

# **Optimierung und vage Daten im Entwicklungsprozess**

**Optimierungsmethode zur Auswahl  
wirtschaftlichkeitsoptimaler Standard-  
Kraftfahrzeugkomponenten und Berücksichtigung von  
vagen Daten während des Entwicklungsprozesses**



# Optimierung und vage Daten im Entwicklungsprozess

Optimierungsmethode zur Auswahl wirtschaftlichkeitsoptimaler  
Standard-Kraftfahrzeugkomponenten und Berücksichtigung von  
vagen Daten während des Entwicklungsprozesses

---

Vom Fachbereich Sicherheitstechnik der  
**Bergischen Universität Wuppertal**  
zur Erlangung des akademischen Grades  
Doktor-Ingenieur  
genehmigte Dissertation  
von  
Diplom-Ingenieur Martin Döring  
aus Braunschweig

Gutachter:

Univ.-Prof. Dr.-Ing. A. Meyna  
Univ.-Prof. Dr. rer. nat. P. C. Müller

Tag der mündlichen Prüfung:

17. Juni 2003

## Impressum

Bibliografische Information der Deutschen Bibliothek:

*Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <<http://dnb.ddb.de>> abrufbar.*

**Copyright** © 2003 by Martin Döring

Alle Rechte liegen beim Autor.

Das Werk einschließlich aller Abbildungen ist urheberrechtlich geschützt. Jede Verwertung außerhalb der Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Autors unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Bearbeitung in elektronischen Systemen.

**Herstellung:** Books on Demand GmbH, Norderstedt

**Verlag:** Eigenverlag, Braunschweig 2003

ISBN 3-8311-4915-1

Für meine Frau Yvonne und meinen Sohn Aaron Thore,  
die mein Leben so richtig wertvoll machen.

## **Danksagung**

Die vorliegende Arbeit wurde im Rahmen eines mir durch die Volkswagen AG gewährten Promotionsstipendiums in der Elektronik-Forschung in Wolfsburg in Zusammenarbeit mit dem Fachbereich für Sicherheitstheorie und Verkehrstechnik der Bergischen Universität Wuppertal erstellt.

Ich danke meinem Doktorvater Univ.-Prof. Dr.-Ing. Arno Meyna von der BU Wuppertal. Er hat während meiner Promotionszeit zielgerichtet, jedoch meine Kreativität nicht einschränkend, meine Arbeit mit gelenkt. Seine Vorgaben und die stets sachliche Kritik gepaart mit seinen inhaltlichen Anregungen waren stets angemessen und kamen immer zur richtigen Zeit.

Ein besonderes Dankschön möchte ich an meinen Betreuer Dr.-Ing. Bernd Rech von der Volkswagen AG richten. Durch seine fachliche und menschliche Kompetenz und seine Hilfe bei der Anfertigung dieser Arbeit habe ich weitaus mehr, als nur das wissenschaftliche Arbeiten erlernt, nämlich fürs Leben.

Weiterhin habe ich durch diverse Diskussionen mit meinen Kollegen und Freunden wesentliche Anregungen hinsichtlich der praxisrelevanten Inhalte dieser Arbeit bekommen. Stellvertretend für alle möchte ich hier Martin Gestwa nennen und an ihn und alle die gemeint sind ein Dankeschön aussprechen.

Letztlich möchte ich mich auch ganz besonders bei meiner Frau Yvonne bedanken. Sie gibt mir den Rückhalt, den man für eine solche Arbeit benötigt. Motivierend, aufmunternd und mich vorbehaltlos unterstützend, stellt Sie Ihre eigenen Interessen immer wieder in den Hintergrund.

Martin Döring

# Inhaltsverzeichnis

<b>Abkürzungen und Begriffe</b> .....	<b>IV</b>
<b>Kurzfassung</b> .....	<b>VI</b>
<b>1 Einleitung</b> .....	<b>1</b>
<b>2 Die Optimierungsaufgabe</b> .....	<b>4</b>
2.1 Wirtschaftlichkeitsmaßzahl .....	6
2.2 Ausfallrate .....	7
<b>3 Lösungsraum und dessen Einschränkung</b> .....	<b>9</b>
3.1 Einbauorte .....	11
3.2 Kombinationsverbot und -zuweisung .....	11
3.3 Ausfallrate des Gesamtsystems.....	11
3.4 Offensichtlich schlechtere Typen .....	12
<b>4 Methoden zum Finden der optimalen Variante</b> .....	<b>13</b>
4.1 Methode 1: Graphenalgorithmus.....	13
4.1.1 Phase 1: Erstellung des Graphen und Definition der Lösungsräume mit Auswertung des Kostenvektors.....	17
4.1.2 Phase 2: Einschränkung des Lösungsraumes.....	20
4.1.3 Phase 3: Berechnung aller verbleibenden Wege.....	24
4.1.4 Phase 4: Ermittlung der optimalen Lösung durch Vergleich .....	24
4.1.5 Bewertung der Methode der Graphenalgorithmen.....	25
4.2 Methode 2: Neuro-Fuzzy-Methode .....	25
4.2.1 Neuronales Lernen .....	26
4.2.2 Fuzzy-Logik .....	28
4.2.3 Fuzzy-Regeln .....	30
4.2.4 Neuro-Fuzzy-Optimierungsprozess .....	31
4.2.5 Bewertung der Neuro-Fuzzy-Methode.....	34
4.3 Methode 3: Genetische Algorithmen.....	35
4.3.1 Prinzip 1: Selektion .....	36
4.3.2 Prinzip 2: Mutation .....	39
4.3.3 Prinzip 3: Kreuzung .....	40
4.3.4 Prinzip 4: Geburt.....	41
4.3.5 Abbruchkriterien.....	41
4.3.6 Gezielte Genmanipulation .....	42
4.3.7 Bewertung der Methode des Genetischen Algorithmus.....	42

4.4	Vergleich der Methoden.....	43
<b>5</b>	<b>Bewertung der Unsicherheit von Lösungen (vage Daten).....</b>	<b>45</b>
5.1	Anwendbare Fuzzy-Sets.....	46
5.2	Fuzzy-Vereinigung.....	48
5.3	Zusammenfassung der Methode zur Bewertung der Entwicklungsunsicherheit.....	49
<b>6</b>	<b>Das realisierte Optimierungstool.....</b>	<b>50</b>
6.1	Anbindung externer Datenbasen und Weiterverarbeitung der Ergebnisse.....	52
6.2	Definition der Parameter.....	52
6.3	Definition des Systems.....	52
6.4	Ausführen der Algorithmen.....	53
6.5	Darstellung der Ergebnisse.....	53
<b>7</b>	<b>Beispielapplikation.....</b>	<b>54</b>
7.1	Die elektromechanische Bremse (EMB).....	54
7.2	Einbauorte im Fahrzeug.....	58
7.3	Definition der maximal zulässigen Ausfallrate.....	61
7.4	Generierung der Referenzlösung(en).....	62
<b>8</b>	<b>Umsetzung und Erweiterung des Genetischen Algorithmus.....</b>	<b>64</b>
8.1	Herleitung des Adaptiven Abbruchkriteriums.....	64
8.1.1	Zusammenhang der optimalen Lösung mit der Anzahl der Evolutionen.....	66
8.1.2	Ergebnisstruktur der Evolutionen.....	66
8.1.3	Adaptives Abbruchkriterium.....	71
8.2	Anwendung der Fitness im EMB-System.....	75
8.2.1	Werte der normierten Ausfallrate.....	75
8.2.2	Werte der normierten Wirtschaftlichkeitsmaßzahl.....	77
8.2.3	Fitness.....	79
8.2.4	Praxisrelevanz der Fitness.....	82
8.3	Darstellung der Effektivität der gezielten Genmanipulation.....	83
8.4	Grenzen bei der Beherrschung von Lösungsräumen beliebiger Komplexität.....	84
8.5	Vergleichsmöglichkeit mit ähnlichen Varianten.....	86
8.6	Voraussetzungen der Übertragbarkeit der Methode auf andere Fragestellungen.....	86



---

<b>9</b>	<b>Methode zur Beurteilung der vagen Daten .....</b>	<b>88</b>
9.1	Beeinflussungsmöglichkeit durch die Methode der vagen Daten.....	88
9.2	Einflussfaktoren auf die vagen Daten im Entwicklungsprozess .....	89
9.3	Quantifizierung der vagen Daten .....	93
9.4	Anwendung der Unsicherheitsabschätzung auf das Beispiel der EMB.....	96
9.5	Darstellung der Ergebnisse in den vier Phasen .....	97
9.5.1	Konzeptphase.....	97
9.5.2	Prototypenphase.....	100
9.5.3	Vorserienphase .....	102
9.5.4	Serieneinsatz.....	104
<b>10</b>	<b>Anwendbarkeit und Erweiterbarkeit der dargestellten Optimierungsmethoden.....</b>	<b>106</b>
<b>11</b>	<b>Zusammenfassung .....</b>	<b>107</b>
11.1	Optimierungsmethoden.....	107
11.2	Bewertung der Entwicklungsunsicherheit .....	107
11.3	Weiterführende Aufgaben .....	108
<b>12</b>	<b>Literaturverzeichnis .....</b>	<b>109</b>
<b>13</b>	<b>Anhang.....</b>	<b>112</b>
13.1	Datenbasis .....	112
13.2	Komponentenliste der EMB .....	113
13.3	Werte des optimalen Systems (Gleichgewichtung aller Parameter) bei der Optimierung mit exakten Werten .....	114
13.4	Verwendete Typen des optimalen Systems (Gleichgewichtung aller Parameter) bei der Optimierung mit exakten Werten.....	114
13.5	Lösungen einer Optimierung mit adaptivem Abbruchkriterium .....	116
13.6	Vage Daten der verwendeten Typen in der Konzeptphase.....	117
13.7	Vage Daten der verwendeten Typen in der Prototypenphase .....	118
13.8	Vage Daten der verwendeten Typen in der Vorserienphase .....	119
13.9	Vage Daten der verwendeten Typen im Serieneinsatz.....	120
13.10	Screenshots .....	121

## Abkürzungen und Begriffe

$\alpha$ .....	Startpunkt in einem Graphen
$\lambda, \lambda(t)$ .....	Ausfallrate [1/Zeit]
$\lambda_{\text{soll}}$ .....	Vorgabe für die zulässige Ausfallrate eines Systems
$\lambda$ -Sets .....	Fuzzy-Sets der Ausfallrate zur Bewertung verschiedener Varianten mit den Fuzzy-Mengen
$\mu$ .....	Zugehörigkeit zu einer Fuzzy-Menge
Abbruchvorgabe ....	Anzahl der Generationen, die während der Evolution ohne eine Änderung der besten Individuen durchlaufen werden müssen, um die Optimierung abubrechen
Abbruchzahl.....	Anzahl der Generationen, die tatsächlich während der Evolution durchlaufen wurden, ohne dass sich die jeweils besten Individuen verbessert haben
ABS .....	Anti-Blockier-System
Axon .....	Abgehender (sendender) Reizleiter eines (biologischen) Neuron
Bussystem.....	System zur digitalen (elektrischen oder optischen) Übermittlung von Daten
c.....	Kostenfunktion
Dendrite .....	Empfangender Reizleiter eines (biologischen) Neuron
E .....	Kante (edge) zwischen zwei Knoten eines Graphen
EHB .....	Elektrohydraulische Bremse
EMB.....	Elektromechanische Bremse
EML .....	Elektromechanische Lenkung
ESP .....	Elektronische-Stabilitäts-Programm
EPB .....	Elektrisches Park-Bremsmodul
F(t).....	Ausfallwahrscheinlichkeit
FBA .....	Fehlerbaum-Analyse
FMEA .....	Fehler-Möglichkeiten- und Einfluss-Analyse
FRA .....	Funktionsrisikoanalyse
Fuzzy-Menge.....	Ermöglicht die Überführung eines scharfen Wertes in einen linguistischen Term und die Zugehörigkeit zur der Fuzzy-Menge
Fuzzy-Set .....	Gesamtheit aller Fuzzy-Mengen eines Parameters
G.....	Graph mit Kanten und Knoten
$g(\lambda)$ .....	Qualitätsfunktion (grade-function)
GUI .....	Grafische Benutzeroberfläche (graphical user interface)
$g_G$ .....	Gewichtung des Gewichts nach Gleichung (1)
$g_K$ .....	Gewichtung der Kosten nach Gleichung (1)
$g_V$ .....	Gewichtung des Volumens nach Gleichung (1)

---

H.....	Gesamtheit aller Ebenen eines Graphen
K.....	Anzahl der Komponenten
l.....	Index einer möglichen Lösung
NP-Probleme .....	nicht-deterministisch lösbare Probleme, wobei die Überprüfung, ob eine Lösung gefunden wurde, mit polynomialem Aufwand möglich ist
O.....	Größe des Lösungsraums
P .....	Menge aller gültigen Permutationen
p .....	Anzahl aller gültigen Permutationen aus P
Package .....	Kompakte Bauweise
Q.....	Untermenge von P, für die gilt $\lambda(l) \leq \lambda_{\text{soll}}$
q .....	Anzahl aller gültigen Permutationen aus Q
r .....	gültiger Weg im Graphenalgorithmus
s.....	Fitnesswert, ergibt sich aus der Fitnessfunktion $s(W, \lambda)$
$s(W, \lambda)$ .....	Fitnessfunktion, Gleichung (32)
SNNS .....	Stuttgarter Neuronale Netze Simulator, Tool zur Regelung mit Neuronalen Netzen
Synapse .....	(Elektrochemischer) Übergang zwischen Axon und Dendrite, der eine Gewichtung der zu übertragenden Reize bewirkt
T .....	Anzahl der Typen
$V_{\text{ab}}$ .....	Abbruchvorgabe
V .....	Knoten (vertice) eines Graphen
W.....	Wirtschaftlichkeitsmaßzahl, berechnet nach Gleichung (1)
W-Sets.....	Fuzzy-Sets der Wirtschaftlichkeitsmaßzahl zur Bewertung verschiedener Varianten mit den Fuzzy-Mengen
x.....	Index für einen Knoten in der i-ten Ebene $H_i$
X-by-Wire.....	Oberbegriff für Fahrzeugsysteme, die über ein Bussystem gesteuert werden
$Z_{\text{ab}}$ .....	Abbruchzahl
Zyklusfestigkeit.....	Fahrzeug-Batterien werden oft geladen und wieder entladen. Die Zyklusfestigkeit gibt an, wie viele solcher Zyklen ein Batterietyp sicher übersteht.

## **Kurzfassung**

Viele der zukünftigen Funktionen im Kraftfahrzeug haben Sicherheitsrelevanz und müssen sorgfältig analysiert werden. Die Systeme sollen nicht nur sicher, sondern auch zuverlässig, kostengünstig und leicht sein. Eine Aufgabe der Entwickler ist es daher, die zu entwickelnden Systeme mit größtmöglicher Wirtschaftlichkeit zu dimensionieren.

Neben den für die entsprechende Anwendung neu entwickelten Komponenten werden auch eine Vielzahl von Komponenten eingesetzt, die standardmäßig bereits in anderen Applikationen verwendet werden. Es stehen für diese Standardkomponenten, wie beispielsweise Batterien, Kabel, Steckverbindungen und verschiedene mechanische Komponenten, häufig mehrere unterschiedliche Typen zur Verfügung. Diese erfüllen dieselbe Funktion bei unterschiedlicher Zuverlässigkeit und/oder Wirtschaftlichkeit. Es können unter Umständen für die zu entwickelnde Applikation unüberschaubar viele Varianten entstehen.

Die Anzahl aller möglichen Lösungen steigt exponentiell mit jeder weiteren Komponente. Um unter Berücksichtigung aller Randbedingungen die wirtschaftlichste Lösung zu finden, können verschiedene Optimierungsmethoden rechnergestützt angewendet werden. Diese unterscheiden sich in der Geschwindigkeit der Lösungsberechnung, aber auch hinsichtlich der Qualität ihrer Ergebnisse.

Die anzuwendende Methode soll die Typen finden, mit denen sich das wirtschaftlichste Gesamtsystem ergibt, wobei die Topologie, die Einbauorte und die Soll-Systemzuverlässigkeit vorgegeben sind.

In dieser Arbeit werden drei Optimierungsmethoden einem kritischen Vergleich hinsichtlich ihren Vor- und Nachteile sowie ihrer Anwendungsmöglichkeiten unterzogen.

Weiterhin werden die vagen Daten der Komponenten der gefundenen Lösung ausgewertet. Man hat so eine Möglichkeit, Systeme unterschiedlicher Entwicklungsstufen miteinander zu vergleichen und die Entwicklungsrisiken abzuschätzen.

## ***Abstract***

Many of the future motor vehicle functions are relevant to safety and have to be carefully analysed. The systems should not only be safe, but also reliable, reasonably priced and light. Therefore, one of the developer's tasks is to design the new systems under development for the optimum economy.

Along with the new components developed for a certain application, a range of components will be involved which are already used as standard in other applications. In the case of these standard components, such as batteries, cables, connectors, various mechanical components etc., there are often a number of different types of component available. These all have the same function while varying in reliability and/or economic efficiency. It is possible for an unmanageable number of variations to be created for the application being developed.

The number of possible solutions increases exponentially with each new component. Different computer-aided optimisation methods can be used in order to find the most economical solution, taking all of the constraints into consideration. These differ in solution-finding speed, but also in the quality of their results.

The method used should find the component types, which produce the most economical complete system for a specified topology, installation location and system reliability.

Three optimisation methods undergo a critical comparison in this study, with regards to their advantages and disadvantages, as well as their possible uses.

The vague component data of the solution found are also analysed and evaluated. It is then possible to compare systems at various stages of development and to estimate the development risks.

## ***Résumé***

Un grand nombre des systèmes qui seront montés dans un véhicule à l'avenir sont liés à la sécurité et doivent être analysés avec grand soin. Ces systèmes doivent non seulement être sûrs, mais également fiables, économiques et peu complexes. Une tâche des développeurs consiste de ce fait à mettre au point des systèmes ayant la plus grande marge de rentabilité possible.

En plus des composants nouvellement développés pour l'application en question, on recourt également à un grand nombre de composants standards dans d'autres applications. Il existe souvent plusieurs types différents de composants standards que sont par exemple les batteries, les câbles, les connecteurs ou divers composants mécaniques. Ces composants remplissent la même fonction mais leur fiabilité et/ou rentabilité est différente. Certaines conditions peuvent mener à une profusion non gérable pour l'application à développer.

L'augmentation du nombre de solutions possibles pour chaque nouveau composant est exponentielle. Pour trouver la solution la plus économique en tenant compte de toutes les conditions-cadres, diverses méthodes d'optimisation assistées par ordinateur peuvent être utilisées. Ces méthodes sont plus ou moins rapides dans le calcul de la solution et se distinguent également de par la qualité de leur résultat.

La méthode utilisée doit trouver les types qui regroupés donneront le système global le plus économique ; la topologie, les emplacements de montage et la fiabilité théorique du système étant fixés.

La présente thèse analyse de façon critique trois méthodes d'optimisation et compare leurs avantages et inconvénients ainsi que leurs possibilités d'application.

Les données imprécises des composants de la solution trouvée sont également analysées. Ainsi, il est possible de comparer les systèmes à des stades de développement différents et d'estimer les risques de développement.



# 1 Einleitung

Ende der 30er Jahre bestand das Bremssystem eines VW-Käfers aus nur wenigen Komponenten. Bei diesem ersten Käfer begnügte man sich beispielsweise noch mit einer Einkreis-Bremsanlage mit 4 Trommelbremsen, die mit einem Seilzug betätigt wurden. Dieses System verfügte ausschließlich über die Funktion „bremsen“ und es standen keine Varianten bei der Auswahl der Komponenten zur Verfügung. Die Realisierung der (Haupt-) Funktion stand im Vordergrund.

Heute ist für einen PKW ein hydraulisches Zweikreis-Bremssystem in verschiedenen Varianten der Stand der Technik. Es verfügt oftmals über zusätzliche Funktionen, wie ABS (Anti-Blockier-System) oder ESP (Elektronisches-Stabilitäts-Programm) und wird von diversen Herstellern angeboten.

Im Kfz gibt es für die unterschiedlichen Komponenten oftmals verschiedene Möglichkeiten für die Wahl der Einbauorte. Die jeweiligen Umgebungsparameter stellen individuelle (Mindest-) Anforderungen an die dort einzubauenden Komponenten. Die Folge daraus ist, dass für eine Komponente diverse verschiedene (Standard-) Typen angeboten werden. Sie unterscheiden sich nicht hinsichtlich ihrer Funktion, wohl aber in ihrer Wirtschaftlichkeit und Qualität. Erhöhte Anforderungen bewirken mehr Aufwand bei der Entwicklung der Typen. Durch diesen erhöhten Aufwand ergibt sich nicht nur eine gesteigerte Qualität (= geringere Ausfallrate) der Typen, sondern es kann auch zu erhöhten Kosten, veränderten Ausmaßen und Gewicht kommen. Aus diesen Parametern ergibt sich die Wirtschaftlichkeit der Typen bzw. des Gesamtsystems. Sie sollte im Verhältnis zur Qualität des Gesamtsystems optimal sein. Was darunter zu verstehen ist und wie diese Optimierung realisiert werden kann, wird in dieser Arbeit dargestellt.

Eine zukünftige Elektromechanische Bremse (EMB), wie sie als Beispielsystem nach [OKC01] in dieser Arbeit dient, könnte bereits ca. 135 Komponenten aufweisen. Sie muss neben der Hauptfunktionen des Bremsens eine Vielzahl weiterer Funktionen ausführen. Die bekanntesten sind das bereits aus der hydraulischen Zweikreis-Bremse bekannte ABS, ESP und ASR (Antriebs-Schlupf-Regelung). Weitere Funktionen werden ermöglicht, von denen das sogenannte Hill-Holding oder die Elektronische Parkbremse (EPB) nur zwei Beispiele sind. Jede dieser Funktionen erfordert Sensoren und Aktoren. Steuergeräte und Software koordinieren das Gesamtsystem und stellen eine Verbindung zu den anderen Fahrzeugsystemen her.

Mit jeder neuen Fahrzeuggeneration erhöht sich nicht nur die Anzahl und die Komplexität der einzelnen Systeme und Funktionen, sondern es steigen meist auch die Anforderungen an deren Sicherheit und Zuverlässigkeit. Ein Teil der bekannten Systeme, aber auch zukünftige Fahrzeugsysteme, wie beispielsweise die EMB oder die Elektromechanische Lenkung (EML), werden elektronisch und in steigendem Maße ohne mechanische Redundanz realisiert. Ausgiebige Sicherheits- und Zuverlässigkeitsanalysen sind deshalb nicht ausschließlich für sicherheitsrelevante Funktionen erforderlich.

Die Durchführung von Fehler- Möglichkeits- und –Einflussanalysen (FMEA) und auch Fehlerbaumanalysen (FBA) sind hierzu bereits im Entwicklungsprozess etabliert. Auch die standardmäßige Durchführung von Funktions-Risiko-Analysen (FRA) ist zukünftig geplant [Bin01].

Unter anderem aus diesen Analysen und der steigenden Komplexität der Fahrzeuge, in dem die vielen verschiedene Systeme fehlerfrei miteinander kooperieren müssen, ergeben sich scheinbar erhöhte Anforderungen an die Qualität der

Komponententypen. Auf den ersten Blick deutet das auf eine sinkende Wirtschaftlichkeit der Fahrzeuge (z. B. durch erhöhte Kosten und Gewicht) hin; vielleicht wird auch mehr Bauraum benötigt. Nach wie vor fordert der Kunde jedoch kostengünstige Fahrzeuge bei geringster Emissionsbelastung mit höchstem Sicherheits- und Zuverlässigkeitsstandard.

Ein Widerspruch?

Die Motivation für diese Arbeit resultiert aus der Forderung, das vorhandene Optimierungspotential bei der Auswahl der Typen jeder Komponente systematisch und quantitativ belegbar auszuschöpfen. Auch der Aspekt, dass oftmals während des Entwicklungsprozesses keine genauen Daten über die Typen vorliegen, soll betrachtet werden.

Ein Ziel dieser Arbeit ist es, eine Methode zu finden, die das Problem der optimalen Kombination der unterschiedlichen Typen hinsichtlich der Wirtschaftlichkeit des Gesamtsystems bei gegebener Zuverlässigkeitsanforderung und System-Topologie löst. Einschränkungen bei der Zuverlässigkeit des Gesamtsystems, also dessen Ausfallrate, sollen nicht gemacht werden.

Zunächst wurde eine Vergleichsmöglichkeit hinsichtlich der Wirtschaftlichkeit von Systemen geschaffen. Hierzu wird in Kapitel 2.1 dargestellt, wie die zu vergleichenden unterschiedlichen Parameter auf nur einen Wert reduziert werden können. Es resultiert die Einführung der Wirtschaftlichkeitsmaßzahl  $W$ .

Bei der vorliegenden Aufgabe wird insbesondere der exponentiell mit der Größe der Datenbasis wachsende Lösungsraum und der in diesem Maße steigende Optimierungsaufwand berücksichtigt.

Methoden zum Lösen kombinatorischer Optimierungsprobleme gibt es viele [Pie69], [Mic94], [Mue95]. Gängige Methoden basieren auf einem graphentheoretischen Ansatz. Diese haben im allgemeinen Nachteile bei der Beherrschung hoher Systemkomplexität. Auch die untersuchte Methode, sie ist im Kapitel 4.1 dargestellt, hat daher keine praktische Verwendbarkeit.

Zum Lösen von Aufgaben mit komplexem Lösungsraum werden meist heuristische Methoden angewendet [OtW93] [Pie69]. Zwei Methoden dieser Art werden in den Kapiteln 4.2 und 4.3 untersucht.

Im Kapitel 4.2 wird eine Neuro-Fuzzy-Methode dargestellt. Solche Methoden werden normalerweise zur Regelung komplexer Vorgänge eingesetzt. In dieser Arbeit wurde versucht, diese Fähigkeit der Regelung komplexer Vorgänge für die Optimierung komplexer Lösungsräume zu nutzen. Die ursprüngliche Idee der Regelung wurde also für die Optimierung angepasst. Ihre Stärken liegen im geringen Optimierungsaufwand, aber deutliche Schwächen zeigten sich in der Qualität der Ergebnisse.

Daher wird in Kapitel 4.3 als zweite heuristische Methode ein Genetischer Algorithmus untersucht. Solche evolutionären Algorithmen sind auf dem Gebiet der kombinatorischen Optimierung bereits vielfach erprobt [Lüb99] [Mue95] [Mic94]. Ein auf das in dieser Arbeit untersuchte Problem anwendbarer Genetischer Algorithmus wird im Kapitel 4.3 dargestellt. Dieser ist unabhängig von der Größe des Lösungsraumes einsetzbar und liefert Ergebnisse mit guter Qualität. Genetische Algorithmen verwenden zur Beurteilung gefundener Lösungen sogenannte Fitnessfunktionen. Als Erweiterung der gängigen genetischen Prinzipien wurde eine solche, auf die vorliegende Problematik abgestimmte Fitnessfunktion gefunden.



Weiterhin wurde durch „gezielte Genmanipulation“ der Algorithmus in seiner Effektivität gesteigert.

In den verschiedenen Phasen des Entwicklungsprozesses liegen leider oftmals ungenaue Parameter der verschiedenen Typen vor. Diese Ungenauigkeit kann die Entwicklung verlangsamen, es können unvorhergesehene Kosten entstehen und Nachbesserungen werden nötig. Die sich ergebende Ungenauigkeit konnte bisher nicht bewertet werden.

Bei der Sicherheitsanalyse wird bereits die Unsicherheit von Ergebnissen durch Fuzzy-Mengen beurteilt [Mey00]. Diese Methode wird in dieser Arbeit auf die Zuverlässigkeitsanalyse übertragen. Um eine solche Bewertung der unscharfen Parameter und des sich daraus ergebenden unscharfen Ergebnisses zu ermöglichen, wird die „Theorie der vagen Daten“ [GRS00] auf das Optimierungsergebnis angepasst und angewandt. Die Untersuchungen hierzu sind in Kapitel 5 dargestellt.

Durch diese Methode lassen sich unsichere (vage) Daten der Typen hinsichtlich ihrer Zuverlässigkeit und Wirtschaftlichkeit bewerten. Die Entwicklungsunsicherheit der jeweiligen Typen und des Gesamtsystems kann nun quantifiziert werden. Für ein betrachtetes System lässt sich in jeder Phase des Produktentwicklungsprozesses (PEP) beurteilen, inwiefern sich das Risiko der Entwicklung hinsichtlich der Zuverlässigkeit und der Wirtschaftlichkeit verändert. Das betrachtete technische System ist während des Produktentwicklungsprozesses jederzeit mit Parametern darstellbar, wobei auch die (Aussage-) Qualität dieser Parameter bewertet wird.

## 2 Die Optimierungsaufgabe

Fahrzeugsysteme bestehen aus einer großen Anzahl unterschiedlicher Komponenten. Bei der Neuentwicklung von Applikationen kann oftmals auf eine Vielzahl bereits vorhandener und vielleicht schon in der Serie erprobter Typen der benötigten Komponenten zurückgegriffen werden.

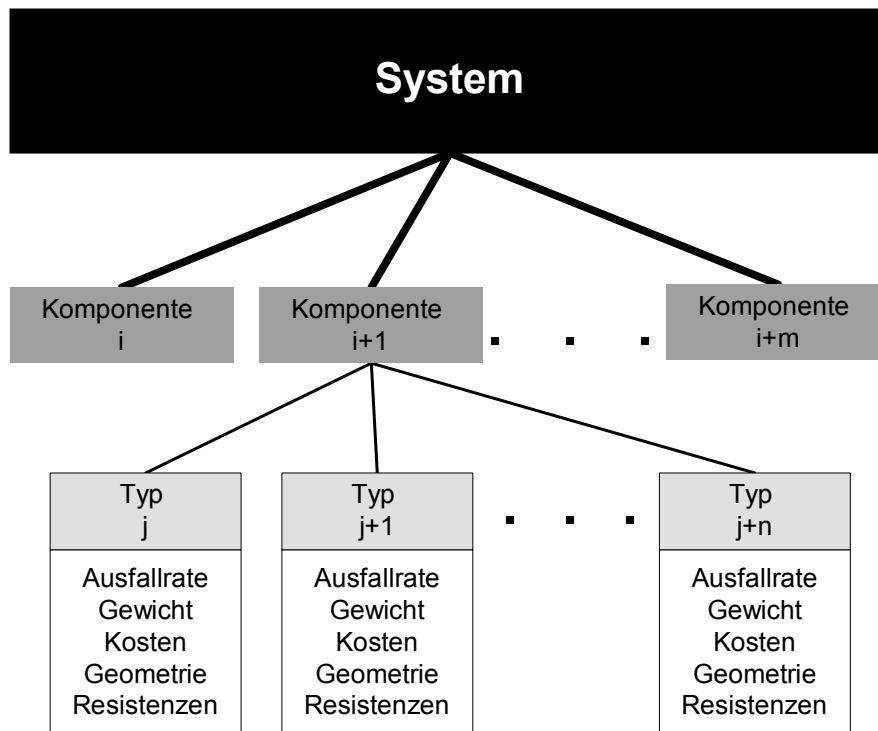
Alle verfügbaren Typen werden in einer Datenbasis zusammengefasst. Die in dieser Arbeit vorgestellten Methoden verwenden diese als Grundlage für die Optimierung. Die darin unter einem Komponentennamen zusammengefassten (Standard-) Typen erfüllen aus technischer Sicht alle dieselbe Funktion (z. B. eine Batterie speichert Energie, unabhängig vom Hersteller). Sie genügen bestimmten minimalen Anforderungen (beispielsweise die minimale Kapazität einer Batterie). Die jeweilige Optimierungsmethode wählt aus der Datenbasis entsprechend der vorgegebenen Komponente einen Typen aus und muss die technische Verwendbarkeit nicht überprüfen. Die wählbaren Typen einer Komponente unterscheiden sich zunächst nur in ihrer Ausfallrate und den Wirtschaftlichkeitsparametern (Gewicht, Kosten, Geometrie).

Nicht jeder Typ darf an jedem beliebigen Ort im Kfz verbaut werden. Die Anforderungen an die Komponenten sind in sogenannten Lastenheften spezifiziert.

Ein Fahrzeug kann in verschiedene Zonen aufgeteilt werden. Für jede existieren bestimmte Anforderungen, die es zu erfüllen gilt. Diese sind in sogenannten Technischen Lieferbedingungen aufgeführt [Vol00].

Ein weiterer Unterschied zwischen den wählbaren Typen einer Komponente resultiert aus der Art, wie die Anforderungen der Einbauorte (Resistenzen) erfüllt werden. Ein Auswahlkriterium für die potenziellen Typen ergibt sich somit aus einem Vergleich der Typen-Spezifikation des Zulieferers der Bauteile mit den Anforderungen des Lastenheftes und den Technischen Lieferbedingungen des Fahrzeugherstellers.

Die Sichtweise des Aufbaus von Systemen, wie sie auch in dieser Arbeit verwendet wird, ist in Abbildung 1 dargestellt.



**Abbildung 1: Darstellung eines Systems, bestehend aus Komponenten und deren jeweiligen Typen**

Die Gesamtheit der Typen aller Komponenten ist die Datenbasis, aufgrund derer die wirtschaftlichste Variante des Systems gefunden werden soll, wobei die Ausfallrate und die Topologie vorgegeben sind. Es handelt sich um ein Problem der kombinatorischen Optimierung.

Die geplanten Einbauorte geben die von den Typen zu erfüllenden Randbedingungen vor. Sie werden mit der Widerstandsfähigkeit der Typen gegen Umwelteinflüsse (Resistenz) verglichen. Auch die Kombinierbarkeit aller Typen ist zu berücksichtigen.

Neben den Wirtschaftlichkeits- und Zuverlässigkeits-Parametern entscheiden somit die strukturellen Kriterien über die Verwendbarkeit bestimmter Typen.

Zur Beurteilung der Wirtschaftlichkeit eines Systems betrachtet man neben den Kosten auch das Gewicht und den Bauraum der zu verwendenden Typen. Wie stark jeder Parameter die Wirtschaftlichkeit beeinflusst, ist auch davon abhängig, für welches Fahrzeug das Zielsystem dimensioniert werden soll. Eventuell muss die Art der Ausstattung des Fahrzeugs berücksichtigt werden.

Die Ausfallrate eines Systems berechnet sich aus den Ausfallraten der verwendeten Typen [PeM85], [Bir91]. Diese hängen wiederum von den Belastungen an ihren Einbauorten ab. Ein System gilt als nicht mehr voll funktionsfähig (ausgefallen), wenn eine seiner Komponenten ausgefallen ist. Es muss repariert werden.

Aus der Vielfalt der Typen, den Randbedingungen der Einbauorte und den unterschiedlichen Wirtschaftlichkeitsparametern resultieren viele Freiheitsgrade bei der Optimierung. Diese zu beherrschen und die zulässige optimale Kombination der Typen zu finden, scheint offenbar nur mit rechnergestützten Methoden möglich zu sein.

*Eine Aufgabe ist somit das Finden einer allgemeingültigen Methode, welche diejenigen Komponententypen auswählt, die insgesamt das wirtschaftlichste System darstellen. Die Topologie und die Ausfallrate des Systems sind vorgegeben und zu berücksichtigen.*

Die Ansprüche an diese Methode sind im Rahmen der gegebenen Aufgabe folgende:

- i. Es soll möglichst die „*optimale*“ Lösung gefunden werden, d.h. die resultierende Ausfallrate muss mindestens der Vorgabe entsprechen und die nachfolgend in Kapitel 2.1 eingeführte Wirtschaftlichkeitsmaßzahl der gefundenen Lösung soll unter dieser Vorgabe minimal sein.
- ii. Die Methode muss „*praktikabel*“, d.h.
  - a) der Berechnungsaufwand muss realisierbar und
  - b) die zur Optimierung benötigten Parameter der potenziellen Typen müssen verfügbar sein.
- iii. Die Anwendbarkeit der Methode soll *unabhängig* von der Architektur des zu optimierenden Systems sein.

