Systemmasse die beste Lösung aus dem betrachteten Architekturraum. Aufgrund der fehlenden Redundanzen sind die Zielwerte bezüglich der beiden Sicherheitsanforderungen jedoch relativ schlecht. Bezüglich der Kernfrage 2 lässt sich aus einer ersten Betrachtung der Konzepte ableiten, dass die Querverbindungen und eventuell ein dritter Generator für die degradierten Systemzustände ausschlaggebend sind.

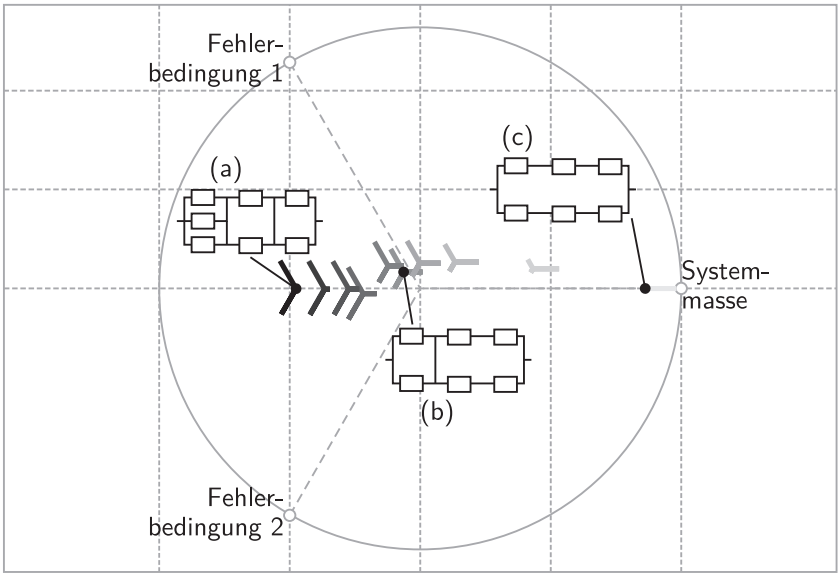


Abb. 6.7: RadViz-Darstellung der Pareto-Front des illustrativen Beispiels

Die nachfolgende Tabelle 6.3 enthält die Architekturvektoren der Pareto-optimalen Architekturen sowie die Bedeutung der einzelnen Einträge des Architekturvektors. In Tabelle 6.4 sind die dazugehörigen vollständigen Zielwerte der nicht-dominierten Menge aufgeführt.

Tab. 6.3: PARETO-Menge des illustrativen Beispiels

Nr.	Architekturvektor x													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	1 GEN 1/B	0	1 GEN 1/B	1 Sek GEN A	0	1 BTC 1	1 BTC 2	1 TRU 1/A	0	1 TRU 2/A	0	1 CBTC 1	1 DBTC 2
2	1 GEN 1/A	0	1 GEN 2/A	0	1 Sek GEN A	0	1 BTC 1	1 BTC 2 2	1 TRU 1/A	0	1 TRU 2/A	0	1 CBTC 1	1 DBTC 2
3	1 GEN 1/A	0	1 GEN 2/A	0	1 Sek GEN A	0	1 BTC 1	1 BTC 2 2	0	1 TRU 1/B	0	1 TRU 2/B	0	0
4	0	1 GEN 1/B	0	1 GEN 1/B	0	0	1 BTC 1	1 BTC 2	0	1 TRU 1/B	0	1 TRU 2/B	0	0
5	0	1 GEN 1/B	0	1 GEN 1/B	0	0	0	0	0	1 TRU 1/B	0	1 TRU 2/B	0	0
6	1 GEN 1/A	0	1 GEN 2/A	0	0	0	0	0	0	1 TRU 1/B	0	1 TRU 2/B	0	0
7	0	1 GEN 1/B	0	1 GEN 1/B	1 Sek GEN A	0	1 BTC 1	1 BTC 2	1 TRU 1/A	0	1 TRU 2/A	0	0	0
8	1 GEN 1/A	0	1 GEN 2/A	0	1 Sek GEN A	0	1 BTC 1	1 BTC 2	1 TRU 1/A	0	1 TRU 2/A	0	0	0
9	0	1 GEN 1/B	0	1 GEN 1/B	0	0	1 BTC 1	1 BTC 2	1 TRU 1/A	0	1 TRU 2/A	0	0	0
10	0	1 GEN 1/B	0	1 GEN 1/B	0	0	0	0	1 TRU 1/A	0	1 TRU 2/A	0	0	0
11	1 GEN 1/A	0	1 GEN 2/A	0	0	0	0	0	1 TRU 1/A	0	1 TRU 2/A	0	0	0

Tab. 6.4: Zielwerte der PARETO-Menge des illustrativen Beispiels

Nr.	$\mathcal{B}_1$ [1/FH] Verlust der elektrischen Energieversorgung	$\mathcal{B}_2$ [1/FH] Verlust der elektrischen Energieversorgung degradiert	$\mathcal{B}_3$ [kg] Systemmasse
1	$5,002 \cdot 10^{-10}$	$1,700 \cdot 10^{-9}$	158
2	$5,006 \cdot 10^{-10}$	$2,700 \cdot 10^{-9}$	148
3	$4,901 \cdot 10^{-9}$	$8,300 \cdot 10^{-9}$	132
4	$1,490 \cdot 10^{-8}$	$1,000 \cdot 10^{-4}$	122
5	$2,890 \cdot 10^{-8}$	$1,700 \cdot 10^{-4}$	112
6	$7,288 \cdot 10^{-8}$	$2,700 \cdot 10^{-4}$	102
7	$9,002 \cdot 10^{-10}$	$2,500 \cdot 10^{-9}$	148
8	$9,006 \cdot 10^{-10}$	$3,500 \cdot 10^{-9}$	138
9	$1,090 \cdot 10^{-8}$	$1,000 \cdot 10^{-4}$	128
10	$1,690 \cdot 10^{-8}$	$1,300 \cdot 10^{-4}$	118
11	$5,289 \cdot 10^{-8}$	$2,300 \cdot 10^{-4}$	108

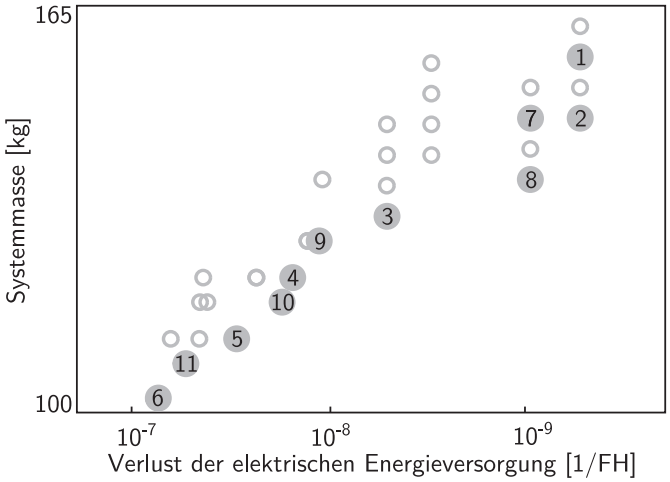


Abb. 6.8: PARETO-Front und Architekturraum des illustrativen Beispiels

Abbildung 6.8 verdeutlicht die Lage der PARETO-Front und die minimalen und maximalen Zielwerte der Systemsicherheit und -masse. Im Folgenden wird anhand dieser Ergebnisse der Zielwertraum soweit reduziert, dass eine Menge an Lösungen bestehen bleibt, die sich für Detailanalysen eignet.

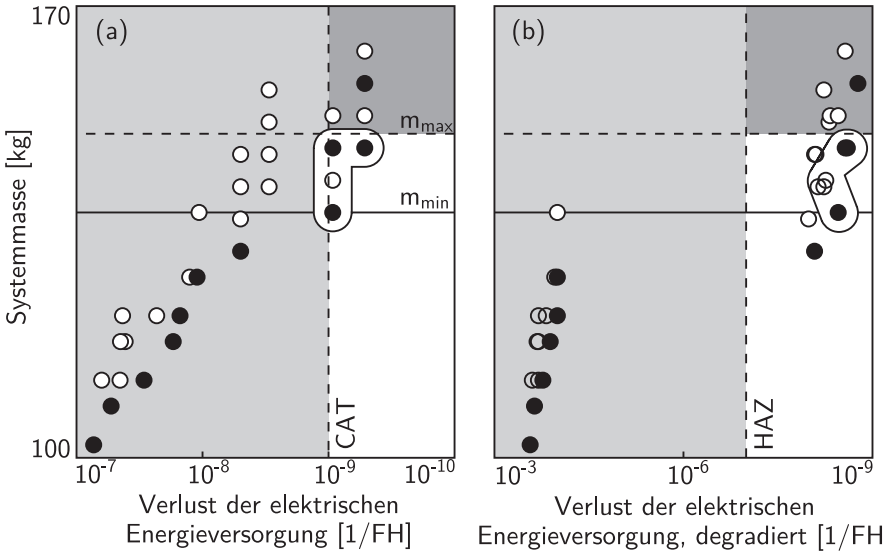


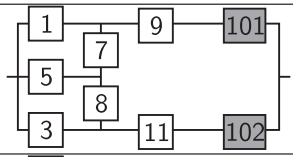
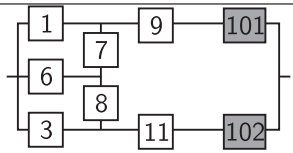
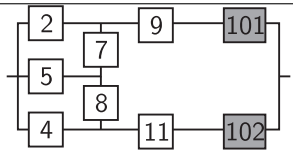
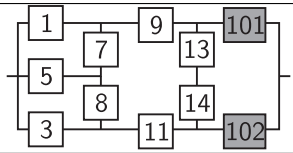
Abb. 6.9: Projektionen des vollständigen Architekturraums des illustrativen Beispiels

In Abbildung 6.9 sind die zwei Projektionen des Architekturraums mit der hervorgehobenen PARETO-Front dargestellt. Die Darstellung der Ausfallwahrscheinlichkeiten für die degradierten Systemzustände zeigt deutlich die unterschiedlichen Architekturen und Redundanzkonzepte, die im linken Teil für die nominellen Systemzustände nicht hervortreten. Durch die Verwendung der beiden Triebwerksgeneratoren und eines weiteren Generators erreicht die rechte Gruppe von Architekturen weit bessere Ergebnisse als die weiteren Konzepte. Die absoluten Zielwerte zeigen bezüglich der Kernfrage 1, dass nur ein Redundanzkonzept mit einem dritten Generator in der Lage ist die Sicherheitsanforderungen zu erfüllen; sowohl für die nominellen als auch für die degradierte Fehlerbedingung. Die weiteren Architekturen können folglich für die weiteren Betrachtungen vernachlässigt werden.

Durch die Klassifizierung der ersten Fehlerbedingung ergibt sich für das analysierte System zudem die minimal mögliche Systemmasse. Diese ist in Abbildung 6.9 dargestellt und verdeutlicht bezüglich Kernfrage 4 zudem den Einfluss der Fehlerklassifizierung auf die Systemmasse. Zudem ist die Reduktion des Architekturraumes durch eine maximale Systemmasse aufgrund der Systemanforderungen dargestellt.

Die nachfolgende Robustheitsuntersuchung mit $\varepsilon = 0,05$ erweitert die ermittelte PARETO-Menge um die in Tabelle 6.5 dargestellten Architekturen. Die Architekturen zeigen unter Berücksichtigung der Kernfrage 3, dass ein sekundärer Generatorpfad aufgrund der Sicherheitsanforderungen vorhanden sein muss. Die Entscheidung welcher Generator, K_5 oder K_6 verwendet wird, jedoch aufgrund der ähnlichen Zielwerte nicht durch die quantitativen Sicherheitanforderungen getrieben wird.

Tab. 6.5: Ausgewählte Architekturen des illustrativen Beispiels

PF_{real}	Struktur	$\mathcal{B}_1$ [1/FH]	$\mathcal{B}_2$ [1/FH]	$\mathcal{B}_3$ [kg]
ja		$9,006 \cdot 10^{-10}$	$3,500 \cdot 10^{-9}$	138
nein		$9,010 \cdot 10^{-10}$	$5,500 \cdot 10^{-9}$	143
ja		$9,002 \cdot 10^{-10}$	$2,500 \cdot 10^{-9}$	148
ja		$5,006 \cdot 10^{-10}$	$2,700 \cdot 10^{-9}$	148

Das illustrative Beispiel verdeutlicht den in dieser Arbeit verfolgten Ansatz der optimalen Redundanzallokation und die Anwendung der zuvor hergeleiteten Methoden: auf der einen Seite die Aufweitung des Architekturraumes durch die Einbringung der Freiheitsgrade, auf der anderen Seite die anschließende Reduktion anhand der Nebenbedingungen, der Ermittlung der Pareto-Menge und abschließend der hierarchischen Architekturauswahl. Im Vergleich zum konventionellen Entwicklungsprozess wird somit im Vorentwurf die vollständige Menge der möglichen Architekturen untersucht. Die Vielfalt ergibt sich hierbei durch die Kombinatorik der lokalen Entscheidungspunkte im Systementwurf; in diesem Beispiel die Entscheidung auf Komponentenebene zu den Generatoren und Umrichtern sowie die sekundären Redundanzkonzepte zum dritten Generator und den Querverbindungen.

6.5 Implementierung der Redundanzallokation

Die vorgestellte Assistenzfunktion zur Redundanzallokation komplexer Flugzeugsysteme wurde als Erweiterung der bestehenden Software SyRelAn (engl. *System Reliability Analysis*) des Instituts für Flugzeug-Systemtechnik umgesetzt [102, 126]. Nachfolgend wird der Aufbau der programmtechnischen Umsetzung erläutert. Dieses umfasst neben den Schnittstellen zu den bereits vorhandenen Programmkomponenten die prinzipielle Umsetzung der neuen Komponente CoSyOp (engl. *Complex System Safety and Reliability Optimization*).

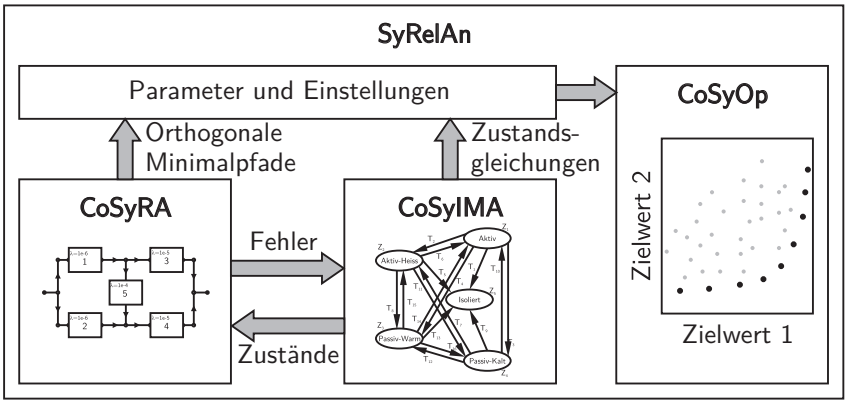


Abb. 6.10: Modularer Aufbau des Analysewerkzeugs SyRelAn

In Abbildung 6.10 ist der Aufbau der systemtechnischen Assistenzfunktion SyRelAn dargestellt, wie es den Funktionalitäten aus Abschnitt 3.1 und dem neuen Optimierungsverfahren entspricht. Weitere Funktionen, wie z.B. unterschiedliche Importanzanalysen oder die Leistungsdegradationsanalyse zustandsdiskreter Systemmodelle, sind in dieser Übersicht nicht dargestellt [94].

Das Programm SyRelAn koordiniert die unteren Module CoSyRA[1], CoSyIMA[2] und CoSyOp und kann als Schnittstelle zwischen den Modulen genutzt werden. Das Modul CoSyRA basiert auf den Arbeiten von Heidtmann, Merkel und Vahl und enthält die Modellierung mit Hilfe von Zuverlässigkeitsblockdiagrammen sowie unterschiedliche Analysefähigkeiten [43, 75, 126]. Das hybride Systemmodell unter Verwendung von nebenläufigen, endlichen Zustandsautomaten ist in dem Modul CoSyIMA umgesetzt und ermöglicht entsprechend Abschnitt 3.1.2 die vollständige Zustandsraumanalyse in Abhängigkeit logischer Abhängigkeiten der modellierten Ereignisse.

Das neu erstellte Modul CoSyOp ist in Abbildung 6.11 detailliert dargestellt und bildet die Umsetzung des zuvor vorgestellten Verfahrens zur Redundanzallokation komplexer Systemstrukturen.

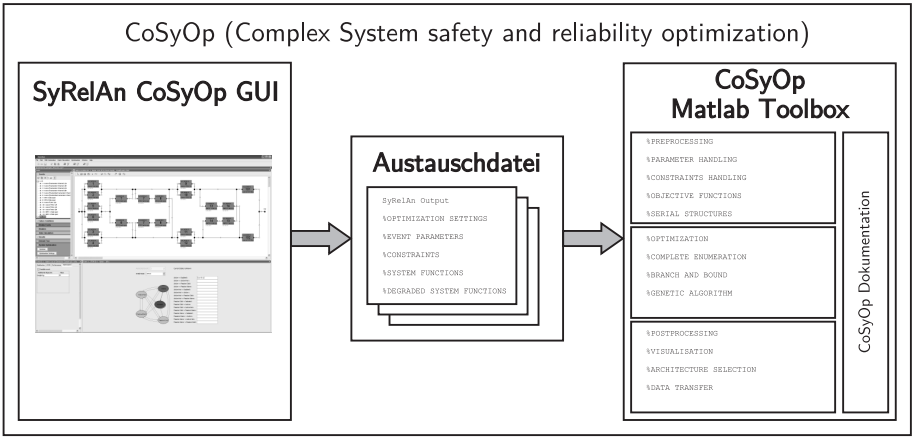


Abb. 6.11: Aufbau des Optimierungsmoduls CoSyOp

[1] CoSyRA: engl. *Complex System Reliability Analysis*
[2] CoSyIMA: engl. *Complex System Analysis of Integrated Modular Avionics based Systems*

Das Programm SyRelAn mit dem Modul CoSyOp wird dabei als grafische Oberfläche zur Definition des Optimierungsproblems und zur Eingabe aller Ereignis- und Problemparameter genutzt. Zudem werden innerhalb von SyRelAn die orthogonalisierten Minimalpfade und die vollständigen Zustandsraumgleichungen bestimmt. Die weitere Architekturoptimierung findet in dem gleichnamigen Werkzeug CoSyOp innerhalb des Programms Mathworks Matlab® statt. Als Schnittstelle zwischen SyRelAn und dem Werkzeug in Mathworks Matlab® wird eine Textdatei genutzt, die alle Problem- und Ereignisparameter enthält, ebenso die Nebenbedingungen und die orthogonalisierten Minimalpfade bzw. die Zustandsgleichungen. Das Matlab-Werkzeug CosyOp kann zudem über eine bereits vorhandenen Schnittstellendatei direkt angesprochen werden, ebenso die Visualisierung der Ergebnisse.

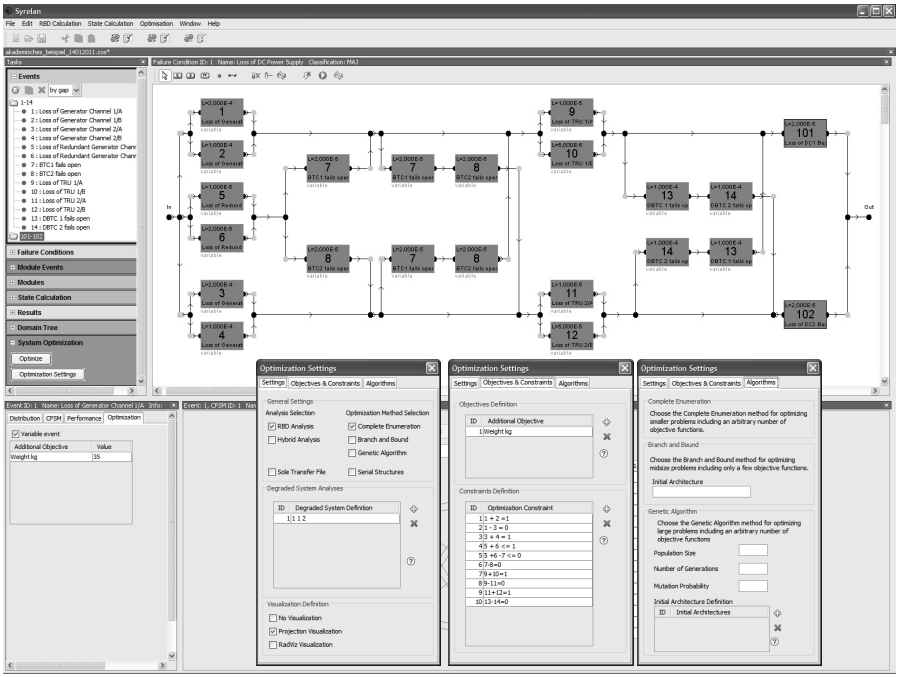


Abb. 6.12: Grafische Oberfläche des Werkzeugs SyRelAn und mögliche Einstellungen für die Erweiterung CoSyOp

In Abbildung 6.12 ist die grafische Oberfläche von SyRelAn sowie das mehrfach-redundante Systemmodell des illustrativen Beispiels aus dem vorherigen Abschnitt abgebildet. Zudem sind die einzelnen Einstellmöglichkeiten der CoSyOp-Erweiterung dargestellt. Diese umfassen links die Auswahl des zu verwendenen Algorithmus, die Definition degradierter Systemanalysen sowie die Auswahl der Visualisierungsmethode. In der Mitte ist die Eingabe der weiteren Zielgrößen entsprechend Abschnitt 4.5 dargestellt und die Eingabe der Nebenbedingungen nach Abschnitt 4.1. Rechts davon sind die spezifischen Einstellungen der Algorithmen aus Kapitel 5 enthalten, wobei die Eingabe von Anfangslösungen für das *Branch & Bound* Verfahren und des Genetischen Algorithmus hervorzuheben sind.

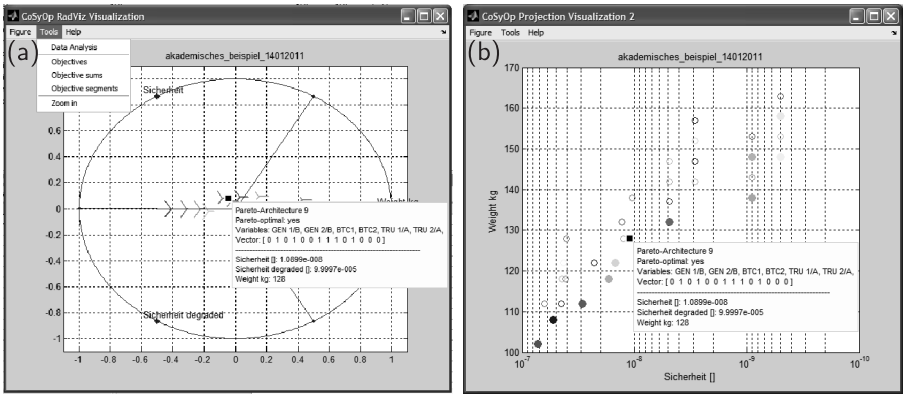


Abb. 6.13: Visualisierung der Optimierungsergebnisse mit Hilfe der Erweiterung CoSyOp, a) RadViz Darstellung, b) Projektion des Zustandsraums

In Abbildung 6.13 sind die Ergebnisse der beiden implementierten Visualisierungsmethoden anhand des illustrativen Beispiels dargestellt. Der linke Teil zeigt die Pareto-Menge visualisiert mit der RadViz-Methode aus Abschnitt 6.1. Neben der nichtlinearen Darstellung des $n_{\mathcal{B}}$-dimensionalen Architekturraums sind zudem weitere Informationen zu jeder Lösung darstellbar, zum Beispiel die absoluten Werte der Zielgrößen, um neben der relativen Darstellung der Visualisierung auch die absoluten Werte überprüfen zu können. Zudem lassen sich die Zielgrößen als Bezeichnung der Ankerpunkte, die Sektoren der Ankerpunkte und die Summen der ungewichteten Schenkellängen einblenden.

Letzteres unterstützt den Nutzer vor allem bei der Suche und Auswahl ausgeglichener Lösungen aus der Knieregion, vergleiche Abschnitt 6.2. In Abbildung 6.13 (b) ist eine Projektion der Ergebnisse dargestellt, wie sie von der CoSyOp-Erweiterung ausgegeben wird. Wie bereits zuvor sind auch hier als zusätzliche Informationen sämtliche Zielwerte zu einer Lösung abrufbar. Neben den absoluten Zielwerten umfasst dieses auch Informationen zur Dominanz der Lösung und eine Beschreibung der Architektur. Mit Hilfe der Architekturnummer ist es im Anschluss möglich über weitere Befehle den Architekturraum zusätzlich einzuschränken.

7 Anwendung der Methode zur optimalen Redundanzallokation

Als Nachweis der Funktions- und Leistungsfähigkeit der entwickelten Methode wird im Folgenden die Redundanzallokation auf ein industrielles Beispiel angewendet. Für ein zukünftiges Verkehrsflugzeug wird hierfür die Architektur des elektrischen Erzeugungs- und Verteilungssystems[1] unter Berücksichtigung eines Brennstoffzellensystems untersucht. Beginnend mit einer Einführung in den Aufbau des betrachteten elektrischen Netzes wird anschließend die Modellbildung mittels des entwickelten MRS und die Auswahl des Optimierungsverfahrens betrachtet. Am Ende des Kapitels folgt die Interpretation der Optimierungsergebnisse. Diese enthält auch die Diskussion und Auswahl der Systemarchitekturen und nachfolgend eine Betrachtung des Verhaltens des ausgewählten Optimierungsverfahrens.

Das hier betrachtete zweistrahlige Kurzstreckenflugzeug soll über eine erweiterte elektrische Systemversorgung (engl. *More Electric Aircraft*) verfügen, wie sie seit einigen Jahren in unterschiedlichen Forschungsvorhaben und mit der Boeing 787 auch industriell verfolgt wird [32, 52, 79, 112]. Zur Abdeckung des hohen elektrischen Leistungsbedarfs von Klimaanlagen ohne Zapfluftversorgung und des Enteisungssystems versorgen die Triebwerksgeneratoren primär ein Hochspannung-Gleichstromnetzwerk (engl. *High Voltage Direct Current, HVDC*) mit beispielsweise 270 VDC [33, 99]. Mit Hilfe von Umrichtern werden zwei 28 VDC Sammelschienen versorgt, die beispielsweise die Avionik versorgen. Neben diesem normalen Netzwerk ist zudem ein essentielles Netzwerk vorgesehen, das die Versorgung der kritischsten Verbraucher sicherstellt und das auch nach einem Funktionsverlust der Triebwerksgeneratoren von den sekundären Generatoren versorgt wird. Das essentielle Netzwerk verfügt ebenfalls über eine *HVDC* und eine 28 VDC Sammelschiene. Abbildung 7.1 veranschaulicht den prinzipiellen Systemaufbau mit den primären Generatoren im oberen, den sekundären Generatoren im unteren Teil. Die betrachteten Sammelschienen sind im mittleren Teil dargestellt. Weitere Sammelschienen, wie zum Beispiel mit

[1] Die Bezeichnung der Subsysteme ist an die englische Bezeichnung *Electrical Generation and Distribution System* angelehnt.

einer 115/200 VAC, 400 Hz Versorgung, sind ebenfalls vorgesehen, werden aufgrund der dimensionierenden Fehlerfälle jedoch nicht betrachtet. Erzeugerseitig werden die Batterien zudem nicht für den Architekturentwurf berücksichtigt, da diese zeitlich limitierte Generatoren darstellen und somit nur kurzzeitig elektrische Leistung im Flug zur Verfügung stellen, beispielsweise für die Dauer des Ausfahrens und Anlaufens einer elektrischen Staulufturbine [94].

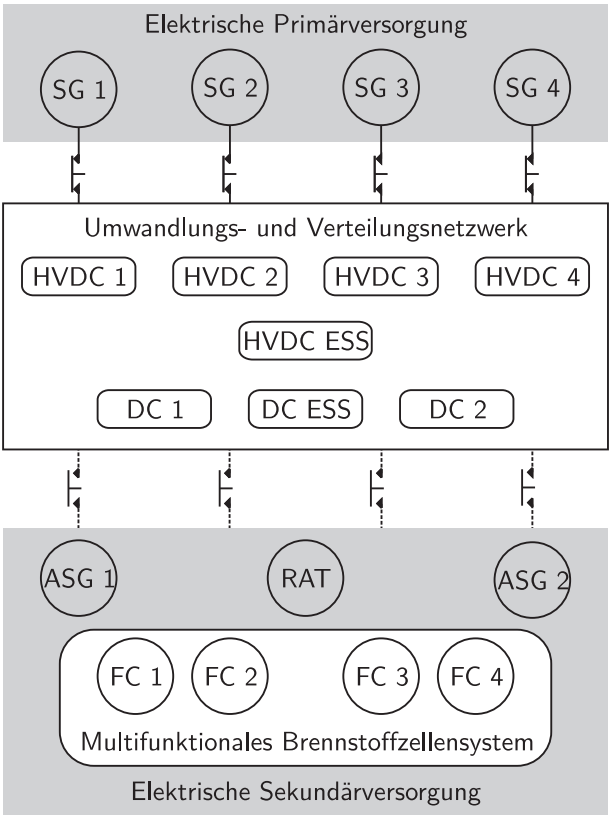


Abb. 7.1: Primäre und sekundäre Leistungsversorgung und -verteilung

Das zu untersuchende Energieversorgungssystem besteht primärseitig aus vier Starter/Generatoren (SG), die über Gleichrichter (engl. *Rectifier Units, RU*)

in das *HVDC*-Netz speisen [33, 99]. Die primäre Energieversorgung wird für die spätere Optimierung als fest definiert. Im Rahmen der Architekturuntersuchungen werden unterschiedliche Konzepte zur sekundären Energieversorgung betrachtet. Dieses umfasst zwei elektrische Generatoren, versorgt über eine konventionelle Hilfsgasturbine (engl. *Auxiliary Power Unit, APU*), und einen Notgenerator angetrieben von der Staulufttturbine (engl. *Ram Air Turbine, RAT*). Hierbei handelt es sich um eine konventionelle Architektur, wie sie bisher in zahlreichen elektrischen Systemen umgesetzt wurde. Der Einsatz der *APU* im Flug ist jedoch nicht bei jeder Realisierung dieser Systeme vorgesehen [79, 124]. Neben diesem konventionellen Konzept wird der Übergang zur Einbringung einer zu definierenden Anzahl von Brennstoffzellen (engl. *Fuel Cells, FC*) zur Ergänzung beziehungsweise zum Ersatz der bisherigen Generatoren untersucht [71]. Die erzeugerseitige Referenzarchitektur stellt dabei die konventionelle Lösung mit der genannten Primärseite und sekundärseitig mit zwei APU-Generatoren (engl. *Auxiliary Starter/Generators, ASG*) und einer elektrischen Staulufttturbine ohne Brennstoffzellensystem dar [52].

Neben der Untersuchung unterschiedlicher Erzeugerkonzepte wird zudem das Energieverteilungssystem analysiert. Dieses dient gemäß EASA CS §25.1351 zur Verteilung der elektrischen Lasten auf die primären und sekundären Generatoren und gewährleistet eine Rekonfiguration des Netzes zur Abdeckung von Fehlerfällen [31]. Das Verteilungsnetzwerk ist in Abbildung 7.1 nicht dargestellt, es handelt sich hierbei um eine übliche *Split-Bus* Architektur, wie sie auch in bisherigen Flugzeugen verwendet wird [124]. Im Normalfall versorgt jeder Triebwerksgenerator einen Leistungspfad, wobei die einzelnen Pfade galvanisch voneinander getrennt sind. Das essentielle Netzwerk wird in diesem Fall von einem der Systempfade über eine Querverbindung gespeist. Daneben bestehen im Fehlerfall unterschiedliche Möglichkeiten elektrische Leistung über Querverbindungen von einem Erzeugerpfad zu einen anderen zu transferieren. Hierbei wird jede Querverbindung durch zwei Schalter realisiert, so dass kein einfacher Fehler zu einer Verbindung von zwei Systemseiten führen kann, was vor allem bei Netzen variabler Frequenz zum Systemverlust führen kann. Die Rekonfigurationsmöglichkeiten im Verteilungsnetz sind neben dem sekundären Erzeugerkonzept Gegenstand des Architekturentwurfs.

7.1 Modellbildung

Gemäß Kapitel 2 basieren die betrachteten Zielgrößen der Architekturoptimierung auf den Ergebnissen einer vorläufigen Gefahrenanalyse (*PFHA*), die auf Grundlage der Systemfunktionen und möglicher Realisierungen die wesentlichen Fehlerbedingungen klassifiziert. In diesem Fall wurden die nachfolgenden acht dimensionierenden Fehlerbedingungen identifiziert und gemäß EASA CS §25.1309 klassifiziert. Hierbei handelt es sich um sieben nominelle Systemzustände und zur Überprüfung der Fehlertoleranz des Systems bezüglich der primärseitigen Generatoren wird als Kandidat für eine *MMEL* Bedingung der degradierte Zustand mit einem ausgefallenen Triebwerksgenerator betrachtet. Neben den allgemeinen Systemanforderungen entsprechend EASA CS §25.1309 sind die Zulassungsvorschriften elektrischer Energieversorgungssysteme in den Paragraphen EASA CS §25.1351 bis §25.1365 geregelt.

Tab. 7.1: Dimensionierende Fehlerbedingungen des industriellen Beispiels

Nr.	Fehlerbedingung	Klass. EASA CS §25.1309
1	Verlust der elektrischen Energieversorgung	CAT
2	Verlust der normalen Energieversorgung	MAJ
3	Verlust der Notfall-Energieversorgung	MAJ
4	Verlust eines Systempfades	MAJ
5	Verlust der Sammelschiene DC ESS	MAJ
6	Verlust der Sammelschienen HVDC ESS	MAJ
7	Verlust einer normalen Sammelschiene	MIN
8	Verlust der elektrischen Energieversorgung, $\mathbf{K}(SG1) = 0$	CAT

Zusätzlich zu den Sicherheitsanforderungen wird zudem die operationalle Zuverlässigkeit des Systems abgeschätzt, was als wesentliches Merkmal für einen wirtschaftlich erfolgreichen Flottenbetrieb auch einen Einfluss auf die Systemarchitektur hat [80]. Als konträre Zielgröße zu den Redundanzkonzepten wird desweiteren die Systemmasse abgeschätzt.

Nochfolgend sind einige der prinzipiellen Zuverlässigkeitsblockdiagramme des mehrfach-redundanten Systemmodells der genannten Fehlerbedingungen dargestellt. Diese umfassen die erste Ebene des MRS und verdeutlichen die Ausfalllogik auf Subsystemebene. Die detaillierten Konzepte und deren Fehlerauswirkungen können aus Gründen der Vertraulichkeit nicht dargestellt werden. Die Zuverlässigkeitsblockdiagramme der weiteren Fehlerbedingungen sind in Anhang B enthalten, wobei die Systemstrukturfunktion für die Fehlerbedingung 8 automatisch aus der Fehlerbedingung 1 entsprechend des als ausgefallen deklarierten Generators abgeleitet wird.