## **2.1 Wirtschaftlichkeitsmaßzahl**

Um eine Vergleichsmöglichkeit der betrachteten Wirtschaftlichkeitsparameter zu erhalten, wurde im Rahmen dieser Arbeit eine Maßzahl entwickelt, die nachfolgend Wirtschaftlichkeitsmaßzahl  $W$  genannt wird und in Gleichung (1) dargestellt ist.

Jeder Typ einer Komponente verursacht individuelle Kosten, hat ein Gewicht und eine individuelle Geometrie. Alle diese Parameter beeinflussen die Wirtschaftlichkeit eines Systems. Da sie in dieser Form nicht miteinander vergleichbar sind, muss eine Beurteilungsmöglichkeit für die Wirtschaftlichkeit des Gesamtsystems gefunden werden. Wird ein System wirtschaftlicher, wenn billigere Typen gewählt werden, sich dafür aber ein höheres Gesamtgewicht ergibt?

Die Wirtschaftlichkeitsmaßzahl  $W$  ermöglicht einen quantifizierten Vergleich dieser Parameter und berechnet sich, wie nachfolgend dargestellt:

$$W_i = g_K \frac{K_i}{K_{ref}} + g_G \frac{G_i}{G_{ref}} + g_V \frac{V_i}{V_{ref}} \quad (1)$$

mit

$K$  = Kosten,  $G$  = Gewicht,  $V$  = Volumen,

$i$  = Typenindex,  $ref$  = Referenztypindex,

$g$  = Gewichtung.

$W$  wird für jeden Typ separat berechnet und ist eine dimensionslose Maßzahl. In ihr wird jeder der drei bei dem vorliegenden Problem zu beurteilenden Wirtschaftlichkeitsparameter ( $K$ ,  $G$ ,  $V$ ) berücksichtigt.

Jeder Wirtschaftlichkeitsparameter eines betrachteten Typs wird normiert. Der Normierungswert entspricht dem höchsten Wert der Typen der zugehörigen Komponente. Der sich ergebende Quotient kann somit höchstens  $W_{max} = 1$  sein.

Jeder der resultierenden drei Quotienten wird anschließend mit dem entsprechenden Gewichtungsfaktor  $g$  multipliziert, wobei gilt

$$g_K + g_G + g_V = 1. \quad (2)$$

Wie stark jeder Gewichtungsfaktor die Wirtschaftlichkeit beeinflusst, kann nicht pauschal definiert werden. Dies ist abhängig vom Ziel-Fahrzeug und evtl. auch von dessen Ausstattungsgrad. Während in dem einen Fahrzeug wenig Bauraum vorhanden ist, muss für die gleiche Applikation in einem anderen Fahrzeug vielleicht das Gesamtgewicht der Applikation höher bewertet werden. Der Entwickler muss daher eine Möglichkeit haben, seine individuellen Wirtschaftlichkeitsanforderungen in die Optimierung einzubringen. Hierzu dienen die Gewichtungsfaktoren.

Nach der Normierung und Gewichtung liegen die Werte der verschiedenen Parameter in identischen Wertebereichen und sind ohne Einheit (dimensionslos). Der Vergleich unterschiedlicher Parameter wird möglich.

Für die resultierende Wirtschaftlichkeitsmaßzahl  $W$  jedes Typs gilt folgender Wertebereich

$$0 \leq W \leq W_{\max} = 1. \quad (3)$$

Die höchstmögliche Wirtschaftlichkeitsmaßzahl ( $W_{\max}=1$ ) ergibt sich für den unwirtschaftlichsten Typ der betrachteten Komponente.

Zur Berechnung der Wirtschaftlichkeitsmaßzahl wird das „Bauvolumen“ jedes Typs aus seiner jeweiligen Geometrie (z. B. Länge, Breite, Höhe) berechnet. Für Typen mit „zusammengesetzten“ Geometrien (Freiformkörper) darf vereinfachend auch die „Umhüllende“ berechnet werden, da es sich um eine Betrachtung der Wirtschaftlichkeit handelt. Nur wenn die Einbaubarkeit an bestimmten Einbauorten beurteilt werden müsste, wären die Form, die Ausdehnung und auch die Einbaulage betrachteter Typen unter Berücksichtigung anderer Komponenten relevant.

Durch die Einführung der Wirtschaftlichkeitsmaßzahl reduziert sich die Anzahl der Freiheitsgrade, durch die das System beurteilt wird, von 6 (Kosten, Gewicht, Länge, Breite, Höhe, Ausfallrate) auf 2 (Wirtschaftlichkeitsmaßzahl, Ausfallrate).

Nach der Berechnung von  $W$  ist jeder Typ durch die beiden Werte Wirtschaftlichkeitsmaßzahl  $W$  und Ausfallrate  $\lambda$  hinreichend für die anschließende Optimierung definiert.

## 2.2 Ausfallrate

Nach [PeM85] lässt sich das Ausfallverhalten von Komponenten auf zwei verschiedene Arten beschreiben:

- durch ihre Ausfallrate und
- durch ihre Ausfallwahrscheinlichkeit je Anforderung.

Die Ausfallrate ( $\lambda(t)$ ) ist darin definiert als „... die relative Abnahme des Bestandes an noch nicht ausgefallenen Komponenten pro Zeiteinheit ...“.

Weiterhin wird als Ausfallwahrscheinlichkeit ( $F(t)$ ) die Wahrscheinlichkeit definiert, „... dass bei Anforderung der Komponente ein Ausfall vorliegt, die Komponentenfunktion also in dem vor der Anforderung liegenden Zeitraum ausgefallen ist oder spätestens zum Anforderungszeitpunkt ausfällt.“

In dieser Arbeit dient nur die Ausfallrate  $\lambda(t)$  als Berechnungsgrundlage. Sie ist abhängig von ihrer Betriebszeit  $t$  der Systeme. Die Ausfall-Charakteristik während der Lebensdauer von Systemen zeigt die sogenannte „Badewannenkurve“ [BGU95] in Abbildung 2.



**Abbildung 2: Ausfallcharakteristik von Bauteilen während ihrer Lebensdauer**

Zu einem frühen Zeitpunkt (Phase 1 in Abbildung 2) fallen Systeme vermehrt aus. Diese sogenannten Frühausfälle resultieren z. B. aus Fertigungstoleranzen und Einbaufehlern.

Nachdem diese erfasst sind, sollten die verbleibenden Systeme so ausgelegt sein, dass sie während der geplanten Lebensdauer bzw. innerhalb der geplanten Wartungsintervalle nicht ausfallen. Leider ist dies dennoch nicht immer zu vermeiden. In dieser Nutzungsphase, der dargestellten Phase 2, fallen die Systeme mit einer annähernd konstanten Ausfallrate aus. Man spricht von den sogenannten Nutzungsausfällen.

Nach der Nutzungsphase steigt die Ausfallrate bei den verbleibenden Systemen wieder an. Diese Ausfälle resultieren aus dem Verschleiß und der Alterung der Bauteile. Sie werden daher Verschleißausfälle (Phase 3) genannt.

In der Automobilindustrie werden die Systeme einer Endkontrolle unterzogen, regelmäßig gewartet und auf die Lebensdauer (ausgenommen Verschleißteile, wie z. B. Bremsbeläge) hin dimensioniert. Daher legt man für Berechnungen vereinfachend die konstante Ausfallrate der Nutzungsausfälle zugrunde. Die Ausfallrate  $\lambda$  ist somit nicht mehr zeitabhängig (Gleichung (4)) und es folgt

$$\lambda(t) = \lambda. \quad (4)$$

Die Ausfallrate des Gesamtsystems als logisches Seriensystem ergibt sich nach [Bir91] als Summe der Ausfallraten aller verwendeten Bauteile (Typen  $i$ ) zu

$$\lambda_{ges} = \sum_i \lambda_i. \quad (5)$$

Die Ausfallrate für ein System ist in der Regel auf eine beliebige Zeiteinheit bezogen (hier: 1 Stunde), z. B.

$$\lambda_{ges} = 5,2 \cdot 10^{-7} \frac{1}{h}. \quad (6)$$

Das Beispiel aus Gleichung (6) kann so interpretiert werden, dass im Schnitt ein betrachtetes System laut Statistik nach 52.000.000 Stunden ausfällt, bzw. von 10.000.000 Systemen innerhalb einer Stunde 5,2 Systeme ausfallen.

### 3 Lösungsraum und dessen Einschränkung

Zu lösende Probleme werden in der Mathematik/Informatik hinsichtlich ihrer Komplexität in unterschiedliche Klassen unterteilt:

- **P-Probleme** sind deterministisch und in **polynomialem** Aufwand lösbar.
- **NP-Probleme** sind nicht-deterministisch lösbar, aber die Überprüfung, ob eine Lösung gefunden wurde, ist mit **polynomialem** Aufwand möglich.
- **NP-vollständige Probleme** sind NP-Probleme, auf die jedes andere Problem mit polynomialem Aufwand reduziert werden kann.

Bei einem NP-Problem wird eine Lösung „geschätzt“ und in polynomialem Aufwand überprüft, ob die Lösung gültig ist. Ist ein NP-vollständiges Problem mit polynomialem Aufwand lösbar, so sind auch alle anderen Probleme dieser Klasse lösbar. Der Aufwand für das Auffinden einer optimalen Lösung wächst bei NP-vollständigen Problemen schneller (exponentiell) als polynomial mit der Größe der Aufgabe. Lösungen werden daher nur für kleine Systeme gefunden. Solche Systeme können in Ihrer Komplexität beliebig gesteigert werden, bis sie aufgrund ihres hohen Aufwandes nicht mehr lösbar sind. Für diese muss dann eine möglichst gute *Näherungslösung* gesucht werden.

Bei der vorliegenden Arbeit ist die Größe des Lösungsraums ( $O$ ) (die Anzahl der Varianten) abhängig von der Anzahl der verfügbaren Typen ( $T$ ) und der Anzahl der Komponenten ( $K$ ) und steigt exponentiell. Für den in Abbildung 3 dargestellten Verlauf gilt vereinfacht die Annahme, dass je Komponente jeweils dieselbe Anzahl Typen existieren. In der dargestellten Schar wurde die Typenanzahl jedoch gleichermaßen verändert.

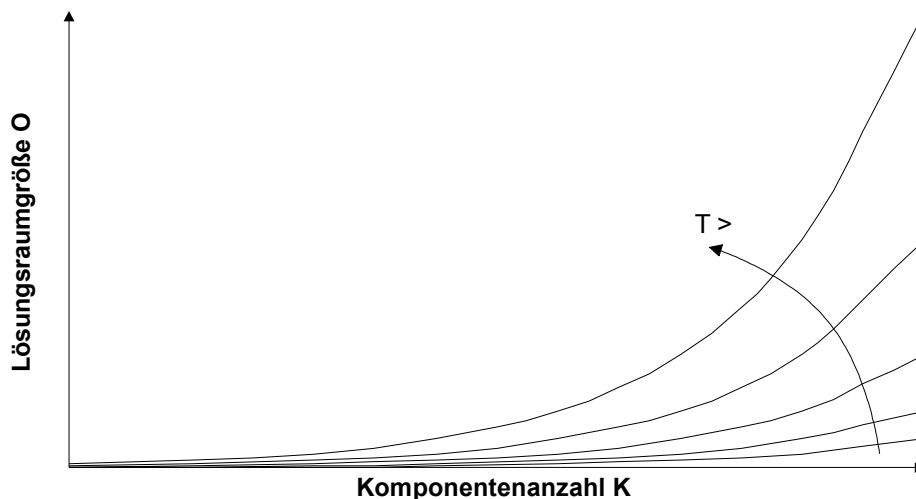


Abbildung 3: Schar des exponentiellen Verhaltens der Größe des Lösungsraumes ( $O$ ) in Abhängigkeit von der Anzahl der Komponenten ( $K$ ) bei jeweils konstanter Typen-Anzahl

Die Lösungsraumgröße berechnet sich demnach gemäß Gleichung (7) zu

$$O = T^K \quad (7)$$

mit

$$T, K \in \mathbb{N} .$$

Besteht ein System beispielsweise aus 10 Komponenten mit jeweils 2 möglichen Typen je Komponente, so ergeben sich

$$2^{10} = 1024 \text{ Varianten} . \quad (8)$$

Die Gleichung (7) lässt sich anwenden, wenn für jede Komponente die gleiche Anzahl an Typen zur Verfügung steht. Praktisch ist dies aber nicht der Fall. Die resultierende Anzahl der möglichen Lösungen berechnet sich durch Multiplikation der Anzahl der jeweils verfügbaren Typen. Es sei angenommen, dass eine Anzahl von  $n$  Komponenten  $K$  mit jeweils  $T_i$  Typen existieren, wobei  $i$  der Index der Komponente  $K$  ist. Die Größe des Lösungsraumes ergibt sich dann zu

$$O = \prod_{i=1}^n T_i . \quad (9)$$

Bereits bei Systemen mit „überschaubarer“ Lösungsraumgröße ist die Berechnung und der Vergleich aller Varianten sehr aufwendig. Bei sehr komplexen Systemen lässt sich ein solcher Vergleich aller Varianten nicht mehr realisieren, denn es handelt sich um ein NP-vollständiges Problem. Dieses Verhalten ist unabhängig von der gewählten Optimierungsmethode [Rüd95].

Die Datenbasis besteht aus diskreten Bauteilen (Typen), so dass durch ihre beliebige Kombination eine diskrete Lösung  $I$  entsteht. Die Ausfallrate  $\lambda(I)$  der gefundene Lösung sollte mindestens so gut wie die Forderung  $\lambda_{soll}$  sein, d. h.

$$\lambda(I) \leq \lambda_{soll} . \quad (10)$$

Die optimale Lösung ist somit in der Teilmenge des Lösungsraumes zu suchen, die alle die Forderungen aus Gleichung (10) erfüllenden Lösungen enthält. Die wirtschaftlichste Lösung dieser Teilmenge ist das gesuchte Optimum.

Es wird vorausgesetzt, dass alle Komponententypen der Datenbasis die technischen Anforderungen des Systems erfüllen. Eine diesbezügliche Überprüfung ist nicht die Aufgabe der Optimierungsmethoden.

Es ist denkbar den Lösungsraum vor der eigentlichen Optimierung einzuschränken, um so den Berechnungsaufwand zu verringern. Dies lässt sich durch Reduzierung der Datenbasis erreichen.

Typen, die offensichtlich nicht Teil der optimalen Lösung sein können, dürfen bereits vor der eigentlichen Optimierung aus der Datenbasis gelöscht werden. Die Komplexität des Lösungsraumes wird verringert. Die Suche nach den zu löschenden Typen darf jedoch den Gesamtaufwand der Methode nicht vergrößern.



Die Komplexität des Lösungsraumes und somit der Berechnungsaufwand kann reduziert werden, falls die Datenbasis Komponententypen enthält, die nicht die Anforderungen

- der Einbauorte,
- der Kombinationsfähigkeit mit anderen Typen oder
- der Ausfallrate des Gesamtsystems

erfüllen.

### **3.1 Einbauorte**

Da die Topologie des Systems festgelegt ist, sind die Einbauorte der Komponenten und die dort vorherrschenden Bedingungen bekannt. Diese sind z. B. die Umgebungs- und Betriebstemperatur an den Einbauorten, der vorhandene Einbauraum, aber auch die mechanischen (Vibrationsfestigkeit) oder chemischen (Resistenz gegen verschiedene Stoffe) Anforderungen. Diese Randbedingungen werden in sogenannten Lastenheften und „Technischen Lieferbedingungen“ festgehalten.

Alle Typen, welche nicht die Mindestanforderungen der relevanten Einbauorte erfüllen, dürfen aus der Datenbasis gelöscht werden. Dies ist unabhängig von der anzuwendenden Optimierungsmethode.

### **3.2 Kombinationsverbot und -zuweisung**

Verschiedene Komponententypen dürfen u. U. nicht, andere müssen miteinander kombiniert werden. Im Kapitel 4.1 wird die Methode des Graphenalgorithmus (vgl. [OtW93]) dargestellt. Bei dieser lässt sich das Kombinationsverbot und auch die Kombinationszuweisung gut umsetzen, denn alle Kombinationen werden bereits vor der eigentlichen Optimierung tabellarisch erstellt. Durch einen leicht zu realisierenden Vergleich lassen sich bei dieser Methode, mit geringem Aufwand, die verbotenen Varianten finden und löschen. Dem geringen Mehraufwand durch den Vergleich steht eine größere Aufwands-Ersparnis bei der Berechnung der Lösungen und den nachfolgenden (mehrfachen) Vergleichen gegenüber.

In den Kapiteln 4.2 und 4.3 werden zwei weitere Methoden (Neuro-Fuzzy-Methode, Genetischer Algorithmus) dargestellt. Bei diesen kann erst nach der Erstellung einer Variante beurteilt werden, ob diese erlaubt ist. Da die Erstellung der Varianten sehr aufwendig ist, bewirkt die Berücksichtigung von Kombinationsverboten oder von Kombinationszuweisungen keine nennenswerte Aufwandsreduzierung.

Der Lösungsraum wird nicht verkleinert, jeder Typ ist zu berücksichtigen.

### **3.3 Ausfallrate des Gesamtsystems**

Eine einfache Begründung für das Löschen von Typen aus der Datenbasis ist bei der vorliegenden Problemstellung in der Berücksichtigung der Ausfallraten zu sehen. Es ist denkbar, dass einzelne Typen eine Ausfallrate haben, die schlechter als die geforderte des Gesamtsystems ist. Solche Typen sollten aus der Datenbasis gelöscht werden.

### **3.4 Offensichtlich schlechtere Typen**

Weiterhin können Typen, die unwirtschaftlicher als andere Typen derselben Komponente sind und deren Ausfallrate auch höher als bei diesen liegt, gelöscht werden.

Wie in Kapitel 2.1 dargestellt, lässt sich die Wirtschaftlichkeit nicht so einfach beurteilen. Es müssen die Wirtschaftlichkeitsmaßzahlen aller Typen berechnet und miteinander verglichen werden.

Diese Art der Einschränkung des Lösungsraumes erfolgt, im Unterschied zu den zuvor dargestellten Methoden, nicht durch einfache Betrachtung der Typenparameter, sondern durch explizite Bestimmung.

## 4 Methoden zum Finden der optimalen Variante

Um die optimale Lösung zu finden, scheint es notwendig, den gesamten Lösungsraum zu durchsuchen. Für diese Vorgehensweise bietet sich zunächst die Anwendung eines **Graphenalgorithmus** [OtW93], [Die00], [Vor98] an.

In Abschnitt 3 wurde dargestellt, dass der Lösungsraum ab einer bestimmten Komplexität des zu betrachtenden Systems zu groß wird, um ihn komplett durchsuchen zu können. In diesem Fall müssen heuristische Optimierungsmethoden angewendet werden.

Der Begriff Heuristik beschreibt die Lehre vom Auffinden geeigneter Lösungen in den verschiedensten Wissensgebieten. Heuristiken sind „Regeln“ zum Herbeiführen einer Lösung in komplexen Denkbereichen, wobei das Auffinden der (optimalen) Lösung nicht garantiert werden kann. Heuristiken werden u. a. bei Optimierungsproblemen, die mit klassischen Optimierungsverfahren einen mit der Problemgröße exponentiell ansteigenden Lösungsaufwand bedeuten, eingesetzt. Dies trifft auch auf die vorliegende Problemstellung zu. Das Ziel der Anwendung von Heuristiken ist es, zu Gunsten einer schnelleren Berechnung den Lösungsraum systematisch zu reduzieren. Um das Risiko des Findens suboptimaler Lösungen zu verringern, müssen die angewandten Heuristiken schlüssig sein. Sie dürfen nur unrelevante Lösungen ausschließen. Die Anwendung von Heuristiken soll hier also das Auffinden der (optimalen) Lösung bewirken, obwohl nur ein Teil des Lösungsraumes betrachtet wird.

Als zweite Methode wird in dieser Arbeit die **Neuro-Fuzzy-Methode** [NaK95] [NaK99] betrachtet. Sie ist heuristisch und deterministisch.

Die dritte untersuchte Methode ist ein **Genetischer Algorithmus** [Lüb99] [Mue95]. Solche Algorithmen basieren auf den Prinzipien der (natürlichen) Evolution. Die realisierte Methode ist heuristisch und zufallsbasiert.

Nachfolgend werden diese drei Methoden näher dargestellt.

### 4.1 Methode 1: Graphenalgorithmus

Ein Graph, wie er in der Abbildung 4 dargestellt ist, besteht aus Knoten und Kanten [MeM02]. Die nummerierten Kreise stellen die Knoten dar. Die verbindenden Pfeile sind die Kanten.

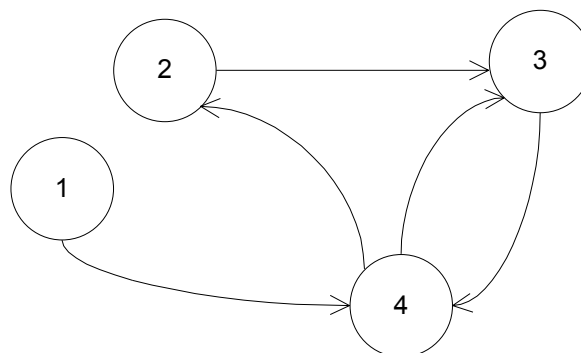


Abbildung 4: Graph

Für das untersuchte System wird jeder Komponententyp der Datenbasis durch einen

Knoten dargestellt und seine zugehörigen Parameter sind in der Kante enthalten. Bei der Suche nach der optimalen Variante mit Hilfe von Graphen sind in der Literatur verschiedene Lösungsstrategien bekannt. Dabei repräsentiert der Graph die wesentliche Struktur des Problems, befreit von „unbedeutenden“ Nebenaspekten. In [Vor98] sind verschiedene graphenbasierte Lösungsansätze dargestellt, die in [Akl89] erklärt werden:

- **Greedy:** sukzessive Bestimmung der Lösungsvariablen,
- **Branch & Bound:** Teillösungen an der vielversprechendsten Stelle verändern,
- **Backtracking:** systematische Erweiterung von Teillösungen,
- **Dynamic Programming:** Berechnung einer Folge von Teillösungen,
- **Divide & Conquer:** aufteilen, lösen, Lösungen vereinigen,
- **Enumeration:** Erzeugung aller Permutationen mit anschließender Überprüfung.

Bei Greedy-Algorithmen (greedy = gierig) wird die Gesamtlösung erstellt, indem eine Teillösung durch die jeweils „günstigste“ nächste Komponente ergänzt wird. Beim auf den Graph angewandten Greedy-Algorithmus wird eine Lösung erstellt, indem, ausgehend von einem Startknoten, der jeweils wirtschaftlichste nächste Knoten zur Lösung hinzugefügt wird. Diesen erkennt man durch den Vergleich der nächsten Kanten. Aus der Summe der wirtschaftlichsten Knoten ergibt sich beim vorliegenden Problem im Normalfall nicht die optimale Lösung, die auch die Anforderungen an die Ausfallrate zu erfüllen hat. Daher wird die Greedy-Methode, die auch mit „Hill-Climbing“ bezeichnet wird, nicht weiter betrachtet.

Beim Branch & Bound Verfahren (Verzweigung und Beschränkung) wird, ausgehend von einer vorhandenen Lösung, die Grenze eines Optimierungsparameter angepasst und die Lösung zum „vielversprechendsten“ Knoten hin verzweigt. Hieraus resultiert ein neues Unterproblem, dessen Umfang um einen Optimierungsparameter geringer ist. Die gefundene (Zwischen-) Lösung wird fixiert (bound) und der nächste Optimierungsparameter angepasst. Dieses Lösungsverfahren verwendet das Prinzip der Aufteilung und Begrenzung des Lösungsraumes, um das Durchsuchen des vollständigen Lösungsraumes zu vermeiden. Die Schwierigkeit bei der Anwendung auf das vorliegende Problem ist, dass die korrekten Grenzen der Optimierungsparameter nicht bekannt sind. Daher ist dieses Verfahren nicht anwendbar.

Beim Backtracking-Verfahren wird versucht, eine Teillösung bis zur optimalen Gesamtlösung auszubauen. Hierbei kann es vorkommen, dass der Weg in einer „Sackgasse“ endet. Einer oder mehrere der letzten Teilschritte werden dann rückgängig (backtracking) gemacht und von dort aus ein anderer Weg beschritten. Entweder wird irgendwann eine Lösung gefunden oder die Optimierung abgebrochen. Die Konvergenz ist also nicht garantiert.

Beim Dynamischen Programmieren (Dynamic Programming) wird das Problem in Teilprobleme zerlegt. Diese werden gelöst und aus den optimalen Teillösungen wird die Gesamtlösung zusammengefügt. Die Teillösungen müssen dabei unabhängig voneinander sein. Neben der Optimierung der Wirtschaftlichkeit, müssen auch die Anforderungen an die maximal zulässige Ausfallrate und die in Kapitel 3 genannten Randbedingungen erfüllt sein. Aus diesem Grund ist eine Aussage über die Parameter-Güte von Teillösungen nicht oder nur sehr ungenau möglich. Die Optimierungs-Parameter der Teillösungen sind schwer zu finden. Daher entspricht die Gesamtlösung oft nicht dem Optimum.

Eine Erweiterung des Dynamischen Programmierens ist das „Divide & Conquer“. Hier wird der Graph zunächst in Teilgraphen aufgeteilt und die Qualität der verwendeten Typen mit dem Mittelwert der Lösung verglichen. Die schlechtesten Typen in jedem Block werden ersetzt und die Teillösungen wieder zusammengesetzt. Die Teillösungen sollten möglichst gleichwertig aufgeteilt werden und dennoch unabhängig voneinander sein. Die Werte der Komponententypen der einzelnen Blöcke müssten sich dann um Größenordnungen voneinander unterscheiden. Sind diese Voraussetzungen erfüllt, so erhält man nach der Teilloptimierung der Blöcke und deren Vereinigung letztlich das optimale System. Divide & Conquer wäre dann anwendbar. Anderenfalls würden voraussichtlich suboptimale Lösungen entstehen.

Beispiel:

Die in einem Fahrzeug-Bordnetz eingesetzten Kabel besitzen typischer Weise eine sehr geringe Ausfallrate. Sie werden daher kaum die Auswahl einer Batterie des gleichen Systems, die sehr oft der Grund für Ausfälle sein kann, beeinflussen. Für solche Komponenten mit deutlich unterschiedlichen Größenordnungen, ist die Aufteilung in einzelne Optimierungsblöcke denkbar. Sollte bei dem betrachteten Beispielsystem eine weitere Komponente existieren, deren Typen Ausfallraten besitzen, die den gesamten Bereich zwischen den Ausfallraten der Batterien und denen der Kabel abdecken, so ist eine Aufteilung des Lösungsraums nicht mehr möglich. Diese neue Komponente lässt sich nicht mehr eindeutig genau einer der beiden Klassen zuordnen. Sie beeinflusst beide.

Die Anwendbarkeit des Lösungsansatzes „Divide & Conquer“ ist also abhängig vom zu untersuchenden System und somit nicht immer anwendbar.

In dieser Arbeit wird daher der Graphenalgorithmus<sup>1)</sup> nach der Methode der Enumeration untersucht. Hierbei wird der gesamte Lösungsraum betrachtet. Dieser enthält garantiert die optimale Lösung und findet sie auch, sofern die Größe des Lösungsraums berechenbar ist.

Bei der vorliegenden Problemstellung wird das System durch einen gerichteten, nicht zyklischen Graphen (Abbildung 6) nachgebildet.

Bei diesem ist es möglich jede denkbare Lösung zu erstellen, indem der Graph systematisch vollständig durchlaufen wird. Werden anschließend alle möglichen Lösungen berechnet und miteinander verglichen, so wird die optimale Lösung gefunden.

Für das vorliegende Problem der Ausfallrate-Wirtschaftlichkeits-Optimierung werden im Kapitel 4.1.2 dieser Arbeit verschiedene Möglichkeiten zur Beschränkung des zu untersuchenden Lösungsraumes dargestellt. Die dargestellten Kriterien sind primär für das vorliegende Problem gültig, aber eine Anpassung auf ähnliche Aufgaben ist denkbar.

Der Berechnungsaufwand dieser Methode ist vorhersagbar.

---

<sup>1)</sup> Wird in dieser Arbeit der Begriff „Graphenalgorithmus“ genannt, sei nachfolgend immer die Methode „Enumeration“ gemeint.

Der realisierte Graphenalgorithmus gliedert sich in 4 Phasen:

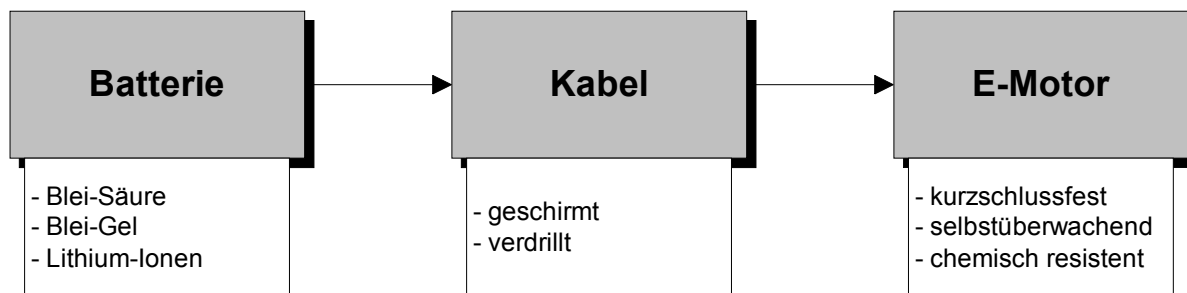
Phase 1: Erstellung des Graphen und Definition der Lösungsräume.

Phase 2: Einschränkung der Lösungsräume.

Phase 3: Berechnung der verbleibenden Lösungen.

Phase 4: Ermittlung der optimalen Lösung.

Diese Phasen werden nachfolgend anhand eines einfachen Beispiel erklärt, das aus Gründen der Nachvollziehbarkeit nur drei Komponenten enthält (Batterie, Kabel, E-Motor). Für jede dieser drei Komponenten sind mehrere verschiedene Typen verfügbar (z. B. 3 Batterien, 2 Kabel und 3 E-Motoren).



**Abbildung 5: Beispielsystem mit drei Komponenten und 3, 2 und 3 Typen je Komponente**

Es ergeben sich für dieses Beispiel nach Gleichung (9) insgesamt

$$3 \cdot 2 \cdot 3 = 18 \text{ Lösungen .} \quad (11)$$

#### 4.1.1 Phase 1: Erstellung des Graphen und Definition der Lösungsräume mit Auswertung des Kostenvektors

Die theoretischen Grundlagen dieses Kapitels entstammen aus [Die00] und sind nachfolgend in vereinfachter Form dargestellt.

Die Basis für den vorliegenden Graphenalgorithmus bildet ein gerichteter, nicht zyklischer Graph  $G$ . Dieser besteht aus einer Menge von Knoten  $V$  und einer Menge gerichteter Kanten  $E$ . Der Graph des Beispiels aus Abbildung 5 hat demnach folgende Form (Abbildung 6).

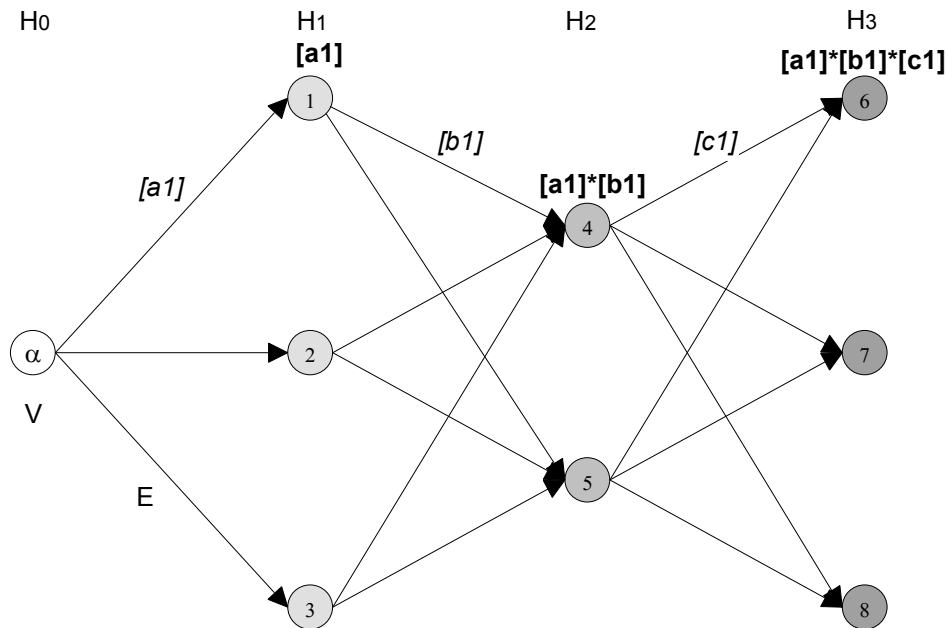


Abbildung 6: Der gerichtete, nicht zyklische Graph  $G$  (mit Kostenfunktion für einen möglichen Weg)

Definiert wird ein Graph durch das 2-Tupel

$$G = (V, E). \quad (12)$$

Für die Knoten  $V$  gilt

$$V = \{1, 2, \dots, |V|\}. \quad (13)$$

Sie sind im Beispiel mit  $(\alpha, 1, 2, \dots, 8)$  nummeriert. Die Kantenmenge  $E$  entsteht als Teil des kartesischen Produktes der Knoten

$$E \subseteq V \times V. \quad (14)$$

Man spricht auch von bipartiten Graphen.

Für jede geplante Komponente des Systems existiert eine Ebene  $H_i$ , in der die Typen der entsprechenden Komponente als Knoten  $V_x^i$  repräsentiert sind. Der Index  $x$  stehe für einen Knoten der  $i$ -ten Ebene  $H_i$ . Das Beispiel enthält vier Ebenen, die Startebene  $H_0$  und die Ebenen der drei Komponenten ( $H_1, H_2, H_3$ ).

Für die Gesamtheit  $H$  aller Ebenen  $H_i$  gilt

$$H = \bigcup_{i=0}^n H_i, \quad (15)$$

wobei  $n$  die Anzahl der Komponenten des Zielsystems ist. Die  $i$ -te Ebene  $H_i$  besteht

aus

$$H_i = (V_1^i, V_2^i, \dots, V_k^i, \dots, V_m^i) \quad (16)$$

mit

$$\begin{aligned} i, n, m &\in \mathbb{N} \wedge n, m \geq 1 \\ \forall i \in \{1, \dots, n\} \wedge \forall x \in H_i & \\ \Rightarrow x \notin H \setminus H_i & \end{aligned} \quad (17)$$

Eine Kante E ist definiert durch

$$E = \{(x, y) \mid x \in H_i \Rightarrow y \in H_{i+1} \vee x = \alpha \Rightarrow y \in H_1 \vee x \in H_n \Rightarrow y = \omega\}. \quad (18)$$

In Gleichung (18) ist ein Endknoten  $\omega$  berücksichtigt. Dieser wird beim vorliegenden Graphenalgorithmus vernachlässigt. Er stellt nur den Schlussknoten des Systems dar und enthält keine zu berücksichtigenden Werte.