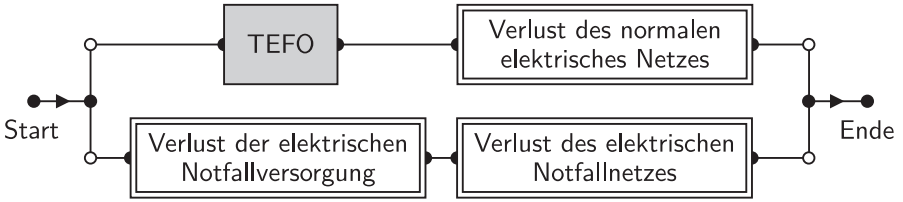


Abb. 7.2: Zuverlässigkeitsblockdiagramm für die Fehlerbedingung *Verlust der elektrischen Energieversorgung*

Die Fehlerbedingung 1 in Abbildung 7.2 enthält neben dem variablen Zuverlässigkeitsblockdiagrammen der Fehlerbedingungen 2 im oberen Pfad zudem das externe Ereignis eines Funktionsverlust aller Triebwerke (engl. *Total Engine Flame-Out, TEFO*). Das *TEFO*-Ereignis wird dabei als externes Ereignis betrachtet, da verschiedene externe Ereignisse (*Particular Risks*), beispielsweise Vulkanasche oder Vogelschlag, zu einem Verlust aller Triebwerke führen können. Die hierfür akzeptierten Eintrittswahrscheinlichkeiten werden entweder von der Zulassungsbehörde vorgegeben oder sind mit dieser zu klären. Der untere Pfad betrachtet den simultanen Ausfall der Notenergieversorgung, die je nach Architektur aus einer konventionellen Staulufturbine oder einem Brennstoffzellensystem besteht.

In Abbildung 7.3 ist die Fehlerbedingung *Verlust der normalen Energieversorgung* dargestellt. Dieses Ereignis wird auch häufig auch als *LMES* (engl. *Loss of Main Electrical System)* bezeichnet [103]. Im Gegensatz zu dem oberen Pfad des vorherigen Zuverlässigkeitsblockdiagramms werden in dieser Fehlerbedingung nur die innersystemischen Ereignisse berücksichtigt. Dieses betrifft die nominelle Versorgung der Sammelschienen durch die Triebwerksgeneratoren entspre-

chend der Zuordnung durch die *Split-Bus* Architektur. Zudem berücksichtigt die Versorgung der Sammelschienen die Freiheitsgrade der Leistungsverteilung und sekundären Energieversorgung. Es sind somit mögliche Querverbindungen zwischen den einzelnen Erzeugerpfaden modelliert, die sich durch die jeweiligen Einträge im Architekturvektor addressieren lassen. Zudem sind die Konzepte zur Versorgung der *HVDC* Sammelschienen durch die sekundären Generatoren abgebildet. Bei der abgebildeten Struktur der Fehlerbedingung durch das MRS handelt es sich um eine quadruplex Struktur. Diese kann sich jedoch aufgrund der Ausfalllogiken der Systemfunktionen signifikant reduzieren, so dass im komplexen Systementwurf auch singuläre Ereignisse möglich sind, die zu einem katastrophalen Ereignis führen.

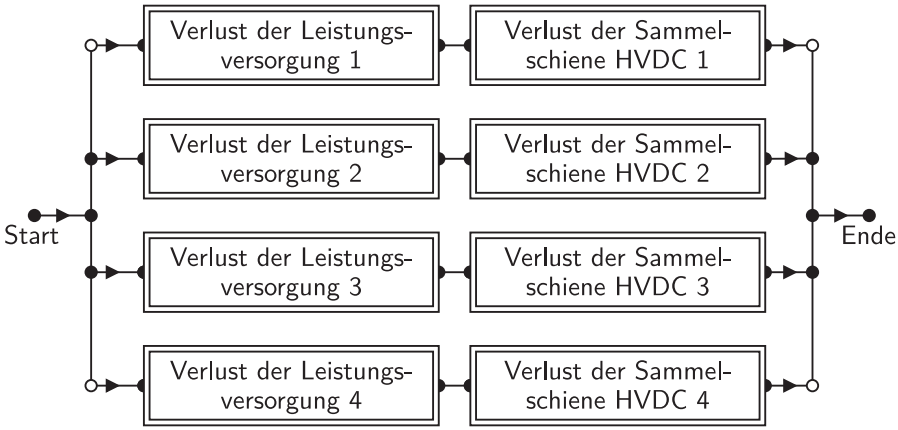


Abb. 7.3: Zuverlässigkeitsblockdiagramm für die Fehlerbedingung *Verlust der normalen Energieversorgung*

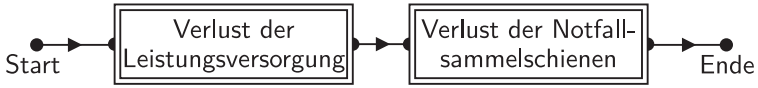


Abb. 7.4: Zuverlässigkeitsblockdiagramm für die Fehlerbedingung *Verlust der Notfall-Energieversorgung*

Entsprechend der Fehlerbedingung 3 ist in Abbildung 7.4 das Zuverlässigkeitsblockdiagramm für den Verlust der elektrischen Notfallversorgung dargestellt.

Diese muss gemäß EASA CS §25.1307(b) über eine Energiequelle verfügen, die unabhängig von den Triebwerken und der Treibstoffversorgung ist [31]. Je nach ausgewähltem Konzept besteht die Versorgung des essentiellen elektrischen Netzes im Notfall aus einer Staulufftturbine oder einer variablen Anzahl von Brennstoffzellen. Im normalen Systemzustand wird das essentielle System jedoch über Querverbindungen oder dedizierte, nicht essentielle Brennstoffzellen versorgt, dieses wird in der Analyse der Fehlerbedingung berücksichtigt. In diesem Fall wird nur der Verlust der Notversorgung im normalen Flugzustand berücksichtigt, es liegt somit noch kein *TEFO*- oder *LMES*-Ereignis vor und die Staulufftturbine darf hierfür nicht berücksichtigt werden. Da die sicherheitskritischen und redundanten Verbraucher stets mindestens über eine normale Sammelschiene und eine essentielle Sammelschiene versorgt werden, ist die Funktionsfähigkeit auch nach Ausfall des essentiellen System gewährleistet. Dieses begründet auch die Klassifizierung *MAJ*.

Die prinzipiellen Zuverlässigkeitsblockdiagramme der weiteren vier Fehlerbedingungen sind in Anhang B enthalten. Die Systemstrukturfunktion für die Fehlerbedingung 8 wird entsprechend der entwickelten Methode zur Ableitung degradierter Systemfunktionen automatisch von der Zielfunktion $\mathcal{A}_1(\mathbf{x}, t)$ abgeleitet.

Neben den sicherheitsrelevanten Zuverlässigkeitsblockdiagrammen wird zudem die operationelle Zuverlässigkeit der primären und sekundären Seite untersucht. Dieses Kriterium ist nicht relevant für die Zulassung des Flugzeugs, es verdeutlicht jedoch die Fehlertoleranz des Systems hinsichtlich eines zuverlässigen Betriebs. Das Ziel der Untersuchung ist somit nicht die Klassifizierung und Quantifizierung hinsichtlich einer Richtlinie, sondern der relative Vergleich der Architekturen zueinander und die Identifizierung singulärer Punkte einer Architektur, die im Fehlerfall zu einer Verspätung oder zu einer Stornierung eines Fluges führen könnten.

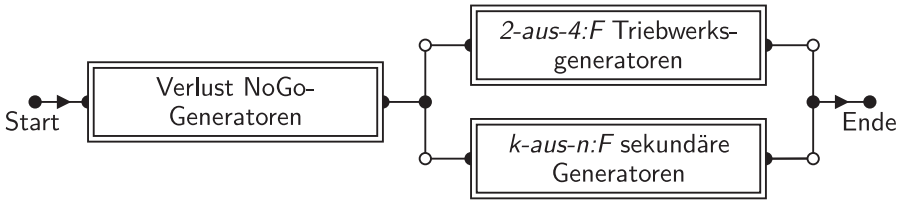


Abb. 7.5: Zuverlässigkeitsblockdiagramm für die Untersuchung der Systemdegradation

Abbildung 7.5 zeigt das Zuverlässigkeitsblockdiagramm für das Systemereignis „*Unzulässige Degradation der primär- und sekundärseitigen Generatoren*“. Singuläre Pfade, wie der Funktionsverlust der Staulufttturbine, führen zu einer unzulässigen Degradation der Generatoren, da eine sichere und unabhängige Versorgung des essentiellen Versorgungssystem nicht mehr gegeben wäre. Die konventionelle Staulufttturbine stellt dabei einen schlafenden Fehler im System dar [56]. Die weiteren Generatoren und Brennstoffzellensysteme hingegen bilden ein *k-aus-n:F* System, hierbei dürfte auch eine Komponente ausgefallen sein, wie es beispielsweise die MMEL für die Airbus A320 regelt [35]. Dieses setzt jedoch voraus, dass mehr als eine Brennstoffzelle das essentielle Netz versorgen kann, da es sich ansonsten ebenfalls um einen singulären Pfad handeln würde. Die variablen singulären Ereignisse werden dabei mittels einer seriellen Struktur gemäß Abschnitt 4.3 modelliert.

Das mehrfach-redundante Systemmodell für diese Optimierung wird über alle Fehlerbedingungen durch 26 variable und 36 feste Ereignisse in acht Zuverlässigkeitsblockdiagrammen beschrieben, wobei zusätzlich eine weitere Fehlerbedingung abgeleitet und die Systemmasse abgeschätzt wird. Der unbeschränkte Architekturraum von $2^{26} \approx 6,7 \cdot 10^7$ Architekturen wird mit Hilfe von 31 Nebenbedingungen auf einen gültigen Raum von 55296 Architekturen beschränkt. Dabei wurden 14 Konzepte der sekundären Erzeugerarchitektur vorgegeben, zusätzlich wurde die Architektur der Leistungsverteilung variiert. Die tatsächliche PARETO-Front PF_{real} mit 226 Lösungen wurde in ca. 27h mit Hilfe der vollständigen Enumeration bestimmt.

Das erstellte mehrfach-redundante Systemmodell wurde mit Hilfe der vorgegebenen Erzeugungskonzepte validiert, wie es bereits in Abschnitt 4.1 erläutert wurde. Werden die Optimierungsvariablen des Verteilungsnetzwerks konstant gehalten, können sowohl die Vollständigkeit der Nebenbedingungen als auch die Sinnhaftigkeit der Modellierung überprüft werden. Unvollständige Nebenbedingungen führen dabei zu einer erhöhten Anzahl von Erzeugungskonzepten. Ebenfalls ist zu überprüfen, ob die richtigen Architekturen erzeugt werden. Neben der Validierung des beschränkten Architekturraums kann zudem die Sinnhaftigkeit der Modellierung überprüft werden. Architekturen mit Zielwerten $\mathcal{B}_i(\mathbf{x}, t = 1\mathsf{FH}) = 0$ verdeutlichen, dass die untersuchte Systemfunktion für die gegebene Architektur nicht funktionsfähig ist und durch das Streichen der orthogonalisierten Minimalpfade im Optimierungsprozess alle Minimalpfade gestrichen wurden. Der letzte Fall kann durch Unterscheidung der seriellen und parallelen Logiken entsprechend Abschnitt 4.3 vermieden werden.

7.2 Lösungsinterpretation

Das Optimierungsproblem wurde aufgrund der Variantenanzahl und der vorherigen Komplexitätsuntersuchungen mit Hilfe des in Abschnitt 5.2.3 vorgestellten und angepassten Genetischen Algorithmus NSGA-II gelöst. Als Referenz zur Entwicklung der Heuristik dient der vollständige Architekturraum, der mit Hilfe der vollständigen Enumeration ermittelt wurde.

Die Interpretation der Optimierungsergebnisse unterteilt sich in zwei Abschnitte. Zunächst wird die Eingrenzung des Architekturraumes und die Auswahl von Pareto-optimalen Architekturen mit Hilfe der berechneten Zielwerte vorgestellt. Anschließend wird die Entwicklung der Lösungsmenge anhand der Populationen des Genetischen Algorithmus und dessen Konvergenzverhalten für das Optimierungsproblem untersucht.

7.2.1 Diskussion der Systemarchitekturen und Optimierungsergebnisse

Das Optimierungsproblem wurde mit dem vorgestellten mehrfach-redundanten Systemmodell modelliert und mittels Heuristik gelöst. Hierfür wurden zur Initialisierung 14 Architekturen vorgegeben, die unterschiedliche Konzepte zur Energieerzeugung darstellen. Das Energieverteilungssystem wurde weitgehend unbeschränkt belassen. Tabelle 7.2 enthält die Parameter und Ergebnisse des Optimierungslaufes im Vergleich zur vollständigen Enumeration.

Tab. 7.2: Parameter und Einstellungen des Algorithmus NSGA-II, erster Lauf

Einstellung/Ergebnis	NSGA-II	VE
Populationsgröße n_P	1000	-
Generationen n_G	50	-
Mutationswahrscheinlichkeit [%]	10	-
Größe PF_1	125	226
Qualität PF_1 [%]	25	100
Anzahl weiterer Lösungen	299	55070
Berechnungszeit [s]	990	97223

In Abschnitt 7.2.2 wird die gute Konvergenz der ermittelten PARETO-Front zur tatsächlichen nicht-dominierten Menge dargestellt, ebenfalls die Divergenz und Streuung der ermittelten PARETO-Front. Abbildung 7.6 zeigt jedoch, dass zahlreiche Lösungen außerhalb des zulässigen Bereichs gemäß der Klassifizierung der Fehlerbedingungen liegen. Gemäß Tabelle 7.2 wurden 125 nicht-dominierte Architekturen vom Genetischen Algorithmus ermittelt. Hiervon erfüllen jedoch nur 77 Architekturen die Sicherheitsanforderungen aufgrund der Fehlerbedingungen *Verlust der Energieversorgung der Sammelschiene DC ESS* und *Verlust einer Systemseite*, die Bereiche sind in der nachfolgenden Grafik entsprechend der Numerierung im Auswahlprozess nach Abschnitt 6.2 hervorgehoben.

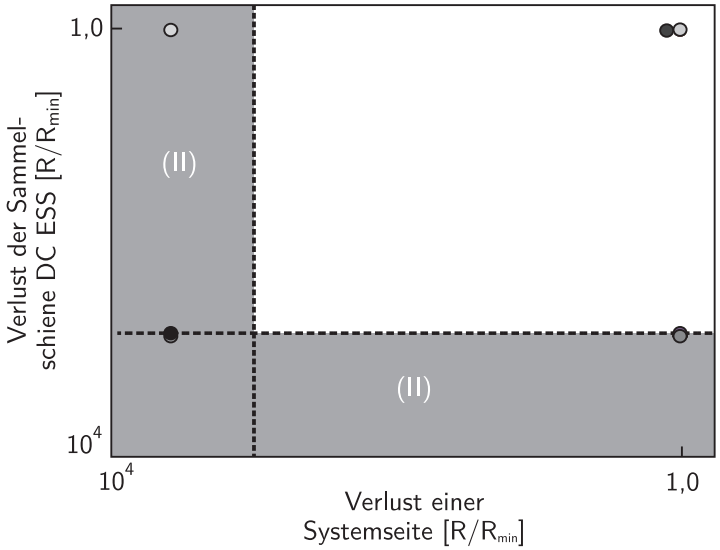


Abb. 7.6: Kartesische Projektion für zwei Zielwerte der ermittelten PARETO-Front mit dem unzulässigen Bereich

Aus diesem Grund wurde ein erneuter Optimierungslauf mit Hilfe der Heuristik gestartet. Für den zweiten Lauf wurde das Versorgungssystem der Initialarchitekturen so variiert, dass die Sicherheitsanforderungen für alle Initialarchitekturen erfüllt sind. Es zeigt sich somit, dass das implementierte Verfahren und die Möglichkeit zur Vorgabe von Initialarchitekturen dem Anwender eine *pro-*

gressive Nutzung des Genetischen Algorithmus ermöglichen. Tabelle 7.3 zeigt die Einstellungen und Ergebnisse für den zweiten Optimierungslauf.

Tab. 7.3: Parameter und Einstellungen des Algorithmus NSGA-II, zweiter Lauf

Einstellung/Ergebnis	NSGA-II	VE
Populationsgröße n_P	1000	-
Generationen n_G	100	-
Mutationswahrscheinlichkeit [%]	10	-
Größe PF_1	115	226
Qualität PF_1 [%]	33	100
Anzahl weiterer Lösungen	174	55070
Berechnungszeit [s]	2201	97223

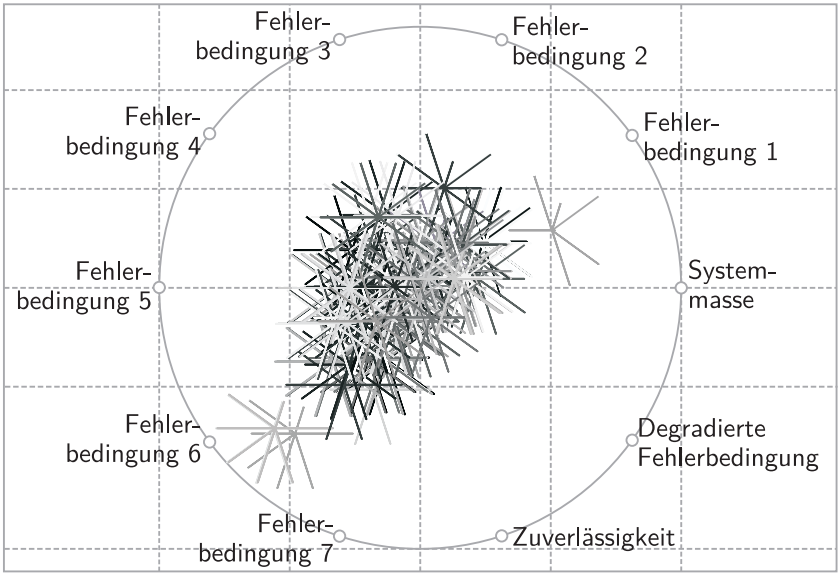


Abb. 7.7: Erweiterte RadViz-Darstellung der vollständigen ermittelten Pareto-Front

Auch wenn in diesem Fall die erste Pareto-Menge kleiner als die vorherige ermittelte Menge ist, zeigt sich, dass zum einen eine bessere Güte im Vergleich zu den Referenzwerten erreicht wurde und zum anderen mehr Lösungen die Sicherheitsanforderungen erfüllen. Dieses wurde hauptsächlich durch die Vorgabe einer Querverbindung im Energieverteilungssystem erreicht.

In Abbildung 7.7 ist die vollständige ermittelte Pareto-optimale Lösungsmenge mittels der RadViz-Darstellung abgebildet. Der unübersichtliche Lösungsraum zeigt deutlich, dass ein hierarchischer Auswahlprozesses notwendig ist, wie er zuvor vorgestellt wurde.