Abgesehen von der Ebene  $H_0$ , deren Knoten  $\alpha$  der Startpunkt ist, repräsentiert jede Ebene  $H_i$  die entsprechende Komponente  $i$  des Systems. Für jeden Typ dieser Komponente existiert ein Knoten  $V$  in dieser Ebene. Über gerichtete Kanten  $E$  ist (nach der Initialisierung) jeder Knoten einer Ebene  $H_i$  mit jedem Knoten der nachfolgenden Ebene  $H_{i+1}$  verbunden. Für den vorliegenden Graphenalgorithmus existiert ein gerichteter, nicht zyklischer Graph, wie er beispielsweise in Abbildung 6 dargestellt ist.

Ein Weg durch den Graphen  $G$  entspricht dem einmaligen Durchlaufen jeder seiner Ebenen. Jeder denkbare Weg (jede Permutation des Graphen), ausgehend vom Startknoten  $\alpha$  (in  $H_0$ ) hin zu einem beliebigen Knoten der letzten Ebene (hier:  $H_3$ ), entspricht einer möglichen Lösung des Problems. Um alle Wege systematisch zu erstellen, werden beim Enumerationsverfahren zunächst fortlaufend alle verwendbaren Knoten in den Hierarchieebenen nummeriert.

Alle möglichen Wege  $r$  des Problems können durch die Berücksichtigung einer in [OtW93] erläuterten Adjazenzmatrix des Graphen generiert werden. Jeder Weg lässt sich danach in Vektorform darstellen. Diese Vektoren werden in eine Liste, die nachfolgend Wege-Liste genannt wird, aufgenommen. Die Gesamtmenge  $P$  aller Permutationen ist die Vereinigungsmenge aller Wege  $r$  mit  $p$  als Anzahl aller gültigen Permutationen, d. h.

$$P = \{r_1\} \cup \{r_2\} \cup \dots \cup \{r_p\}. \quad (19)$$

Die Vektoren enthalten jeweils alle Knotennummern des definierten Weges. Anhand der Abbildung 6 kann ein Beispiel-Weg erstellt werden, indem die jeweils oberste Kante des Graphen gewählt wird. Dieser Weg lässt sich in Vektorform gemäß nachfolgender Gleichung darstellen

$$r_1 = \{\alpha, V_1, V_4, V_6\}. \quad (20)$$

Eine andere Art der Darstellung ist die Angabe der genutzten Kanten. Für den Beispiel-Weg ergibt sich folgende Darstellung

$$r_1 = \{(\alpha, V_1), (V_1, V_4), (V_4, V_6)\}. \quad (21)$$

Da vor der eigentlichen Suche nach dem Optimum bereits alle möglichen Permutationen bekannt sind, lässt sich der Aufwand zur Berechnung des Gesamtsystems  $O(S)$  abschätzen. Er ergibt sich aus dem Aufwand für die



Berechnung eines (beliebigen) Weges  $O(j)$ , multipliziert mit der Anzahl  $m$  aller möglichen Wege zu

$$O(S) = m \cdot O(j). \quad (22)$$

Jede Kante  $E(i, j)$  enthält den Vektor mit den Parametern des nachfolgenden Knotens  $j$ . In diesem ist wiederum die Verknüpfung aller bisher überstrichenen Kanten des Teilweges gespeichert. Jeder weitere Teilweg kann mit dem aktuell überstrichenen Weg verknüpft werden, so dass jeder Knoten der letzten Ebene die Lösungsvektoren mehrerer Wege enthält. Die optimale Lösung ergibt sich durch den Vergleich aller entstandenen Lösungsvektoren.

Die Lösungsvektoren sind „Kostenvektoren“ des Weges. Diese sind nicht zu verwechseln mit dem „Preis“, sondern es handelt sich um Bewertungsmaßzahl(en) zur Beurteilung der Lösungen.

Die Parameter des Beispiels aus der Abbildung 6 werden durch die Kantenvektoren  $[a1]$ ,  $[b1]$  und  $[c1]$  zu  $[a1]*[b1]*[c1]$  zusammengefasst, wobei  $*$  eine beliebige Verknüpfung ist.

Es entsteht der in dieser Arbeit entwickelte Kostenvektor  $\vec{c}$ , der jeweils die Kosten, das Bauvolumen, die Ausfallrate und das Gewicht des betrachteten Weges enthält. Die optimale Lösung ergibt sich hier durch den Vergleich der resultierenden Wirtschaftlichkeitsmaßzahlen  $W$  der verschiedenen Lösungen, wobei die jeweilige Ausfallrate  $\lambda_i$  (einer Lösung  $I$ ) die Forderung aus Gleichung (10) zu erfüllen hat. Der Kostenvektor enthält daher auch die Wirtschaftlichkeitsmaßzahl  $W$ .

Der hier verwendete Kostenvektor  $\vec{c}_I$  einer Lösung  $I$  ermittelt sich bei dem realisierten Graphenalgorithmus durch Addition der jeweiligen Parameter aller überstrichenen Kanten. Es folgt

$$\vec{c}_I = \sum_{i=1}^n \vec{e}_i = \begin{bmatrix} \lambda_1 \\ K_1 \\ G_1 \\ V_1 \\ W_1 \end{bmatrix} + \begin{bmatrix} \lambda_2 \\ K_2 \\ G_2 \\ V_2 \\ W_2 \end{bmatrix} + \dots + \begin{bmatrix} \lambda_n \\ K_n \\ G_n \\ V_n \\ W_n \end{bmatrix} = \begin{bmatrix} \lambda_I \\ K_I \\ G_I \\ V_I \\ W_I \end{bmatrix}. \quad (23)$$

Die Knoten der höchsten Ebene enthalten insgesamt so viele Kostenvektoren, wie es Wege durch den Graph  $G$  gibt. Das Beispiel aus Abbildung 6 enthält daher in der letzten Ebene 18 Kostenvektoren, je 6 pro Knoten.

Die Gesamtkosten

$$K_I = \sum_i K_i, \quad (24)$$

das Gesamtgewicht

$$G_I = \sum_i G_i, \quad (25)$$

das insgesamt benötigt Bauvolumen

$$V_I = \sum_i V_i \quad (26)$$

und die resultierende Ausfallrate

$$\lambda_I = \sum_i \lambda_i \quad (27)$$

jeder Lösung I können direkt aus  $\vec{c}_I$  abgelesen werden.

#### 4.1.2 Phase 2: Einschränkung des Lösungsraumes

Die Anzahl p aller Permutationen (möglichen Wege durch den Graphen, vgl. Gleichung (9)) entspricht der Komplexität des vollständigen Lösungsraumes O. Alle Wege enthalten dieselbe Anzahl an Komponenten, denn sie durchlaufen dieselben Ebenen. Der Berechnungsaufwand eines Weges ist somit nur von der Anzahl der Ebenen  $H_i$  abhängig und für alle Wege gleich. Die Berechnung dieser Kostenvektoren (Gleichung (23)) stellt den wesentlichen Aufwand des Graphenalgorithmus dar.

In dieser Arbeit wurden für das vorliegende Problem verschiedene Vorgehensweisen zur Reduzierung des Lösungsraumes entwickelt. Bevor die Kostenvektoren berechnet werden, können nicht relevante Lösungen durch die Berücksichtigung der Randbedingungen ausgeschlossen werden.

Folgende Ansätze wurden dabei verfolgt:

- Löschen von Knoten,
- Löschen von Kanten,
- Teilen des Graphen.

Löscht man die falschen Knoten oder Kanten, können sinnvolle Lösungen verloren gehen. Die Löschungskriterien müssen daher auf gesichertem (Experten-) Wissen basieren und sollten allgemeingültig auf das Problem anwendbar sein.

##### 4.1.2.1 Löschen von Knoten

Es sei vorausgesetzt, dass alle Typen der Datenbasis die technischen Funktionsanforderungen des Systems erfüllen. Wie in Kapitel 2 dargestellt, sind alle Typen hinsichtlich ihrer Ausfallrate, Geometrie, Gewicht, Kosten und Resistenzen definiert. Da alle Typen einer Komponente dieselbe Funktion erfüllen, müssen bei der vorliegenden Aufgabenstellung die Gründe für das Löschen eines Knotens bei diesen Parametern liegen.

Wird ein Knoten gelöscht, so entspricht das dem Ausschluss eines Typen aus der betrachteten Ebene. Für diese Arbeit wurden für das Löschen von Knoten drei Löschkriterien definiert:

Knoten-Löschkriterium 1:

- *Es kann jeder Knoten gelöscht werden, dessen Ausfallrate  $\lambda_x$  größer, als die Soll-Ausfallrate  $\lambda_{soll}$  des Gesamtsystems ist.*

Knoten-Löschkriterium 2:

- *Wird bei der Betrachtung des bestmöglichen Weges  $W_{best}$  ein beliebiger Knoten durch einen schlechteren ersetzt und ist die daraus resultierende Ausfallrate  $\lambda_{ist}$  höher als die Soll-Ausfallrate  $\lambda_{soll}$ , so darf der ersetzende Knoten gelöscht werden.*

Um das zweite Kriterium anwenden zu können, muss zunächst ein Ranking der Typen in der Datenbasis erstellt und anschließend eine mehr oder weniger aufwendige Betrachtung des gesamten Systems durchgeführt werden.

Ab wann dieses Löschkriterium und der damit verbundene Mehraufwand tatsächlich zu einer Reduzierung des Berechnungsaufwands führt, ist abhängig von der Anzahl der tatsächlich gelöschten Knoten und somit von den Maßzahlen der Kanten und den Vorgaben des Systems.

Knoten-Löschkriterium 3:

- *Erfüllt ein Knoten die Anforderung des Einbauortes [Vol00] nicht, so darf er aus dem Graphen gelöscht werden.*

Es ist denkbar, dass nach dem Löschen von Knoten kein unterbrechungsfreier Weg mehr vom Start- zum Zielknoten gefunden werden kann. An mindestens einem Einbauort ist dann kein verwendbarer Typ mehr verfügbar. Es müsste dann entweder ein anderer Einbauort gefunden (Topologieänderung) oder ein neuer Typ entwickelt werden (Technologieänderung).

#### 4.1.2.2 Löschen von Kanten

Bei der Betrachtung der Kanten werden für diese Arbeit „reale“ und „virtuelle“ Kanten definiert und zwischen diesen unterschieden:

*Reale* Kante:

- Liegen die betrachteten Knoten in aufeinanderfolgenden Ebenen, so ist die Verbindung zwischen diesen eine *reale* Kante. Die Abbildung 7 zeigt eine reale Kante zwischen den Knoten 1 und 4 (durchgezogener Pfeil). Unter der Annahme, dass diese Typen nicht miteinander kombiniert werden dürfen, kann man diese Kante als „verbotene Kante“ betrachten. Löscht man sie, werden keine relevanten Lösungen verworfen. Alle gelöschten Lösungen enthalten garantiert die verbotene Kombination der Knoten 1 und 4. Das Löschen realer Kanten wird durch die Veränderung des entsprechenden Eintrags in der Adjazenzmatrix realisiert.

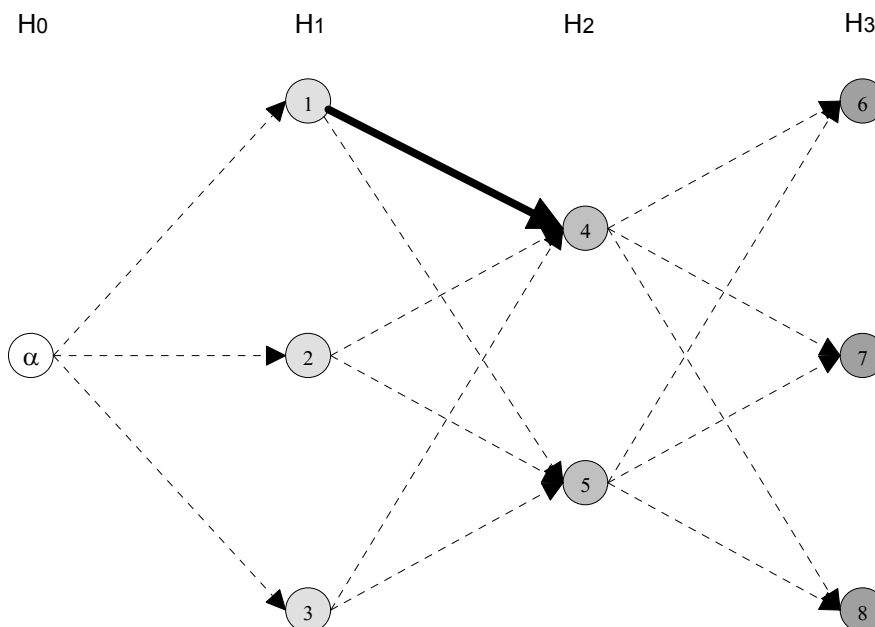
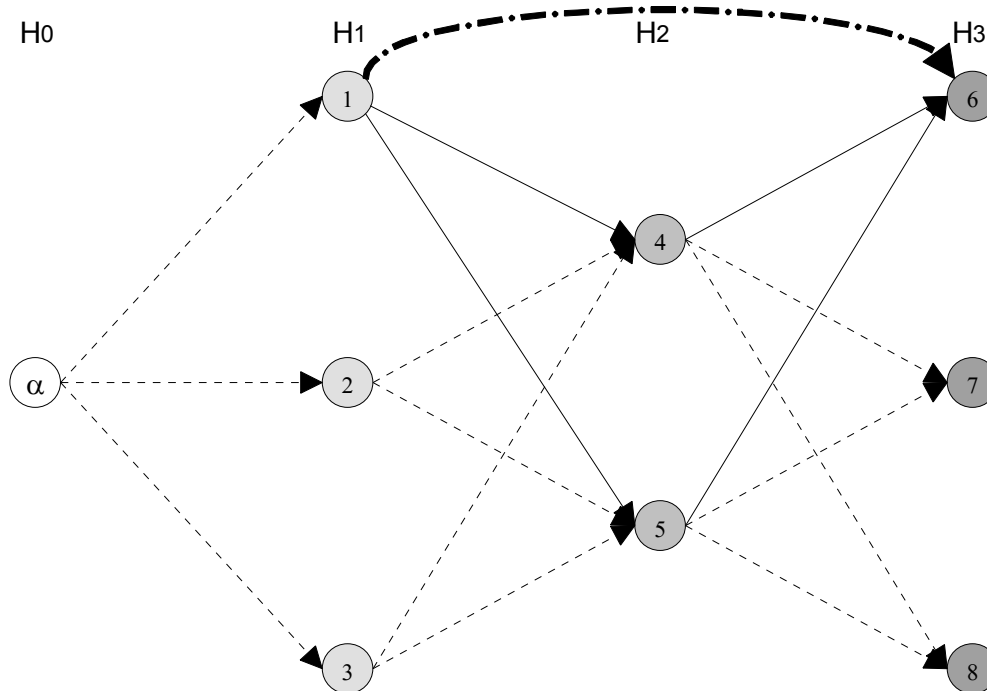


Abbildung 7: *Reale* Kante zwischen den Knoten 1 und 4

*Virtuelle* Kante:

- Folgen die Ebenen der betrachteten, nicht kombinierbaren Knoten nicht aufeinander, so existiert zwischen diesen Knoten eine *virtuelle* Kante. Die Abbildung 8 zeigt eine virtuelle Kante zwischen den Knoten 1 und 6 (virtueller Pfeil mit Strich-Punkt-Linie). Sie besteht aus insgesamt 4 realen Kanten (Pfeile mit durchgezogener Linie). Wollte man diese virtuelle Kante löschen, müsste jede reale Kante, die eine Verbindung zwischen den betrachteten Knoten ermöglicht, gelöscht werden. Es müssten also vier reale Kanten gelöscht werden. Bei dieser Vorgehensweise würden aber voraussichtlich neben der Variante mit der verbotenen Kombination auch lösungsrelevante Varianten aus dem Lösungsraum ausgeschlossen werden.

Das Löschen virtueller Kanten entspricht daher dem Löschen der entsprechenden Wege aus der Wegeliste (vgl. Kapitel 4.1.1).



**Abbildung 8: Virtuelle Kante zwischen den Knoten 1 und 6**

Im Rahmen dieser Arbeit wurden für das Löschen von (realen oder virtuellen) Kanten drei Löschkriterien definiert:

Kanten-Löschkriterium 1:

- Die (reale) Kante zwischen zwei aufeinanderfolgenden, nicht kombinierbaren Knoten kann aus dem Graph gelöscht werden.

Kanten-Löschkriterium 2:

- Besteht zwischen zwei nicht kombinierbaren Knoten eine virtuelle Kante (Verbindung), so können alle Wege mit dieser Kombination gelöscht werden.

Ein Sonderfall der beiden genannten Löschkriterien liegt vor, wenn ein Knoten immer mit einem bestimmten weiteren Knoten zu kombinieren ist.

Kanten-Löschkriterium 3:

- Muss ein Knoten mit einem anderen Knoten kombiniert werden, so dürfen alle Wege gelöscht werden, die ausschließlich den zuweisenden (abgehenden) Knoten enthalten. Das bedeutet, dass solche Wege, die ausschließlich den zugewiesenen (ankommenden) Knoten, alle beide Knoten oder keinen der beiden Knoten enthalten, nicht gelöscht werden dürfen.

Die Effektivität der Variantenreduzierung ist abhängig von der Anzahl der dem zuweisenden Knoten nachfolgenden Knoten.

#### 4.1.2.3 Teilen des Graphens

Lässt man eine Abweichung vom Enumerationsverfahren zu, so ist für das vorliegende Problem die Aufteilung des Lösungsraumes in Teilprobleme möglich. Diese Vorgehensweise entspricht dem „Divide & Conquer“ Lösungsansatz [Vor98], [Akl89]. Die Ausfallraten der Komponenten in einem System unterscheiden sich zum

Teil um mehrere Größenordnungen. Die Komponenten mit großen Ausfallraten haben den maßgeblichen Anteil an der sich ergebenden Ausfallrate einer Lösung. Berücksichtigt man dies, so lässt sich eine Reduzierung des Aufwands zur Berechnung der optimalen Lösung wie folgt realisieren:

Man teilt die Komponenten des betrachteten Systems in (zwei) Oberklassen ein. Die erste enthält die Komponenten mit den höheren Ausfallraten, die zweite die restlichen. Nun optimiert man die Klasse der Komponenten mit den höheren Ausfallraten. Von der Klasse mit den geringeren verwendet man anschließend die jeweils wirtschaftlichsten Typen jeder Komponente.

Die Reduzierung des Lösungsraumes wird an dem nachfolgenden Beispiel erläutert:

Ein System bestehe aus 15 Komponenten mit jeweils 3 Typen. Es müssen somit theoretisch

$$O = 3^{15} = 14348907 \quad (28)$$

unterschiedliche Lösungen betrachtet werden.

Zur Verkleinerung des Variantenreichtums, teilt man das System nun in die beiden Klassen der hohen und niedrigen Ausfallraten ein. Als Folge müssen nun zwei Systeme betrachtet werden, eins mit 5 und ein weiteres mit 10 Komponenten. Man vergleicht dann

$$O = 3^5 + 3^{10} = 243 + 59049 = 59292 \quad (29)$$

Varianten, also deutlich weniger als zuvor.

Die Idee der Optimierung besteht darin, unwirtschaftliche Komponententypen durch wirtschaftlichere mit voraussichtlich höherer Ausfallrate zu ersetzen. Die Ausfallrate des Gesamtsystems verschlechtert sich. Dies gilt es durch die Verwendung besserer Typen von anderen Komponenten zu kompensieren. Diese Kompensation ist möglich, wenn die substituierenden Typen Ausfallraten in ähnlicher Größenordnung wie die substituierten haben. Die Lösung wird verbessert, wenn die neue Lösung insgesamt wirtschaftlicher ist und (maximal) die gleiche Ausfallrate wie die Vergleichslösung hat. Somit ist die Vorgehensweise der Klassenbildung zulässig, wenn sich die Ausfallraten der Oberklassen um Größenordnungen voneinander unterscheiden. Ist dies nicht der Fall, so könnte eine suboptimale Lösung gefunden werden.

Die Graphenteilung sollte jedoch nur dann angewendet werden, wenn der Lösungsraum aus rechentechnischen Gründen zu komplex ist.

#### **4.1.3 Phase 3: Berechnung aller verbleibenden Wege**

Nachdem der Graph nun um nicht relevante Knoten und Kanten reduziert ist, bzw. der zuvor geteilte Graph wieder zusammengeführt wurde, erfolgt in der dritten Phase die Berechnung des Kostenvektors  $\vec{c}_x$  für alle verbleibenden Varianten.

Alle in dieser Phase existierenden Wege sind zwar hinsichtlich der jeweiligen Randbedingungen gültig, müssen aber keinesfalls die aus der Soll-Ausfallrate resultierende Anforderung erfüllen.

#### **4.1.4 Phase 4: Ermittlung der optimalen Lösung durch Vergleich**

In der vierten Phase werden nun die Kosten der verbleibenden Varianten miteinander verglichen. Es sei  $Q$  die Untermenge von  $P$ , deren Ausfallrate maximal

gleich der Vorgabe (siehe Gleichung (10)) ist. Die optimale Lösung wird in ihr gesucht.

Die Anzahl aller in  $Q$  enthaltenen Permutationen ist  $q$ . Die optimale Lösung  $r_{opt}$  ist diejenige aus  $Q$ , deren resultierende Wirtschaftlichkeitsmaßzahl  $W_i$  am geringsten ist. Dieser Zusammenhang ist in Gleichung (30) dargestellt. Es gilt

$$r_{opt} = \{r_i | W_i = \min(W_i) \forall 1 \leq i \leq q\}. \quad (30)$$

#### 4.1.5 Bewertung der Methode der Graphenalgorithmen

Beim Graphenalgorithmus wird zunächst der Lösungsraum durch den Vergleich der Randbedingungen des Einbauortes mit der Spezifikation des potenziellen Typs reduziert. Die beim Löschen der Knoten und Kanten angewandte Reduktionsmethode ist schlüssig. Es werden nur Wege (Varianten) gelöscht, die garantiert nicht Teil der optimalen Lösung sind. Diese Vorgehensweise kann teilweise auch bei den beiden nachfolgend betrachteten Methode angewendet werden. Wenn alle potenziellen Typen die Anforderungen ihrer Einbauorte erfüllen wird der Lösungsraum nicht reduziert.

*Ein Vorteil der Methode des Graphenalgorithmus ist, dass der betrachtete Lösungsraum garantiert die ideale Lösung enthält.*

Bei komplexen Systemen steigt die Größe des Lösungsraumes und somit der Berechnungsaufwand exponentiell. Bei komplexen Systemen mit einer großen Datenbasis ist es denkbar, dass die optimale Lösung nicht mit vertretbarem Aufwand gefunden werden kann. Die Methode wäre nicht anwendbar.

*Ein Nachteil der Methode des Graphenalgorithmus ist daher, dass Sie für komplexe Systeme mit einer großen Datenbasis nicht anwendbar ist.*

Die Komplexität und somit die Lösbarkeit der Optimierungsaufgabe ist vorweg jedoch mit geringem Aufwand berechenbar (siehe Gleichung (9)).

#### 4.2 Methode 2: Neuro-Fuzzy-Methode

Die Neuro-Fuzzy-Methode ist die Kombination von Lernfähigkeit (Neuronale Netze) mit der Fähigkeit zur Bewertung unscharfer Parameter (Fuzzy-Logik) [NaK99], [Dör01], [YaZ87].

Verschiedene wissensbasierte Vorgänge können durch die Lernfähigkeit automatisiert und die Einstellungen verschiedener Parameter dem Anwender abgenommen werden.

Neuronale Netze sind vor ihrer Anwendung zu trainieren. Daher dürfen die notwendigen Entwickler-Vorgaben von allgemeiner Art sein. Sie können sich innerhalb gewisser Grenzen sogar widersprechen.

Fuzzy-Logik wird zur Klassifizierung von Mustern genutzt. Kombiniert man Neuronale Netze mit der Fuzzy-Logik, erhält man eine Möglichkeit zur lernfähigen Musterklassifizierung [NaK95].

Neuro-Fuzzy-Systeme werden oftmals für selbstadaptierende Regelvorgänge eingesetzt. Die theoretischen Grundlagen dieses Kapitels beziehen sich größtenteils auf Veröffentlichungen von Prof. Kruse et. al. von der TU-Magdeburg (z. B. [NaK95], [NaK99], [Lnk01]).

In der vorliegenden Arbeit wurde versucht, deren bekannte Mechanismen der Neuro-

Fuzzy-Logik für Optimierung diskreter Systeme einzusetzen.

#### 4.2.1 Neuronales Lernen

Vorbild für Neuronales Lernen ist das Gehirn. Eine Gehirnzelle (Neuron), wie sie in Abbildung 9 dargestellt ist, besteht aus einem Zellkörper mit mehreren Eingängen (Dendriten) und einem Ausgang (Axon). Die Eingangsreize gelangen über die Synapse und die Dendriten zum Zellkörper. Dort werden sie zusammengefasst. Die Summe der Eingangsreize entscheidet über den Zustand des Ausgangs, der nur aktiv oder passiv sein kann [Has98].

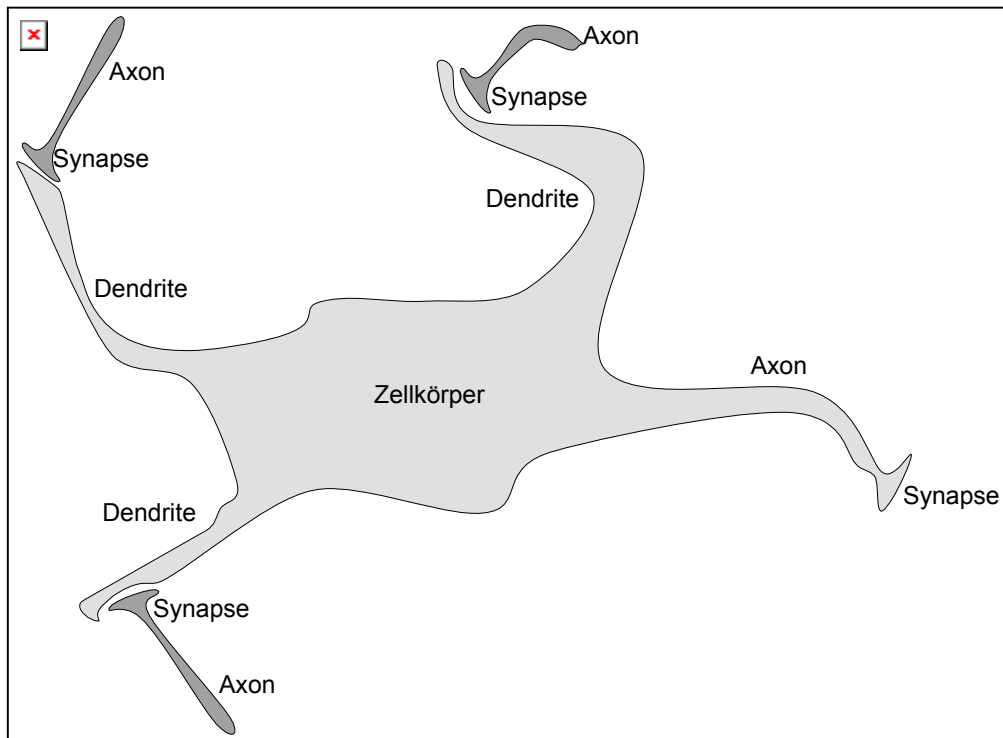


Abbildung 9: Biologisches Neuron

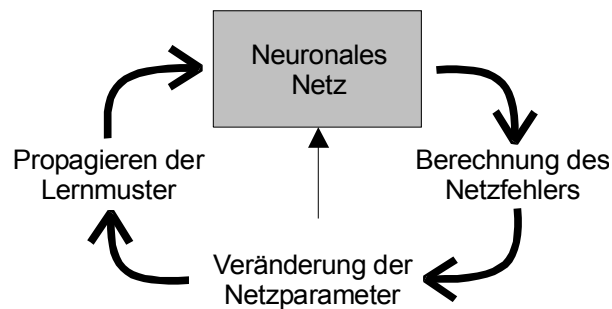
Das Axon ist an (beliebig viele) Dendriten nachfolgender Neuronen(schichten) gekoppelt. Durch diese Verknüpfungen entsteht ein Netzwerk von Neuronen. Erweitert man dieses um eine Eingabe- und eine Ausgabeschicht, so entsteht ein sogenanntes Neuronales Netz.

An der Synapse, dem Übergang eines Axons zu einer Dendrite, werden die Eingangsreize mit einem elektrochemischen Vorgang über Neurotransmitter gewichtet. Die Stärke der Gewichtung ist variabel und stellt den Wissenspeicher Neuronaler Netze dar. Nach [Heu97], [Roj93] unterscheidet man zwei Lernverfahren, „Verstärkungslernen“ und „korrigierendes Lernen“. Die realisierte Neuro-Fuzzy-Methode wird durch die Anwendung des korrigierenden Lernens trainiert.

Hierzu werden in einer Trainingsepoche verschiedene Lernmuster nacheinander auf das Netz gegeben. Weichen die Reaktionen des Netzes von den vorgegebenen Soll-Reaktionen ab, so werden die Gewichtungen verändert und eine weitere Trainingsepoche durchgeführt.

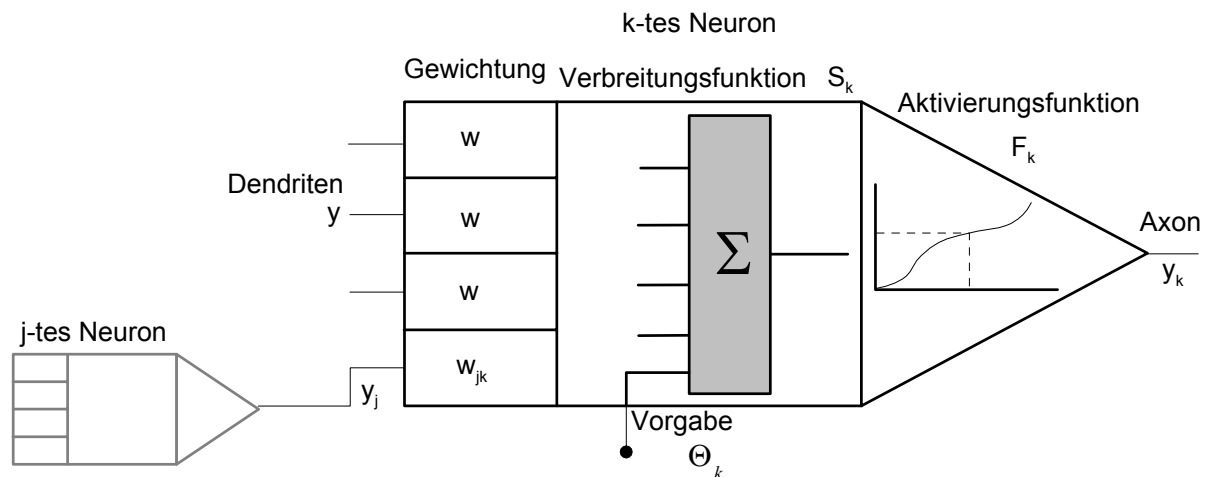


Nach beliebig vielen dieser Trainingsepochen befindet sich der Netzfehler innerhalb einer akzeptablen Abweichung, bzw. verbessert sich nicht weiter. Der Lernvorgang wird abgebrochen und das Neuronale Netz ist trainiert. Der Lernprozess Neuronaler Netze ist in Abbildung 10 dargestellt.



**Abbildung 10: Lernprozess eines Neuronales Netzes**

Der Aufbau künstlicher Neuronaler Netze ist der Verknüpfung biologischer Neuronen des Gehirn nachempfunden [KrS96]. Hierzu werden Neuronen modelliert, deren Funktion ihrem biologischen Vorbild nachempfunden wurde (Abbildung 11).

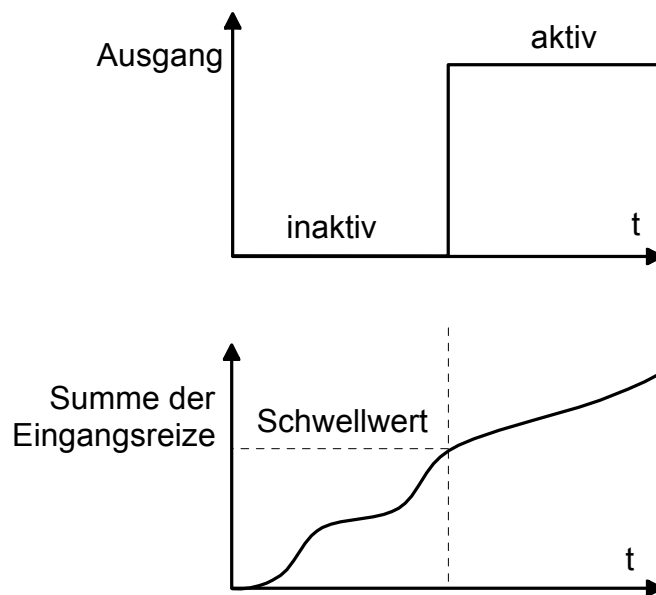


**Abbildung 11: Elektrisches Neuron**

Jedes Neuron innerhalb der Vernetzung hat beliebig viele Eingänge (Dendriten) und erhält über diese die digitalen Eingangsinformationen (1/0). Alle Eingangssignale werden gewichtet und in einer Verbreitungsfunktion zusammengefasst. Im Unterschied zum biologischen Vorbild ist es beim künstlichen Neuronales Netz möglich, die Verbreitungsfunktion mit einem Vorgabewert (Offset) zu justieren.

Nach der anschließenden Bewertung der zusammengefassten Eingangswerte anhand einer Aktivierungsfunktion existieren am Ausgang (Axon) nur zwei Zustände, aktiv oder inaktiv.

Die Abbildung 12 zeigt, dass mit dem Erreichen eines beliebigen Schwellwertes der Ausgang des Neurons aktiv wird. Die Summe der Eingangsreize bildet einen kontinuierlichen Wertebereich, da die Gewichtungen ebenfalls kontinuierliche Werte annehmen.



**Abbildung 12: Zustand des Ausgangs eines Neurons in Abhängigkeit des Schwellwertes der Aktivierungsfunktion**

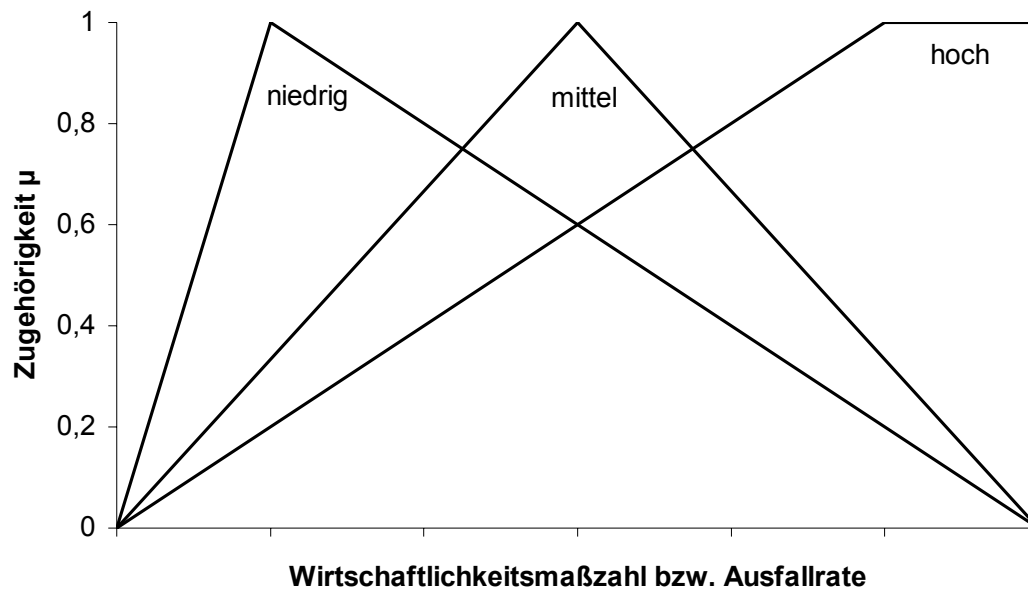
Das Signal am Axon eines Neurons ist wiederum eine Eingangsinformation einer weiteren Neuronenschicht oder, sofern es sich um die letzte Neuronenschicht handelt, eine Ausgangsinformation des gesamten Netzes.

Beim biologischen Neuronalen Netz werden alle Eingangsinformationen in einem parallel ablaufenden Prozess über das Netz propagiert. Im Gegensatz dazu werden beim künstlichen Neuronalen Netz die Daten seriell propagiert. Eher selten werden auch Parallelrechner eingesetzt.

#### 4.2.2 Fuzzy-Logik

Fuzzy-Logik ermöglicht die Auswertung von unscharfen Mengen. Im Anwendungsfall sind die Parameter (Ausfallrate, Wirtschaftlichkeitsmaßzahl) jedes Typs von jeder Komponente durch scharfe Werte definiert. Die Fuzzy-Logik überführt diese in unscharfe linguistische Terme. Diesen Vorgang nennt man Fuzzyfizierung.

Sowohl für die Wirtschaftlichkeitsmaßzahl (W-Set) als auch für die Ausfallrate ( $\lambda$ -Set) werden dazu jeweils die Fuzzy-Mengen niedrig, mittel und hoch erstellt (Abbildung 13).



**Abbildung 13: Formen des Fuzzy-Sets für die Wirtschaftlichkeitsmaßzahl bzw. die Ausfallrate**

Die Mengen für „niedrig“ und „mittel“ haben eine Dreiecksform, „hoch“ ist rechtsseitig „geschultert“. Die Gesamtheit dieser drei Fuzzy-Mengen wird unter dem Begriff Fuzzy-Set zusammengefasst.

Anhand der Fuzzy-Sets lassen sich alle Typen in linguistische Terme überführen. In der vorliegenden Aufgabe ist jeder Typ durch eine Ausfallrate und eine Wirtschaftlichkeitsmaßzahl definiert. Diese lassen sich an der Abszisse auftragen. An der Ordinate ist über die Fuzzy-Menge die Zugehörigkeit  $\mu$  des Abszissenwertes zu der entsprechenden Fuzzy-Menge ablesbar. Die höchste Zugehörigkeit ( $\mu=1$ ) der Fuzzy-Menge „niedrig“ wird bei der Initialisierung des Fuzzy-Sets dem Wert des kleinsten scharfen Parameters aller Typen einer Komponente entsprechen. Für die Fuzzy-Menge „hoch“ wird eine rechtsseitig geschulterte Funktion verwendet. Das Intervall  $\mu=1$  beginnt bei dem größten scharfen Parameter und ist nach rechts unendlich. Voraussichtlich sind alle Typen, die den höchsten Zugehörigkeitswert jeweils zu den Fuzzy-Mengen  $\lambda$ =hoch oder  $W$ =hoch haben „ungeeignet“. Daher werden auch nach einer späteren Verschiebung dieses Fuzzy-Sets alle Typen rechts des Intervalls verminderter Zugehörigkeit ( $\mu < 1$ ) der Fuzzy-Menge „hoch“ mit ungeeignet klassifiziert. Würde auch „hoch“ eine Dreiecksfunktion haben, so würden Typen bei unpassender Verschiebung der Sets nicht klassifiziert werden können. Ihre Zugehörigkeit zu jeder Fuzzy-Menge wäre  $\mu=0$ .

Die Abbildung 14 zeigt das initialisierte W-Set einer beispielhaften Komponente, die 7 Typen (a...g) enthält. Auf gleiche Weise werden diese Typen im  $\lambda$ -Set hinsichtlich ihrer Ausfallrate fuzzyfiziert.

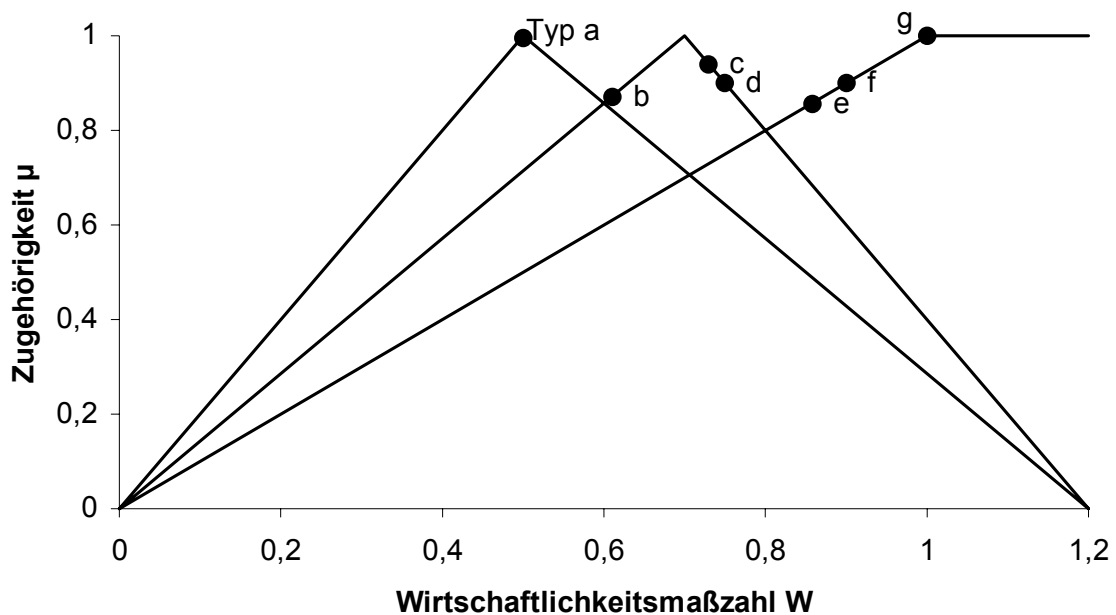


Abbildung 14: Fuzzy-Set für die Wirtschaftlichkeitsmaßzahl  $W$  einer Komponente mit 7 Typen

Nach der Fuzzyfizierung sind alle Typen aller Komponenten mit linguistischen Termen definiert (z. B. „Ausfallrate niedrig“, „Wirtschaftlichkeitsmaßzahl mittel“). Anhand dieser Terme werden die Typen später klassifiziert.

#### 4.2.3 Fuzzy-Regeln

Bei der Neuro-Fuzzy-Methode erfolgt die Klassifizierung von fuzzyfizierten Typen durch Fuzzy-Regeln [NaK99]. Diese können bei der realisierten Methode erlernt werden und haben folgende Form:

**WENN** Prämisse1 **UND** Prämisse2 **DANN** Konklusion.

Es entstehen für die vorliegende Problemstellung Regeln, wie sie in Tabelle 1 beispielhaft dargestellt sind.

Prämisse 1			Prämisse 2			Konklusion	
<b>WENN</b>	W =	<i>niedrig</i>	<b>UND</b>	$\lambda =$	<i>niedrig</i>	<b>DANN</b>	<i>geeignet</i>
<b>WENN</b>	W =	<i>niedrig</i>	<b>UND</b>	$\lambda =$	<i>mittel</i>	<b>DANN</b>	<i>geeignet</i>
<b>WENN</b>	W =	<i>niedrig</i>	<b>UND</b>	$\lambda =$	<i>hoch</i>	<b>DANN</b>	<i>ungeeignet</i>
<b>WENN</b>	W =	<i>mittel</i>	<b>UND</b>	$\lambda =$	<i>niedrig</i>	<b>DANN</b>	<i>geeignet</i>
<b>WENN</b>	W =	<i>mittel</i>	<b>UND</b>	$\lambda =$	<i>mittel</i>	<b>DANN</b>	<i>ungeeignet</i>
<b>WENN</b>	W =	<i>mittel</i>	<b>UND</b>	$\lambda =$	<i>hoch</i>	<b>DANN</b>	<i>ungeeignet</i>
<b>WENN</b>	W =	<i>hoch</i>	<b>UND</b>	$\lambda =$	<i>niedrig</i>	<b>DANN</b>	<i>ungeeignet</i>
<b>WENN</b>	W =	<i>hoch</i>	<b>UND</b>	$\lambda =$	<i>mittel</i>	<b>DANN</b>	<i>ungeeignet</i>
<b>WENN</b>	W =	<i>hoch</i>	<b>UND</b>	$\lambda =$	<i>hoch</i>	<b>DANN</b>	<i>ungeeignet</i>

**Tabelle 1: Regeln mit je zwei Prämissen führen zu einer Konklusion**

Anhand dieser Regeln werden anschließend alle Typen der Datenbasis klassifiziert. Nach der Fuzzyfizierung sind alle Typen durch unscharfe linguistische Terme beschrieben. Diese Terme werden als Prämissen auf alle erlernten Regeln angewendet und es erfolgt die *Klassifizierung der Typen*. So wird für jeden Typ ermittelt, ob er geeignet ist und wie stark bei der aktuellen Dimensionierung der Fuzzy-Sets der Grad der Eignung des Typen im Vergleich zu den anderen Typen ist. Alle Typen grenzen sich dadurch hinsichtlich ihrer Eignungsgüte untereinander ab. Anschließend werden nur noch die am besten geeigneten Typen berücksichtigt. Der Lösungsraum verkleinert sich.

#### 4.2.4 Neuro-Fuzzy-Optimierungsprozess

Um die wirtschaftlichste Kombination der Typen zu finden, wurde in dieser Arbeit der in Abbildung 15 dargestellte Optimierungsprozess entwickelt. Er basiert auf dem Prozess der Neuro-Fuzzy-Regelung. Ein vollständiger Durchlauf der vier dargestellten Phasen nach der Initialisierung entspricht einer Optimierungsepoche. Die vier Phasen sind:

- Fuzzyfizierung,
- Auswahl,
- Defuzzyfizierung mit anschließendem Abbruchtest und
- Anpassung aller Fuzzy-Sets.

Die Abbildung 15 zeigt den Ablauf dieser vier Phasen.

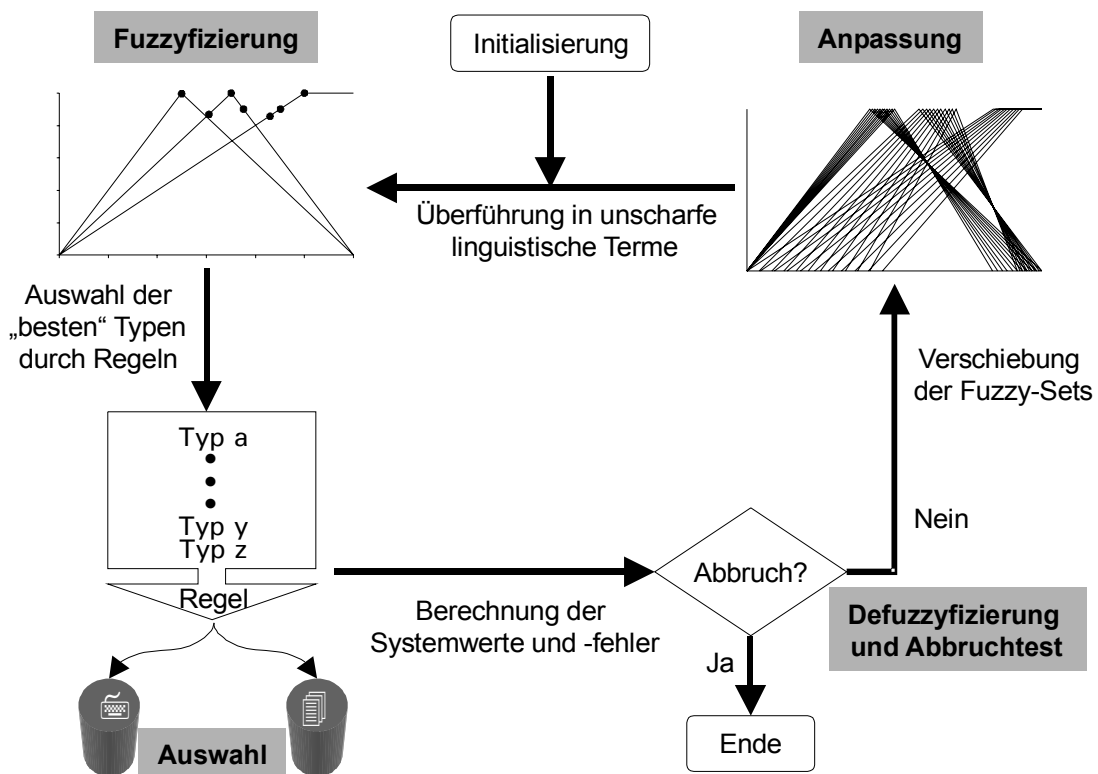


Abbildung 15: Neuro-Fuzzy Optimierungsprozess

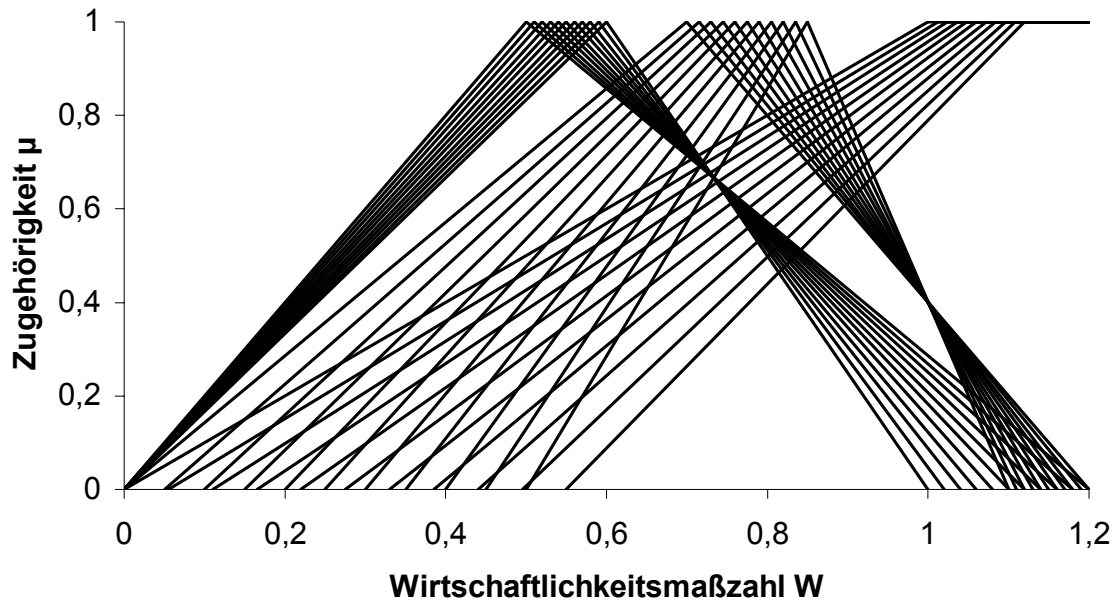
Nach der *Initialisierung* des Systems werden alle Typen *fuzzyfiziert*, d.h. ihre scharfen Werte werden in unsharp linguistische Terme überführt (vgl. 4.2.2). Die Abbildung 14 zeigt beispielhaft den Initialisierungszustand einer Komponente mit 7 Typen.

Anschließend werden die Typen durch die Regeln klassifiziert und die „Besten“ *ausgewählt*. Nun existiert eine Zwischenlösung des Systems, d. h. jeder Komponente des Systems wurde ein passender Typ zugeordnet. Das System könnte so realisiert werden; diese Zwischenlösung muss jedoch die Randbedingungen nicht erfüllen.

Für die Zwischenlösung lassen sich anhand der scharfen Werte der Typen die Ausfallrate und Wirtschaftlichkeitsmaßzahl berechnen. Falls es sich nicht um die erste Optimierungsepoche (nach der Initialisierung) handelt, werden die aktuellen Werte der Zwischenlösung mit zuvor erstellten Lösungen, auf jeden Fall aber mit den Zielvorgaben verglichen. Die beste Lösung wird gespeichert und die Erfüllung der *Abbruchkriterien* geprüft. Diese können beispielsweise berücksichtigen ob sich (in mehreren Optimierungsschleifen) noch Veränderungen ergeben oder bereits alle Typen mit  $\lambda$ =niedrig und  $W$ =niedrig fuzzyfiziert sind.

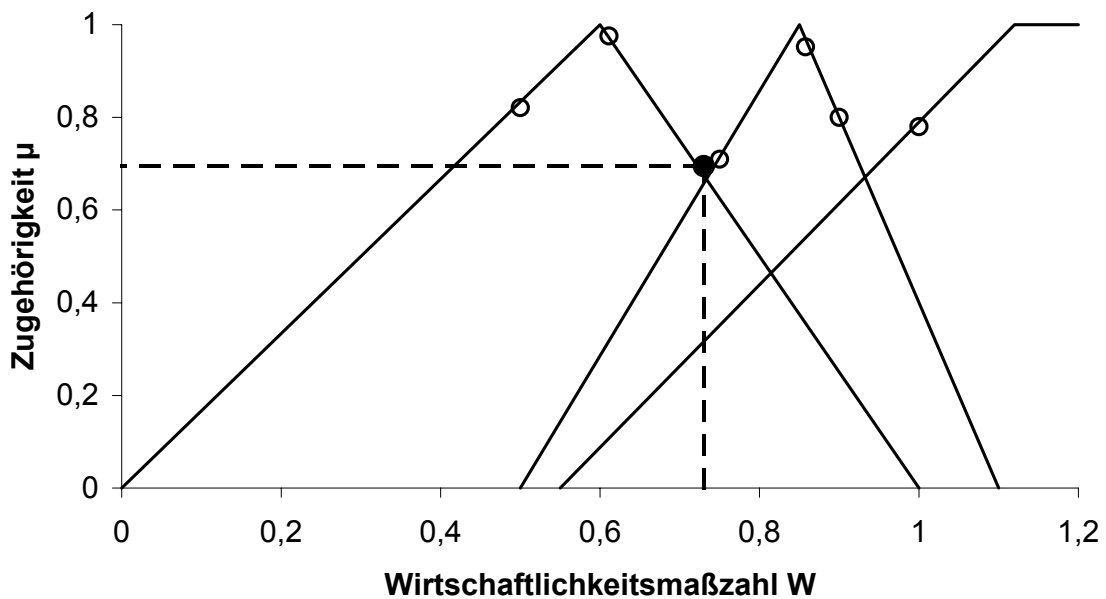
Bei Nichterfüllung der Abbruchkriterien werden die Fuzzy-Sets aller Komponenten durch *Verschiebung angepasst* und die Auswahl- und Bewertungsprozedur nochmals durchlaufen. Die Art der Verschiebung hängt von der Höhe und dem Vorzeichen der aktuellen Abweichung von den Soll-Werten ab. Die Dimensionierung der Fuzzy-Sets in den verschiedenen Optimierungsepochen stellt, ähnlich den Gewichtungen bei Neuronalen Netzen, den Wissensspeicher der Methode dar, also den Lerneffekt.

Die überlagerten Fuzzy-Sets mehrerer Optimierungsepochen sind in der Abbildung 16 dargestellt.



**Abbildung 16: Verschiebung der Fuzzy-Sets einer Komponente in mehreren Optimierungsepochen**

Die Abbildung 17 zeigt das nach mehreren Optimierungsepochen angepasste Fuzzy-Set der Komponente aus Abbildung 14. Ein Vergleich ergibt, dass die Typen nun andere Zugehörigkeiten zu den Fuzzy-Mengen haben oder sogar einer anderen Menge zugeordnet sind.



**Abbildung 17: Klassifizierung der Typen nach der (mehrfachen) Anpassung des Fuzzy-Sets**

Beispielhaft sei die Veränderung des in Abbildung 14 mit c bezeichneten Typs betrachtet. Seine Wirtschaftlichkeitsmaßzahl beträgt  $W=0,73$ . Ursprünglich wurde er

nach der Initialisierung dieser Komponente in der Abbildung 14 dem W-Set „W mittel“ mit der Zugehörigkeit  $\mu = 0,94$  zugeordnet. Nach mehreren Optimierungsepochen, in denen die Fuzzy-Mengen wie in Abbildung 16 angepasst wurden, gehört er in der Abbildung 17 nun der Fuzzy-Menge „W niedrig“ mit  $\mu = 0,695$  an. Diese veränderte Zuordnung könnte nach der erneuten Klassifizierung durch die Fuzzy-Regeln eine Änderung der Konklusion (z. B. ungeeignet  $\rightarrow$  geeignet) ergeben. Dieser Typ könnte nun Teil einer Zwischenlösung sein.

Die Fuzzy-Mengen müssen in einer geeigneten Auflösung, also in kleinen Schritten angepasst werden. Wäre die Auflösung zu grob, werden evtl. relevante Lösungen übergangen. Bei einer zu feinen Auflösung, werden hingegen zu viele Schritte benötigt. Die Optimierung würde entweder zu lange dauern oder zu früh abgebrochen werden.

Der Abbruch der Optimierung erfolgt, wenn mindestens eines der zu definierenden Abbruchkriterien erfüllt wird. Die optimale Lösung sollte dann gefunden sein.

#### **4.2.5 Bewertung der Neuro-Fuzzy-Methode**

Bei der Neuro-Fuzzy-Methode ist die Heuristik zur Verkleinerung des Lösungsraums eine regelbasierte Entscheidungslogik. Vorausgesetzt es existieren Regeln mit der Konklusion „ungeeignet“ und es gibt mehr als einen unterschiedlichen Typ pro Komponente, so wird der Lösungsraum zuverlässig verkleinert. Ebenso kann evtl. die Optimierung durch Berücksichtigung des Optimierungsverlaufs vorzeitig abgebrochen werden. Ein (großer) Teil der Datenbasis wird gar nicht erst berücksichtigt. Je größer die Datenbasis, desto größer kann auch die Effizienz dieser Reduzierung sein. Die Auflösung der Optimierung beeinflusst sowohl die Qualität als auch die Dauer der Optimierung. Ist sie zu hoch, könnte sich der Berechnungsaufwand stark vergrößern. Ist sie zu niedrig, könnten relevante Lösungen übergangen werden.

Auch ein Anstieg der Anzahl der Komponenten und deren Typen vergrößert den Berechnungsaufwand.

*Ein Vorteil der Neuro-Fuzzy-Methode ist, dass sich auch für sehr komplexe Systeme bei dem NP-vollständigen Problem Lösungen mit realisierbarem Aufwand finden lassen.*

Es wäre wünschenswert, während der Optimierung nur die Fuzzy-Sets derjenigen Komponenten anzupassen, die sich am stärksten auf den Systemfehler auswirken. Die Komponenten mit dem größten Einfluss auf den Systemfehler könnten durch einen Vergleich mit der optimalen Lösung identifiziert werden. Diese ist jedoch während der Optimierung nicht bekannt. Daher müssen aus dem resultierenden Fehler immer alle Fuzzy-Sets aller Komponenten angepasst werden (globale Anpassung). Durch diese deterministische Verschiebung ist es möglich, dass die optimalen Typen bei der Klassifizierung immer übersprungen werden. Es werden somit oft suboptimale Typen ausgewählt. Daher ist diese Methode nicht anwendbar.

*Ein Nachteil der Neuro-Fuzzy-Methode ist, dass sie oft nur suboptimale Lösungen findet.*

Der Berechnungsaufwand dieser Methode ist aufgrund der Definition der Abbruchkriterien nicht vorhersagbar. Sind diese falsch bemessen, wird diese Methode Lösungen eines lokalen Minimum finden oder den Berechnungsaufwand unnötig vergrößern.



Ein weiterer Nachteil der Neuro-Fuzzy-Methode ist, dass die zu definierenden Abbruchkriterien einen großen Einfluss auf die Qualität und Geschwindigkeit der Optimierung haben.

### 4.3 Methode 3: Genetische Algorithmen

Die Methode der Genetischen Algorithmen bildet die Prinzipien der biologischen Evolution [Dar00] nach [Lüb99], [Mue95]. Daher werden auch die Vokabeln der Evolutionstheorie verwendet.

Ein *Individuum* ist demnach eine konkrete Lösung des Problems und besteht aus verschiedenen *Genen* (= Typen der Komponente). Die Gesamtheit aller zu einem bestimmten Zeitpunkt der Evolution, d.h. in einer *Generation* existierenden Individuen, ist eine *Population*. Die Grundlage der Optimierung bildet eine (Ausgangs-) Population zufällig erstellter Individuen.

Alle jeweils in einer Generation existierenden Individuen bilden die Grundlage der nachfolgenden Optimierung. Individuen sind durch einen Evolutionsschritt nur in begrenztem Maße veränderbar. Es macht wenig Sinn, die Evolution nur auf die besten Individuen anzuwenden. Die Optimierung könnte hierdurch ein Individuum als Optimum darstellen, das durch eine bestimmte, eingeschränkte Sichtweise als Lösung erkannt wurde, insgesamt jedoch suboptimal ist. Das globale Optimum wurde dann nicht gefunden.

Um dies zu verhindern, sollte die Individuen-Struktur in einer Generation möglichst unterschiedlich sein. Durch die Evolution können dadurch wiederum völlig neue Strukturen entstehen.

Diese große Variabilität stellt die wesentliche Leistungsfähigkeit Genetischer Algorithmen dar. Lokale Minima können so verlassen werden. Als solche bezeichnet man in diesem Falle Individuen, in deren Umfeld sich zunächst keine besseren Individuen befinden, die jedoch nicht das Optimum darstellen. Dieses kann erst gefunden werden, wenn eine Optimierung mit einem deutlich unterschiedlichen genetischen Ansatz durchgeführt wird.

Genetische Algorithmen sind heuristisch und zufallsbasiert. Um eine neue Generation zu erstellen, wird ein strukturiertes zufallsbasiertes Verfahren angewendet. Eine neue Generation entsteht durch die Anwendung der Evolutions-Prinzipien Selektion, Mutation, Kreuzung und Geburt.

Die Individuen der neuen Generation entstehen bei der Selektion durch unveränderte Übernahme von Individuen der Vorgängergeneration, bei der Mutation und der Kreuzung durch teilweise Änderung der Gene von Individuen der Vorgängergeneration und bei der Geburt durch Erstellung völlig neuer Individuen. Nach der Erstellung jeder neuen Generation wird die alte vollständig gelöscht. Das Ende der Evolution (mehrfache Anwendung der genannten Prinzipien) wird durch geeignete Abbruchkriterien definiert.

Nach dieser genetischen Evolution wird in dieser Arbeit abschließend eine gezielte Genmanipulation angewendet. Sie stellt eine allgemein bekannte Erweiterung der Evolutions-Prinzipien dar. Hierbei werden diejenigen Gene der besten Individuen gezielt ersetzt, für die in der Datenbasis offensichtlich bessere Gene existieren. Diese könnten bisher in allen Evolutionsschritten „übersehen“ worden sein.

### 4.3.1 Prinzip 1: Selektion

Mit Selektion ist die Auswahl und Übernahme von Individuen in die nächste Generation gemeint. Die Auswahl erfolgt anhand von Kostenfunktionen. Ergänzend zur gängigen Vorgehensweise wurde in dieser Arbeit der Evolutionsschritt „Selektion“ in zwei Teilschritte unterteilt.

Das natürliche Vorbild sieht zunächst die Selektion der besten Individuen vor.

Diese entspricht einer Übernahme der bisher besten Individuen anhand der Ausfallrate und der Wirtschaftlichkeitsmaßzahl. Individuen, deren Ausfallrate höher als die höchstzulässige Ausfallrate ist, werden dabei verworfen

In einer späteren Generation kann das optimale Individuum auch aus den die maximal zulässige Ausfallrate knapp überschreitenden Individuen entstehen. Außerdem ist es sinnvoll, die Individuen nahe des augenblicklich besten Individuums verstärkt zu betrachten.

Daher wurde die bekannte Selektionsmethode erweitert, indem alle Individuen durch eine weitere Funktion bewertet werden. Für jedes Individuum ergibt sich ein Wert, der in dieser Arbeit Fitness genannt wird. Diese Funktion soll daher Fitnessfunktion heißen. Die fittesten Individuen werden in die nächste Generation übernommen.

Diese Methode der Selektion der Fittesten wurde speziell für das in dieser Arbeit vorliegende Problem entwickelt. Es werden die fittesten Individuen nahe des augenblicklich besten (nicht des fittesten!) Individuums übernommen.

#### 4.3.1.1 Prinzip 1a: Selektion der Besten

Ein Teil der Individuen der neuen Generation werden durch die „Selektion der Besten“ (Abbildung 18) unverändert in die neue Generation übernommen.

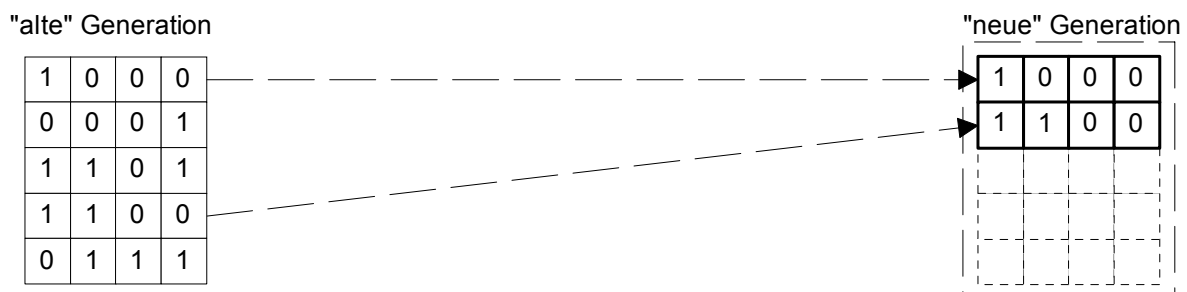


Abbildung 18: Selektion

Zu den Besten gehören diejenigen Individuen, deren Ausfallrate unter dem zulässigen Wert  $\lambda_{\text{soll}}$  liegen und deren Wirtschaftlichkeitsmaßzahl zu den niedrigsten gehört. Nach diesen Kriterien wird auch das optimale Individuum gesucht.

Beide Auswahlkriterien müssen erfüllt werden. Bei ausschließlicher Berücksichtigung der Wirtschaftlichkeit würden voraussichtlich die Anforderungen an die Ausfallrate nicht erfüllt werden, und umgekehrt.

Zur Auswahl der besten Individuen wird die Qualitätsfunktion  $g(\lambda)$  eingeführt.

$$g(\lambda) = \begin{cases} W & \text{für } \lambda_{\text{ist}} \leq \lambda_{\text{soll}} \\ \infty & \text{sonst} \end{cases} \quad (31)$$

Sie berücksichtigt nur die Wirtschaftlichkeitsmaßzahlen derjenigen Individuen, deren Ausfallraten mindestens die Vorgabe erfüllen.

Durch die „Selektion der Besten“ werden also die wirtschaftlichsten der bisher erstellten brauchbaren Individuen gespeichert.

Durch diese Selektion der Besten werden bevorzugt die guten Lösungen in die neue Generation übernommen. Die anderen Prinzipien werden somit in den folgenden Generationen vorzugsweise auf gute Individuen angewendet. Aus ihnen sollen sich später noch bessere ergeben.

Dieser Teil der neuen Generation enthält später auch das optimale Individuum.

#### 4.3.1.2 Prinzip 1b: Selektion der Fittesten

Die „Selektion der Fittesten“ erfolgt ebenfalls ähnlich der Abbildung 18. Auch hier werden Individuen unverändert in die nächste Generation übernommen. Die „Selektion der Fittesten“ beruht auf der Bewertung der Fitness  $s$  aller Individuen der aktuellen Generation anhand einer Fitnessfunktion. Für die vorliegende Aufgabe wurde die Fitnessfunktion

$$s(W, \lambda) = W_{norm} - \lambda_{norm} \quad (32)$$

entwickelt.

##### 4.3.1.2.1 Die normierte Ausfallrate

Zur Berechnung der Fitnessfunktion wurde die normierte Ausfallrate  $\lambda_{norm}$  eingeführt. Sie ist in Gleichung (33) dargestellt und wurde eigens für das vorliegende Problem entwickelt.

$$\lambda_{norm} = \left( \frac{\lambda_{ist}}{\lambda_{best}} \right) \text{SGN}(\lambda_{ist} - \lambda_{best}) \quad (33)$$

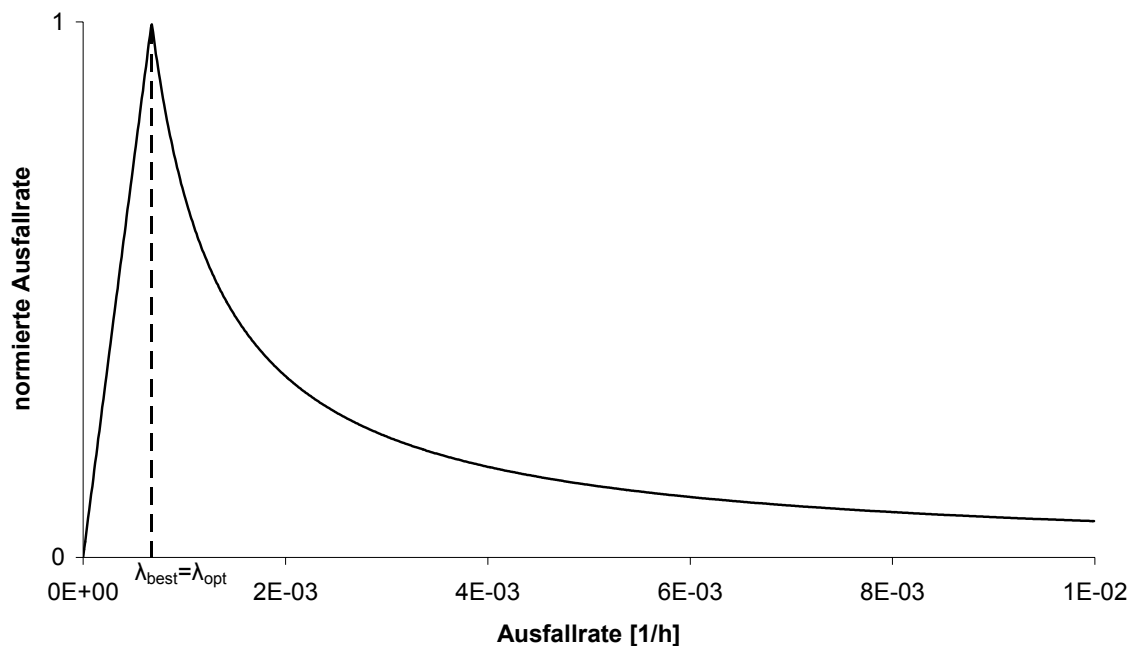
Sie wird berechnet, indem die Ausfallrate  $\lambda_{ist}$  des betrachteten Individuums und die Ausfallrate  $\lambda_{best}$  des besten (nicht des fittesten!) Individuums der vorangegangenen Generation verglichen werden. Wenn gilt:  $\lambda_{ist} > \lambda_{best}$ , dann liegt die normierte Ausfallrate  $\lambda_{norm}$  in dem Wertebereich

$$0 \leq \lambda_{norm} \leq 1. \quad (34)$$

Dieser Wertebereich muss auch für den Fall  $\lambda_{best} > \lambda_{ist}$  eingehalten werden. Daher wird im Exponenten der Normierung die Vorzeichen-Funktion (Signum-Funktion) verwendet. Das Vorzeichen der Abweichung von  $\lambda_{ist}$  wird bewusst vernachlässigt.