Die Erzeugerkonzepte, die in der ermittelten Pareto-Front enthalten sind, sind in Abbildung 7.8 mit identischen Verteilungsnetzen dargestellt und zeigen die konzeptionellen Vor- und Nachteile der Erzeugerarchitekturen. Exemplarisch werden im Folgenden drei Varianten des sekundären Energieerzeugungssystems betrachtet.

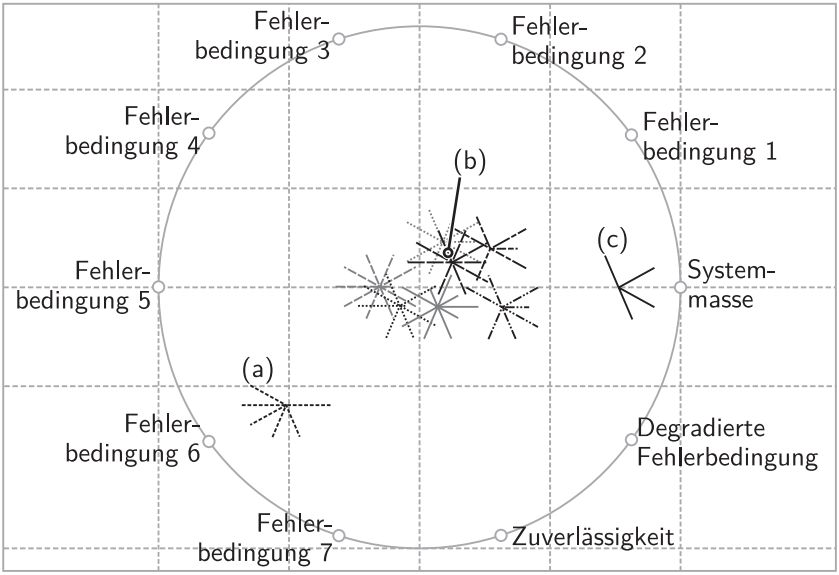


Abb. 7.8: Erweiterte RadViz-Darstellung der enthaltenen Konzepte in der ermittelten Pareto-Front

Konventionelle Architektur: Die Erzeugerarchitektur ist in Abbildung 7.8 als Architektur (a) hervorgehoben und in Abbildung 7.9(a) dargestellt. Die Architektur enthält eine konventionelle Staulufttturbine und zwei Generatoren, angetrieben von der Hilfsgasturbine. Dieses Erzeugerkonzept zeichnet sich durch eine geringe Systemmasse, verglichen mit den weiteren Architekturen, aus. Bezüglich der Systemzuverlässigkeit handelt es sich bei der Staulufttturbine jedoch um einen singulären Punkt, so dass hier die Ausfallrate der RAT aufgrund der hohen marginalen Importanz überwiegt. In der Fehlerbedingung 1 sowie weiteren Zielwerten erreicht dieses Konzept die schlechtesten Werte, solange jedoch die Sicherheitsanforderungen robust erfüllt werden, kann dieser Aspekt für die Architekturauswahl vernachlässigt werden. Die Überprüfung der Anforderungen ist mit Hilfe der hier dargestellten RadViz-Grafik nicht möglich, da die Lösungen nur relativ zueinander betrachtet werden. Stattdessen sollte im Entscheidungsprozess auf die Projektionen des Architekturraums oder den zuvor vorgestellten hierarchischen Auswahlprozess zurückgegriffen werden.

Duplex Brennstoffzellenarchitektur: Das Konzept der Erzeugerarchitektur ist in Abbildung 7.9(b) dargestellt und in Abbildung 7.8 als Konzept (b) markiert. Neben den Starter/Generatoren wird ein Brennstoffzellensystem bestehend aus zwei Brennstoffzellen genutzt, wobei beide nach einem Triebwerksausfall das Notfallsystem versorgen können [71]. Das Konzept zeichnet sich aufgrund der Fehlertoleranz der essentiellen Energieversorgung vor allem durch gute Zuverlässigkeitswerte aus. Zudem zeigt die Lage der Lösung und die Visualisierung der Federkräfte durch die erweiterte RadViz-Methode in Abbildung 7.8, dass es sich hierbei um eine ausgeglichene Lösung handelt. Abbildung 7.9 veranschaulicht dieses durch die Lage in der Knieregion der Projektion.

Hochredundante Brennstoffzellenarchitektur: Das elektrische Netz wird im Normalfall von vier Starter/Generatoren vorsorgt, in bestimmten Flugphasen können bis zu vier Brennstoffzellen das System neben den Starter/Generatoren oder ausschließlich versorgen. Die Architekturen weisen unterschiedliche Charakteristika auf, abhängig von der Anzahl der Brennstoffzellen und deren Möglichkeiten das elektrische Netz zu versorgen. Im Vergleich zu den vorherigen Erzeugerkonzepten weisen sie jedoch größtenteils höhere Systemmassen auf. Das Konzept ist in Abbildung 7.8 als Architektur (c) hervorgehoben, die Defizite durch die hohe Systemmasse werden hierbei durch die nicht vorhandene Schenkellänge in der erweiteren RadViz-Darstellung deutlich.

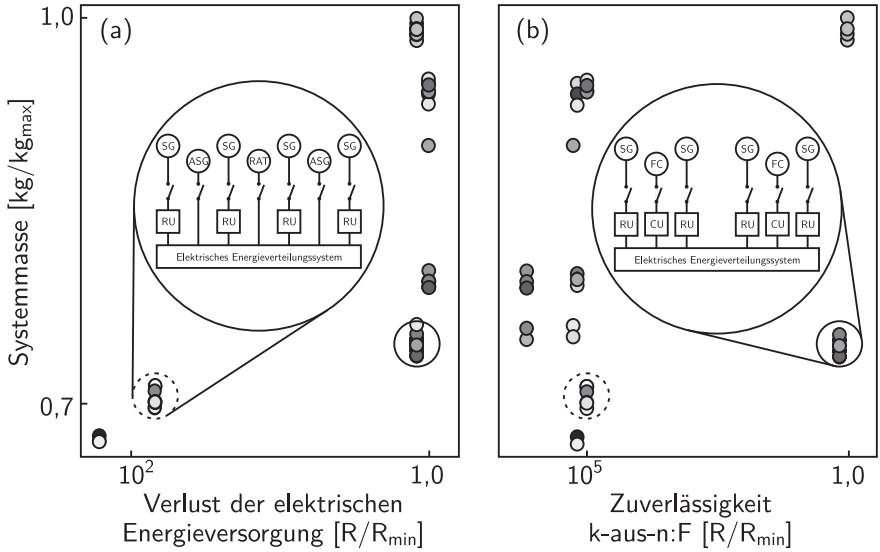


Abb. 7.9: Kartesische Projektion für drei Zielwerte der ermittelten PARETO-Front

Abbildung 7.9(a) zeigt die Projektion der ermittelten PARETO-Front für die Zielgröße „*Verlust der elektrischen Energieversorgung*“ und die relative Systemmasse. Dabei verdeutlichen sich die Eigenschaften der Architekuren anhand des kartesischen Koordinatensystems, wie sie bereits in Abbildung 7.8 gegenüber gestellt wurden. Die Konzepte konventioneller Architekturen mit einer Stauluftturbine und die Verwendung von zwei Brennstoffzellen sind in den Abbildungen 7.9(a) und (b) hervorgehoben.

Basierend auf der vollständigen ermittelten PARETO-Menge wird im Folgenden der hierarchische Auswahlprozess zur Reduktion des Zielwertraumes genutzt. Die Reduktion mit Hilfe der Sicherheitsanforderungen bringt keine weitere Reduktion, da aufgrund der Vorgabe der Initialmenge in der letzten Population nur noch Architekturen enthalten sind, die die Sicherheitsanforderungen erfüllen. Die Steuerung des Algorithmus in ein bevorzugtes Gebiet ist somit möglich, wobei die Streuung der Architekturen in dem nächsten Abschnitt betrachtet wird. Eine weitere Reduktion des Zielwertraumes auf Grundlage der bisherigen

Ergebnisse ist somit nur noch anhand der Massenabschätzung, der Systemzuverlässigkeit und der Robustheit der Sicherheitswerte möglich.

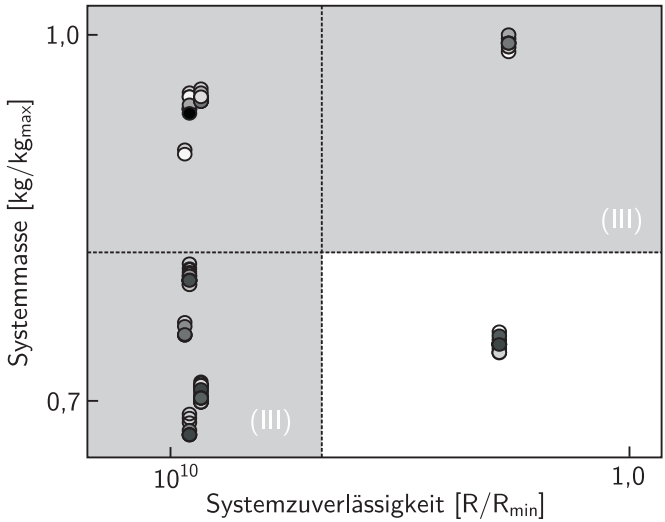


Abb. 7.10: Projektion der abgeschätzten Systemmasse und der Zuverlässigkeitswerte

In Abbildung 7.10 ist die kartesische Projektion für die Abschätzung der Systemmasse und die Berechnung der Systemzuverlässigkeit dargestellt. Für die Zuverlässigkeitswerte wird dabei nur indirekt eine quantitative Anforderung definiert; es darf kein singuläres Ereignis zu einer unzulässigen Degradation der sekundären oder primären Generatoren führen. Das Ziel ist in diesem Fall somit nicht nur eine fehlertolerante Architektur hinsichtlich der Systemsicherheit, sondern auch mit Bezug auf die Zuverlässigkeit und einer entsprechenden *Master Minimum Equipment List*. Aufgrund der klaren Trennung der erreichten Zuverlässigkeitswerte lässt sich somit eine Quantifizierung erreichen und der Zielwertraum teilen. Die Aufteilung zeigt somit auch, dass die Anforderung bezüglich der Sicherheit und Zuverlässigkeit nur mit den enthaltenen notwendigen Redundanzen erreicht werden können, was somit zu einer minimal möglichen Systemmasse führt. Die maximale Systemmasse wurde wiederum anhand der Systemanforderungen definiert und reduziert den Zielraum zusätzlich. Die

weiteren Projektionen des Zielwertraumes und der zugehörigen PARETO-Front sind in Anhang B dargestellt.

Bei den verbleibenden 11 Architekturen handelt es sich vollständig um das bereits vorgestellte Konzept mit zwei Brennstoffzellensystemen, die aufgrund ihrer Anbindung beide in der Lage sind als Notgeneratoren zu funktionieren. Im Gegensatz zu den anderen Konzepten bietet sich hierdurch auch eine gute Umsetzung der hohen Zuverlässigkeitsanforderungen. Die Unterscheidung der ermittelten Konzepte liegt somit in den Energieverteilungssystemen: hierbei zeigt sich mit Hilfe von zwei Sicherheitszielwerten eine weitere Varianz der betrachteten Architekturen, auch wenn bei den betrachteten Architekturen alle Zielwerte erfüllt werden. Auf Grundlage der dargestellten Projektion sollte die Architektur mit der geringsten Masse ausgewählt werden, zuvor sollten jedoch auch die weiteren Projektionen und somit Stärken und Schwächen der ausgewählten Architekturen betrachtet werden.

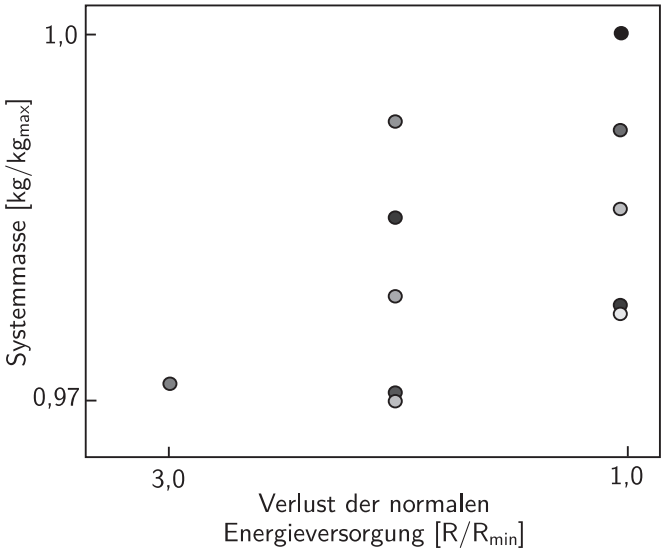


Abb. 7.11: Projektion der abgeschätzten Systemmasse und einer Sicherheitsbewertung für die ausgewählten Konzepte

Auf Grundlage der ausgewählten Konzepte ist eine Fortführung des bereits vorgestellten Entwicklungsprozesses möglich. Da die Varianz nur noch im Bereich

der Energieverteilung liegt, sind bereits für viele Fehlerfälle der primären und sekundären Energieversorgung detaillierte Analysen möglich und somit für die wesentlichen Aspekte des elektrischen Energieversorgungssystems.

7.2.2 Diskussion des Genetischen Algorithmus

Das industrielle Beispiel wurde aufgrund der Problemcharakteristika und den Erkenntnissen aus Kapitel 5 mit Hilfe des vorgestellten Algorithmus NSGA-II gelöst. Hierfür wurden zwei Optimierungsläufe durchlaufen, da sich nach dem ersten Optimierungslauf herausstellte, dass zahlreiche Architekturen nicht die erforderlichen Sicherheitsanforderungen erfüllen, vergleiche Abbildung 7.6.

Nachfolgend wird das Verhalten des Genetischen Algorithmus anhand des Beispielssystems erläutert, da dieses aufgrund der großen Beschränkungen durch die Nebenbedingungen und der Anzahl der Zielgrößen die üblichen Charakteristika von Problemen zur Redundanzoptimierung darstellt.

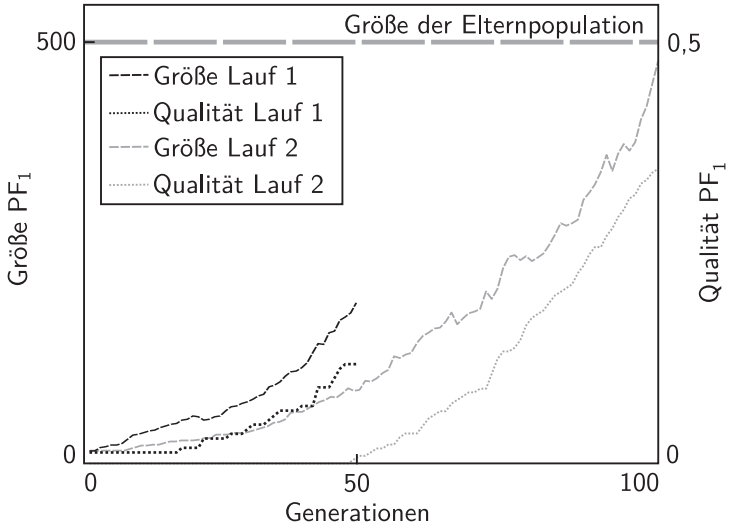


Abb. 7.12: Entwicklung der Anzahl der zulässigen Lösungen und Qualität der ermittelten PARETO-Front

Abbildung 7.12 zeigt für die beiden Optimierungsläufe die Entwicklung der ersten Pareto-Front und prozentual die enthaltenen Lösungen der realen nichtdominierten Menge, d.h. die Qualität der ermittelten Pareto-Front. Der erste Optimierungslauf enthält bereits in der Anfangspopulation Architekturen der realen Pareto-optimalen Menge. Dieses sind vor allem Lösungen, die durch die Initialpopulation vorgegeben wurden und aufgrund der einfachen Energieverteilungssysteme geringe Systemmassen aufweisen. In Abbildung 7.9 wurde jedoch bereits gezeigt, dass viele der ermittelten Architekturen bezüglich der Sicherheitsanforderungen nicht zulässig sind. Der implementierte NSGA-II hat somit bereits nach 992, 52s, vergleiche Tabelle 7.2, erste Ergebnisse geliefert. Ähnliche Ergebnisse hätten sich mit einer vollständigen Enumeration oder dem *Branch & Bound* Algorithmus erst nach dem vollständigen Durchlaufen ergeben, dieses unterstützt die Wahl des Genetischen Algortihmus für das Optimierungsproblem. Zudem ist deutlich zu erkennen, wie die Qualität der nicht-dominierten Menge monoton steigt, so dass nach einer endlichen Anzahl von Generationen die vollständige Pareto-Front ermittelt werden kann.

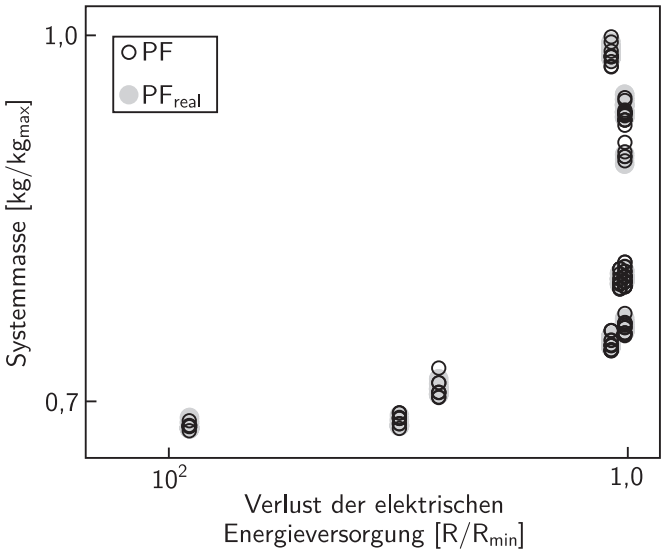


Abb. 7.13: Divergenz der ermittelten Pareto-Front im Vergleich zur tatsächlichen Pareto-Front

Für den zweiten Optimierungslauf wurde die Startpopulation aufgrund der Erkenntisse aus dem ersten Lauf geändert, so dass bereits notwendige Redundanzen eingebracht wurden. Zudem wäre es möglich, die Nebenbedingungen entsprechend der weiteren Redundanzen im Energieverteilungsnetz abzuändern. Dieses hätte jedoch den gültigen Architekturraum verändert und somit die Vergleichbarkeit mit den Ergebnissen der vollständigen Enumeration und des ersten Laufes beeinflusst. Der zweite Optimierungslauf startet ohne Architekturen der realen PARETO-optimalen Menge, findet im weiteren Verlauf jedoch verstärkt die optimalen und bezüglich der Sicherheit auch zulässigen Lösungen. Trotz der starken Beschränkung des vollständigen Architekturraums ermittelt der Algorithmus bereits nach 2201s ca. 33% der realen PARETO-optimalen Menge. In Abbildung 7.13 ist eine Projektion der ermittelten nicht-dominierten Menge mit der realen Menge dargestellt.