Der Wert von  $\lambda_{best}$  ist nicht konstant, sondern entspricht jeweils dem Wert der Ausfallrate des wirtschaftlichsten Individuums der Vorgänger-Generation. Das wirtschaftlichste Individuum kann in der nächsten Generation bereits ein anderes sein. Daher muss  $\lambda_{norm}$  in jeder Generation neu berechnet werden.

Die Abbildung 19 zeigt den theoretischen Verlauf von  $\lambda_{\text{norm}}$ .



**Abbildung 19: Theoretischer Verlauf von  $\lambda_{\text{norm}}$  aus Gleichung (33) bei der Normierung auf die Ausfallrate des optimalen Individuums**

Der Normierungswert ist hier durch die gestrichelte Linie gekennzeichnet. Die Normierung erfolgte in diesem Fall auf die Ausfallrate eines Individuums mit  $\lambda_{\text{best}} = 6,77 \cdot 10^{-4}/\text{h}$ .

Der Verlauf der Kennlinie in dem Bereich  $\lambda_{\text{best}} > \lambda_{\text{ist}}$  beginnt bei der Ausfallrate  $\lambda = 0$ , verläuft linear und endet bei  $\lambda_{\text{best}}$  mit dem Wert  $\lambda_{\text{norm}} = 1$ . Der Verlauf der Kennlinie im Bereich  $\lambda_{\text{best}} < \lambda_{\text{ist}}$  nähert sich asymptotisch dem Wert  $\lambda_{\text{norm}} = 0$  an.

Individuen mit einer schlechteren Ausfallrate als  $\lambda_{\text{best}}$  haben also im Vergleich zu den besseren bei gleichem Abstand zu  $\lambda_{\text{best}}$  einen höheren Wert  $\lambda_{\text{norm}}$ .

#### 4.3.1.2.2 Die normierte Wirtschaftlichkeitsmaßzahl

In der Fitnessfunktion wird die normierte Wirtschaftlichkeitsmaßzahl  $W_{\text{norm}}$  des betrachteten Individuums verwendet. Sie resultiert aus der Summe der Wirtschaftlichkeitsmaßzahlen aller verwendeten Gene, normiert auf die Anzahl der Gene. Jedes einzelne Gen kann die maximale Wirtschaftlichkeitsmaßzahl

$$W_{\text{max}} = 1 \quad (35)$$

aufweisen. Die größtmögliche Wirtschaftlichkeitsmaßzahl des Gesamtsystems ergibt sich somit aus dem Produkt der „Anzahl der Gene“ mit  $W_{\text{max}}$  zu

$$W_{\text{max System}} = \text{Genzahl} * W_{\text{max}} \quad (36)$$

Für  $W_{\text{norm}}$  gilt der Wertebereich

$$0 \leq W_{\text{norm}} \leq 1. \quad (37)$$

Man dividiert daher die tatsächliche Wirtschaftlichkeitsmaßzahl durch  $W_{\text{max System}}$  die nach Gleichung (36) identisch mit der Anzahl der Gene ist.

Die normierte Wirtschaftlichkeitsmaßzahl ergibt sich zu

$$W_{norm} = \frac{W_{System}}{W_{max System}} = \frac{W_{System}}{Genzahl} = \frac{\sum_i W_i}{Genzahl} \quad (38)$$

mit  $i \in \mathbb{N}$

#### 4.3.1.2.3 Der Wertebereich und die Auswirkung der Fitness

Subtrahiert man von der normierten Wirtschaftlichkeitsmaßzahl die normierte Ausfallrate unter Berücksichtigung der Vorzeichen-Funktion, so ergibt sich für die Fitnessfunktion  $s(W, \lambda)$  aus Gleichung (32) ein Wertebereich von

$$-1 \leq s(W, \lambda) \leq 1. \quad (39)$$

Ein niedriger Wert  $s$  steht für eine hohe Fitness, so dass bei der „Selektion der Fittesten“ die Individuen mit den kleinsten Fitness-Werten  $s$  übernommen werden. Das letztlich wirtschaftlichste Individuum muss nicht identisch mit dem Fittesten sein.

Die durch die Fitnessfunktion übernommenen Individuen sollen eine Ausfallrate und eine Wirtschaftlichkeitsmaßzahl nahe des momentan besten (nicht des fittesten!) Individuums haben und die Optimierung in Richtung der Verbesserung der Wirtschaftlichkeit vorantreiben.

Durch die Einführung der Fitnessfunktion, und die daraus resultierende Übernahme einer (größeren) Anzahl von Individuen (bessere und schlechtere) mit Werten nahe dem besten Individuum der aktuellen Generation, ergeben sich im wesentlichen zwei Vorteile:

1. Auch Individuen mit einer Ausfallrate  $\lambda_{ist} > \lambda_{soll}$  können aufgrund ihrer Fitness in die neue Generation übernommen werden und dort beliebig lange „überleben“. Aus ihnen kann sich in späteren Evolutionsschritten das optimale Individuum entwickeln.
2. Die Wahrscheinlichkeit, dass aus einem guten Individuum ein noch besseres wird ist höher als die, dass sich das optimale Individuum aus einem sehr schlechten entwickelt. Daher ist es effizient, wenn der Bereich nahe dem momentan besten Individuum ein höheres Gewicht bei der Erstellung der neuen Generation bekommt. Es werden mehr gute als schlechte Individuen übernommen und der bevorzugte Bereich kann deutlich unter dem Vorgabewert  $\lambda_{soll}$  liegen.

Bilden ausschließlich „gute“ Individuen die Basis der weiteren Optimierung, könnte ein „falscher Weg“ eingeschlagen werden. Beim Genetischen Algorithmus lassen sich die Vorgänger-Individuen nur in begrenztem Maße verändern. Die Vorgänger-Individuen werden daher die Optimierung in eine Richtung beeinflussen. Ein sich später daraus ergebendes Minimum der Wirtschaftlichkeitsmaßzahl könnte einem lokalen Minimum entsprechen. Dieses kann vom absoluten Minimum, also dem Optimum abweichen.

Die Manipulation von „schlechten“ Individuen durch die anderen Prinzipien des Genetischen Algorithmus ist daher notwendig, um lokale Minima verlassen zu können.

#### 4.3.2 Prinzip 2: Mutation

Die Mutation von Individuen ist in Abbildung 20 dargestellt. Hierbei wird das zufällig ausgewählte Gen eines beliebigen Individuums der alten Generation durch ein

beliebiges anderes Gen derselben Komponente der Datenbasis ersetzt. Das Individuum „mutiert“. Von diesen Mutanten werden beliebig viele erstellt und von diesen wiederum eine beliebige Anzahl der Besten in die neue Generation übernommen.

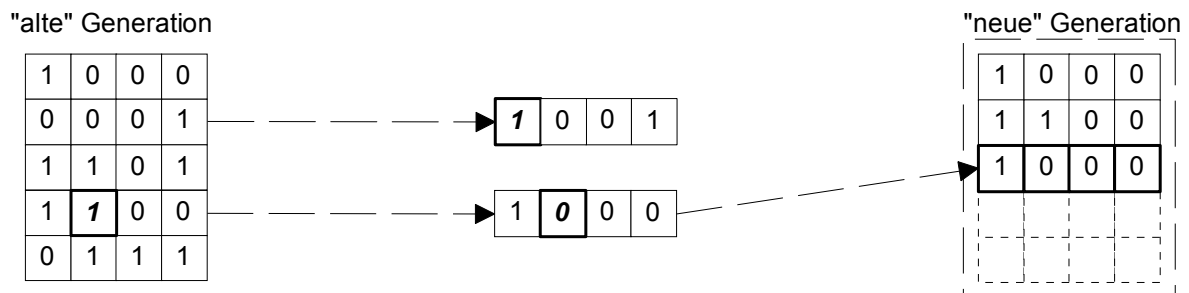


Abbildung 20: Mutation

Da die Qualität des ersetzenden Gens nicht relevant ist, kann das neue Individuum schlechter als das Eltern-Individuum sein und wird dennoch übernommen. Dies ist gewollt, um lokale Minima verlassen zu können. In einer nachfolgenden Generation kann hieraus ein besseres Individuum entstehen. Da jedoch nur ein einziges Gen verändert wird, wird sich das neue Individuum meist nur leicht verändern.

Es ist denkbar, mehrere Gene gleichzeitig zu mutieren. Mit dem Anstieg der Anzahl der Mutationen je Individuum erhöht sich auch die Wahrscheinlichkeit des Verlassens lokaler Minima. Dieser Effekt wird jedoch auch mit dem Evolutionsprinzip der Geburt erzielt. Bei der vorliegenden Arbeit wird jeweils nur ein Gen mutiert, d.h. ein Komponententyp ausgetauscht.

### 4.3.3 Prinzip 3: Kreuzung

Bei der Kreuzung (Abbildung 21) werden neue Individuen erstellt, indem die verwendeten Gene von (meist zwei) Eltern-Individuen gezielt ausgetauscht werden.

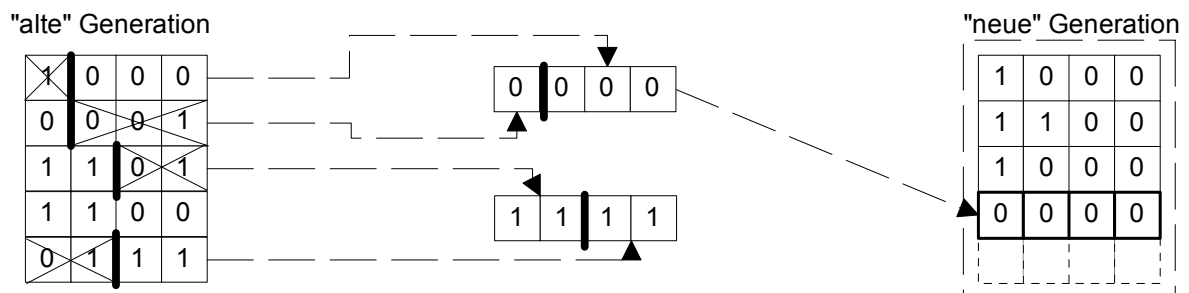


Abbildung 21: Kreuzung

Zunächst werden die zu kreuzenden Individuen zufällig ausgewählt und ein zufälliger Kreuzungspunkt bestimmt. In der Abbildung 21 ist dieser als dicker schwarzer Trennstrich kenntlich gemacht. Anschließend werden die Gene des vorderen Teils des einen Individuums mit den Genen des hinteren Teils des anderen Individuums gekreuzt und es entsteht ein neues Individuum. Auf diese Art lassen sich beliebig viele neue Individuen erstellen, von denen wiederum eine beliebige Anzahl der besten Individuen in die neue Generation übernommen werden. Der Veränderungseffekt der neuen Individuen gegenüber ihren Eltern kann größer sein, als bei der Mutation. Je nach Struktur der verwendeten Eltern-Individuen und Art der Neu-Anordnung können sich hieraus sehr gute, aber auch sehr schlechte Individuen ergeben.

#### 4.3.4 Prinzip 4: Geburt

Bei der Geburt (Abbildung 22) werden Individuen zufällig neu erzeugt.

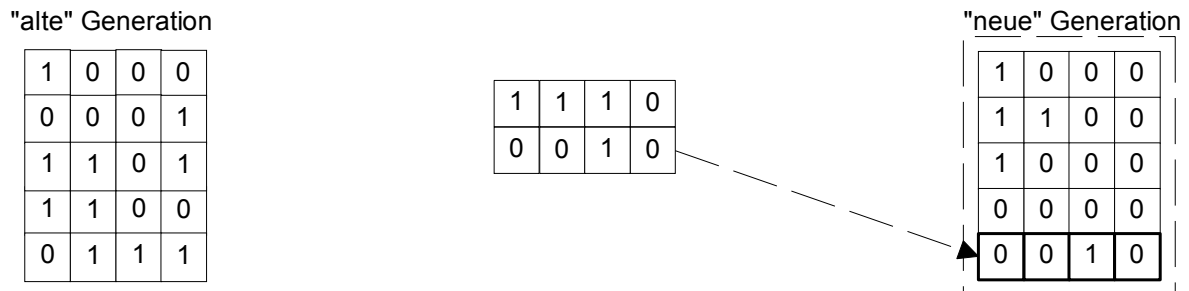


Abbildung 22: Geburt

Bei ausschließlicher Anwendung der drei zuvor dargestellten Prinzipien kann es lange dauern, bis sich eine Generation so deutlich von einer Ursprungs-Generation unterscheidet, dass lokale Minima verlassen werden können. Zufällig neu erstellte (geborene) Individuen können bisher unbekannte Evolutionsansätze für Nachfolgenerationen bieten und so die genannten Probleme umgehen.

#### 4.3.5 Abbruchkriterien

Es ist nicht immer sinnvoll, eine konstante Anzahl an Individuen zu übernehmen. Für diese Arbeit wurde daher eine Methode umgesetzt, bei der die Anzahl der erstellten und zu übernehmenden Generationen variiert.

Bei den zuvor dargestellten vier Evolutionsprinzipien kann es vorkommen, dass exakte Kopien von bereits bestehenden Individuen entstehen. Diese werden aussortiert, denn identische Individuen innerhalb einer Generation haben keinen für die Optimierung verwendbaren Informationsgehalt.

Das beste Individuum einer Kind-Generation kann nicht schlechter sein, als das beste der Eltern-Generation. Mit steigender Evolutionsdauer nähert sich die Qualität des besten Individuums der jeweils aktuellen Generation dem Optimum. Meist werden zu einem frühen Zeitpunkt der Evolution stärkere Verbesserungen erzielt als später. Die Evolution muss irgendwann durch die Erfüllung geeigneter Kriterien abgebrochen werden.

Da die Methode der Genetischen Algorithmen zufallsbasiert ist, ist es nicht sicher, dass nach dem Abbruch der Optimierung tatsächlich das wirtschaftlichste Individuum gefunden wurde. Mit zunehmender Evolutionsdauer steigt einerseits die Qualität der besten Individuen, weiterhin aber auch die Wahrscheinlichkeit dieses zu finden. Selbst wenn nach einer großen Anzahl an Generationen nicht das wirtschaftlichste Individuum gefunden wurde, sollte es diesem schon sehr nahe kommen.

Das Abbruchkriterium der realisierten Methode beruht auf der Beobachtung der Historie der Veränderungen der besten Individuen. Es wird mehr als ein bestes Individuum in die neue Generation übernommen.

Nachfolgend werden zur Vereinfachung dieser Betrachtung zwei Begriffe definiert, die auch im weiteren Verlauf dieser Arbeit verwendet werden, die **Abbruchzahl** und die **Abbruchvorgabe**.

Die *Abbruchzahl* sei die Anzahl der Generationen, die während der Evolution durchlaufen wurden, ohne dass sich die jeweils besten Individuen verbessert haben.

Die *Abbruchvorgabe* ist die vorgegebene Anzahl der Generationen, die ohne Änderung der besten Individuen durchlaufen werden müssen, um die Optimierung abzubrechen.

Abbruchzahl und Abbruchvorgabe werden nach jeder Generation miteinander verglichen. Die Optimierung wird abgebrochen wenn gilt

$$\text{Abbruchzahl} = \text{Abbruchvorgabe} . \quad (40)$$

Aufgrund des zufallsbasierten Ansatzes von Genetischen Algorithmen können nach wie vor die besten Gene „übersehen“ worden sein. Manchmal können diese ganz offensichtlich sein, so dass abschließend die Datenbasis nach ihnen durchsucht wird. Diese Verfahrensweise wird „gezielte Genmanipulationen“ genannt.

#### 4.3.6 Gezielte Genmanipulation

Die Qualität der nach der gesamten Evolution verbleibenden Individuen ist von den gewählten Optimierungsparametern abhängig. Wie in 4.3.5 erwähnt, wird das wirtschaftlichste Individuum nicht mit Sicherheit gefunden. Daher wird nach Abbruch der Optimierung eine gezielte Genmanipulation durchgeführt. Dieses Verfahren ergänzt die bekannten Evolutionsprinzipien.

Es kann sein, dass die Datenbasis noch bessere Gene als die des scheinbar optimalen Individuums enthält und dieses während der gesamten Evolution „übersehen“ wurde. Abschließend wird daher das vermeintlich beste Individuum gezielt genmanipuliert (Abbildung 23).

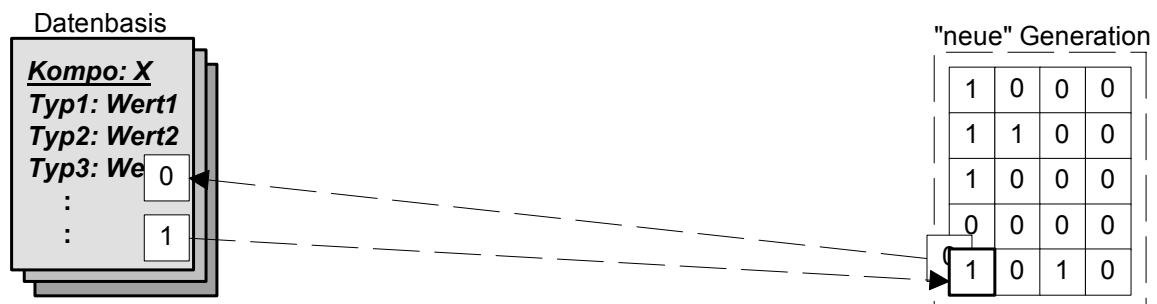


Abbildung 23: Gezielte Genmanipulation nach Abschluss der Evolution

Hierzu wird jedes im besten Individuum enthaltene Gen mit jedem relevanten Gen der Datenbasis verglichen. Findet sich in der Datenbasis ein besseres, so ersetzt es das entsprechende Gen im Individuum.

Ein Gen ist besser, wenn es bei gleicher oder geringerer Ausfallrate bei allen Wirtschaftlichkeitsparametern mindestens von der Qualität des zu ersetzenden Gens ist.

Dieses Verfahren bedeutet nur wenig Mehraufwand, denn der Vergleich des besten Individuums mit der Datenbasis muss nur einmal durchgeführt werden.

#### 4.3.7 Bewertung der Methode des Genetischen Algorithmus

Bereits mit der ersten erstellten Generation lässt sich beim Genetischen Algorithmus eine (meist suboptimale) Lösung angeben. Im weiteren Verlauf der Evolution steigert sich die Qualität der Lösung. Da nicht der gesamte Lösungsraum betrachtet wird, können mit realisierbarem Aufwand auch akzeptable Lösungen für NP-vollständige Probleme gefunden werden.



Nach jedem Evolutionsschritt müssen die existierenden Individuen einer Generation bewertet werden. Voraussetzung für die Anwendbarkeit dieser Methode ist, dass diese Bewertung mit geringem Aufwand geschehen kann. Da hierzu nur wenige Additionen und Multiplikationen durchgeführt werden müssen, ist diese Anforderung erfüllt. Dieser Aufwand, multipliziert mit der Anzahl der Evolutionsschritte und der Anzahl der Individuen in jeder Generation, ergibt den Gesamtaufwand der Optimierung. Hierzu addiert sich gegebenenfalls noch der Aufwand der gezielten Genmanipulation, die abschließend durchgeführt wird.

*Ein Vorteil der Methode des Genetischen Algorithmus ist die Anwendbarkeit auch bei Systemen mit hoher Komplexität.*

Die Prinzipien der Genetischen Algorithmen wurden der biologischen Evolution nachempfunden. Diese Methode findet daher, ähnlich der biologischen Entwicklung, meist sehr gute, jedoch nicht immer optimale Lösungen. Die Prinzipien der Genetischen Algorithmen sind zwar strukturiert, die Erstellung neuer Individuen jedoch zufallsbasiert. Durch die Anwendung der vier dargestellten Prinzipien entstehen in jeder neuen Generation Individuen, die sich einerseits von ihren Vorgängern deutlich unterscheiden können, deren Qualität der besten Individuen sich andererseits stetig verbessert. Lokale Minima können verlassen werden.

*Weitere Vorteile sind das Finden einer Lösung nahe dem Optimum und das Überspringen lokaler Minima.*

*Ein Nachteil der Methode des Genetischen Algorithmus ist, dass das Optimum nicht mit Sicherheit gefunden wird.*

#### **4.4 Vergleich der Methoden**

Jede der drei dargestellten Methoden hat ihre jeweiligen Stärken und Schwächen.

Da der Graphenalgorithmus den gesamten Lösungsraum nach der optimalen Lösung durchsucht, scheint er die richtige Optimierungsmethode zu sein. Er ist jedoch nur anwendbar, sofern der Lösungsraum nicht zu groß ist. Seine Schwäche ist die Nicht-Beherrschung von Lösungsräumen hoher Komplexität. Trotzdem der Lösungsraum beim Graphenalgorithmus mit eindeutigen Regeln verkleinert werden kann, kann die Komplexität der zu untersuchenden Systeme schnell in nicht mehr berechenbare Dimensionen steigen. Der Graphenalgorithmus ist dann nicht mehr für alle Probleme anwendbar.

Lösungsräume mit hoher Komplexität werden durch die Anwendung von Heuristiken beherrscht. Daher wurden zwei heuristische Methoden untersucht.

Die realisierte und ursprünglich aus der Regelungstechnik stammende Neuro-Fuzzy-Methode ist heuristisch und deterministisch. Sie findet schnell auch für hoch komplexe Aufgaben eine Lösung. Ihre Anpassung der Fuzzy-Sets erfolgt aufgrund eines „globalen“ Systemfehlers und mit vom Anwender (Programmierer) einzustellender Auflösung. Da nicht festgestellt werden kann, welche Komponente primär für den Systemfehler verantwortlich ist, werden immer alle Fuzzy-Sets angepasst. Aus diesem Grund werden oftmals relevante Kombinationen „übersehen“ und die Optimierung endet meist in lokalen Minima. Die gefundene Lösung ist häufig suboptimal und von unzureichender Qualität. Daher wurde als dritte Methode ein Genetischer Algorithmus untersucht.

Der realisierte Genetische Algorithmus ist ebenfalls heuristisch, jedoch im Unterschied zur Neuro-Fuzzy-Methode zufallsbasiert. Seine Heuristik erlaubt die

Behandlung beliebig komplexer Lösungsräume. Durch den zufallsbasierten Ansatz können lokale Minima verlassen werden. Die gefundene Lösung nähert sich mit steigender Optimierungsdauer dem Optimum an. Das Finden des Optimums ist zwar nicht sichergestellt, die gefundene Lösung liegt jedoch nahe (abhängig von der Optimierungsdauer) am Optimum.

Aufgrund des heuristischen Ansatzes und der sehr guten Lösungsqualität des Genetischen Algorithmus kann dessen Anwendbarkeit auf das in dieser Arbeit untersuchte Problem als allgemeingültig angenommen werden. Für das Optimierungsproblem ist daher der Genetische Algorithmus zu wählen.

## 5 Bewertung der Unsicherheit von Lösungen (vage Daten)

Nach der Anwendung einer der zuvor dargestellten Optimierungsmethoden liegt die wirtschaftlichste Lösung des Problems vor. Bis zum Abschluss der Optimierung wurden alle Typen der Datenbasis durch exakt definierte (scharfe) Werte beschrieben. Die verschiedenen Lösungsvorschläge wurden durch den Vergleich der sich ergebenden exakten Werte bewertet.

Während des Entwicklungsprozesses von Systemen sind jedoch viele Werte der Typen in der Datenbasis, aufgrund unzureichender Praxiserfahrung, als unsicher anzusehen. Sie beruhen oft auf Schätzungen. Diese Datenunsicherheit führt dazu, dass die Optimierungsergebnisse häufig, auf Grund ihrer unzuverlässigen Zielwertermittlung, nicht verwendbar sind. Die tatsächlichen Werte können von den Prädiktwerten sehr stark abweichen.

Zur Bewertung der Datenunsicherheit kann Fuzzy-Logik (siehe Kapitel 4.2) verwendet werden. Theoretische Grundlagen hierzu sind beispielsweise in [Zad65],[ZaK92] und [TFL83] erläutert. In [Mey00] wird die Möglichkeit der Bewertung unscharfer und aus Fehlerbäumen resultierender Ergebnisse dargestellt.

Hierauf basierend wurde im Rahmen dieser Arbeit eine Möglichkeit zur Bewertung der Datenunsicherheit während des Entwicklungsprozesses gefunden.

Die Anwendung dieser Methode ist auf zweierlei Arten denkbar:

Vorgehensweise 1: Bereits während der Optimierung wird ausschließlich mit den vagen Daten gerechnet.

Ein *Vorteil* der Vorgehensweise 1 ist, dass die Optimierung gezielt auch hinsichtlich des geringsten Entwicklungsrisikos durchgeführt werden kann. Die gefundene Lösung könnte dann gleichzeitig diejenige mit dem geringsten Entwicklungsrisiko sein. Ein *Nachteil* der Vorgehensweise 1 ist, dass eine (aufwendige) Anpassung der in Kapitel 4 dargestellten Methoden nötig wäre.

Vorgehensweise 2: Nach der (mit den scharfen Werten durchgeführte) Optimierung liegt eine Lösung vor. Diese wird hinsichtlich ihrer Datenunsicherheit bewertet und somit nur das Entwicklungsrisiko dieser einen Lösung quantifiziert.

Ein Vorteil der Vorgehensweise 2 ist, dass nur für die Typen in den zu untersuchenden Lösungen vage Daten angegeben werden müssen. Zur Umsetzung ist nur ein geringer Aufwand nötig, da die Methoden nicht angepasst und nur die gefundene(n) Lösung(en) betrachtet werden müssen.

Ein Nachteil der Vorgehensweise 2 ist, dass zum Finden der wirtschaftlichsten Lösung mit dem geringsten Entwicklungsrisiko mehrere verschiedene Lösungen zu betrachten sind. Ihre Auswahl erfolgt „manuell“ durch den Entwickler. Unter der Annahme, dass die optimale Lösung mit den scharfen Werten nahe der optimalen Lösung mit den vagen Daten liegt, lässt sich auch auf diese Art die wirtschaftlichste Lösung finden, die das relativ geringste Entwicklungsrisiko aufweist.

Es wurde daher die Vorgehensweise 2 umgesetzt.

Für die Optimierungsphase wurde die Wirtschaftlichkeitsmaßzahl (vgl. Kapitel 2.1) eingeführt. Zur Bewertung der Entwicklungsunsicherheit müssen jedoch die individuellen Risiken jedes Parameters des Gesamtsystems abgeschätzt werden. Die Wirtschaftlichkeitsmaßzahl mit ihrem einheitslosen Wertebereich (0 bis 1), lässt jedoch keinen Rückwärtsschluss auf die Kosten, das Gewicht und das Volumen der

Gesamtlösung zu. Daher kann die Wirtschaftlichkeitsmaßzahl nicht mehr verwendet und die Kosten, das Gewicht und das Volumen müssen wieder getrennt voneinander betrachtet werden.

Die Geometrie-Daten der entsprechenden Typen werden weiterhin zum Volumen zusammengefasst, denn eine Betrachtung der einzelnen Parameter (Längen, Breiten Höhen) lässt sich nicht sinnvoll begründen. Der mathematische Nachweis ergibt sich durch die Berücksichtigung des Distributivgesetzes [BrS87], aus dem sich folgende Ungleichung ergibt

$$l_1 \cdot b_1 \cdot h_1 + l_2 \cdot b_2 \cdot h_2 \neq (l_1 + l_2) \cdot (b_1 + b_2) \cdot (h_1 + h_2). \quad (41)$$

Die Terme  $l$ ,  $b$  und  $h$  stehen für die einzelnen Geometrie-Parameter Länge, Breite und Höhe. Diese sind aufgrund der Ungleichung nur über das Volumen vergleichbar. Der Schluss auf das (unscharfe) Gesamtvolumen des Systems ist (auch mathematisch) nur über die Summe der einzelnen Volumen zulässig. Es gilt

$$V_{ges} = \sum_i V_i. \quad (42)$$

### 5.1 Anwendbare Fuzzy-Sets

Zunächst muss der Entwickler alle Typen der Datenbasis hinsichtlich ihrer vagen Daten bewerten. Er beurteilt also die Verlässlichkeit der vorliegenden Daten indem er bewertet, in welche Richtung und wie stark der tatsächliche Wert von der Vorgabe abweichen kann. Diese Bewertung muss für jeden mit Entwicklungsrisiko behafteten Wert der Datenbasis vorgenommen werden. Die Parameter werden fuzzyfiziert.

Bei der realisierten Methode ergeben sich für die unscharfen Parameter Fuzzy-Sets in der Form von Trapezen und deren Sonderformen. Die möglichen Fuzzy-Sets sind am Beispiel der Ausfallrate in der Abbildung 24 dargestellt.

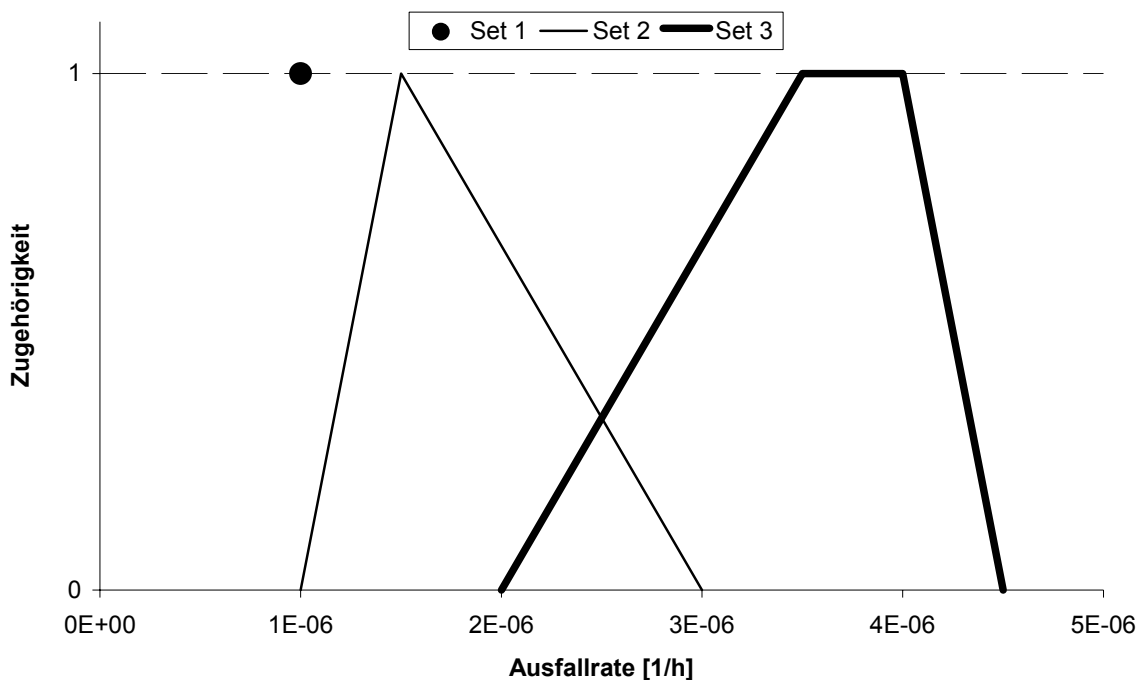


Abbildung 24: Darstellung verschiedener Fuzzy-Sets am Beispiel der Ausfallrate

Die Trapezform des Set 3 ist definiert durch seine vier Eckkoordinaten (von links: a, b, c, d). Die Darstellung der unscharfen Parameter erfolgt also durch die Angabe von vier Werten in der Form

$$\tilde{M}_i = (m_{ai}, m_{bi}, m_{ci}, m_{di}). \quad (43)$$

M ist hierin der Platzhalter für die Parameterart ( $\lambda$ , K, G, V) des Typs i und das Tilde-Zeichen über der Parameterart gibt an, dass es sich um die Fuzzy-Darstellung handelt. Daher wird folgende Definition getroffen:

$$\text{Wenn } \rightarrow m_{ai} = m_{bi} = m_{ci} = m_{di} \quad \text{dann } \rightarrow \text{Es handelt sich um die Punktform} \quad (44)$$

$$\text{und es gilt: } \mu_{ai} = \mu_{bi} = \mu_{ci} = \mu_{di} = 1$$

$$\text{Wenn } \rightarrow m_{ai} < m_{bi} = m_{ci} < m_{di} \quad \text{dann } \rightarrow \text{Es handelt sich um die Dreiecksform} \quad (45)$$

$$\text{und es gilt: } \begin{aligned} \mu_{ai} &= \mu_{di} = 0 \\ \mu_{bi} &= \mu_{ci} = 1 \end{aligned}$$

$$\text{Wenn } \rightarrow m_{ai} < m_{bi} < m_{ci} < m_{di} \quad \text{dann } \rightarrow \text{Es handelt sich um die Trapezform} \quad (46)$$

$$\text{und es gilt: } \begin{aligned} \mu_{ai} &= \mu_{di} = 0 \\ \mu_{bi} &= \mu_{ci} = 1 \end{aligned}$$

Das Set 1 (scharfer Wert,  $a=b=c=d$ ) und das Set 2 (Dreieck,  $b=c$ ) sind die Sonderformen des Set 3.

Set 1 ist durch einen scharfen Wert definiert. Es hat immer genau die Zugehörigkeit  $\mu=1$  zum angegebenen Wert (hier z. B. Ausfallrate  $\lambda=1 \cdot 10^{-6}/h$ ). Zu allen anderen Werten im betrachteten Wertebereich hat es die Zugehörigkeit  $\mu=0$ .

Ist die Datenbasis gesichert, so kann der entsprechende Parameter des betrachteten Typs durch einen scharfen Wert angegeben werden. Die Datenbasis eines Typs darf als gesichert angenommen werden, wenn dieser beispielsweise in einem ähnlichen System an gleicher Stelle im Kfz bereits im Serieneinsatz ist.

Set 2 ist durch ein Intervall beschrieben, das seine höchste Zugehörigkeit ( $\mu=1$ ) beim angegebenen scharfen Wert des betrachteten Typs hat (hier:  $\lambda=1,5 \cdot 10^{-6}/h$ ). Rechts und links hiervon sind Intervalle definiert, die mit einer geringeren Zugehörigkeit für den betrachteten Typ gültig sind. Diese Intervalle auf der linken und der rechten Seite des scharfen Wertes können von unterschiedlicher Größe sein.

Dieses Set ist zu wählen, wenn der betrachtete Typ zwar bereits in Serie eingesetzt wird (Maximalwert), das neue System oder der neue Einbauort sich jedoch deutlich von den bisherigen Bedingungen unterscheiden. Sind die neuen Anforderungen härter als bisher, so wird das Intervall in Richtung der Verschlechterung der Werte (hier: rechts) breiter sein, und umgekehrt.

Set 3 hat wie Set 2 ebenfalls Intervalle mit verminderten Zugehörigkeiten. Im Unterschied zu Set 2 wurde hier jedoch der scharfe Wert bei  $\mu=1$  durch ein Intervall mit  $\mu=1$  ersetzt.

Dieses Set ist zu wählen, wenn der betrachtete Typ eine Neuentwicklung ist, über deren scharfen Wert wenig oder gar kein Vorwissen (Schätzung) existiert.

Nachdem die Unsicherheit jedes Typs definiert und das Gesamtsystem optimiert ist, kann durch die Vereinigung der Fuzzy-Sets aller relevanten Parameter auf die Unsicherheit der Gesamtlösung geschlossen werden.

## 5.2 Fuzzy-Vereinigung

Zadeh entwickelte bereits im Jahre 1965 das Erweiterungsprinzip [Zad65]. In diesem wird die Übertragbarkeit der mathematischen Konzepte der scharfen Zahlen auf die Fuzzy-Mengen bewiesen. Bei der Vereinigung der Fuzzy-Sets wird dieselbe mathematische Funktion angewendet, die zur Zusammenfassung der scharfen Werte nötig ist.

Zadehs Beweise sind für beliebige Funktionen gültig. Bei der vorliegenden Aufgabe verlaufen die betrachteten Funktionen stückweise linear. Es lassen sich so nach [Mey00] die grundlegenden mathematischen Konzepte wie folgt vereinfachen:

$$A \oplus B = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 + b_1 \\ a_2 + b_2 \end{pmatrix}, \quad (47)$$

$$A \otimes B = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} * \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 * b_1 \\ a_2 * b_2 \end{pmatrix}, \quad (48)$$

$$A \simeq B = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} - \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 - b_1 \\ a_2 - b_2 \end{pmatrix}, \quad (49)$$

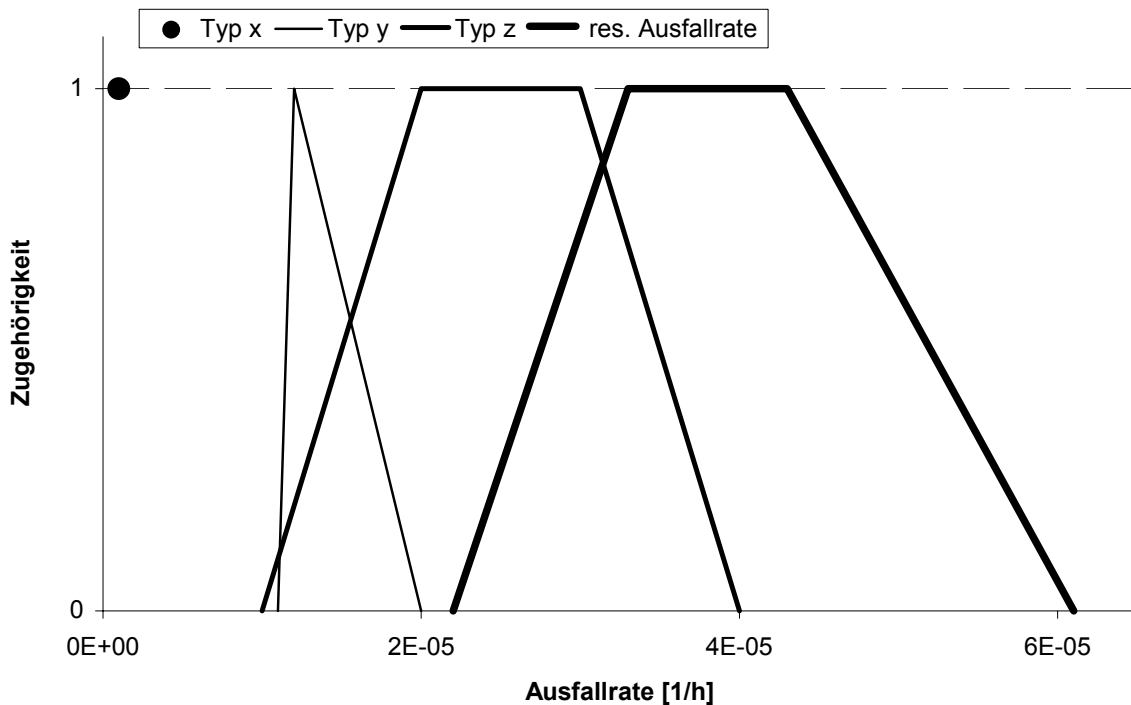
$$A \tilde{\div} B = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \div \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 \div b_1 \\ a_2 \div b_2 \end{pmatrix}. \quad (50)$$

Die relevanten Parameter Ausfallrate, Kosten, Gewicht und Volumen werden durch Addition zusammengefasst, so dass lediglich die Gleichung (47) angewendet werden muss. Diese Vorgehensweise entspricht der bereichsweisen Addition diskreter Werte. Zwei trapezförmige Fuzzy-Mengen werden wie folgt addiert:

$$\tilde{M}_1 \oplus \tilde{M}_2 = \begin{pmatrix} m_{a1} \\ m_{b1} \\ m_{c1} \\ m_{d1} \end{pmatrix} + \begin{pmatrix} m_{a2} \\ m_{b2} \\ m_{c2} \\ m_{d2} \end{pmatrix} = \begin{pmatrix} m_{a1} + m_{a2} \\ m_{b1} + m_{b2} \\ m_{c1} + m_{c2} \\ m_{d1} + m_{d2} \end{pmatrix}. \quad (51)$$

Somit lassen sich alle unsicheren Parameter des Gesamtsystems zusammenfassen. Eine Bewertung der resultierenden Entwicklungsunsicherheit wird möglich.

Die Abbildung 25 zeigt diese Bewertungsart am Beispiel der Ausfallrate von drei fiktiven Typen (Typen x, y und z). Der Typ x ist durch einen scharfen Wert (vgl. Set 1) beschrieben. Typ y entspricht einem in ähnlicher Form bereits in Serie verwendeten Typen (vgl. Set 2). Der Typ z ist eine Neuentwicklung, dessen Ausfallrate nur geschätzt (vgl. Set 3) werden kann.



**Abbildung 25: Zusammensetzung der System-Ausfallrate mit unscharfer Datenbasis**

Ist die Datenbasis (Vorwissen) unscharf, wird auch die Summe (z. B. resultierende Ausfallrate) unscharf sein.

Die Form des sich ergebenden Fuzzy-Intervalls resultiert aus den Fuzzy-Sets der verwendeten Typen. Die Breite ist ein Indiz für die Aussagequalität des berechneten Ergebnisses. Mit zunehmender Breite verringert sich dessen Güte und die Entwicklungsunsicherheit steigt. Welcher Ansatz im Entwicklungsprozess weiterverfolgt wird, bleibt letztlich wiederum dem Expertenwissen des Entwicklers überlassen. Er muss entscheiden, welche Unsicherheit tragbar ist. Es kann sinnvoll sein, eine suboptimale Lösung mit hoher Ergebnislösungsgüte aber gegenüber dem Optimum verringerter Wirtschaftlichkeit zu verwenden. Der Entwickler muss die Unsicherheit gegenüber dem möglichen Wirtschaftlichkeitsgewinn abwägen. Die dargestellte Methode der Beurteilung der vagen Daten erleichtert ihm diese Entscheidung.

### **5.3 Zusammenfassung der Methode zur Bewertung der Entwicklungsunsicherheit**

Durch die Bewertung und Zusammenfassung der unsicheren Daten der verwendeten Typen wird ein Rückschluss auf die Entwicklungsunsicherheit des Gesamtsystems hinsichtlich der möglichen Kosten, des Gewichts, der Ausfallrate und des resultierenden Gesamtvolumens möglich.

Die resultierende Entwicklungsunsicherheit wird quantifiziert.

Es ist jetzt eine Beurteilung möglich, inwiefern sich die Entwicklungsunsicherheit nach Änderung der System-Auslegung (beispielsweise bei Verwendung anderer Typen, Modifikation der Topologie) verändert hat. Weiterhin kann das Risiko zum Erlangen eines Zielwertes abgeschätzt werden (Risiko-Nutzen-Analyse).

## 6 Das realisierte Optimierungstool

Zur Validierung der theoretischen Erkenntnisse wurden die Algorithmen anhand eines geeigneten Tools umgesetzt. Für ähnliche Optimierungs- oder Regelungsaufgaben sind verschiedene Tools, wie beispielsweise das kommerzielle Matlab/ Simulink oder auch der universitäre „Stuttgarter Neuronale Netze Simulator“ (SNNS) [ZeA95], verfügbar.

Die Funktionsweise verfügbarer Tools ist selten offengelegt, so dass bei diesen nicht oder nur sehr aufwendig analysiert werden kann, wie und warum sich die jeweiligen Lösungen in der entsprechenden Güte und Performance ergeben. Leider lassen sich diese Tools auch nicht oder nur begrenzt auf das vorhandene Problem und die individuellen Methoden anpassen. Aufgrund dieses „Black-Box-Effektes“ verfügbarer Tools wurde in dieser Arbeit auf die Verwendung fertiger Tools verzichtet.

Zur Validierung der vorgestellten Optimierungsmethoden wurden im Rahmen dieser Arbeit drei Optimierungstools erstellt.

Je ein Tool wurde für den Graphenalgorithmus und die Neuro-Fuzzy Methode erstellt.

Im dritten Tool wurde der Genetische Algorithmus realisiert und angepasst. Nachdem dieser die gewünschten Optimierungsergebnisse lieferte, wurde das Tool um die Methode zur Beurteilung vager Daten erweitert. Alle Tools basieren auf der objektorientierten Programmiersprache C++.

Die Strukturen aller drei Tools sind sehr ähnlich. Die grafische Benutzeroberfläche (GUI = graphical user interface) ist bei allen Tools weitestgehend, sofern es die Methode zuließ, identisch. Der Programmieraufwand konnte dadurch primär in die Hauptaufgabe dieser Arbeit, die Realisierung und Verbesserung der Algorithmen, investiert werden.

Dem Aufwandsnachteil durch die Erstellung eines eigenen Tools stehen im wesentlichen drei Vorteile gegenüber:

- Möglichkeit der Umsetzung und des Tests eigener Methoden,
- Möglichkeit des Imports und Exports von Daten in beliebiger Form und
- Vergleichbarkeit der resultierenden Ergebnisse.

Die Struktur und Peripherie ist in jedem der erstellten Tools weitgehend identisch und unterscheidet sich nur durch die realisierten Optimierungsalgorithmen. Daher resultieren die unterschiedlichen Ergebnisse der verschiedenen Methoden direkt aus deren Optimierungsalgorithmen.



Die Abbildung 26 zeigt den für alle drei Tools gültigen schematischen Aufbau.

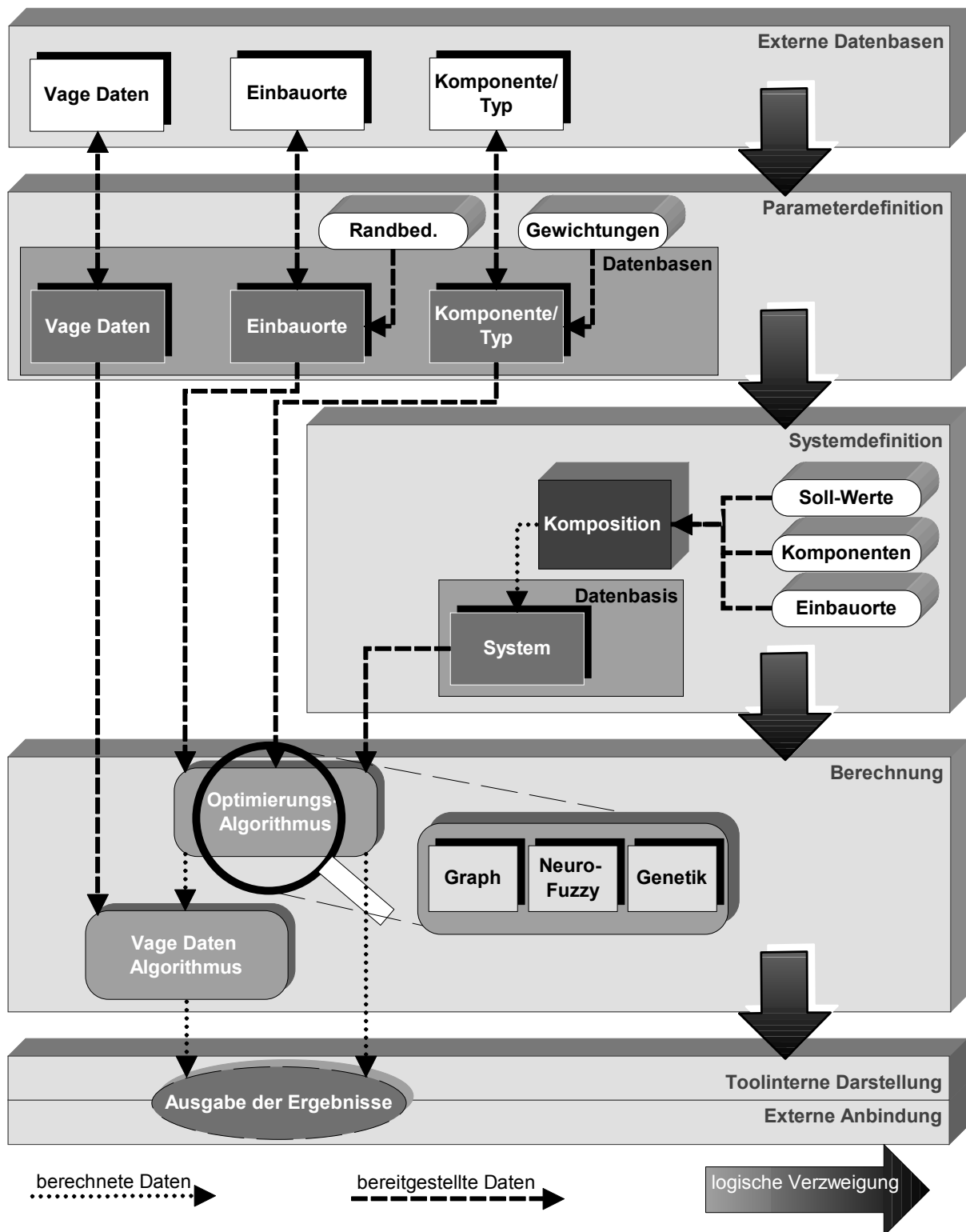


Abbildung 26: Aufbau der Optimierungstools

Die Tools sind im Wesentlichen in sechs Logik-Blöcke unterteilt. Diese und ihre integrierten Bausteine sind durch drei verschiedene Verzweigungsarten verbunden.

Die gepunktete Verzweigung steht für die Übergabe zuvor berechneter Daten. Diese Berechnung führt das erstellte Tool durch.

Die gestrichelte Verzweigung steht für die Übergabe von Daten aus den verschiedenen Datenbasen. Diese Daten werden durch das Tool inhaltlich nicht verändert, eventuell aber Algorithmengerecht aufbereitet.

Der dicke Strukturpfeil weist auf die logische Abfolge zur Durchführung einer Optimierung hin. Er verbindet daher folgende Logik-Blöcke miteinander:

1. Externe Datenbasen,
2. Parameterdefinition,
3. Systemdefinition,
4. Berechnung,
5. toolintern Darstellung und
6. externe Anbindung.

Im Anhang 13.10 sind einige Screenshots des realisierten Tools mit der Methode des Genetischen Algorithmus dargestellt.

### **6.1 Anbindung externer Datenbasen und Weiterverarbeitung der Ergebnisse**

Der erste Block enthält die externen Datenbasen. Diese sind über Im- und Exportfilter an den Block der Parameterdefinition angebunden

Verschiedene für die Optimierung benötigte Daten sind in externen Programmen und Dateien bereits verfügbar. Es ist sinnvoll diese Daten anhand geeigneter Importfilter einzubinden. Durch die Verwendung von Import- und dementsprechend auch Exportfiltern wird die Möglichkeit geschaffen, bereits vorhandene Programme zu nutzen, beispielsweise zum Erstellen oder Editieren von Datenbanken. Es lässt sich Arbeitsaufwand sparen.

Weiterhin ist es notwendig, die gefundenen Ergebnisse bearbeiten und in einer geeigneten Form darstellen zu können. Hierfür wurde eine Exportmöglichkeit in das Microsoft Excel-Datenformat geschaffen.

Die meisten Diagramme und Tabellen dieser Arbeit resultieren aus diesen exportierten Daten des sechsten Blocks „Externe Datenanbindung“.

### **6.2 Definition der Parameter**

Der zweite Block entspricht einer toolinternen Datenbasis. In dieser werden alle benötigten Daten aus Dateien (mit tooleigenen und fremden Datenbasen) entnommen, aufbereitet und in geeigneten Datenstrukturen im Speicher des Systems abgelegt. Eine Aufbereitung der Daten besteht zunächst in der Formatierung der Datenformate. Weiterhin wird anhand der vorgegebenen Gewichtungen die Wirtschaftlichkeitsmaßzahl berechnet und die „Komponente/Typ“ Datenbasis durch diese ergänzt. Den Einbauorten werden ihre Randbedingungen zugeordnet.

Durch diesen Block wird ein schneller Zugriff der Optimierungsalgorithmen auf die benötigten Daten möglich.

### **6.3 Definition des Systems**

Im dritten Block, der Systemdefinition, wird die zu optimierende Applikation mit allen zu beachtenden Randbedingungen definiert.

Hier werden der Applikation ihre Komponenten mit den geplanten Einbauorten zugeordnet und die Zielwerte (Ausfallrate, Gewichtung) zugeordnet. Diese sind abhängig von der Art der Applikation (Ist sie sicherheitsrelevant?), ihrem Status (In welcher Entwicklungsphase befindet sie sich? Wurde ihre Topologie geändert?) und auch dem Zielfahrzeug (Einbauorte). Es existieren noch beliebig viele weitere, die Komposition des Zielsystems beeinflussende Randbedingungen.

Nach der Systemkomposition werden diese Daten wiederum in einer Datenbasis hinterlegt, die einen schnellen Zugriff auf die Daten ermöglicht.

#### **6.4 Ausführen der Algorithmen**

Nachdem alle benötigten Daten aufbereitet bzw. berechnet sind, werden nun die Algorithmen zur Optimierung und Ermittlung der vagen Daten ausgeführt.

Zunächst muss das optimale System ermittelt werden. Der verwendete Optimierungsalgorithmus stellt den wesentlichen Unterschied der drei verschiedenen Tools dar.

Nachdem die optimale Lösung des Problems gefunden wurde, kann die Bewertung der Entwicklungsunsicherheit durchgeführt werden. Da sich die Methode der Genetischen Algorithmen am besten bewährt hat, wurde der vage Daten Algorithmus ausschließlich in diesem Tool implementiert. Er bewertet die ermittelte optimale Lösung. Prinzipiell ist dieser Algorithmus jedoch in identischer Form auch bei den anderen beiden Methoden einsetzbar.

#### **6.5 Darstellung der Ergebnisse**

Beide ermittelten Lösungen (Optimum, Entwicklungsunsicherheit) werden abschließend dargestellt. Ein Vergleich mit anderen Lösungen wird durch die Möglichkeit zum Export der Ergebnisse möglich.

## 7 Beispielapplikation

In Kapitel 4 wurden verschiedene Optimierungsmethoden auf ihre Anwendbarkeit beim vorliegenden Problem untersucht. Die für das vorliegende Problem am besten verwendbare Methode ist demnach der Genetische Algorithmus. In diesem Kapitel soll er nun an einem Beispiel verifiziert werden. Als Beispielapplikation wurde eine elektromechanische Bremse (EMB) gewählt.

Genetische Algorithmen sind zufallsbasiert. Sie können bei identischen Abbruchkriterien und verschiedenen Optimierungsdurchläufen unterschiedliche Ergebnisse bringen. Zur Bewertung der Qualität der Optimierung und zur Dimensionierung der Abbruchkriterien ist der Vergleich mit Referenzwerten nötig. Die optimale Lösung muss also bekannt sein.

Für Probleme begrenzter Komplexität liefert der Graphenalgorithmus immer die optimale Lösung. Daher wurde der Lösungsraum dieses Beispiels bewusst gerade so groß gehalten, dass mit den zur Verfügung stehenden Ressourcen auch der Graphenalgorithmus angewendet werden konnte. Auf diese Art wurde(n) die Referenzlösung(en) gefunden.

Bei komplexeren Problemen liegen diese Referenzwerte nicht mehr vor. Durch die zufallsbasierte Struktur Genetischer Algorithmen wird das optimale Ergebnis nur mit einer gewissen Wahrscheinlichkeit gefunden. Daher wurde in diesem Kapitel ein adaptives Abbruchkriterium entwickelt, durch das mit einer sehr hohen Wahrscheinlichkeit, bei beliebig komplexen Problemen, das Optimum gefunden wird.

### 7.1 Die elektromechanische Bremse (EMB)

Bereits das erste Automobil, der 1886 von Karl Benz entwickelte „Patent-Motorwagen“, wurde über ein mechanisches Bremssystem gebremst. Hierbei handelte es sich um eine auf eine Vorgelege-Riemenscheibe einwirkende Handbremse.

Seitdem wurden die Fahrzeuge und natürlich auch die Bremssysteme kontinuierlich weiterentwickelt. Der Stand der Technik bei PKW ist ein hydraulisches Zweikreis-Bremssystem. Bei diesem wird die durch den Fahrer aufgebrachte Pedalkraft verstärkt und wirkt direkt über die Bremsbeläge auf die Brems scheiben ein. Der Fahrer hat einen direkten mechanischen Durchgriff vom Bremspedal bis zum bremsenden Rad.

Im Jahr 2001 wurde erstmals eine Pkw-Bremse ohne mechanischen Durchgriff in Serie gebaut (Mercedes SLK). Bei dieser elektrohydraulischen Bremse (EHB) wird der Bremswunsch des Fahrers im Bremspedal in elektrische Signale umgesetzt und an die entsprechenden Steuergeräte, die im Fahrzeug beliebig verteilt sein können, gesendet. Die Steuergeräte kennen den Zustand und die Lage des gesamten Fahrzeugs und werten viele weitere Signale aus. Sie steuern letztlich die Hydraulik-Ventile und den Bremsbooster, der zur Erzeugung der hydraulischen Bremskraft dient.

Die Bremswunsch-Übermittlung wird aus dem hydraulischen in den elektrischen Signalbereich verlagert. Es ergeben sich viele Vorteile, beispielsweise bei der Systemauslegung und -anordnung, bei der Signalmanipulation oder auch der Funktionsergänzung.

Zur Zeit wird die elektromechanische Bremse (EMB) entwickelt. Ähnlich wie bei der EHB, hat die EMB keinen mechanischen Durchgriff der von dem Fahrer erzeugten Bremskraft auf die Räder. Im Unterschied zur EHB ist die Art der Bremsenergie

jedoch nicht hydraulisch, sondern elektrisch. Die Energie kann durch Kabel übertragen werden und treibt letztlich elektromechanische Bremsaktoren an. Bei diesen werden Elektromotoren betrieben, die über ein Getriebe die Bremsbacken bewegen und so die nötige Verzögerung der Räder bewirken. Durch den Verzicht auf die Hydraulik, wird dieses System auch „trockenes Bremssystem“ genannt.

Die EMB hat alle Vorzüge der EHB hinsichtlich der Systemanordnung, der Signalübertragung und der einfachen Funktionsumsetzbarkeit. Außerdem lässt sich bei ihr der vorhandene Bauraum noch besser nutzen (Steigerung der Wirtschaftlichkeit). Da keine Hydraulikflüssigkeit verwendet wird, ist sie umweltverträglicher als die EHB. Leider ist die Erzeugung und Speicherung von ausreichend (elektrischer) Energie aufwendiger als bei der Hydraulik. Hier hat die EMB gegenüber der EHB wirtschaftliche Nachteile.

Die Abbildung 27 zeigt das vereinfachte Prinzip einer EMB. Die Daten- und die Energieleitungen sind nicht getrennt eingezeichnet, d. h. es handelt sich bei den dargestellten Verbindungen jeweils um Energie und Datenleitungen.

Da die Abbildung 27 lediglich eine prinzipielle Darstellung ist, sind auch keine Redundanzanforderungen oder Gesetzesvorschriften darin berücksichtigt. Aus Redundanzgründen könnte z. B. die Bereitstellung eines zweiten Energiespeichers nötig sein. Gesetzesvorschriften könnten die Umsetzung einer Zweikreis-Anordnung, ähnlich wie bei der hydraulischen Bremse, von Energie- und Datenleitungen fordern.

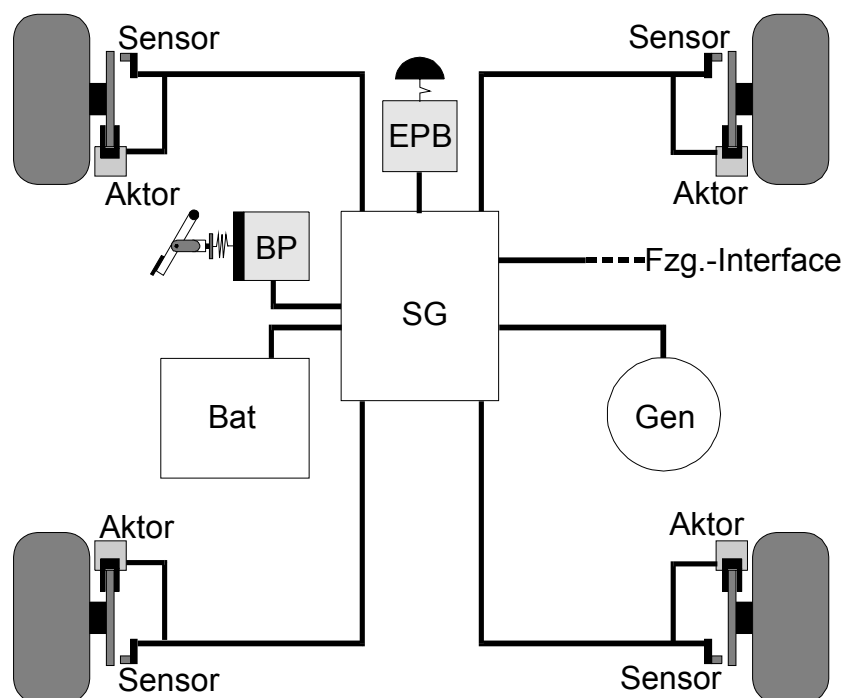


Abbildung 27: Vereinfachtes Schema einer EMB

Die Abkürzungen der Komponenten der EMB aus Abbildung 27 haben folgende Bedeutung (Tabelle 2):

<b>Abkürzung</b>	<b>Bedeutung</b>
Aktor	Elektromechanischer Bremsaktor, wandelt die elektrische Energie in mechanische Bremsenergie
Bat	Batterie, stellt den Energiespeicher des Systems dar für den Fehler- oder Hochlast-Fall
EPB	Elektronische Parkbremsbetätigung (Handbremse), wandelt die Parkbrems-Anforderung des Fahrers in elektrische Signale
Fzg.-Interface	Schnittstelle zum restlichen Fahrzeug-Netzwerk, Verschiedene Informationen werden hierüber ausgetauscht (z. B. Diagnose, ausgelagerte Sensorwerte)
Gen	Generator, angetrieben vom Verbrennungsmotor des Fahrzeugs und erzeugt die elektrische (Brems-) Energie
Sensor	verschiedene Radsensoren, erfassen z. B. die Raddrehzahl, die Temperaturen, den Verschleiß
SG	Steuergerät, Auswertung von Signalen und Umsetzung von Funktionen

**Tabelle 2: Abkürzungen und Bedeutungen der EMB-Komponenten aus Abbildung 27**

Die Energie für das System wird durch den Generator (Gen) bereitgestellt. Fällt er aus, muss mindestens ein Speicher (Bat) vorgesehen werden. Dieser versorgt dann das System und ermöglicht so die Berücksichtigung von Ausfallszenarien, also Reaktionen auf Fehler in der Applikation.

Im Bremspedal-Modul (BP) wird der Fahrerwunsch erfasst und über Datenleitungen an das Steuergerät weitergegeben. Dieses wertet die zur Verfügung stehenden Daten der EMB und die über das Interface bereitgestellten Fahrzeug-Daten aus und koordiniert alle Funktionen. Im Modul Elektronische Parkbremsbetätigung (EPB) wird der Wunsch der Betätigung der Feststellbremse (Handbremse) erfasst. Das Steuergerät leitet diesen zu den Bremsaktoren an die Räder weiter. Dort wird ein Feststellmechanismus ausgelöst, der die Bremsen im angezogenen Zustand arretiert. Diese Arretierung hält auch im stromlosen Zustand.

An den Rädern sind neben den Aktoren diverse Sensoren vorhanden. Sie erfassen z. B. die Raddrehzahl oder den Verschleißzustand der Bremsen und deren Temperaturen. Auch der Verschleiß und der Zustand der Aktoren kann überwacht werden.

In Abbildung 28 ist das Prinzip einer weiteren EMB dargestellt, die als Grundlage für die nachfolgenden Untersuchungen diente. Es entstammt einer Analyse des in [OKC01] vorgestellten EU-Projektes „BRAKE“ und enthält mehr bzw. andere Komponenten als das in Abbildung 27 dargestellte System. Die Funktion ist identisch, die Zuverlässigkeit und Sicherheit des Systems ist jedoch höher.

Ein Vergleich der Komponenten der beiden EMB-Systeme aus Abbildung 27 und Abbildung 28 zeigt Unterschiede. Eine wie in der Abbildung 27 dargestellte EMB erfüllt nicht die Anforderungen an die Ausfallsicherheit und Verfügbarkeit an die Bremse. Um die Erfüllung dieser Anforderungen nachzuweisen, bzw. eventuelle Schwachstellen der Systeme zu analysieren und zu verbessern, werden in der Automobilindustrie im Entwicklungsprozess für (sicherheitsrelevante) Systeme

verschiedene Sicherheits- und Zuverlässigkeitsanalysemethoden eingesetzt. Gängige Methoden sind hier neben der Fehlerbaum-Analyse (FBA) auch die Fehler-Möglichkeiten- und Einfluss-Analyse (FMEA) und die Markov-Analyse [Bin01].

Ist eine Schwachstelle analysiert, gibt es verschiedene Möglichkeiten das System zu verbessern. Oft wird die Auslegung des Systems verändert:

- Fehlerverursachende Komponenten können doppelt ausgeführt und parallel angeordnet werden (Redundanz).
- Die Qualität der Komponenten kann erhöht werden.

Die unterschiedliche Anzahl und Art der Komponenten der EMB-System aus Abbildung 27 und dem Beispiel aus [OKC01] können Ergebnisse durchgeführter Sicherheits- und Zuverlässigkeitsanalysen sein.

In dieser Arbeit soll nachfolgend eine EMB nach dem in Abbildung 28 dargestellten Schema betrachtet werden.

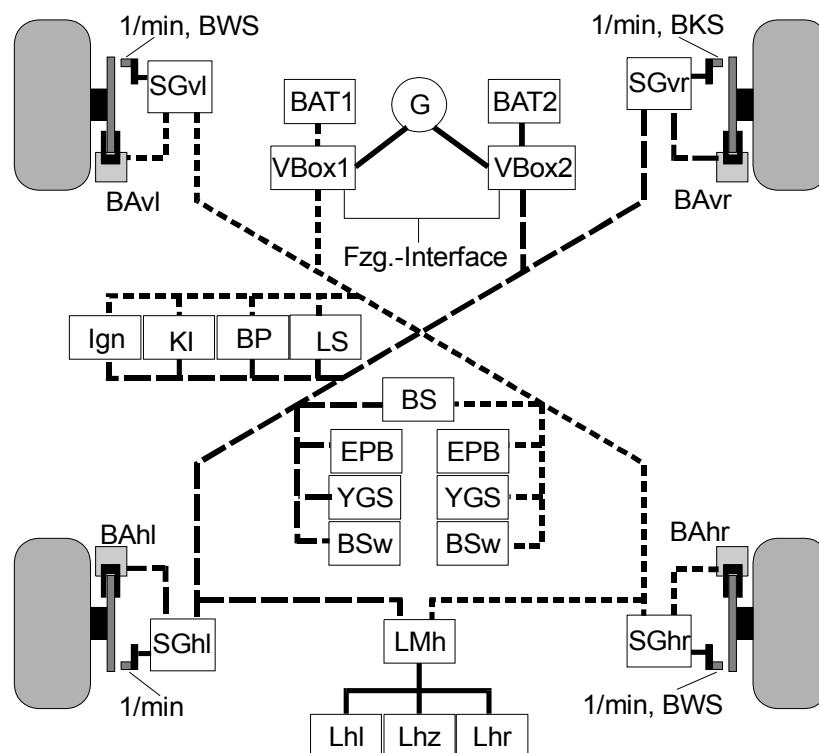


Abbildung 28: Schema einer EMB

Die Abkürzungen der Komponenten der EMB aus Abbildung 28 haben folgende Bedeutung (Tabelle 3):

<b>Abkürzung</b>	<b>Bedeutung</b>
1/min	Drehratensensor
BAT1, BAT2	Batterien 42
BAvr, BAhr, BAvl, BAhl	Bremsaktoren und deren Lagen (z. B. vl = vorne links)
BKS	Bremsbacken-Kraft-Sensor
VBOX1, VBOX2	Energieverteilung und -überwachung
BP	Bremspedal
BS	Beschleunigungs-Sensor
BSw	Bremsschalter (brake switch)
BWS	Bremsbacken-Weg-Sensor
EPB	Elektronische Parkbremse
G	Generator
Ign	Zündung (Ignition)
KI	Kombi-Instrument
Lhr, Lhz, Lhl	Lampe (Lage: hinten links, zentral, rechts )
LMh	Lichtmodul (hinten)
LS	Lenkwinkel-Sensor
Fzg.-Interface	Schnittstelle zum restlichen Fahrzeug-Netzwerk
YGS	Gierraten-Sensor (senkrechte Fahrzeug-Achse)

**Tabelle 3: Abkürzungen und Bedeutungen der EMB-Komponenten aus Abbildung 28**

Die Verbindungen im betrachteten EMB-Schema sind durch (gestrichelte und durchgezogene) Linien dargestellt. Sie stehen sowohl für Daten- als auch für Energieleitungen. In diesem EMB-Schema wurde das Redundanzprinzip einer hydraulischen Zweikreis-Bremse nachempfunden. Jeder der beiden Kreise betreibt (weitestgehend) autark jeweils eine „Diagonale“ des Fahrzeugs. Die beiden Kreise sind in der Abbildung 28 durch unterschiedliche gestrichelte Linien dargestellt.

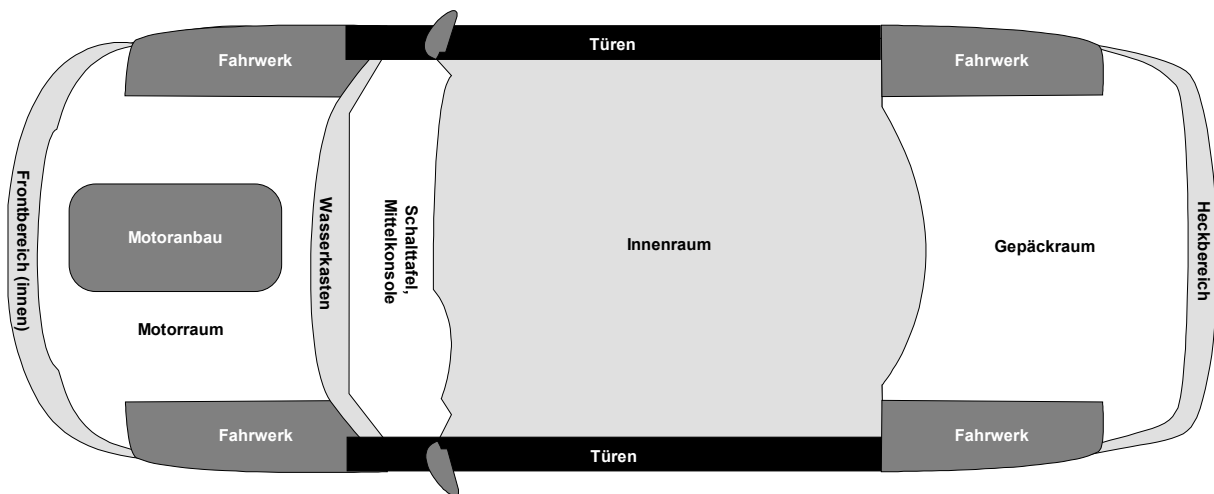
Ausnahmen des Diagonal-Schemas sind die Energieversorgung durch den Generator, der beide Kreise speist, und die hinteren Lampen, die aus einem durch beide Kreise gesteuertes Lichtmodul betrieben werden.