Es zeigt sich, dass der ermittelte Architekturraum nicht frühzeitig konvergiert ist, sondern die Randbereiche sehr gut abdeckt sind. Neben dem Konvergenzverhalten ist zudem die Streuung über den Architekturraum entscheidend, damit neben den Extremwerten auch die so genannte Knieregion gut abgedeckt ist. Auch hier zeigt sich, dass alle Erzeugerkonzepte der PARETO-Front, wie sie in Abbildung 7.8 identifiziert wurden, gefunden wurden. Der Unterschied zwischen PF_{real} und PF_{akt} liegt somit nur in den ermittelten Energieverteilungssystemen.

8 Zusammenfassung und Ausblick

Die gesteigerten Anforderungen an moderne Flugzeugsysteme führen zu einer erhöhten inner- und intersystemischen Komplexität, sowohl der Systeme selbst, als auch deren Entwicklung. Neben den funktionalen Systemforderungen müssen Flugzeugsysteme auch den Anforderungen bezüglich Sicherheit und Zuverlässigkeit genügen. Die quantitativen und qualitativen Anforderungen durch die Zulassungsvorschriften erfordern dabei für sicherheitskritische Funktionen eine fehlertolerante Systemarchitektur. Die möglichen Redundanzkonzepte dieser Architekturen unterscheiden sich durch verschiedene, lokale Strategien primärer, sekundärer und schadensvermeidender Redundanz. Hierbei wirkt sich jede Redundanzstrategie nicht nur auf sicherheits- und zuverlässigkeitstechnische Aspekte aus, sondern auch auf weitere Aspekte des Systementwicklungsprozesses und die Bewertung von Technologien. Der Entwicklungsprozess von Flugzeugsystemen betrachtet hierfür nach SAE ARP 4754 den Zusammenhang zwischen der Systementwicklung und der begleitenden Sicherheitsbewertung, die in der Vorentwicklung maßgeblichen Einfluss auf die Systemarchitektur hat.

Zusammen mit dem komplexen Entwicklungsprozess ist die Luftfahrtindustrie gefordert, herausfordernde Leistungs- und Umweltziele zu erreichen. Ein Weg hierfür ist die Entwicklung neuer Systemtechnologien, wie die Elektrifizierung von Flugzeugsystemen oder die Integration neuer Technologien, beispielsweise von Brennstoffzellen. Hierbei betrachtet die Bewertung der Technologien neben einer ökologischen auch stets eine ökonomische Verbesserung. Eine mögliche Reduktion der installierten Sekundärleistung an Bord führt dabei zu einer Zentralisierung von Flugzeugsystemen, wie sie auch aus ökonomischen Gründen in anderen Systembereichen untersucht wird. Die zukünftigen Flugzeugsysteme können aus diesen Gründen nur noch bedingt aus den vorherigen, konventionellen Flugzeugsystemen abgeleitet werden. Somit müssen auf der Ebene der Systemarchitekturen neue Konzepte entwickelt werden, die den Anforderungen hinsichtlich Sicherheit und Zuverlässigkeit gerecht werden. Aufgrund der unterschiedlichen Systemfunktionen und durch die Kombinatorik der möglichen Konzepte zur Umsetzung dieser Funktionen ergibt sich ein exponentiell wachsender Architekturraum.

Die Größe des potentiellen Lösungsraums und die Komplexität des umgebenden Entwicklungsprozesses erfordert daher eine Methode, die den Vorentwurf von fehlertoleranten Systemarchitekturen unterstützt und die Treiber der Systemarchitektur berücksichtigt: Sicherheit und Zuverlässigkeit. Der Vorentwurf erfordert dabei eine Quantifizierung der Architekturparameter, die eine transparente Entscheidungsfindung zulassen und anhand derer ein Systemingenieur die Menge der möglichen Architekturen derart reduzieren kann, dass für die verbleibenden Architekturen detaillierte Analysen zur Bewertung genutzt werden können, die auch systemspezifische Aspekte berücksichtigen. Hierbei zeigt sich, dass für die sicherheits- und zuverlässigkeitstechnische Bewertung von Systemarchitekturen vor allem komplexe Ausfalllogiken zu betrachten sind.

Im Rahmen dieser Arbeit wurde ein Bewertungswerkzeug entwickelt, das auf Grundlage eines erweiterten, variablen Strukturmodells die Zielwerte Sicherheit und Zuverlässigkeit in einer Architekturoptimierung berücksichtigt. Das verwendete Systemmodell nutzt hierfür das hybride Analysemodell von Vahl und Rehage unter Verwendung von Zuverlässigkeitsblockdiagrammen in einer oberen Modellierungsebene und hinterlegten nebenläufigen, endlichen Zustandsautomaten, die das zustandsdiskrete Verhalten und somit die Möglichkeiten zur Rekonfiguration des betrachteten Systems abbilden. Das Systemmodell wurde für die Optimierungsumgebung um die Möglichkeit zur Einbindung von Freiheitsgraden mit beliebigen Systemstrukturen erweitert. Anhand des entwickelten mehrfach-redundanten Systemmodells werden anschließend variable Systemstrukturfunktionen ausgelesen und im Verlauf der Optimierung unter Berücksichtigung konjunktiver Nebenbedingungen auf gültige Lösungen reduziert. Entsprechend des Entwurfsprozesses sicherheitskritischer Systeme werden hierbei nicht nur singuläre Fehlerbedingungen berücksichtigt, sondern die Menge der dimensionierenden Fehlerereignisse auf Grundlage der systemspezifischen vorläufigen, funktionalen Gefahrenanalyse und -klassifizierung. Neben den hieraus abgeleiteten Sicherheitsanforderungen können zudem Zuverlässigkeitsaspekte und konträre summative Systemparameter abgeschätzt werden. Desweiteren erlaubt der Modellierungsansatz die Ableitung degradierter Systemstrukturfunktionen, die es ermöglichen, für dedizierte Fehlerkombinationen die Fehlertoleranz zu überprüfen und somit die Kopplung von Sicherheit und Zuverlässigkeit bei redundanten Strukturen.

Auf Grundlage der Eigenschaften der Modellgleichungen des mehrfach-redundanten Systemmodells und der Beschreibung der gültigen Architekturen durch Nebenbedingungen wurde die Redundanzallokation als ein diskretes Optimierungsproblem definiert. Das Ziel hierbei ist die Ermittlung der nicht-

dominierten Architekturmenge, d.h. der PARETO-Menge. Somit kann der Systemingenieur die vollständige Architekturmenge derart reduzieren, dass eine überschaubare Variantenanzahl verbleibt, die für Detailanalysen geeignet ist. Als diskrete Optimierungsverfahren wurden hierbei gemäß des *No Free Lunch* Theorems von WOLPERT drei Ansätze untersucht. Für kleine bis mittlere Architekturräume eignet sich das entwickelte Verfahren zur vollständigen Enumeration auf Grundlage einer Breitensuche, wobei hier nur der vollständige, architekturell gültige Architekturraum untersucht wird. Als Verfahren für mittlere Architekturräume wurde ein mehrkriterielles *Branch & Bound* Verfahren entwickelt, das somit eine unvollständige Enumeration darstellt. Für große Optimierungsprobleme wurde aufgrund der Problemkomplexität als Heuristik ein Genetischer Algorithmus auf Grundlage des NSGA-II von DEB konditioniert. Der Ansatz angepasster, problemspezifischer Algorithmen wurde abschließend anhand eines variablen Beispielssystems validiert und sinnvolle Grenzen zur Verwendung der Algorithmen wurden angenähert.

Der Auswahlprozess von Systemarchitekturen ist nach SAE ARP 4754 stark an die Sicherheitsbewertung gekoppelt und integraler Bestandteil der Systementwicklung. Es wurde daher die Integration der entwickelten Methode in den weiteren Entwicklungsprozess untersucht. Dieses umfasst die Aufbereitung und Visualisierung mehrdimensionaler Architekturparameter, die Unterstützung des Systemingenieurs bei der weiteren Architekturraumreduktion sowie die Nutzung der Optimierungsergebnisse für den funktionalen Entwicklungsprozess. Die Visualisierungsmethode muss die ermittelte PARETO-optimale Menge dabei derart aufbereiten, dass eine Unterstützung des Entscheidungsfindungsprozesses möglich ist. Hierfür wurden mehrkriterielle Visualisierungsmethoden untersucht und eine bestehende nichtlineare Methode zum Konzeptvergleich verbessert. Der Schwerpunkt hierbei liegt auf der Darstellung der Stärken und Schwächen einer Architekur im Vergleich zur weiteren PARETO-Front. Neben der Visualisierungsmethode unterstützt zudem ein hierarchischer Auswahlprozess die weitere Entscheidungsfindung. Dieses umfasst zunächst eine Unsicherheitsbetrachtung mittels einer ε-PARETO-Front und anschließend die Reduktion des Zielwertraumes mittels der diskreten Sicherheitsanforderungen und den weiteren Anforderungen hinsichtlich Zuverlässigkeit und der abgeschätzten Systemparameter. Desweiteren wurde die Integration des Verfahrens in weitere Entwicklungsumgebungen betrachtet und eine Schnittstelle zum Auslesen der Architekturparameter geschaffen. Die Untersuchungen zur Integration des entwickelten Verfahrens schließen mit der Optimierung eines offen zugänglichen Beispielsystems.

Zur Validierung der Anwendbarkeit und Leistungsfähigkeit wurde abschließend ein industrielles Anwendungsbeispiel eines elektrischen Energieversorgungssystems untersucht. Die Freiheitsgrade des Systems stellen dabei die sekundärseitigen Generatoren dar, neben der konventionellen Versorgung mittels Stauluftturbine und Generatoren der Hilfsgasturbine wurden unterschiedliche Konzepte zur Integration einer Brennstoffzellenarchitektur in die Systemarchitektur betrachtet. Das Optimierungsproblem wurde durch insgesamt zehn Zielfunktionen definiert. Dabei handelte es sich um sieben dimensionierende Sicherheitfunktionen als Ergebnis der vorläufigen Gefahrenanalyse, um die Bewertung der Fehlertoleranz mittels einer abgeleiteten degradierten Sicherheitsfunktion, um die Untersuchung der erzeugerseitigen Systemzuverlässigkeit und um die Abschätzung der Systemmasse. Unter Verwendung des Genetischen Algorithmus konnte mit zwei Optimierungsläufen ein Drittel der tatsächlichen nichtdominierten Menge ermittelt werden. Die anschließende Reduktion des Zielwertraumes mit Hilfe des hierarchischen Auswahlprozesses hat elf Systemarchitekturen identifiziert, wobei diese jeweils über ein identisches Konzept zur Sekundärversorgung verfügen. Die wesentlichen Kriterien zur Auswahl des Architekturen waren, nach der Erfüllung der Zulassungsvorschriften, die Anforderung, dass kein singuläres Ereignis einen Start verzögert und die Begrenzung der maximalen Systemmasse. Eine abschließende Betrachtung zum Konvergenzverhalten des Algorithmus und zur Diversität des Zielwertraumes haben die Eignung der vorherigen Anpassungen am Algorithmus verifiziert.

Eine Fortsetzung des entwickelten Ansatzes stellt die vollständige Integration der Optimierungsumgebung in den weiteren Entwicklungsprozess dar. Somit könnte für sicherheitskritische Systeme der iterative Prozess zwischen Architekturdefinition, Sicherheitsnachweisen und Parametrisierung der Komponenten teilweise automatisiert und somit beschleunigt werden. Der Zusammenschluss funktionaler, dynamischer Simulationsmodelle mit Aspekten der Systemsicherheit birgt jedoch stets das Risiko das Prinzip einer unabhängigen Sicherheitsbewertung abzuschaffen. Unabhängig von den verwendeten Methoden und Werkzeugen ist eine unabhängige Sicherheitsbewertung für den Entwurf komplexer Flugzeugsysteme unabdingbar. Eine Möglichkeit die Unabhängigkeit der Entwurfsdisziplinen zu bewahren, wurde bereits in der vorliegenden Arbeit vorgestellt. Diese sieht die Nutzung variabler Architekturmodelle unterschiedlicher Disziplinen vor, die jeweils durch einen identischen Architekturvektor addressiert werden. Die Zusammenführung der einzelnen Ergebnisse disziplinspezifischer Analyseverfahren würde somit den Zielwertraum abbilden und beliebige

Zielwertanalysen gestatten. Die Eigenständigkeit der Sicherheitsanalysen bleibt durch die unabhängige Modellierung der variablen Systemmodelle bewahrt.

Neben der Berücksichtigung weiterer Systemparameter ist zudem eine Weiterentwicklung bezüglich der Sicherheitsbewertung sinnvoll. Die aktuelle Implementierung der Optimierungsmethodik ermöglicht bereits die weitere Verwendung der erstellten mehrfach-redundanten Systemmodelle im nachfolgenden Entwicklungsprozess. Eine stärkere Einbindung der Dokumentation und auch zur frühzeitigen Bewertung von Konzepten aufgrund von Anforderungen durch *Common Cause* Ereignisse würde den komplexen Entwicklungsprozess noch weitgehender unterstützen. Dieses würde somit den Übergang von einer sicherheits- und zuverlässigkeitstechnischen Architektur- hin zu einer Topologiebewertung darstellen.

A Illustratives Beispiel

Nachfolgend sind die orthogonalisierten Minimalpfade des illustrativen Beispiels aufgeführt. Anhand dieser Minimalpfade ist eine Aufstellung der benötigten Systemstrukturfunktion möglich und ein vollständiges Nachvollziehen des Beispiels.

$K_1 K_9 K_{101}$
$K_1 K_{10} K_{101} \bar{K}_9$
$K_2 K_9 K_{101} \bar{K}_1$
$K_2 K_{10} K_{101} \bar{K}_1 \bar{K}_9$
$K_3 K_{11} K_{102} \bar{K}_{101}$
$K_3 K_{11} K_{101} K_{102} \bar{K}_1 \bar{K}_2$
$K_2 K_3 K_{11} K_{101} K_{102} \bar{K}_1 \bar{K}_9 \bar{K}_{10}$
$K_1 K_3 K_{11} K_{101} K_{102} \bar{K}_9 \bar{K}_{10}$
$K_3 K_{12} K_{102} \bar{K}_{101} \bar{K}_{11}$
$K_3 K_{12} K_{101} K_{102} \bar{K}_1 \bar{K}_2 \bar{K}_{11}$
$K_2 K_3 K_{12} K_{101} K_{102} \bar{K}_1 \bar{K}_9 \bar{K}_{10} \bar{K}_{11}$
$K_1 K_3 K_{12} K_{101} K_{102} \bar{K}_9 \bar{K}_{10} \bar{K}_{11}$
$K_4 K_{11} K_{102} \bar{K}_{101} \bar{K}_3$
$K_4 K_{11} K_{101} K_{102} \bar{K}_1 \bar{K}_2 \bar{K}_3$
$K_2 K_4 K_{11} K_{101} K_{102} \bar{K}_1 \bar{K}_9 \bar{K}_{10} \bar{K}_3$
$K_1 K_4 K_{11} K_{101} K_{102} \bar{K}_9 \bar{K}_{10} \bar{K}_3$
$K_4 K_{12} K_{102} \bar{K}_{101} \bar{K}_3 \bar{K}_{11}$
$K_4 K_{12} K_{101} K_{102} \bar{K}_1 \bar{K}_2 \bar{K}_3 \bar{K}_{11}$
$K_2 K_4 K_{12} K_{101} K_{102} \bar{K}_1 \bar{K}_9 \bar{K}_{10} \bar{K}_3 \bar{K}_{11}$
$K_1 K_4 K_{12} K_{101} K_{102} \bar{K}_9 \bar{K}_{10} \bar{K}_3 \bar{K}_{11}$
$K_5 K_7 K_9 K_{101} \bar{K}_1 \bar{K}_2 \bar{K}_{102}$
$K_5 K_7 K_9 K_{101} K_{102} \bar{K}_1 \bar{K}_2 \bar{K}_3 \bar{K}_4$
$K_4 K_5 K_7 K_9 K_{101} K_{102} \bar{K}_1 \bar{K}_2 \bar{K}_3 \bar{K}_{11} \bar{K}_{12}$
$K_3 K_5 K_7 K_9 K_{101} K_{102} \bar{K}_1 \bar{K}_2 \bar{K}_{11} \bar{K}_{12}$
$K_5 K_7 K_{10} K_{101} \bar{K}_1 \bar{K}_2 \bar{K}_{102} \bar{K}_9$
$K_5 K_7 K_{10} K_{101} K_{102} \bar{K}_1 \bar{K}_2 \bar{K}_3 \bar{K}_4 \bar{K}_9$
$K_4 K_5 K_7 K_{10} K_{101} K_{102} \bar{K}_1 \bar{K}_2 \bar{K}_3 \bar{K}_{11} \bar{K}_{12} \bar{K}_9$
$K_3 K_5 K_7 K_{10} K_{101} K_{102} \bar{K}_1 \bar{K}_2 \bar{K}_{11} \bar{K}_{12} \bar{K}_9$