## **7.2 Einbauorte im Fahrzeug**

Die dargestellten Komponenten werden verschiedenen Einbauorten zugeordnet. Nach [Vol00] lässt sich das Fahrzeug in 15 verschiedene Anforderungsbereiche aufteilen. Für diese Arbeit sind nur 10 der dort genannten Anforderungsbereiche relevant.



Die Abbildung 29 zeigt die 14 relevanten Einbauorte mit den 10 verschiedenen Anforderungsbereichen.



**Abbildung 29: Relevante Einbauorte**

Jedem Einbauort wird ein Anforderungsbereich zugeordnet. Unterschiedliche Einbauorte können aber identische Anforderungen haben (hier: 4 Fahrwerk-Einbauorte mit identischen Anforderungen).

Für jeden dieser Bereiche ist definiert, welche Randbedingungen die dort einzubauenden Komponenten erfüllen müssen. Folgende Anforderungsarten müssen nach [Vol00] mindestens berücksichtigt werden:

- Schwingungsbelastungsschärfegrad,
- Betriebstemperaturbereich,
- Lagertemperaturbereich,
- Umweltbeständigkeiten nach DIN EN 60068-2-30Db (innen) und DIN 50 021-SS (außen),
- Beständigkeit gegen Reagenzien (z. B. Kraftstoffe, Säuren, Öle, Getränke usw.),
- Staub- und Spritzwasserdichtheit nach DIN 40 050-9.

In der Abbildung 30 sind die 35 wesentlichen Komponenten der Beispiel-EMB aus [OKC01] an ihren denkbaren Einbauorten dargestellt. Diese sind auch für die nachfolgende Optimierung vorgesehen.

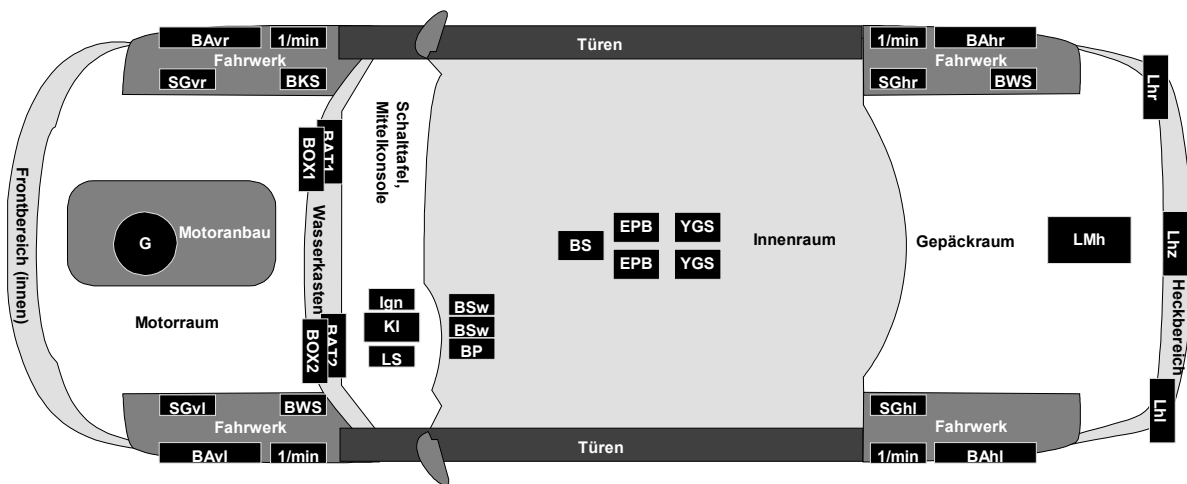


Abbildung 30: Lagegerechte Darstellung der Komponenten einer EMB

Es enthält 17 unterschiedliche Komponenten, die sich auf die insgesamt 10 Einbauorte mit 7 verschiedenen Anforderungsbereichen verteilen. Berücksichtigt man zusätzlich zu diesen Komponenten die Verkabelung mit allen Steckern und Kabeln für die Kommunikation und die Übertragung der Energie, so verteilen sich die insgesamt 134 resultierenden Komponenten auf 14 Einbauorte in 10 unterschiedlichen Anforderungsbereichen. Vereinfachend wurde angenommen, dass jede Komponente der Abbildung 30 jeweils nur eine Stecker/Buchse-Kombination als verbindendes Element hat. Die Abbildung 31 zeigt das sich ergebende Verkabelungsschema.

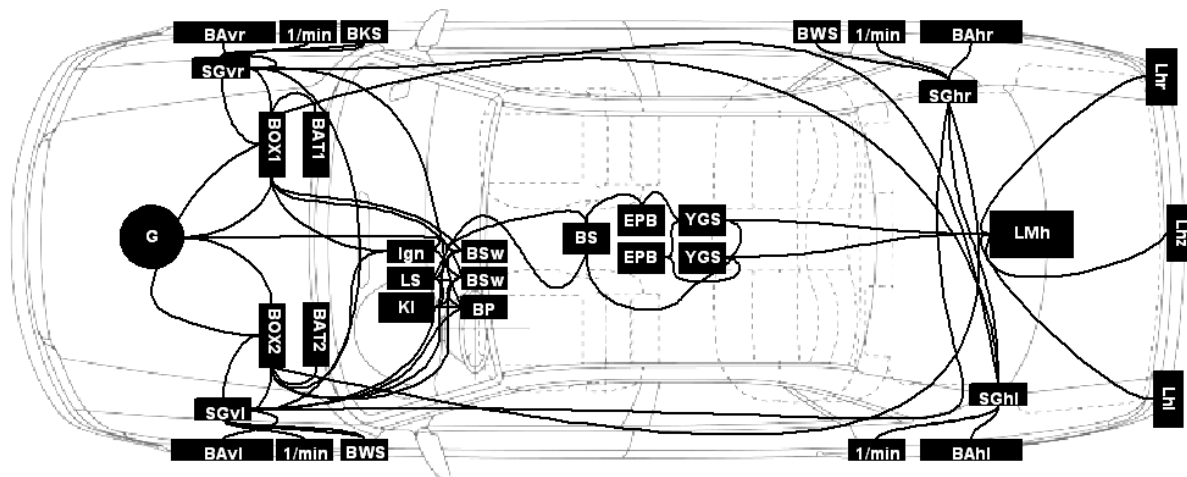


Abbildung 31: Verkabelung der EMB

Elektromechanische Bremssysteme sind zur Zeit noch Forschungsgegenstand. Alle realisierten Fahrzeuge sind Prototypen. Es liegen daher keine exakten Daten für die Wirtschaftlichkeitszahlen und Ausfallraten der Typen vor. Somit beruhen die Daten der für die Betrachtung verwendeten Komponente/Typ-Datenbasis weitestgehend auf Schätzungen.

Die Ausfallraten der betrachteten Typen wurden zum Teil aus den im Militärhandbuch [Dod91] dargestellten Anforderungen generiert. Es wurde jedoch

versucht, mit diesen Schätzungen die möglichen Relationen der späteren realen Daten zu wahren. Der Rückschluss von den in dieser Arbeit gefundenen Ergebnissen auf die späteren Daten der realisierten Applikation ist aber nicht möglich. Für den Nachweis der Anwendbarkeit der Methode ist die Betrachtung der Relationen jedoch ausreichend.

Die Datenbasis der Komponenten/Typen ist im Abschnitt 13.1 des Anhangs dargestellt. Es ergeben sich für die betrachtete EMB 442368 mögliche Lösungen. Aus diesen soll die wirtschaftlichste Variante im Bezug auf die gegebene Soll-Ausfallrate  $\lambda_{\text{soll}}$  gefunden werden.

### 7.3 Definition der maximal zulässigen Ausfallrate

Einer der Optimierungsparameter ist die maximal zulässige Ausfallrate. Auch für die Beispiel-EMB muss diese angegeben werden. Die Randbedingungen der nachfolgenden Herleitung wurden willkürlich gewählt. Die berechnete Ausfallrate ist deutlich größer, als bei einer möglichen Serienanwendung. Es soll an dieser Stelle jedoch nur die Vorgehensweise der Bestimmung der Soll-Ausfallraten dargestellt werden. Für die Betrachtung der Beispiel-EMB wird die Ausfallrate folgendermaßen hergeleitet:

Es sei angenommen, dass ein Fahrzeug pro Jahr insgesamt 200 Betriebsstunden erreicht. Dieses Fahrzeug soll ein Lebensalter von 15 Jahren erreichen, so dass sich insgesamt 3000 Betriebsstunden je Fahrzeugleben ergeben. In dieser Zeit darf die Bremsanlage des Fahrzeugs nicht ausfallen. Das bedeutet, eine EMB sollte eine Ausfallrate von

$$\lambda_{\text{soll}} < \frac{1 \text{ Ausfall}}{3000 \text{ Stunden}} = 3,33 \cdot 10^{-4} \frac{1}{h} \quad (52)$$

nicht überschreiten. Nicht jeder Ausfall bewirkt einen sicherheitskritischen Zustand, dieser sollte jedoch erkannt werden. Der Anteil der sicherheitskritischen Ausfälle ist sehr gering, da durch die erwähnten Redundanzprinzipien und eine ständige Eigendiagnose während des Betriebs des Systems solche Ausfälle weitestgehend abgefangen werden. Unter der Voraussetzung der regelmäßigen Wartung der EMB können Verschleißteile frühzeitig ausgetauscht und somit die tatsächliche Ausfallrate deutlich verbessert werden. Mögliche Verschleißausfälle, wie z. B. die Abnutzung der Bremscheiben, können vermieden werden.

Die Wartung und Eigendiagnose erlaubt es daher, die zu verwendenden Komponenten auf eine etwas höhere Ausfallrate des Gesamtsystems auszulegen. Für die nachfolgenden Untersuchungen sei daher als Vorgabe für die maximal zulässige Ausfallrate im zuverlässigkeitstechnischen Sinne folgender Wert angenommen

$$\lambda_{\text{soll}} = 8 \cdot 10^{-4} [\text{Ausfälle} / h]. \quad (53)$$

Dieser Wert gilt für das nicht gewartete und eigen-diagnostizierte System. Die tatsächliche Ausfallrate ist aber deutlich niedriger, so dass das System während seiner angenommenen Lebensdauer statistisch gesehen nicht ausfällt

$$\lambda_{\text{tatsächlich}} \ll \lambda_{\text{soll}}. \quad (54)$$

Bei dieser Auslegung ist eine Unsicherheit der Daten gegenüber den tatsächlichen Bedingungen berücksichtigt.

### 7.4 Generierung der Referenzlösung(en)

Um eine Bewertung der mit Genetischen Algorithmen gefundenen Lösungen vornehmen zu können, ist es nötig, die optimalen Lösungen zu kennen. Diese lassen sich durch die Anwendung des in Kapitel 4.1 dargestellten Graphenalgorithmus generieren.

Die Datenbasis bei den Untersuchungen zur betrachteten EMB ist kleiner, als bei einem tatsächlichen System. Der Lösungsraum wurde so in seiner Komplexität beschränkt, denn der vollständige Graph sollte berechenbar bleiben.

Für das Beispielsystem der EMB benötigte, bei der zugrundegelegten Datenbasis, ein schneller Rechner mit dem Graphenalgorithmus einige Minuten Berechnungszeit. Bei verdoppelter Lösungsraumgröße sollte sich auch die doppelte Berechnungszeit ergeben usw.. Eine Verdoppelung des Lösungsraums resultiert bereits bei der Ergänzung des Systems um eine Komponente mit nur 2 verschiedenen Typen (vgl. Gleichung (9)).

Bei diesem Beispiel wurde von nur einem Stecker je Komponente ausgegangen. Ebenfalls wurden redundant verlegte Kabel nicht berücksichtigt. Da eine Ergänzung dieser EMB mit weiteren Komponenten und Typen beliebig vorstellbar ist, wird die Praxisrelevanz des Problems der Größe des Lösungsraumes schnell deutlich.

Zur Bewertung des Genetischen Algorithmus werden durch Anwendung des Graphenalgorithmus Referenzlösungen generiert. Es wurden insgesamt 4 Optimierungsziele definiert. Diese sind das Gewicht, die Kosten, das Volumen und bei der Gleichgewichtung dieser Parameter, die Wirtschaftlichkeitsmaßzahl.

Abhängig vom Ziel der Optimierung ergeben sich für die jeweils optimale Variante der EMB folgende Werte (Tabelle 4):

	$\lambda$ [1/h]	W	Gewicht [Kg]	Kosten [€]	Volumen [ccm]	$g_k$	$g_G$	$g_v$
<b>Lösung 1</b>	6,77E-04	0,808620	50,32	<b>2080,60</b>	77895,79	1	0	0
<b>Lösung 2</b>	5,78E-04	0,941321	<b>49,32</b>	2153,40	134391,79	0	1	0
<b>Lösung 3</b>	6,32E-04	0,812014	50,67	2095,40	<b>77429,79</b>	0	0	1
<b>Lösung 4</b>	6,77E-04	<b>0,808620</b>	50,32	2080,60	77895,79	1/3	1/3	1/3

Tabelle 4: Optimale Lösungen für die Forderung aus Gleichung (53)

Die optimale Lösung wird jeweils anhand der geringsten Wirtschaftlichkeitsmaßzahl ermittelt. Die letzten drei Spalten der Tabelle 4 geben die gewählten Gewichtungen (Gleichung (1)) der Wirtschaftlichkeitsmaßzahl, also das Ziel der Optimierung an.

Die Lösung 1 ist somit die kostengünstigste, die Lösung 2 die leichteste und die Lösung 3 die raumsparendste Variante. Mit der Lösung 4 wurde in einem vierten Optimierungsdurchgang die wirtschaftlichste Variante gefunden, unter der Vorgabe der identischen Gewichtung aller drei Optimierungsparameter.

Verwendet man für jede Komponente den jeweils billigsten Typ, so ergibt sich die nachfolgend dargestellte Lösung 5 (Tabelle 5).

	$\lambda$ [1/h]	W	Gewicht [Kg]	Kosten [€]	Volumen [ccm]
<b>Lösung 5</b>	1,08E-03	0,91762	50,72	2000,60	91895,79

**Tabelle 5: Billigste Lösung, unabhängig von der Forderung aus Gleichung (53)**

Diese Variante überschreitet die in Gleichung (53) geforderte Soll-Ausfallrate. Sie würde weniger Kosten als die Lösungen aus Tabelle 4, jedoch etwas mehr wiegen und deutlich mehr Bauraum benötigen. Da die Ausfallrate aber eine harte Grenze darstellt, die nicht überschritten werden darf, würden solche Systeme voraussichtlich keine Zulassung bekommen.

Die Ausfallraten der optimierten Lösungen (Tabelle 4) unterschreiten die Forderung aus Gleichung (53) deutlich. Die in Tabelle 6 dargestellte Lösung 6 ist diejenige, deren Ausfallrate die Forderung aus Gleichung (53) „gerade eben“ unterschreitet. Man erkennt, dass dies dennoch nicht die wirtschaftlichste Lösung ist (Tabelle 6).

	$\lambda$ [1/h]	W	Gewicht [Kg]	Kosten [€]	Volumen [ccm]
<b>Lösung 6</b>	7,97E-04	0,902971	50,67	2087,70	118153,79

**Tabelle 6: Lösung, deren Ausfallrate die Forderung aus Gleichung (53) am geringsten unterschreitet**

## 8 Umsetzung und Erweiterung des Genetischen Algorithmus

Die Effizienz der Genetischen Algorithmen ist abhängig von deren Auslegung. In Abschnitt 4.3 wurden verschiedene Prinzipien der Genmanipulation vorgestellt. Bei der Umsetzung in dieser Arbeit hat es sich als sinnvoll erwiesen, jeweils mehr Individuen zu mutieren, zu kreuzen und zu gebären als letztlich übernommen werden.

Durch die Anwendung jedes der Prinzipien Mutation, Kreuzung oder Geburt können beliebig viele Individuen erstellt werden. Würden alle Individuen übernommen werden, so wüchse der Berechnungsaufwand in ungewollte Dimensionen. Dieses „Überangebot“ an neuen Individuen wird reduziert, indem in einer Vorauswahl nur die jeweils besten der durch das entsprechende Prinzip erstellten Individuen in die neue Generation übernommen werden.

Die Verringerung des Berechnungsaufwandes ist das Ziel der Reduzierung des Überangebots. Da nicht die absolut besten, sondern nur die besten der durch das betrachtete Prinzip erstellten Individuen übernommen werden, werden dennoch lokale Maxima übersprungen. Bei den Untersuchungen zeigte sich, dass sich das Optimum, trotz der eingebauten Zufallsmechanismen (Kreuzung, Mutation), nicht aus den sehr schlechten Individuen ergab. Diese wurden daher vernachlässigt.

Sollten die sehr schlechten Individuen dennoch nötig sein, so können sie in einer späteren Generation - wiederum zufällig - erneut erstellt werden. Dann gehört das entsprechende Individuum vielleicht zu den aktuell besten und wird übernommen.

Durch die Vorauswahl wird also der Berechnungsaufwand reduziert und dennoch das Optimum schnell erreicht.

### 8.1 Herleitung des Adaptiven Abbruchkriteriums

Eine der wesentlichen Erweiterungen der bisher bekannten Genetischen Algorithmen ist die Anwendung des nachfolgend hergeleiteten adaptiven, also sich an die Gegebenheiten anpassenden Abbruchkriteriums.

Die Qualität der Optimierung ist bei Genetischen Algorithmen abhängig von der Evolutionsdauer, also von der Anzahl der durchlaufenen Generationen. Mit steigender Evolutionsdauer sollte sich die Abweichung des aktuell wirtschaftlichsten Individuums vom Optimum verringern. Um dies darzustellen, wurden verschiedene Evolutionen an der Beispiel-EMB durchgeführt. Der Begriff „Evolution“ beschreibt einen Optimierungsvorgang, von der Initialisierung bis zur Erfüllung der Abbruchkriterien, an dessen Ende die (optimale) Lösung steht.

In Kapitel 4.3.5 wurde ein erstes Abbruchkriterium dargestellt. Für dieses gilt:

- Die Optimierung wird abgebrochen, sobald die als „Abbruchvorgabe“ definierte Anzahl an Generationen ohne Verbesserung des „besten“ Individuums durchlaufen wurde.

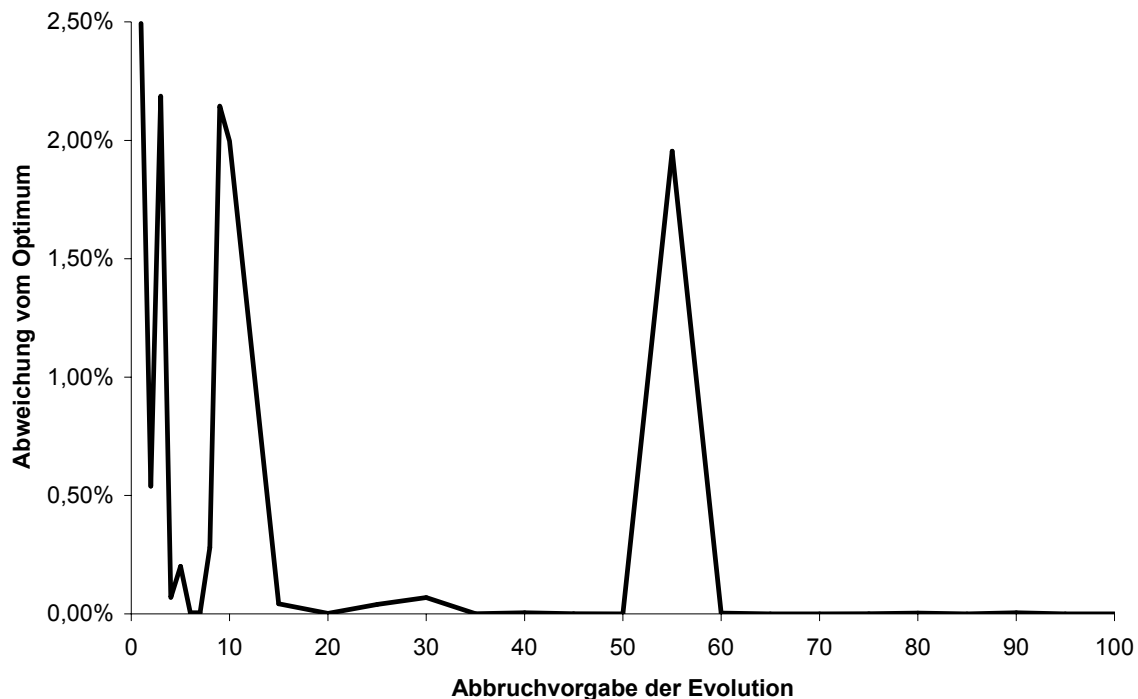
In einer ersten Messreihe galt dieses erste Abbruchkriterium. Hierbei wurde nach jeder Evolution die Abbruchvorgabe wie folgt erhöht:

- Für Abbruchvorgaben  $V_{ab} \leq 10$  war der Iterationsschritt gleich 1.
- Für alle anderen war der Iterationsschritt der Abbruchvorgabe gleich 5.

Es ergaben sich Abweichungen der Wirtschaftlichkeitsmaßzahlen von der im Kapitel 7.4 gefundenen Referenzlösung. Da alle Wirtschaftlichkeits-Parameter (vgl.

Gleichung (1)) gleich gewichtet wurden, diente die „Lösung 4“ (vgl. Kapitel 7.4, Tabelle 4) als Referenz. Die jeweilige Wirtschaftlichkeitsmaßzahl wurde somit auf den dort dargestellten Referenzwert  $W=0,808620$  des EMB-Systems bezogen.

Die jeweilige prozentuale Abweichung der Evolutionen bei den verschiedenen Abbruchvorgaben ist in der Abbildung 32 dargestellt. Je geringer die Abweichung ist, desto höher ist die Qualität der Lösung.



**Abbildung 32: Prozentuale Abweichungen der Wirtschaftlichkeitsmaßzahlen vom Optimum von jeweils einer Evolutionen je Abbruchvorgabe**

Es lässt sich die Tendenz der steigenden Qualität der gefundenen Individuen mit erhöhter Evolutionsdauer erkennen.

Es zeigt sich weiterhin, dass sich die Qualität der gefundenen Individuen bei der dargestellten Messreihe nicht stetig verbessert. Es gibt immer wieder Werte, die eine höhere Abweichung haben, als vergleichbare Evolutionen mit einer kleineren Abbruchvorgabe. Dies lässt sich mit dem zufallsbasierten Ansatz Genetischer Algorithmen erklären. Das optimale Ergebnis kann daher nur mit einer Wahrscheinlichkeit gefunden werden.

Aus der Methode der Genetischen Evolution ergibt sich, dass das beste Individuum einer Kind-Generation nicht schlechter als das beste Individuum der Eltern-Generation sein kann. Daher sollte sich die Wirtschaftlichkeitsmaßzahl des jeweils besten Individuums innerhalb einer Evolution dem Optimum asymptotisch annähern. Das Optimum würde dann bei sehr großer Evolutionsdauer ( $t \rightarrow \infty$ ) sicher gefunden werden. Für das Finden des Optimums steht jedoch nicht unendlich viel Zeit zur Verfügung.

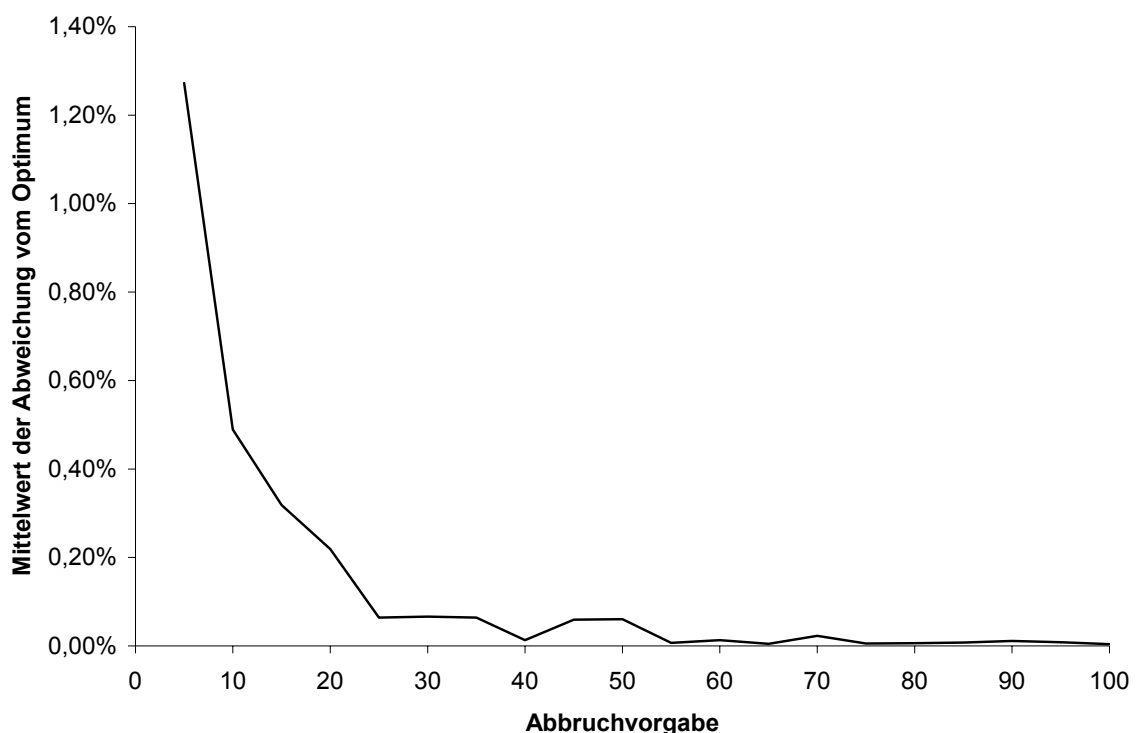
In dieser Arbeit wurde daher das adaptive Abbruchkriterium entwickelt. Durch dieses wird trotz begrenztem (Zeit-) Aufwand mit sehr hoher Wahrscheinlichkeit das Optimum gefunden. Dieses Abbruchkriterium wird nachfolgend hergeleitet und erläutert.

Zunächst wird dafür die Abhängigkeit der Optimallösung und der Ergebnisstruktur von der Anzahl der Evolutionslänge untersucht.

### 8.1.1 Zusammenhang der optimalen Lösung mit der Anzahl der Evolutionen

Im vorangegangenen Abschnitt wurde dargestellt, dass die Lösungsqualität von der Abbruchvorgabe abhängt (vgl. Abbildung 32). Da nur eine Stichprobe je Abbruchvorgabe ausgewertet wurde, könnte diese Tendenz zufällig entstanden sein. Daher wurden in einer weiteren Messreihe je Abbruchvorgabe 100 Evolutionen durchgeführt.

In der Abbildung 33 wurde von den je 100 sich ergebenden Individuen der arithmetische Mittelwert der Wirtschaftlichkeitsmaßzahl ausgewertet und deren prozentuale Abweichung dargestellt.



**Abbildung 33: Arithmetischer Mittelwert der Abweichung aus je 100 Evolutionen bei steigender Abbruchvorgabe**

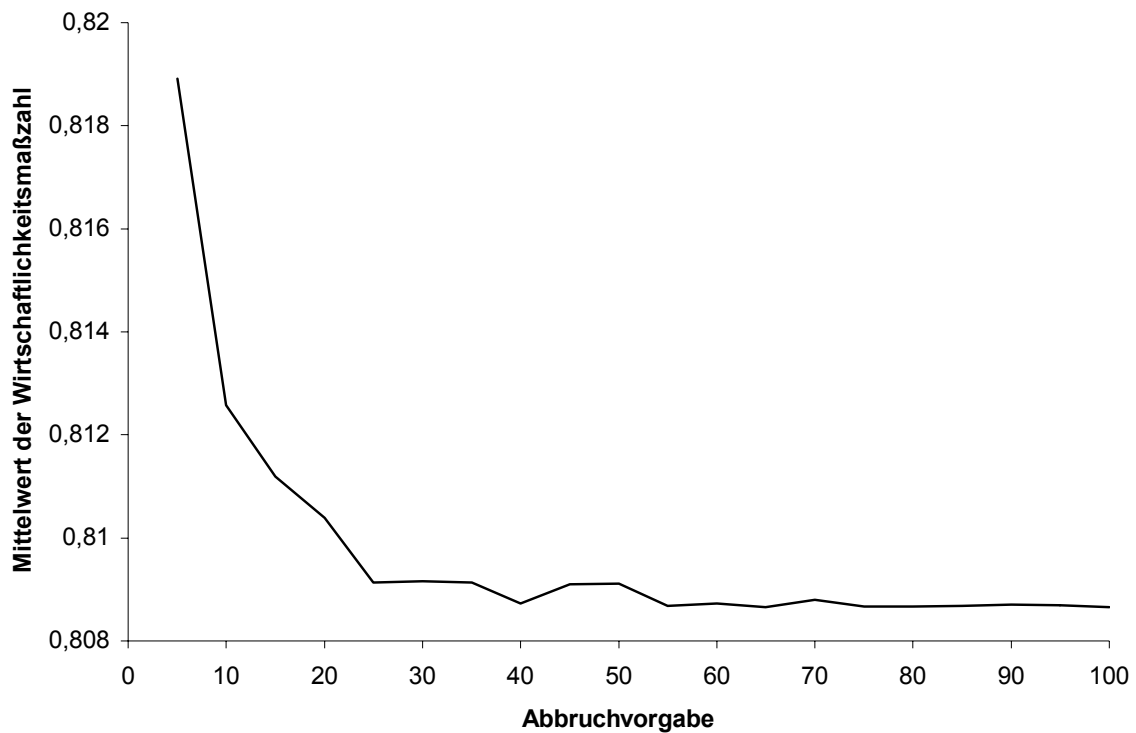
Es ergibt sich dieselbe Tendenz wie bereits in Abbildung 32 dargestellt. Mit steigender Abbruchvorgabe steigt auch die Wahrscheinlichkeit, das optimale Individuum zu finden. Der prozentuale Fehler nähert sich mit steigender Abbruchvorgabe asymptotisch, jedoch aufgrund des zufallsbasierten Ansatzes, nicht stetig fallend, dem Optimum (0%) an.

### 8.1.2 Ergebnisstruktur der Evolutionen

Für die Untersuchung der in Abbildung 33 dargestellten Ergebnisse wurden die Mittelwerte aller 100 Individuen je Abbruchvorgabe ausgewertet.



Diese Mittelwertdarstellung der Wirtschaftlichkeitsmaßzahlen zeigt die Abbildung 34.

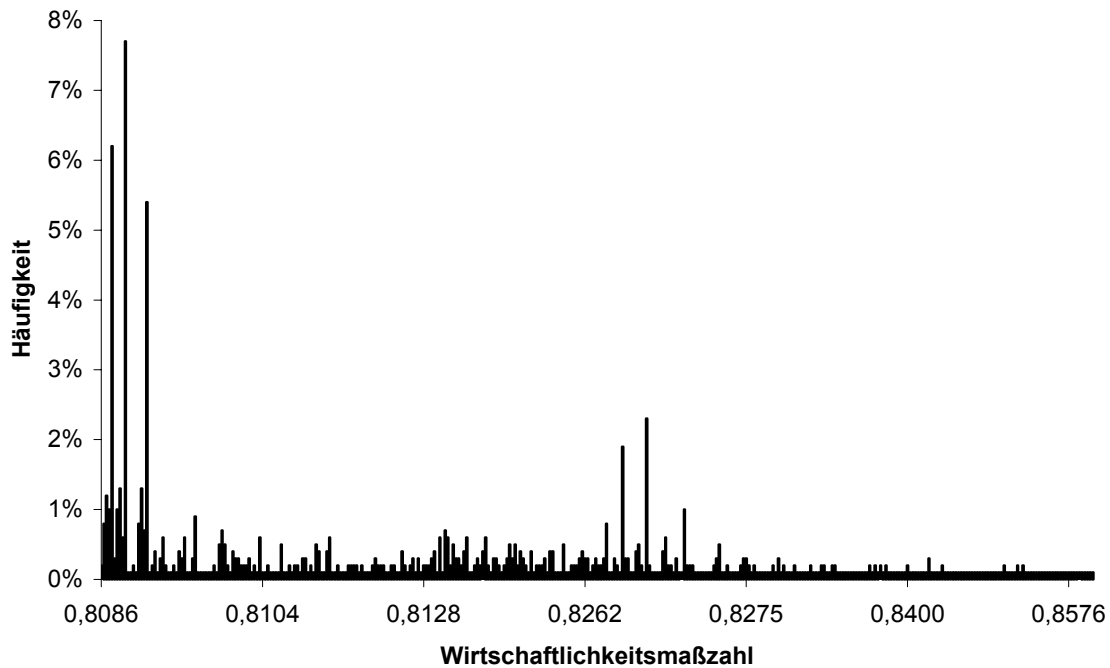


**Abbildung 34: Arithmetischer Mittelwert der Wirtschaftlichkeitsmaßzahlen aus 100 Evolutionen je Abbruchvorgabe**

Von Mittelwerten vieler Individuen ist jedoch kein Rückschluss auf eine diskrete Lösungen (Individuum) möglich. Der Lösungsraum ist nicht kontinuierlich. Ein Individuum muss aus diskreten Komponententypen kombiniert werden.

Nachfolgend wird daher die Struktur der berücksichtigten Individuen näher betrachtet.

Die Abbildung 35 zeigt die Struktur von insgesamt 1000 durchgeführten Evolutionen mit der Abbruchvorgabe  $V_{ab}=5$ .



**Abbildung 35: Häufigkeitsverteilung von 1000 Individuen mit der Abbruchvorgabe  $V_{ab}=5$**

Aus den 1000 durchgeführten Evolutionen resultierten insgesamt 370 verschiedene Individuen. Nur 5 Individuen wurden häufiger als bei 2% aller Evolutionen gefunden. Das Optimum mit der Wirtschaftlichkeitsmaßzahl  $W=0,80862$  wurde überhaupt nur zweimal gefunden. Das entspricht einem Anteil von 0,2%.

In der Abbildung 36 sind die Wirtschaftlichkeitsmaßzahlen der 100 Individuen dargestellt, die aus den Evolutionen mit der Abbruchvorgabe  $V_{ab}=20$  resultieren.