$K_5K_8K_{11}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4$
$K_5K_8K_{11}K_{101}K_{102}\bar{K}_1\bar{K}_2\bar{K}_3\bar{K}_4\bar{K}_7$
$K_5K_7K_8K_{11}K_{101}K_{102}\bar{K}_1\bar{K}_2\bar{K}_3\bar{K}_4\bar{K}_9\bar{K}_{10}$
$K_2K_5K_8K_{11}K_{101}K_{102}\bar{K}_1\bar{K}_9\bar{K}_{10}\bar{K}_3\bar{K}_4$
$K_1K_5K_8K_{11}K_{101}K_{102}\bar{K}_9\bar{K}_{10}\bar{K}_3\bar{K}_4$
$K_5K_8K_{12}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_{11}$
$K_5K_8K_{12}K_{101}K_{102}\bar{K}_1\bar{K}_2\bar{K}_3\bar{K}_4\bar{K}_7\bar{K}_{11}$
$K_5K_7K_8K_{12}K_{101}K_{102}\bar{K}_1\bar{K}_2\bar{K}_3\bar{K}_4\bar{K}_9\bar{K}_{10}\bar{K}_{11}$
$K_2K_5K_8K_{12}K_{101}K_{102}\bar{K}_1\bar{K}_9\bar{K}_{10}\bar{K}_3\bar{K}_4\bar{K}_{11}$
$K_1K_5K_8K_{12}K_{101}K_{102}\bar{K}_9\bar{K}_{10}\bar{K}_3\bar{K}_4\bar{K}_{11}$
$K_6K_7K_9K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_5$
$K_6K_7K_9K_{101}K_{102}\bar{K}_1\bar{K}_2\bar{K}_3\bar{K}_4\bar{K}_5$
$K_4K_6K_7K_9K_{101}K_{102}\bar{K}_1\bar{K}_2\bar{K}_3\bar{K}_{11}\bar{K}_{12}\bar{K}_5$
$K_3K_6K_7K_9K_{101}K_{102}\bar{K}_1\bar{K}_2\bar{K}_{11}\bar{K}_{12}\bar{K}_5$
$K_6K_7K_{10}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_5\bar{K}_9$
$K_6K_7K_{10}K_{101}K_{102}\bar{K}_1\bar{K}_2\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_9$
$K_4K_6K_7K_{10}K_{101}K_{102}\bar{K}_1\bar{K}_2\bar{K}_3\bar{K}_{11}\bar{K}_{12}\bar{K}_5\bar{K}_9$
$K_3K_6K_7K_{10}K_{101}K_{102}\bar{K}_1\bar{K}_2\bar{K}_{11}\bar{K}_{12}\bar{K}_5\bar{K}_9$
$K_6K_8K_{11}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_5$
$K_6K_8K_{11}K_{101}K_{102}\bar{K}_1\bar{K}_2\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_7$
$K_6K_7K_8K_{11}K_{101}K_{102}\bar{K}_1\bar{K}_2\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_9\bar{K}_{10}$
$K_2K_6K_8K_{11}K_{101}K_{102}\bar{K}_1\bar{K}_9\bar{K}_{10}\bar{K}_3\bar{K}_4\bar{K}_5$
$K_1K_6K_8K_{11}K_{101}K_{102}\bar{K}_9\bar{K}_{10}\bar{K}_3\bar{K}_4\bar{K}_5$
$K_6K_8K_{12}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_{11}$
$K_6K_8K_{12}K_{101}K_{102}\bar{K}_1\bar{K}_2\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_7\bar{K}_{11}$
$K_6K_7K_8K_{12}K_{101}K_{102}\bar{K}_1\bar{K}_2\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_9\bar{K}_{10}\bar{K}_{11}$
$K_2K_6K_8K_{12}K_{101}K_{102}\bar{K}_1\bar{K}_9\bar{K}_{10}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_{11}$
$K_1K_6K_8K_{12}K_{101}K_{102}\bar{K}_9\bar{K}_{10}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_{11}$
$K_1K_7K_8K_{11}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_6$
$K_1K_7K_8K_{11}K_{101}K_{102}\bar{K}_9\bar{K}_{10}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_6$
$K_1K_7K_8K_{12}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_6\bar{K}_{11}$
$K_1K_7K_8K_{12}K_{101}K_{102}\bar{K}_9\bar{K}_{10}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_6\bar{K}_{11}$
$K_1K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_8$
$K_1K_8K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_6\bar{K}_7$
$K_1K_7K_8K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_6\bar{K}_{11}\bar{K}_{12}$
$K_1K_6K_8K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_{11}\bar{K}_{12}$
$K_1K_5K_8K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_{11}\bar{K}_{12}$
$K_1K_4K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_{11}\bar{K}_{12}$
$K_1K_3K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_{11}\bar{K}_{12}$

$K_1K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_8\bar{K}_9$
$K_1K_8K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_6\bar{K}_7\bar{K}_9$
$K_1K_7K_8K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_6\bar{K}_{11}\bar{K}_{12}\bar{K}_9$
$K_1K_6K_8K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_{11}\bar{K}_{12}\bar{K}_9$
$K_1K_5K_8K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_{11}\bar{K}_{12}\bar{K}_9$
$K_1K_4K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_{11}\bar{K}_{12}\bar{K}_9$
$K_1K_3K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_{11}\bar{K}_{12}\bar{K}_9$
$K_2K_7K_8K_{11}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_6\bar{K}_1$
$K_2K_7K_8K_{11}K_{101}K_{102}\bar{K}_1\bar{K}_9\bar{K}_{10}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_6$
$K_2K_7K_8K_{12}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_6\bar{K}_1\bar{K}_{11}$
$K_2K_7K_8K_{12}K_{101}K_{102}\bar{K}_1\bar{K}_9\bar{K}_{10}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_6\bar{K}_{11}$
$K_2K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_8\bar{K}_1$
$K_2K_8K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_6\bar{K}_1\bar{K}_7$
$K_2K_7K_8K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_6\bar{K}_1\bar{K}_{11}\bar{K}_{12}$
$K_2K_6K_8K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_{11}\bar{K}_{12}\bar{K}_1$
$K_2K_5K_8K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_{11}\bar{K}_{12}\bar{K}_1$
$K_2K_4K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_{11}\bar{K}_{12}\bar{K}_1$
$K_2K_3K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_{11}\bar{K}_{12}\bar{K}_1$
$K_2K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_8\bar{K}_1\bar{K}_9$
$K_2K_8K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_6\bar{K}_1\bar{K}_7\bar{K}_9$
$K_2K_7K_8K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_6\bar{K}_1\bar{K}_{11}\bar{K}_{12}\bar{K}_9$
$K_2K_6K_8K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_{11}\bar{K}_{12}\bar{K}_1\bar{K}_9$
$K_2K_5K_8K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_{11}\bar{K}_{12}\bar{K}_1\bar{K}_9$
$K_2K_4K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_{11}\bar{K}_{12}\bar{K}_1\bar{K}_9$
$K_2K_3K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_{11}\bar{K}_{12}\bar{K}_1\bar{K}_9$
$K_3K_7K_8K_9K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_5\bar{K}_6$
$K_3K_7K_8K_9K_{101}K_{102}\bar{K}_1\bar{K}_2\bar{K}_{11}\bar{K}_{12}\bar{K}_5\bar{K}_6$
$K_3K_7K_8K_{10}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_5\bar{K}_6\bar{K}_9$
$K_3K_7K_8K_{10}K_{101}K_{102}\bar{K}_1\bar{K}_2\bar{K}_{11}\bar{K}_{12}\bar{K}_5\bar{K}_6\bar{K}_9$
$K_3K_{11}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_7$
$K_3K_7K_{11}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_5\bar{K}_6\bar{K}_8$
$K_3K_7K_8K_{11}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_5\bar{K}_6\bar{K}_9\bar{K}_{10}$
$K_3K_6K_7K_{11}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_5\bar{K}_9\bar{K}_{10}$
$K_3K_5K_7K_{11}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_9\bar{K}_{10}$
$K_2K_3K_{11}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_9\bar{K}_{10}\bar{K}_{102}$
$K_1K_3K_{11}K_{13}K_{14}K_{101}\bar{K}_9\bar{K}_{10}\bar{K}_{102}$
$K_3K_{12}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_7\bar{K}_{11}$
$K_3K_7K_{12}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_5\bar{K}_6\bar{K}_8\bar{K}_{11}$
$K_3K_7K_8K_{12}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_5\bar{K}_6\bar{K}_9\bar{K}_{10}\bar{K}_{11}$

$K_3K_6K_7K_{12}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_5\bar{K}_9\bar{K}_{10}\bar{K}_{11}$
$K_3K_5K_7K_{12}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_9\bar{K}_{10}\bar{K}_{11}$
$K_2K_3K_{12}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_9\bar{K}_{10}\bar{K}_{102}\bar{K}_{11}$
$K_1K_3K_{12}K_{13}K_{14}K_{101}\bar{K}_9\bar{K}_{10}\bar{K}_{102}\bar{K}_{11}$
$K_4K_7K_8K_9K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_5\bar{K}_6\bar{K}_3$
$K_4K_7K_8K_9K_{101}K_{102}\bar{K}_1\bar{K}_2\bar{K}_3\bar{K}_{11}\bar{K}_{12}\bar{K}_5\bar{K}_6$
$K_4K_7K_8K_{10}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_5\bar{K}_6\bar{K}_3\bar{K}_9$
$K_4K_7K_8K_{10}K_{101}K_{102}\bar{K}_1\bar{K}_2\bar{K}_3\bar{K}_{11}\bar{K}_{12}\bar{K}_5\bar{K}_6\bar{K}_9$
$K_4K_{11}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_7\bar{K}_3$
$K_4K_7K_{11}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_5\bar{K}_6\bar{K}_3\bar{K}_8$
$K_4K_7K_8K_{11}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_5\bar{K}_6\bar{K}_3\bar{K}_9\bar{K}_{10}$
$K_4K_6K_7K_{11}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_5\bar{K}_9\bar{K}_{10}\bar{K}_3$
$K_4K_5K_7K_{11}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_9\bar{K}_{10}\bar{K}_3$
$K_2K_4K_{11}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_9\bar{K}_{10}\bar{K}_{102}\bar{K}_3$
$K_1K_4K_{11}K_{13}K_{14}K_{101}\bar{K}_9\bar{K}_{10}\bar{K}_{102}\bar{K}_3$
$K_4K_{12}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_7\bar{K}_3\bar{K}_{11}$
$K_4K_7K_{12}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_5\bar{K}_6\bar{K}_3\bar{K}_8\bar{K}_{11}$
$K_4K_7K_8K_{12}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_5\bar{K}_6\bar{K}_3\bar{K}_9\bar{K}_{10}\bar{K}_{11}$
$K_4K_6K_7K_{12}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_5\bar{K}_9\bar{K}_{10}\bar{K}_3\bar{K}_{11}$
$K_4K_5K_7K_{12}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_9\bar{K}_{10}\bar{K}_3\bar{K}_{11}$
$K_2K_4K_{12}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_9\bar{K}_{10}\bar{K}_{102}\bar{K}_3\bar{K}_{11}$
$K_1K_4K_{12}K_{13}K_{14}K_{101}\bar{K}_9\bar{K}_{10}\bar{K}_{102}\bar{K}_3\bar{K}_{11}$
$K_5K_7K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_8\bar{K}_1\bar{K}_2$
$K_5K_7K_8K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_{11}\bar{K}_{12}\bar{K}_1\bar{K}_2$
$K_4K_5K_7K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_{11}\bar{K}_{12}\bar{K}_1\bar{K}_2$
$K_3K_5K_7K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_{11}\bar{K}_{12}\bar{K}_1\bar{K}_2$
$K_5K_7K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_8\bar{K}_1\bar{K}_2\bar{K}_9$
$K_5K_7K_8K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_{11}\bar{K}_{12}\bar{K}_1\bar{K}_2\bar{K}_9$
$K_4K_5K_7K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_{11}\bar{K}_{12}\bar{K}_1\bar{K}_2\bar{K}_9$
$K_3K_5K_7K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_{11}\bar{K}_{12}\bar{K}_1\bar{K}_2\bar{K}_9$
$K_5K_8K_{11}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_7\bar{K}_3\bar{K}_4$
$K_5K_7K_8K_{11}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_9\bar{K}_{10}\bar{K}_3\bar{K}_4$
$K_2K_5K_8K_{11}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_9\bar{K}_{10}\bar{K}_{102}\bar{K}_3\bar{K}_4$
$K_1K_5K_8K_{11}K_{13}K_{14}K_{101}\bar{K}_9\bar{K}_{10}\bar{K}_{102}\bar{K}_3\bar{K}_4$
$K_5K_8K_{12}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_7\bar{K}_3\bar{K}_4\bar{K}_{11}$
$K_5K_7K_8K_{12}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_9\bar{K}_{10}\bar{K}_3\bar{K}_4\bar{K}_{11}$
$K_2K_5K_8K_{12}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_9\bar{K}_{10}\bar{K}_{102}\bar{K}_3\bar{K}_4\bar{K}_{11}$
$K_1K_5K_8K_{12}K_{13}K_{14}K_{101}\bar{K}_9\bar{K}_{10}\bar{K}_{102}\bar{K}_3\bar{K}_4\bar{K}_{11}$
$K_6K_7K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_8\bar{K}_1\bar{K}_2\bar{K}_5$

$K_6K_7K_8K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_{11}\bar{K}_{12}\bar{K}_1\bar{K}_2$
$K_4K_6K_7K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_{11}\bar{K}_{12}\bar{K}_1\bar{K}_2\bar{K}_5$
$K_3K_6K_7K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_{11}\bar{K}_{12}\bar{K}_1\bar{K}_2\bar{K}_5$
$K_6K_7K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_8\bar{K}_1\bar{K}_2\bar{K}_5\bar{K}_9$
$K_6K_7K_8K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_{11}\bar{K}_{12}\bar{K}_1\bar{K}_2\bar{K}_9$
$K_4K_6K_7K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_{11}\bar{K}_{12}\bar{K}_1\bar{K}_2\bar{K}_5\bar{K}_9$
$K_3K_6K_7K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_{11}\bar{K}_{12}\bar{K}_1\bar{K}_2\bar{K}_5\bar{K}_9$
$K_6K_8K_{11}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_7\bar{K}_3\bar{K}_4\bar{K}_5$
$K_6K_7K_8K_{11}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_5\bar{K}_9\bar{K}_{10}\bar{K}_3\bar{K}_4$
$K_2K_6K_8K_{11}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_9\bar{K}_{10}\bar{K}_{102}\bar{K}_3\bar{K}_4\bar{K}_5$
$K_1K_6K_8K_{11}K_{13}K_{14}K_{101}\bar{K}_9\bar{K}_{10}\bar{K}_{102}\bar{K}_3\bar{K}_4\bar{K}_5$
$K_6K_8K_{12}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_7\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_{11}$
$K_6K_7K_8K_{12}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_2\bar{K}_{102}\bar{K}_5\bar{K}_9\bar{K}_{10}\bar{K}_3\bar{K}_4\bar{K}_{11}$
$K_2K_6K_8K_{12}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_9\bar{K}_{10}\bar{K}_{102}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_{11}$
$K_1K_6K_8K_{12}K_{13}K_{14}K_{101}\bar{K}_9\bar{K}_{10}\bar{K}_{102}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_{11}$
$K_1K_7K_8K_{11}K_{13}K_{14}K_{101}\bar{K}_9\bar{K}_{10}\bar{K}_{102}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_6$
$K_1K_7K_8K_{12}K_{13}K_{14}K_{101}\bar{K}_9\bar{K}_{10}\bar{K}_{102}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_6\bar{K}_{11}$
$K_2K_7K_8K_{11}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_9\bar{K}_{10}\bar{K}_{102}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_6$
$K_2K_7K_8K_{12}K_{13}K_{14}K_{101}\bar{K}_1\bar{K}_9\bar{K}_{10}\bar{K}_{102}\bar{K}_3\bar{K}_4\bar{K}_5\bar{K}_6\bar{K}_{11}$
$K_3K_7K_8K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_{11}\bar{K}_{12}\bar{K}_1\bar{K}_2\bar{K}_5\bar{K}_6$
$K_3K_7K_8K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_{11}\bar{K}_{12}\bar{K}_1\bar{K}_2\bar{K}_5\bar{K}_6\bar{K}_9$
$K_4K_7K_8K_9K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_{11}\bar{K}_{12}\bar{K}_1\bar{K}_2\bar{K}_5\bar{K}_6$
$K_4K_7K_8K_{10}K_{13}K_{14}K_{102}\bar{K}_{101}\bar{K}_3\bar{K}_{11}\bar{K}_{12}\bar{K}_1\bar{K}_2\bar{K}_5\bar{K}_6\bar{K}_9$

B Industrielles Beispiel

Nachfolgend sind weitere Informationen zu dem industriellen Beispiel aus Kapitel 7 enthalten. Dieses umfasst die variablen Zuverlässigkeitsblockdiagramme auf Subsystemebene und weitere Projektionen des Architekturraumes. Die vollständige Menge der variablen orthogonalisierten Minimalpfade ist aufgrund der resultierenden 28 000 Pfade nicht aufgeführt. Zudem würden die Minimalpfade einen Rückschluss auf alle Systemarchitekturen zulassen, was aufgrund der verwendeten vertraulichen Informationen nicht zulässig ist.

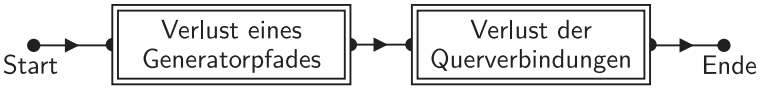


Abb. B.1: Zuverlässigkeitsblockdiagramm für die Fehlerbedingung *Verlust eines Systempfades*

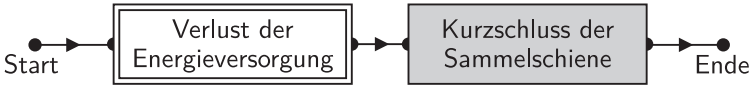


Abb. B.2: Zuverlässigkeitsblockdiagramm für die Fehlerbedingung *Verlust der Sammelschiene DC ESS*

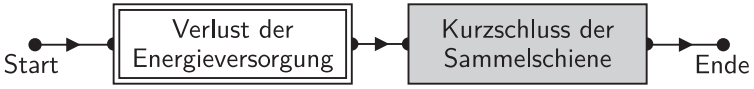


Abb. B.3: Zuverlässigkeitsblockdiagramm für die Fehlerbedingung *Verlust der Sammelschiene HVDC ESS*

Abb. B.4: Zuverlässigkeitsblockdiagramm für die Fehlerbedingung *Verlust einer normalen Sammelschiene*

Die folgenden Abbildungen verdeutlichen den vollständigen, zulässigen Architekturraum des industriellen Beispiels, wobei die nicht-dominierten Architekturen hervorgehoben sind. Aufgrund der lokalen Entscheidungspunkte der variablen Strukturmodelle, wirken sich Änderungen des Architekturvektors nicht auf alle berücksichtigten Zielgrößen aus, so dass in diesem Fall die Varianz des Zielwertraumes sinkt. Die resultierenden *Cluster* sind in den nachfolgenden Abbildungen zu erkennen.

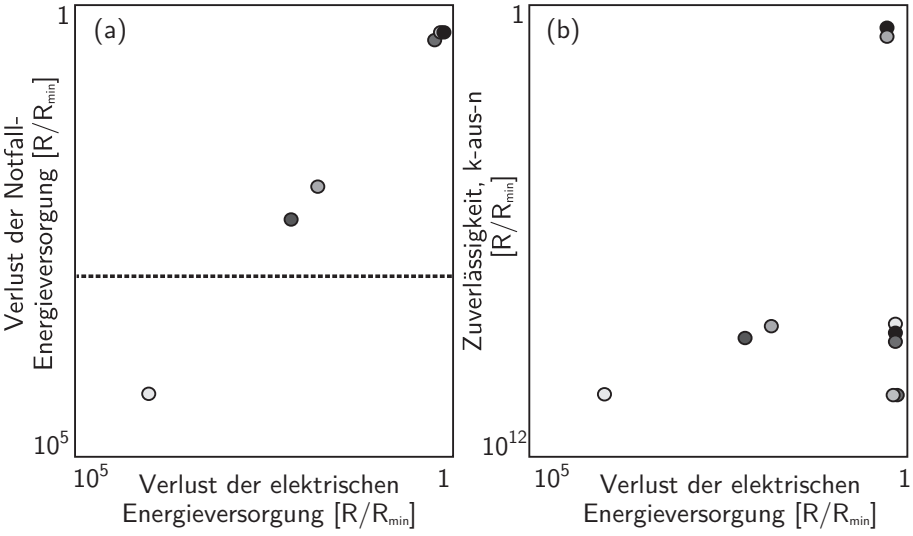


Abb. B.5: Projektionen des Zielwertraumes des Anwendungsbeispiels

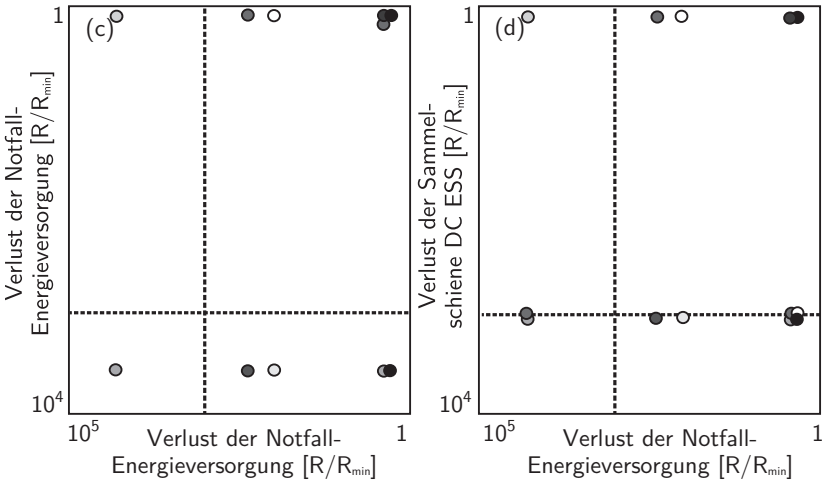


Abb. B.6: Projektionen des Zielwertraumes des Anwendungsbeispiels, Fortsetzung 1

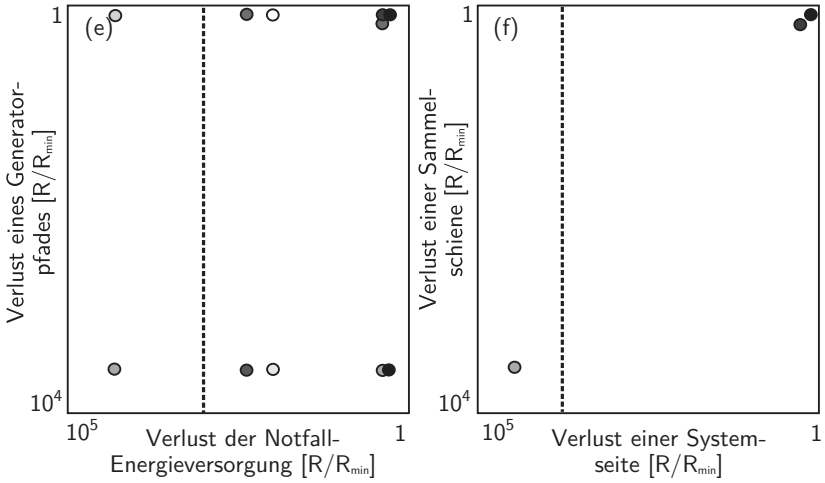


Abb. B.7: Projektionen des Zielwertraumes des Anwendungsbeispiels, Fortsetzung 2

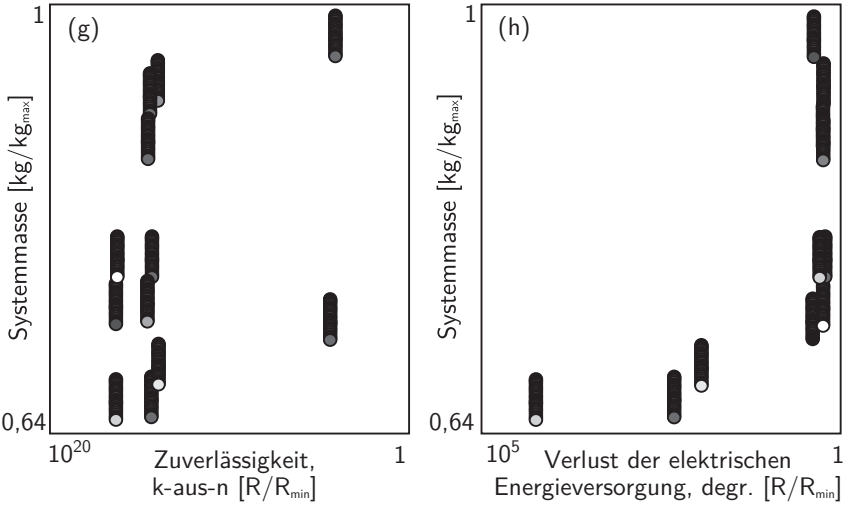


Abb. B.8: Projektionen des Zielwertraumes des Anwendungsbeispiels, Fortsetzung 3

Literaturverzeichnis

[1] Abele, Marcus: *Modellierung und Bewertung hochzuverlässiger Energiebordnetz-Architekturen für sicherheitskritische Verbraucher in Kraftfahrzeugen.* Dissertation, Universität Kassel, Kassel University Press, 2008

[2] ACARE - Advisory Council for Aerospace Research in Europe: *SRA2: Strategic Research Agenda - Edition 2.* 2004

[3] Aggarwal, K. K.: Redundany Optimization in General Systems. In: *IEEE Transaction on Reliability* 25 (1976), S. 330–332

[4] Airbus S.A.S.: *Getting to Grips with MMEL and MEL.* 2005

[5] Andrews, Joseph: System Reliability Modelling: The Current Capability and Potential Future Developments. In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 223 (2009), Nr. 12, S. 2881–2897

[6] Annighöfer, Björn ; Kleemann, Ernst ; Thielecke, Frank: Model-Based Development of Integrated Modular Avionics Architectures on Aircraft-Level. In: *3rd CEAS European Air and Space Conference, 24. - 28. Oktober 2011, Venedig, Italien*, 2011

[7] Axelsson, Jakob: Architecture Synthesis and Partitioning of Real-Time Systems: A Comparision of Three Heuristic Search Strategies. In: *5. International Workshop on Hardware/Software Codesign, 24. - 26. März 1997, Braunschweig*, 1997

[8] Axmann, Joachim K.: *Paralleles Optimieren technischer Systeme: adaptive evolutionäre Algorithmen auf Workstation-Clustern und Mehrprozessor-Systemen.* Habilitationsschrift, Technische Universität Braunschweig, Shaker, Aachen, 1999

[9] Bauer, Christophe ; Lagadec, Kristen ; Bes, Christian ; Monega, Marcel: Flight-Control System Architecture Optimization for Fly-by-Wire Airliners / Laboratoire d'Analyse et d'Architecture des Systemes,

N.06476 LAASCNRS. 2006. – Forschungsbericht

[10] Birolini, Alessandro: *Reliability Engineering: Theory and Practice*. 5. Auflage. Berlin, Heidelberg, Springer Verlag, 2007

[11] Burke, Michael: *Mehrkriterielle Redundanzoptimierung komplexer Flugzeugsystemarchitekturen unter Verwendung des Branch-and-Bound Verfahrens*. Studienarbeit, Institut für Flugzeug-Systemtechnik, Technische Universität Hamburg-Harburg, Hamburg, 2011

[12] Busacca, P. G. ; Marseguerra, M. ; Zio, Enrico: Multi-objective Optimization by Genetic Algorithms: Application to Safety Systems. In: *Reliability Engineering and System Safety* 72 (2001), S. 59–74

[13] Butz, Henning: Open Integrated Modular Avionic (IMA): State of the Art and Future Development Road Map at Airbus Deutschland. In: *Aircraft Systems Technologies Workshop, Hamburg*, 2007

[14] *Kapitel* Genetic Algorithms. In: Bäck, Thomas ; Fogel, David B. ; Michalewicz, Zbigniew: *Evolutionary Computation 1 - Basic Algorithms and Operators*. Institute of Physics Publishing, Bristol, Großbritannien, 2000, S. 64–80

[15] Cadini, F. ; Zio, E. ; Petrescu, C.A.: Multi-Objective Genetic Algorithm Optimization of Electrical Power Transmission Networks. In: *European Safety and Reliability Conference, 7. - 10. September 2009, Prag, Tschechische Republik*, 2009

[16] Carl, Udo B.: Moderne Flugzeugsysteme - Anforderungen an Technologien und Entwicklungsmethoden. In: *2. Symposium Flugzeug-Systemtechnik, DGLR-Fachausschuß S2.1 Starrflügelsysteme, Entwicklungstrends bei Basissystemen und ihre Wechselwirkungen zum Flugzeugentwurf, 15. - 16. September 1997, Hamburg*, 1997

[17] Coello, Carlos A. C.: A Short Tutorial on Evolutionary Multiobjective Optimization. In: *Evolutionary Multi-Criterion Optimization, 1. International Conference, 7.-9. März 2001, Zürich, Schweiz*, 2001

[18] Coello Coello, Carlos A. ; Van Veldhuizen, David A. ; Lamont, Gary B.: *Evolutionary Algorithms for Solving Multi-Objective Problems*. 2. New York : Kluwer Academic Publishers, New York, U.S.A., 2007

[19] Cohon, Jared L. ; Marks, David H.: A Review and Evaluation of Multiobjective Programming Techniques. In: *Water Resources Research* 11 (1975), April, Nr. 2, S. 208–220

[20] Cormen, Thomas H. ; Leiserson, Charles E. ; Rivest, Ronald L. ; Stein, Clifford: *Algorithmen - eine Einführung*. 2. Auflage. München, Oldenbourg Verlag, 2004

[21] Deb, Kalyanmoy: *Multi-Objective Optimization using Evolutionary Algorithms*. 2. New York : New York, U.S.A., Wiley-Interscience Series in Systems and Optimization, 2004

[22] Deb, Kalyanmoy D. ; Pratap, Amrit ; Agarwal, Sameer ; Meyarivan, T.: A fast and elitist multiobjective genetic algorithm : NSGA-II. In: *IEEE Transactions on Evolutionary Computation* 6 (2002), S. 182–197

[23] Domschke, Wolfgang ; Drexl, Andreas: *Einführung in Operations Research*. 7. Auflage. Berlin, Heidelberg, Springer Verlag, 2007

[24] Dueck, Gunter: New Optimization Heuristics The Great Deluge Algorithm and the Record-to-Record Travel. In: *Journal of Computational Physics* 104 (1993), S. 86–92

[25] Dussa-Zieger, Klaudia: *Model-Based Scheduling and Configuration of Heterogeneous Parallel Systems*. Dissertation, Universität Erlangen-Nürnberg, Arbeitsberichte des Instituts für Mathematische Maschinen und Datenverarbeitung, Erlangen, 1998

[26] Ebendt, Rüdiger ; Drechsler, Rolf ; Fey, Görschwin: *Advanced BDD Optimization*. 5. Auflage. Boston, USA, Springer Verlag, 2005

[27] Echtle, Klaus: *Fehlertoleranzverfahren*. 1. Auflage. Berlin, Heidelberg, Springer Verlag, 1990

[28] Ehrgott, Matthias: *Multicriteria Optimization*. 5. Auflage. Berlin, Heidelberg, Springer Verlag, 2005

[29] Emadi, A. ; Ehsani, M.: Aircraft Power Systems: Technology, State of the Art and Future Trends. In: *IEEE Aerospace and Electronic Systems Magazine* 15 (2000), Nr. 1, S. 28–32

[30] European Aviation Safety Agency: *Certification Specification for Large Airplanes, Acceptable Means of Compliance, AMC-25*. 2003

[31] European Aviation Safety Agency: *Certification Specification for Large Airplanes, Airworthiness Code, CS-25*. 2003

[32] Faleiro, Lester: Beyond the More Electric Aircraft. In: *Aerospace America* 43 (2005), Nr. 9, S. 35–40

[33] Faucher, Jerome: Simulation Study of New Aircraft Electrical Power Network Performances. In: *MOET Forum, 8.-11. September 2009, Barcelona, Spanien*, 2009

[34] Fayyad, Usama ; Grinstein, Georges ; Wierse, Andreas: *Information Visualization in Data Mining and Knowledge Discovery*. 1. Auflage. San Francisco, USA, Morgan Kaufmann Publishers, 2002

[35] Federal Aviation Administration: *Master Minimum Equipment List Airbus A318/A319/A320/A321*. 21. 2009

[36] Garey, Michael R. ; Johnson, David D.: *Computers and Intractability - A Guide to the Theory of NP-Completeness*. 1. Auflage. Englewood Cliffs, USA, Prentice-Hall, 1979

[37] Geilsdorf, Hendrik ; Raksch, Christian ; Carl, Udo B.: Analyse der Zustandserreichbarkeit zur Fehlergrenzwertbestimmung in Hochauftriebssystemen. In: *at - Automatisierungstechnik* 55 (2007), S. 561–567

[38] Giese, Tim ; Oehler, Bettina ; Sielemann, Michael: A Systematic Approach to Optimise Conventional Environmental Control Architectures. In: *Deutscher Luft- und Raumfahrtkongress, 31. August -2. September 2010, Hamburg*, 2010

[39] Glover, Fred ; Knox, John: Tabu Search: An Effective Heuristic for Combinatorial Optimization Problems. In: *3. Annual Rocky Mountain Conference on Artificial Intelligence, Denver, USA, 13.-15. June 1988*, 1988, S. 437–441

[40] Goupil, Philippe: Industrial Practices in Fault Tolerant Control. In: *Fault Tolerant Flight Control: A Benchmark Challenge, Lecture Notes in Control and Information Sciences* 399 (2010), S. 157–167

[41] Gutkowski, Witold: *Discrete Structural Optimization*. Berlin, Heidelberg, Springer Verlag, 1993

[42] Haitao, Q. I. ; Yongling, F. U. ; Xiaoye, Q. I. ; Lang, Y.: Architecture Optimization of More Electric Aircraft Actuation System. In: *Chinese*

Journal of Aeronautics 24 (2011), S. 506–513

[43] Heidtmann, Klaus: Smaller Sums of Disjoint Products by Subproduct Inversion. In: *IEEE Transaction on Reliability* 38 (1989), S. 305–311

[44] Heinze, Wolfgang ; Österheld, C.M. ; Horst, Peter: Multidisziplinäres Flugzeugentwurfsverfahren PrADO - Programmentwurf und Anwendung im Rahmen von Flugzeug-Konzeptstudien. In: *Deutscher Luft- und Raumfahrtkongress, 17. - 20. September 2001, Hamburg*, 2001. – Bonn

[45] Hoffmann, G.G. ; Grinstein, Georges ; Marx, K. ; Grosse, I. ; Stanley, E.: DNA Visual and Analytic Data Mining. In: *8. IEEE Visualization, Phoenix, USA, 19. -24. Oktober 1997*, 1997, S. 437–441

[46] Holert, Ben: *Eine Methode zum mehrkriteriellen Entwurf von Führungsmechanismen in Hochauftriebssystemen von Transportflugzeugen.* Dissertation, Technische Universität Hamburg-Harburg, Schriftenreihe Flugzeug-Systemtechnik, Nr. 03-2005, Shaker, Aachen, 2006

[47] Holland, John H.: Genetische Algorithmen. In: *Spektrum der Wissenschaft* 9 (1992), S. 44–51

[48] Howard, Ron W.: Planning for Super Safety: the Fail-Safe Dimension. In: *The Aeronautical Journal* (2000), S. 517–555

[49] Hörnle, Felix: *Optimization of Aircraft System Architectures considering System Reliability.* Bachelorarbeit, Institut für Flugzeug-Systemtechnik, Technische Universität Hamburg-Harburg, Hamburg, 2009

[50] Institute of Electrical and Electronics Engineers: *IEEE Standard Glossary of Software Engineering Terminology, IEEE Standard 610.12-1990.* 1990

[51] International Council on Systems Engineering: *Systems Engineering Handbook - A guide for system life cycle processes and activities.* Seattle, Washington, U.S.A, 2006

[52] Jomier, Thomas: MOET Public Technical Report / MOET Projekt Konsortium. 2009. – Forschungsbericht

[53] Joos, Hans-Dieter ; Bals, Johann ; Looye, Gertjan ; Schnepper, Klaus ; Varga, Andras: A Multi-Objective Optimisation-Based Software En-

vironment for Control Systems Design. In: *IEEE CADCS Symposium 2002, Glasgow, Großbritannien*, 2002

[54] Kazeminia, Amir ; Junglas, Marco ; Söffker, Dirk: Optimization of System Component Reliability Characteristics at Early Design Stage with Economically Reasonable Uncertainty Level. In: *European Safety and Reliability Conference, 7. - 10. September 2009, Prag, Tschechische Republik*, 2009

[55] Kirkpatrick, S. ; Gelatt, C. D. ; Vecchi, M. P.: Optimization by Simulated Annealing. In: *Science* 220 (1983), S. 671–680

[56] Knepper, Roger: Sicherheits- und Zuverlässigkeitsanalyse eines mechanisch-hydraulischen Standby-Systems, am Beispiel der A321 Stauluftturbine. In: *Erfolg durch zuverlässige Technik - Erfolgsfaktor Zuverlässigkeit für wettbewerbsfähige Produkte und Dienstleistungen, Fulda, 26. und 27. September 1995*, 1995

[57] Knepper, Roger: Der Sicherheits- und Zuverlässigkeitsprozess in der zivilen Luftfahrtindustrie. In: *Sicherheit komplexer Verkehrssysteme, Königswinter, 25. und 26. Mai 2000*, 2000

[58] Knowles, Joshua D. ; Corne, David W.: Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. In: *Evolutionary Computation* 8 (2000), S. 149–172

[59] Koeppen, Carsten: *Methodik zur modellbasierten Prognose von Flugzeugsystemparametern im Vorentwurf von Verkehrsflugzeugen*. Dissertation, Technische Universität Hamburg-Harburg, Schriftenreihe Flugzeug-Systemtechnik, Nr. 01-2006, Shaker, Aachen, 2006

[60] Koeppen, Carsten ; Carl, Udo B.: Erfassung und Bewertung von Systemen im Flugzeugentwurf. In: *Deutscher Luft- und Raumfahrtkongress, Stuttgart*, 2002

[61] Kritzinger, Duane: *Aircraft System Safety - Military and Civil Aeronautical Applications*. 1. Auflage. Woodhead Publishing, Cambridge, Großbritannien, 2006

[62] Kuo, Way ; Prasad, V. R. ; Tillman, Frank A. ; Hwang, Ching-Lai: *Optimal Reliability Design: Fundamentals and Applications*. 1. Auflage. Cambridge, UK, Cambridge University Press, 2001

[63] Land, A. H. ; Doig, A. G.: An Automatic Method of Solving Discrete Programming Problems. In: *Econometrica* 28 (1960), S. 497–520

[64] Langton, Roy ; Clark, Chuck ; Hewitt, Martin ; Richards, Lonnie: *Aircraft Fuel Systems*. 1. Auflage. London, UK, Wiley, 2009

[65] Levitin, Gregory ; Lisnianski, Anatoly: A New Approach to Solving Problems of Multi-State System Reliability Optimization. In: *Quality and Reliability Engineering International* 17 (2001), S. 93–104

[66] Liscouet-Hanke, Susan: *A model-based methodology for integrated preliminary sizing and analysis of aircraft power system architectures*. Dissertation, Institut National des Sciences Appliquées de Toulouse, Frankreich, 2007

[67] Lloyd, E. ; Tye, W.: *Systematic Safety - Safety Assessment of Aircraft Systems*. 2. New York : Civil Aviation Authority, Gatwick, Großbritannien, 1982

[68] Lohn, Jason D. ; Kraus, William F. ; Haith, Gary L.: Comparing a Coevolutionary Genetic Algorithm for Multiobjective Optimization. In: *IEEE Congress on Evolutionary Computation, Hawaii, USA, May 2002*, 2002

[69] Lubis, Aldamanda A.: *Zur Optimierung des multidisziplinären Entwurfs von Verkehrsflugzeugen in der parallelen Rechenumgebung*. Dissertation, Technische Universität Braunschweig, Forschungsbericht des Zentrums für Luft- und Raumfahrtechnik, 94-02, Braunschweig, 1994

[70] Lübke, Andreas: *Systematischer Bordnetzentwurf - Optimierung der Bordnetzarchitektur mit Hilfe von genetischen Algorithmen*. Dissertation, Technische Universität Dresden, Wissenschaftsverlag Mainz, Aachen, 1999

[71] Lüdders, Hauke ; Claussen, Stefan ; Grymlas, Jan ; Vredenborg, Enno ; Thielecke, Frank: Challenges Faced by the Integration of a Fuel Cell System in Aircraft System Architectures. In: *H2-Expo 2011, Hamburg, 8.-9. Juni 2011*, 2011

[72] Lüdders, Hauke ; Grymlas, Jan ; Vredenborg, Enno ; Thielecke, Frank: A Methodology for Rapid Evaluation and Sizing of Fuel Cell System Architectures for Commercial Aircraft. In: *SAE 2011 AeroTech Congress & Exhibition, 18.-21. Oktober 2011, Toulouse, Frankreich*, 2011

[73] Marseguerra, Marzio ; Zio, Enrico ; Podofillini, Luca ; Coit, David W.: Optimal Design of Reliable Network Systems in Presence of Uncertainty. In: *IEEE Transactions on Reliability* 54 (2005), S. 243–253

[74] McLeavy, D.W. ; J.A., McLeavy: Optimization of System Reliability by Branch-and-Bound Technique. In: *IEEE Transaction on Reliability* 25 (1976), S. 327–329

[75] Merkel, Maximillian: *Konzeption eines interaktiven Graphical User Interface Systems zur Optimierung der Anwendung zuverlässigkeitstechnischer Algorithmen.* Studienarbeit, Institut für Flugzeug-Systemtechnik, Technische Universität Hamburg-Harburg, Hamburg, 2001

[76] Meyna, Arno: *Zuverlässigkeitsbewertung zukunftsorientierter Technologien.* 1. Auflage. Braunschweig, Vieweg Verlag, 1994

[77] Meyna, Arno ; Pauli, Bernhard: *Taschenbuch der Zuverlässigkeits- und Sicherheitstechnik - Quantitative Bewertungsverfahren.* 1. Auflage. München, Hanser Verlag, 2003

[78] Moir, Ian: More-Electric Aircraft - System Considerations. In: *IEE Colloquium on Electrical Machines and Systems for the More Electric Aircraft, London, Großbritannien, 9. November 1999*, 1999

[79] Moir, Ian ; Seabridge, Allan: *Aircraft Systems: Mechanical, Electrical and Avionics Subsystems Integration.* 1. Auflage. Reston, USA, American Institute of Aeronautics and Astronautics, 2001

[80] Moir, Ian ; Seabridge, Allan: *Design and Development of Aircraft Systems.* 1. Auflage. London, UK, Professional Engineering Publishing, 2004

[81] Nakagawa, Yuji. ; Nakashima, Kyoichi ; Hattori, Yoshio: Optimal Reliability Allocation by Branch-and-Bound Technique. In: *IEEE Transaction on Reliability* 27 (1978), S. 31–38

[82] National Aeronautics and Space Administration: *NASA Systems Engineering Handbook.* Washington, D.C., U.S.A, 2007

[83] Nemhauser, George L. ; Wolsey, Laurence A.: *Integer and Combinatorial Optimization.* New York, U.S.A., Wiley, 1988

[84] Novakova, Lenka ; Stepankova, Olga: Multidimensional Clusters in RadViz. In: *6. WSEAS International Conference on Simulation, Control,*

Modelling and Optimization, Lissabon, Portugal, 22.-24. September 2006, 2006, S. 470 – 475

[85] Owsnicki-Klewe, Bernd: *Algorithmen und Datenstrukturen.* 3. Auflage. Augsburg, Wißner Verlag, 1999

[86] Padula, Sharon L. ; Kincaid, Rex K.: Optimization Strategies for Sensor and Actuator Placement / National Aeronautics and Space Administration (NASA), TM-1999-209126. 1999. – Forschungsbericht

[87] Pfennig, Malte: *Methodik zum wissensbasierten Entwurf der Antriebssysteme von Hochauftriebssystemen.* Dissertation, Technische Universität Hamburg-Harburg, Schriftenreihe Flugzeug-Systemtechnik, Nr. 01-2012, Shaker, Aachen, 2012

[88] Pfennig, Malte ; Carl, Udo B. ; Thielecke, Frank: Recent Advances Towards an Integrated and Optimized Design of High Lift Actuation Systems. In: *SAE International Journal of Aerospace* 3 (2010), Nr. 1, S. 55 – 64

[89] Pham, Hoang ; Pham, Hoang (Hrsg.): *Reliability Engineering.* Berlin, Heidelberg, Springer Verlag, 2003

[90] Pohlheim, Hartmut: *Evolutionäre Algorithmen - Verfahren, Operatoren und Hinweise für die Praxis.* 5. Auflage. Berlin, Heidelberg, Springer Verlag, 1999

[91] Radio Technical Commission for Aeronautics: *Software Considerations in Airborne Systems and Equipment Certifications, RTCA/DO-178B.* 1992

[92] Radio Technical Commission for Aeronautics: *Design Assurance Guidance for Airborne Electronic Hardware, RTCA/DO-254.* 2000

[93] Rakowsky, Uwe K.: *System-Zuverlässigkeit - Terminologie, Methoden, Konzepte.* 1. Auflage. Hagen, LiLoLe Verlag, 2002

[94] Raksch, Christian ; Maanen, Rachel van ; Rehage, Dominick ; Thielecke, Frank ; Carl, U. B.: Performance Degradation Analysis of Fault-Tolerant Aircraft Systems. In: *1st CEAS European Air and Space Conference, 10.-13. September 2007, Berlin*, 2007

[95] Raksch, Christian ; Rehage, Dominick ; Thielecke, Frank: Reliability Analysis of Aircraft Power Supply Systems. In: *Aerospace System*

Technologies Workshop, 26.- 27. März 2009, Hamburg, 2009

[96] Raksch, Christian ; Thielecke, Frank: Multiobjective optimization of fault-tolerant aircraft systems considering system degradation. In: *European Safety and Reliability Conference, 7. - 10. September 2009, Prag, Tschechische Republik*, 2009

[97] Raksch, Christian ; Thielecke, Frank: Optimierung fehlertoleranter Flugzeugsysteme mit mehrfachen Sicherheits- und Zuverlässigkeitsanforderungen. In: *Deutscher Luft- und Raumfahrtkongress, 31. August -2. September 2010, Hamburg*, 2010

[98] Ramirez-Marquez, Jose ; Coit, David: A Heuristic for Solving the Redundancy Allocation Problem for Multistate Series-Parallel Systems. In: *Reliability Engineering & System Safety* 83 (2004), Nr. 3, S. 341–349

[99] Ravel, Pierre: Electrical Distribution of High Power: Impacts, Technologies. In: *MOET Forum, 8.-11. September 2009, Barcelona, Spanien*, 2009

[100] Ravi, V. ; Murty, B. S. N. ; Reddy, P. J.: Nonequilibrium Simulated Annealing-Optimization in Complex Systems. In: *IEEE Transaction on Reliability* 46 (1997), S. 233–239

[101] Rechtin, Eberhardt: *Systems Architecting - Creating and Building Complex Systems*. 5. Auflage. Englewood Cliffs, USA, Prentice-Hall, 1991

[102] Rehage, Dominick: *Zustandsmodellierung und Zuverlässigkeitsanalyse fehlertoleranter Systemarchitekturen auf Basis Integrierter Modularer Avionik*. Dissertation, Technische Universität Hamburg-Harburg, Schriftenreihe Flugzeug-Systemtechnik, Nr. 01-2009, Shaker, Aachen, 2009

[103] Ross, Richard: Report on the Serious Incident to Airbus A319-111, Registration G-EZAC near Nantes, France on 15 September 2009 / Department for Transportation, Air Accidents Investigation Branch. 2009. – Forschungsbericht

[104] Salomon, Uwe ; Reichel, Reinhard: Automatic Safety Computation for IMA Systems. In: *30th Digital Avionics Systems Conference, 16. - 20. Oktober 2011 , Seattle, USA*, 2011

[105] Schallert, Christian: An Integrated Tool for Aircraft Electric Power Systems Pre-Design. In: *Aerospace System Technologies Workshop, 26.-*

27. März 2009, Hamburg, 2009

[106] Scholz, Dieter: *Entwicklung eines CAE-Werkzeuges zum Entwurf von Flugsteuerungs- und Hydrauliksystemen.* Dissertation, Technische Universität Hamburg-Harburg, Fortschritt-Berichte VDI Reihe 20, Nr. 262, Düsseldorf, 1996

[107] Schulz, Mario ; Zerbe, Volker ; Yang, Kai ; Zimmermann, Armin: Optimization of Avionic System Architectures. In: *CEAS Aeronautical Journal* 1 (2011)

[108] Schwehm, Markus: *Globable Optimierung mit massiv parallelen Genetischen Algorithmen.* Dissertation, Universität Erlangen-Nürnberg, Arbeitsberichte des Instituts für Mathematische Maschinen und Datenverarbeitung, Erlangen, 1997

[109] Seeckt, Kolja ; Scholz, Dieter: Application of the Aircraft Preliminary Sizing Tool PreSTo to Kerosene and Liqiud Hydrogen Fueled Regional Freighter Aircraft. In: *Deutscher Luft- und Raumfahrtkongress, 31. August -2. September 2010, Hamburg*, 2010

[110] Sharma, J. ; Venkateswaran, K. V.: A Direct Method for Maximizing the System Reliability. In: *IEEE Transaction on Reliability* 20 (1971), S. 256–259

[111] Shi, Dinghua H.: A New Heuristic Algorithm for Constrained Redundancy-Optimization in Complex Systems. In: *IEEE Transaction on Reliability* 36 (1987), S. 621–623

[112] Sinnet, Mike: 787 No-Bleed Systems: Saving Fuel and Enhancing, Operational Efficiencies. In: *Boeing Aero Magazine* 4 (2007), S. 6–11

[113] Slomka, Frank: *Mehrkriterienoptimierung verteilter Echtzeitsysteme mit Tabu-Search.* Dissertation, Universität Erlangen-Nürnberg, Fortschritt-Berichte VDI Reihe 20, Nr. 353, Düsseldorf, 2002

[114] Smyth, Richard: Entwicklung von hochintegrierten Flugzeugsystemen - Anforderungen an die nächste Generation von Zivilflugzeugen. In: *2. Symposium Flugzeug-Systemtechnik, DGLR-Fachausschuß S2.1 Starrflügelsysteme, Entwicklungstrends bei Basissystemen und ihre Wechselwirkungen zum Flugzeugentwurf, 15. - 16. September 1997, Hamburg*, 1997

[115] Smyth, Richard ; Roloff, Gerd: Best Practices for Systems Development and Integration. In: *FINSE Seminar, Systems Engineering for Enabling Life-Cycle Management, Helsinki, Finnland, 26. October 2006*, 2006

[116] Society of Automotive Engineers Inc.: *Guidelines and Methods for Conducting the Safety Assessement Process on Civil Airborne Systems and Equipment, SAE ARP 4761.* 1995

[117] Society of Automotive Engineers Inc.: *Certification Considerations for Highly-Integrated or Complex Aircraft Systems, SAE ARP 4754.* 1996

[118] Sourd, Francis ; Spanjaard, Olivier: A Multiobjective Branch-and-Bound Framework: Application to the Biobjective Spanning Tree Problem. In: *INFORMS Journal on Computing* 20 (2008), Nr. 3, S. 472–484

[119] Strunk, Elizabeth A. ; Knight, J. C. ; Aiello, M. A.: Distributed reconfigurable avionics architectures. In: *23. Digital Avionics Systems Conference, 24.28. Oktober 2004, Salt Lake City, U.S.A.*, 2004

[120] Taboada, Heidi A. ; Espiritu, Jose F. ; Coit, David W.: MOMS-GA: A Multiobjective Multi-State Genetic Algorithm for System Reliability Optimization Design Problems. In: *IEEE Transactions on Reliability* 57 (2007), S. 182–191

[121] Thielecke, Frank: *Parameteridentifizierung von Simulationsmodellen für das viskoplastische Verhalten von Metallen - Theorie, Numerik und Anwendung.* Dissertation, Technische Universität Braunschweig, Braunschweiger Schriften zur Mechanik 34, Braunschweig, 1998

[122] Tillman, Frank A. ; Hwang, Ching-Lai ; Kuo, Way: *Optimization of Systems Reliability.* New York, U.S.A., Marcel Dekker Inc., 1980

[123] Tillman, Frank A. ; Liittschwager, J. M.: Integer Programming Formulation of Constrained Reliability Problems. In: *Management Science* 13 (1967), S. 887–899

[124] Tooley, Michael H. ; Wyatt, David: *Aircraft electrical and electronic systems: principles, operation and maintenance.* Amsterdam, Elsevier/Butterworth-Heinemann, 2009

[125] Törn, Aimo ; Zilinskas, Antanas: *Global Optimization.* 5. Auflage. New York, U.S.A., Springer Verlag, 1989

[126] Vahl, Andreas: *Interaktive Zuverlässigkeitsanalyse von Flugzeug-Systemarchitekturen.* Dissertation, Technische Universität Hamburg-Harburg, Fortschritt-Berichte VDI Reihe 12, Nr. 565, Düsseldorf, 1998

[127] Valiant, Leslie G.: The Complexity of Enumeration and Reliability Problems. In: *SIAM Journal of Computing* 8 (1979), S. 410–421

[128] Weener, Earl F.: Integrated Safety System Design and Future Developments of Large Commercial Airplanes. In: *Aviation Safety, Human Factors, System Engineering, Flight Operations, Economics, Strategies, Management* (1997), S. 53–71

[129] Wolpert, David H. ; Macready, William G.: No free lunch theorems for optimization. In: *IEEE Transactions on Evolutionary Computation* 1 (1997), Nr. 1, S. 67–82

[130] Zitzler, Eckart ; Thiele, Lothar: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. In: *IEEE Transactions on Evolutionary Computation* 3 (1999), S. 257–271