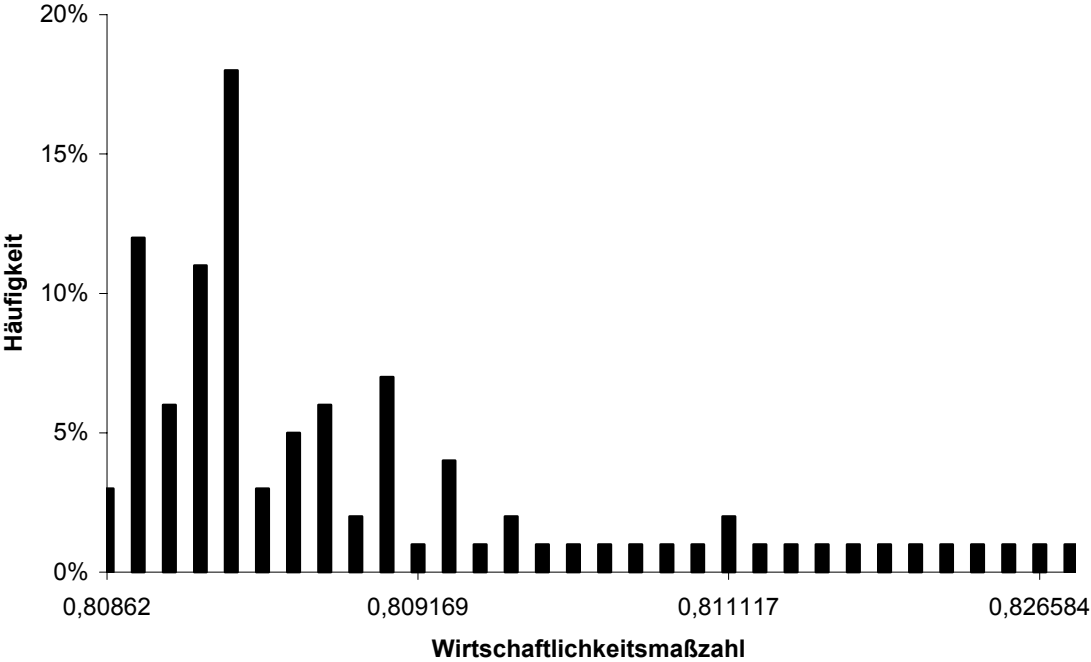
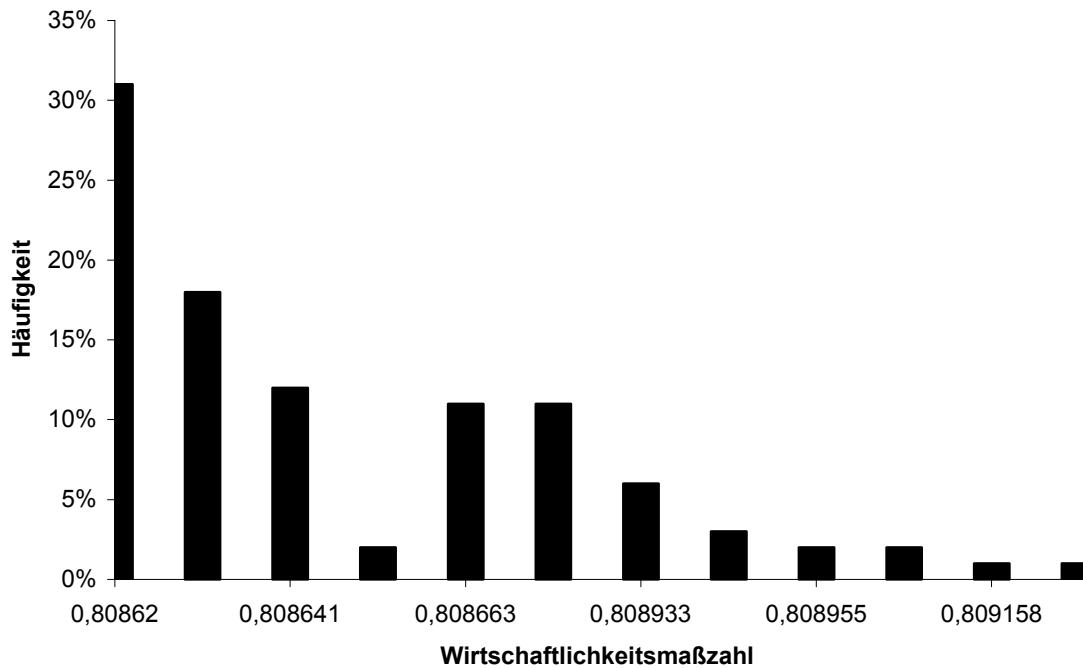


Abbildung 36: Häufigkeitsverteilung von 100 Individuen mit der Abbruchvorgabe  $V_{ab}=20$

Die Wirtschaftlichkeitsmaßzahl der optimalen Lösung liegt bei  $W=0,80862$ . Anhand der in Abbildung 36 dargestellten Lösungsstruktur lässt sich das jedoch nicht erkennen. Die Evolutionen ergaben insgesamt 32 verschiedene Lösungen, wobei die meisten (18 Evolutionen) ein Individuum mit der Wirtschaftlichkeitsmaßzahl  $W=0,808663$  als vermeintlich optimale Lösung fanden. Das optimale Individuum wurde hingegen nur 3 mal gefunden. Wenn die Vermutung, dass mit steigender Evolutionsdauer die Lösungsqualität der Individuen steigt stimmt, sollte sich diese Statistik mit der Erhöhung der Abbruchvorgabe zum Optimum hin verschieben.

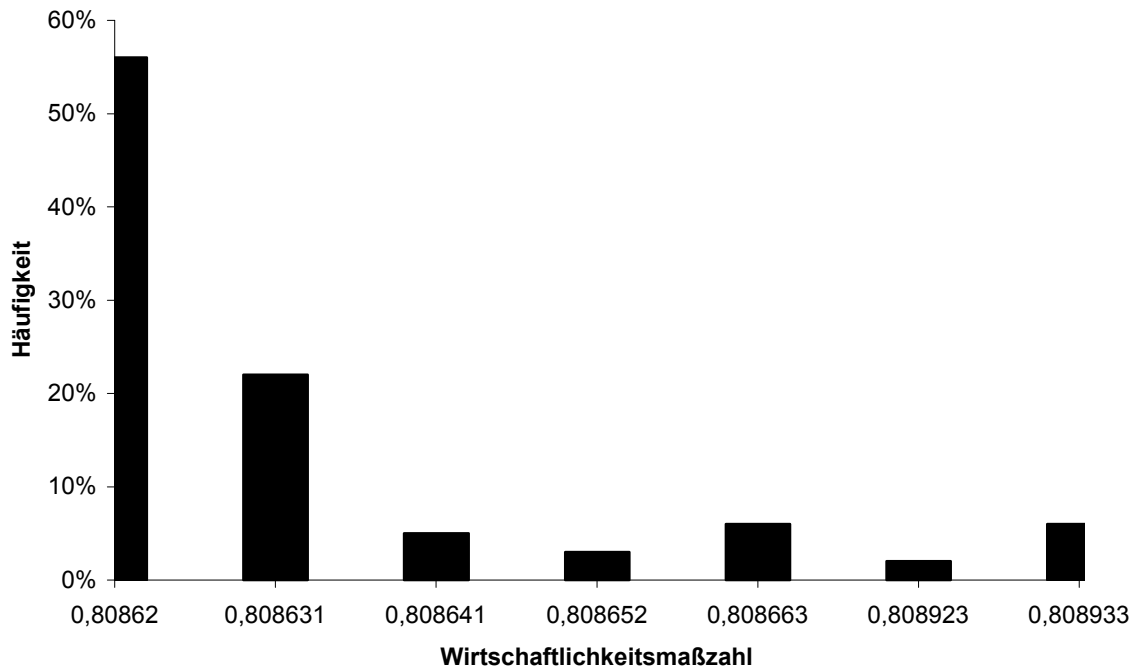
Bereits die Struktur der gefundenen Individuen bei der Abbruchvorgabe  $V_{ab}=60$  bestätigt die vermutete Tendenz (Abbildung 37).



**Abbildung 37: Häufigkeitsverteilung von 100 Individuen mit der Abbruchvorgabe  $V_{ab}=60$**

Bei 31 der 100 Evolutionen mit der Abbruchvorgabe  $V_{ab}=60$  wurde das optimale Individuum gefunden. Des Weiteren wurden nur 13 verschiedene Individuen als vermeintliches Optimum gefunden.

Noch deutlicher wird diese Tendenz bei den gefundenen Individuen mit der Abbruchvorgabe  $V_{ab}=100$ . Diese sind in der Abbildung 38 dargestellt.



**Abbildung 38: Häufigkeitsverteilung von 100 Individuen mit der Abbruchvorgabe  $V_{ab}=100$**

Es wurden insgesamt nur 7 verschiedene Individuen gefunden, und mit Abstand die meisten Evolutionen (55 v. H.) wurden beim absoluten Optimum abgebrochen.

Diese Untersuchungen zeigen, dass der entwickelte Genetische Algorithmus tatsächlich mit steigender Evolutionshäufigkeit und Abbruchvorgabe Individuen höherer Qualität findet.

Wie kann man sich aber sicher sein, das Optimum gefunden zu haben?

Das entwickelte adaptive Abbruchkriterium liefert hierzu gute Lösungen und wird nachfolgend erläutert.

### 8.1.3 Adaptives Abbruchkriterium

Bei der Anwendung einer Optimierungsmethode muss die optimale Lösung auch sicher, bzw. mit sehr hoher Wahrscheinlichkeit gefunden werden. Selbst bei sehr hoch gewählter Abbruchvorgabe (beim Beispiel wäre eine Abbruchvorgabe von  $V_{ab} \sim 1000$  als sehr hoch anzusehen) ist es bei einmaliger Anwendung der Evolution nicht sicher, auch das Optimum gefunden zu haben.

Aus den zuvor dargestellten Untersuchungen geht hervor, dass suboptimale Lösungen zwar möglich sind, jedoch mit steigender Abbruchvorgabe unwahrscheinlicher werden.

Die Idee des adaptiven Abbruchkriteriums greift diese Erkenntnis auf:

- Das optimale Individuum soll gefunden werden, indem nicht nur die aktuelle Evolution berücksichtigt, sondern mehrere Evolutionen durchführt und ausgewertet werden.

Es wurde daher ein Abbruch-Algorithmus entwickelt, der auf der Speicherung der Wirtschaftlichkeitsmaßzahlen der Individuen der letzten 10 Evolutionen bei einer Abbruchvorgabe basiert. Sie werden in einer Tabelle gespeichert, die hier W-Tabelle genannt werden soll. Die Optimierung wird abgebrochen, wenn alle Einträge in der W-Tabelle identisch sind. Sie sollten dann alle den Wert der Wirtschaftlichkeitsmaßzahl des optimalen Individuums haben.

Das adaptive Abbruchkriterium findet mit folgender Vorgehensweise das optimale Individuum:

Zunächst wird eine (kleine) Abbruchvorgabe (z. B.  $V_{ab}=10$ ) definiert. Mit dieser werden mehrere Evolutionen durchgeführt. Nach jeder Evolution wird die Wirtschaftlichkeitsmaßzahl des besten Individuums in die W-Tabelle geschrieben.

Falls sich die Einträge unterscheiden, wird der beste Eintrag der W-Tabelle gespeichert und diese vollständig gelöscht. Da mit der gewählten Abbruchvorgabe unterschiedliche Werte gefunden wurden, war diese offensichtlich zu klein dimensioniert. Die Abbruchvorgabe wird daher (z. B. um den Wert 10) erhöht.

Nun werden wieder Evolutionen durchgeführt.

Da eine „beste“ Wirtschaftlichkeitsmaßzahl bereits aus den vorangegangenen Evolutionen vorliegt, können alle Wirtschaftlichkeitsmaßzahlen der neu gefundenen, besten Individuen, mit diesem Referenzwert verglichen werden. Wird in einer folgenden Evolution ein besserer Wert gefunden, wird dieser als neuer Referenzwert gespeichert und die W-Tabelle gelöscht. Die Abbruchvorgabe wird nicht erhöht.

Wird ein schlechterer Wert gefunden, wird wiederum die W-Tabelle gelöscht und nun die Abbruchvorgabe erhöht.

Die Optimierung wird abgebrochen, wenn die Tabelle in Ihrer vordefinierten Größe (z. B. 10 Einträge) vollständig mit der besten Wirtschaftlichkeitsmaßzahl gefüllt ist. Berücksichtigt man die vorangegangenen Untersuchungen der Kapitel 8.1.1 und 8.1.2, ist es sehr unwahrscheinlich, dass 10 von 10 Evolutionen in einem lokalen Minimum enden und das Optimum übersehen wird.

Dieser Algorithmus für den Abbruch der Optimierung ist adaptiv, d.h. er passt die Optimierung dem Lösungsraum an. Für Probleme hoher Komplexität wird die Optimierung voraussichtlich mit einer höheren Abbruchvorgabe beendet, als bei kleinen Lösungsräumen.

Der entwickelte Algorithmus ist nachfolgend dargestellt:

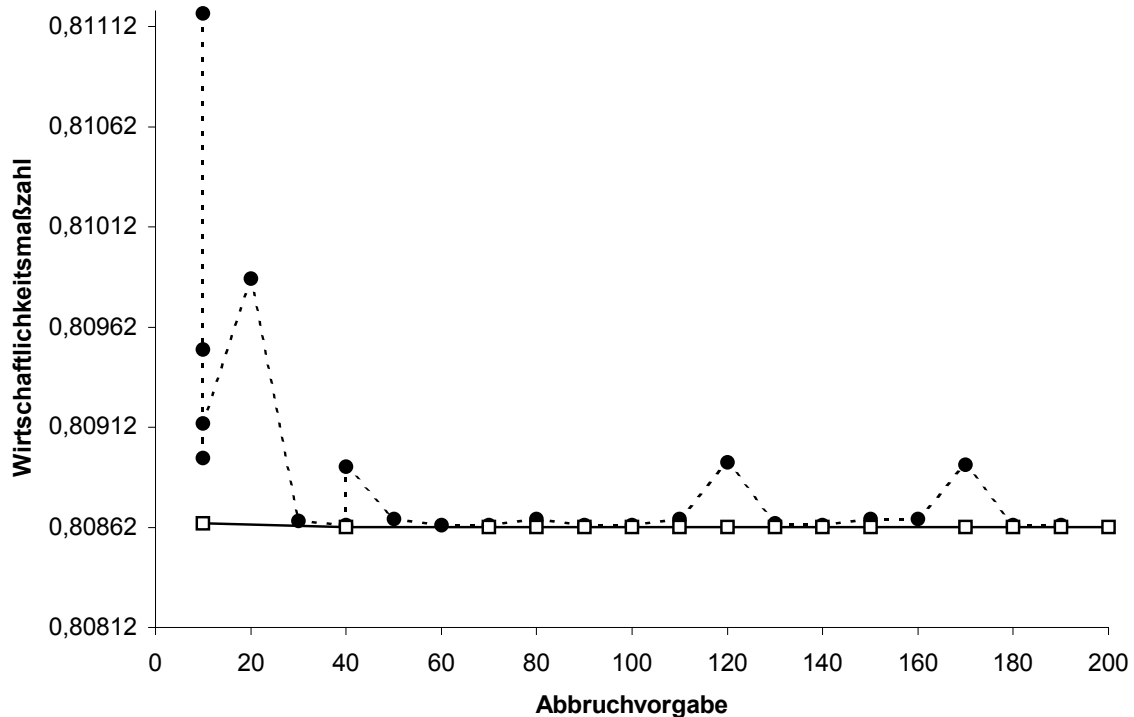
```

/////////////////////////////////////////////////////////////////
// Abv ist die Abbruchvorgabe der Evolution
// Wbest ist die aktuell beste Wirtschaftlichkeitsmaßzahl aller vorheriger Evolutionen
// W ist die Wirtschaftlichkeitsmaßzahl der aktuellen Evolution
// Wtable ist die Tabelle mit den gespeicherten Wirtschaftlichkeitsmaßzahlen
/////////////////////////////////////////////////////////////////
begin
{
  Abv = 10;
  Wbest = 1;
  while (size_of(Wtable) < 10)
  {
    Evolution;
    W = Wmz_of_best_Individuum;
    if ( ( W < Wbest ) OR ( W > Wbest ) )
    {
      if ( W < Wbest )
      {
        Wbest = W;
        if (size_of(Wtable) > 1)
        {
          loesche Wtable;
          Abv += 10;
        }
      }
      else
      {
        loesche Wtable;
        Abv += 10;
      }
    }
    if ( W == Wbest )
    {
      write_to_Wtable(W);
    }
  }
}
end;
/////////////////////////////////////////////////////////////////

```

Nach dem Abbruch der Optimierung liegt nur noch ein Individuum vor. Es ist das optimale Individuum des betrachteten Systems (hier: EMB). Mit diesem Algorithmus ergaben sich die im Anhang 13.5 dargestellten Lösungen.

Die Abbildung 39 zeigt die Struktur dieser Lösungen der Optimierung, die sich durch Anwendung des adaptiven Abbruchkriterium ergab.



**Abbildung 39: Verlauf der Optimierung mit dem adaptiven Abbruchkriterium**

In Abbildung 39 sind zwei verschiedene Verläufe dargestellt. Der gestrichelte Verlauf entspricht den Lösungen mit den jeweils suboptimalen, also vom aktuellen Optimum abweichenden Lösungen. Der durchgezogene Verlauf enthält die Lösungen, die zum jeweiligen Zeitpunkt die niedrigste Wirtschaftlichkeitsmaßzahl haben und der absolut optimalen Lösungen entsprechen könnten.

Die Optimierung begann mit der Abbruchvorgabe  $V_{ab}=10$ . Bei dieser ergaben sich 5 verschiedene Lösungen. Mit den ersten vier Evolutionen verbesserte sich die Wirtschaftlichkeitsmaßzahl der Lösung jedes mal. In der vierten ergab sich der beste Wert  $W=0,808641$  dieser Abbruchvorgabe. In der fünften Evolution folgte eine Verschlechterung auf den Wert  $W=0,809137$ .

Dies widerspricht der Forderung, den Wert stetig zu verbessern und letztlich 10 mal nacheinander den besten Wert zu berechnen. Daher wurde die Wertetabelle gelöscht, die beste Wirtschaftlichkeitsmaßzahl gespeichert, die Abbruchvorgabe um 10 erhöht und die Evolution erneut begonnen.

Bei der Abbruchvorgabe  $V_{ab}=40$  wurde erstmals das optimale Individuum mit dem Wert  $W = 0,80862$  gefunden. Zu diesem Zeitpunkt war jedoch noch nicht bekannt, dass es sich hierbei um das Optimum handelte. Auch diese Wirtschaftlichkeitsmaßzahl wurde gespeichert und war nun die Referenz für alle nachfolgenden Evolutionen.

Mit der Abbruchvorgabe  $V_{ab}=200$  wurde 10 mal nacheinander (ausschließlich) dieses beste Individuum gefunden. Unter Berücksichtigung der vorangegangenen Überlegungen ist die Annahme zulässig, dass es sich hierbei um das optimale Individuum handelt. Der Vergleich mit der Lösung 4 aus Tabelle 4 (Referenzlösung)



bestätigt dies. Die Optimierung wurde abgebrochen. In dieser Optimierung wurden insgesamt 6432 verschiedene Generationen erzeugt und 82 Evolutionen durchlaufen.

Mit demselben System wurde diese Optimierung mehrfach durchgeführt. Es wurde erwartungsgemäß immer dasselbe Optimum gefunden. Aufgrund des zufallsbasierten Ansatzes des Genetischen Algorithmus unterschied sich jedoch jeweils der Aufwand.

## **8.2 Anwendung der Fitness im EMB-System**

Eines der genetischen Evolutions-Prinzipien ist die Selektion. In dieser Arbeit wird das bekannte Prinzip der Selektion der besten Individuen um eine weitere Selektionsart erweitert, die Selektion der fittesten Individuen. Die Besten-Selektion basiert auf der Übernahme der besten Individuen in die nächste Generation, die Fittesten-Selektion auf der Einführung und Berücksichtigung einer Individuen-Fitness.

Die Selektion der besten Individuen entspricht einem Gradienten-Verfahren, bei dem sich die Wirtschaftlichkeit der selektierten Individuen stetig verbessern soll. Die zu übernehmenden Individuen müssen mindesten genauso gut sein, wie das aktuell beste Individuum.

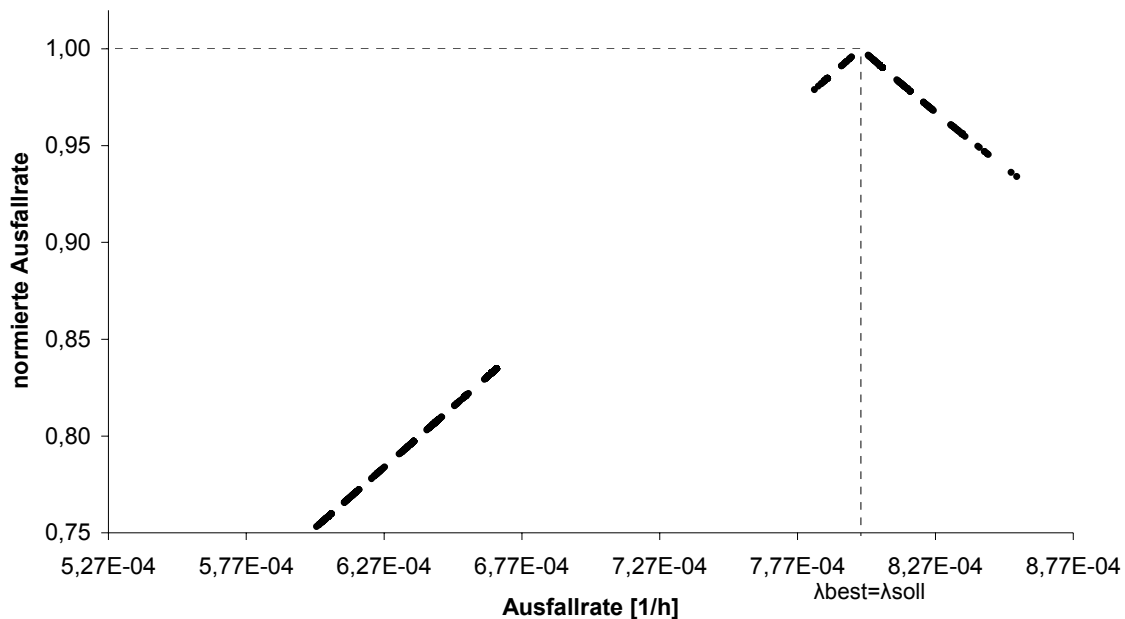
Diejenigen Individuen, die (geringfügig) schlechter als das wirtschaftlichste Individuum sind, haben keine Chance sich weiter fortzupflanzen. Aus eben jenen Individuen könnte aber letztlich das optimale Individuum entstehen. Daher wurde in dieser Arbeit mit der Selektion der fittesten Individuen eine Methode gefunden, die diesen Aspekt berücksichtigt.

In Kapitel 4.3.1.2 wird die Berechnung der Fitness von Individuen und die Zusammensetzung der enthaltenen Terme erläutert. Die Fitnessfunktion  $s(W,\lambda)$  (Gleichung (32)) wurde speziell für die vorliegende Aufgabenstellung entwickelt.

### **8.2.1 Werte der normierten Ausfallrate**

Im Kapitel 4.3.1.2.1 wurde in der Abbildung 19 der theoretische Verlauf der normierten Ausfallrate für  $\lambda_{\text{best}}=\lambda_{\text{opt}}$  dargestellt. Da die vorliegende Optimierungsaufgabe ein Problem mit diskretem Lösungsraum ist, ergeben sich folglich diskrete Werte für  $\lambda_{\text{norm}}$ . Es ist außerdem nicht der gesamte Ausfallraten-Bereich relevant; nur die Werte der jeweils aktuellen Individuen müssen berechnet werden.

Die Abbildung 40 zeigt die sich ergebenden normierten Ausfallraten der Beispiel-EMB zu einem frühen Zeitpunkt der Optimierung.



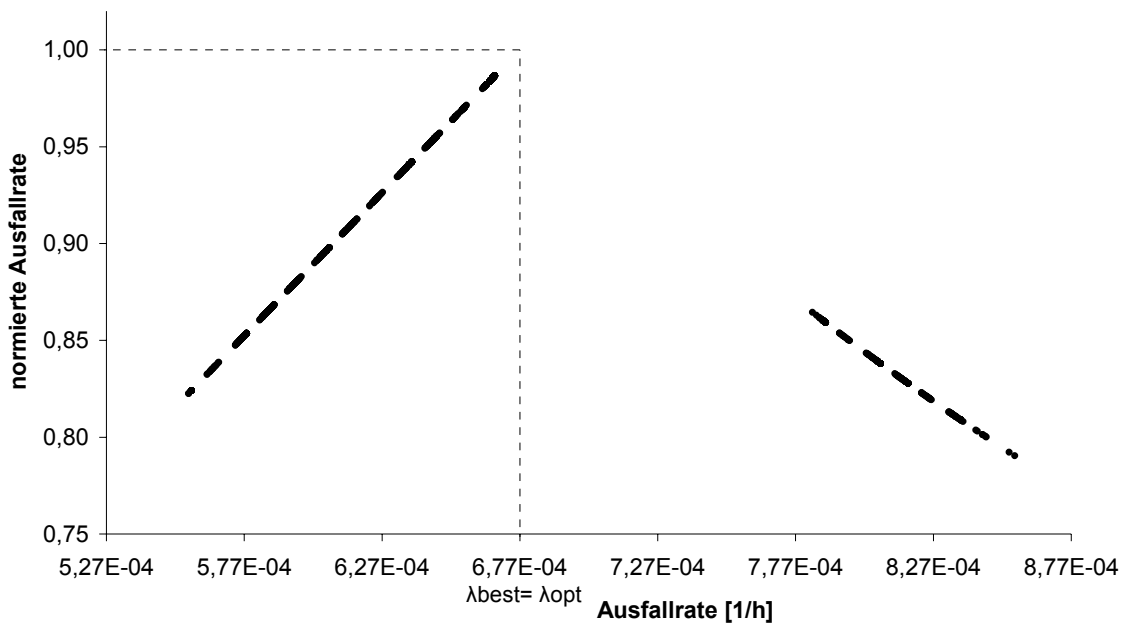
**Abbildung 40: Relevanter Bereich der normierten Ausfallrate der resultierenden Individuen der EMB für  $\lambda_{\text{best}} = \lambda_{\text{soll}}$**

Die Ausfallrate des besten Individuums der aktuellen Generation ist hier identisch mit der Soll-Vorgabe ( $\lambda_{\text{best}} = \lambda_{\text{soll}}$ ). Man sieht, dass sich der höchste Wert für  $\lambda_{\text{norm}} = 1$  erwartungsgemäß beim Normierungswert  $\lambda_{\text{soll}}$  ergibt. Auch Individuen mit  $\lambda > \lambda_{\text{soll}}$  werden betrachtet und haben nahe  $\lambda_{\text{soll}}$  einen relativ hohen Wert.

Die normierte Ausfallrate der Individuen mit  $\lambda < \lambda_{\text{opt}}$  ( $\lambda_{\text{opt}}$  ist an dieser Stelle der Optimierung noch nicht bekannt) sind deutlich kleiner. Sollte ein Individuum in diesem Bereich gefunden werden, so wird es durch die Selektion der Besten, und nicht durch die Selektion der Fittesten in die nächste Generation übernommen.

In einer späteren Generation kann das optimale Individuum bereits gefunden worden sein. Seine Ausfallrate dient als Normierungswert für anschließende Ausfallraten-Normierungen.

Es ergeben sich normierte Ausfallratenwerte, wie sie in Abbildung 41 dargestellt sind.



**Abbildung 41: Relevanter Bereich der normierten Ausfallrate der resultierenden Individuen der EMB für  $\lambda_{\text{best}} = \lambda_{\text{opt}}$**

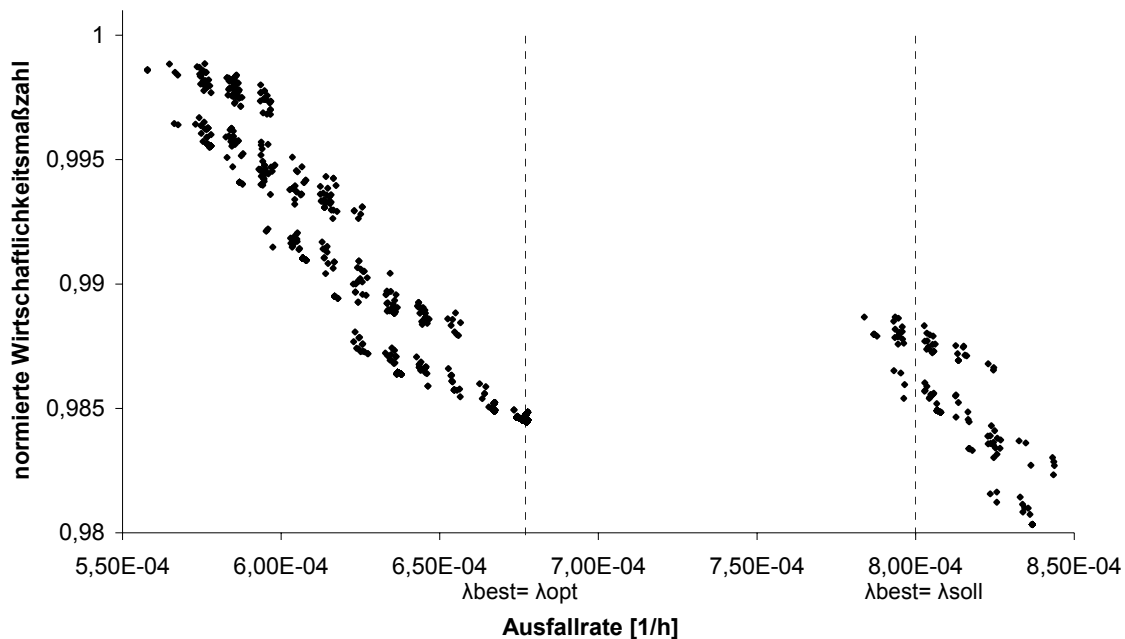
Man erkennt, dass die „schlechten“ Individuen nahe  $\lambda_{\text{soll}}$  nun einen deutlich geringeren Wert aufweisen.

### 8.2.2 Werte der normierten Wirtschaftlichkeitsmaßzahl

Die normierte Wirtschaftlichkeitsmaßzahl  $W_{\text{norm}}$  ist nach Gleichung (38) die Summe aller Wirtschaftlichkeitsmaßzahlen, dividiert durch die maximal mögliche Summe aller Wirtschaftlichkeitsmaßzahlen der Gene.

Die Normierung ermöglicht einen Vergleich aller Individuen des Lösungsraumes und liegt gleichzeitig im Wertebereich der normierte Ausfallrate  $\lambda_{\text{norm}}$ . Da die beiden normierten Werte dimensionslos sind und in vergleichbaren Wertebereichen liegen, dürfen sie in der Fitnessfunktion in Relation zueinander gesetzt werden.

Der relevante Bereich der normierten Wirtschaftlichkeitsmaßzahlen verschiedener Individuen der EMB ist nachfolgend dargestellt (Abbildung 42).



**Abbildung 42: Relevanter Bereich der normierten Wirtschaftlichkeitsmaßzahlen verschiedener Individuen der EMB**

Jeder Punkt der Abbildung 42 stellt den diskreten Wert  $W_{\text{norm}}$  der normierten Wirtschaftlichkeitsmaßzahl von mindestens einem Individuum im Bezug zu seiner Ausfallrate dar. Die linke gestrichelte Linie zeigt die Ausfallrate des optimalen Individuums, die rechte die maximal zulässige Ausfallrate.

Man erkennt in Abbildung 42 eine strukturierte Anordnung der Werte. Zunächst lassen sich die beiden weit auseinanderliegenden Bereiche in jeweils einer „Einhüllenden“ zusammenfassen.

Die linke enthält Individuen, deren Ausfallrate teilweise geringfügig schlechter, im wesentlichen aber besser als die des Optimums sind. Die Ausfallraten der rechten „Einhüllenden“ resultieren aus Individuen, deren Ausfallraten teilweise geringfügig besser, aber überwiegend schlechter als die maximal zulässige Ausfallrate sind. Diese Strukturierung resultiert aus den für das betrachtete EMB-System benötigten Komponenten.

Es sind zwei Batterien erforderlich, für die drei verschiedene Typen zur Verfügung stehen. Die Batterie-Typen 2 und 3 haben hohe Ausfallraten. Diese sind deutlich schlechter als die Ausfallraten aller anderen Komponententypen. Jede Kombination mit mindestens einer dieser beiden „dominanten“ Batterien führt zu einem Individuum in der rechten „Einhüllenden“. Alle anderen Kombinationen liegen in der linken „Einhüllenden“.

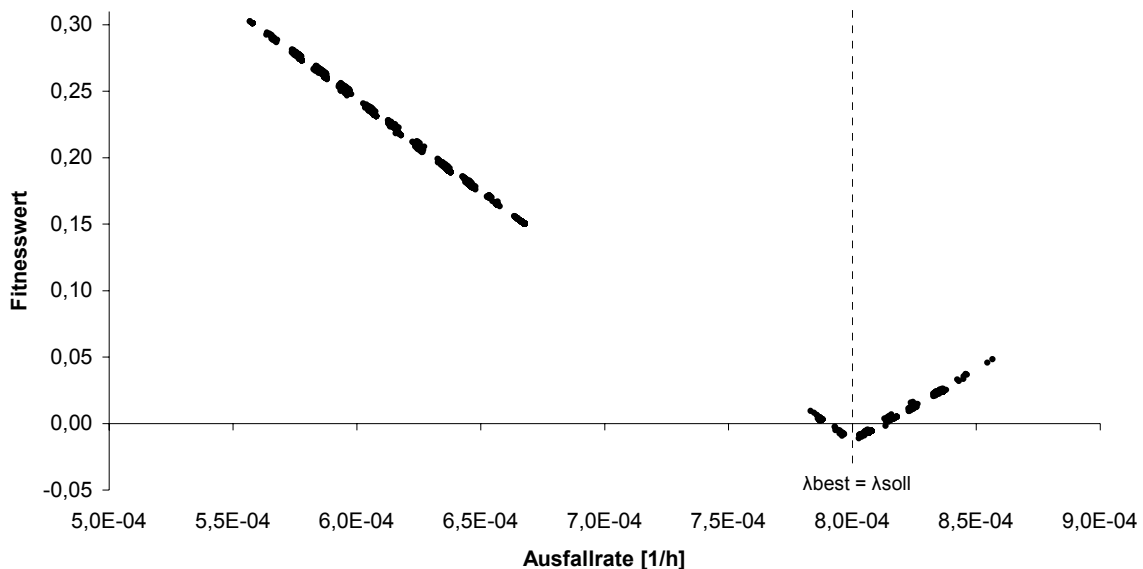
In diesen beiden „Einhüllenden“ lassen sich weitere Unterstrukturen erkennen. Betrachtet man beispielsweise die linke „Einhüllende“, so könnte man diese in vier weitere Unterstrukturen aufteilen. Die sich ergebenden „Einhüllenden“ fallen jeweils von links nach rechts leicht ab. Ähnlich wie die beiden durch die Batterien dominierten Bereichen, resultieren diese Unterstrukturen ebenfalls aus „dominanten“

Komponententypen. Auch in diesen Unterstrukturen lassen sich mit der gleichen Begründung kleinste „Einhüllende“ erkennen.

Alle Individuen rechts der maximal zulässigen Ausfallrate hätten bei der ausschließlichen Selektion der besten Individuen keine Chance auf Fortpflanzung über mehrere Generationen hinweg. Ihre niedrige Wirtschaftlichkeitsmaßzahl lässt aber darauf schließen, dass sich aus Ihnen ein gutes Individuum entwickeln kann. Um diese Fortpflanzung zu ermöglichen, wurde die Fitness eingeführt.

### 8.2.3 Fitness

Für das untersuchte EMB-System existieren 442368 unterschiedliche Individuen. Für jedes Individuum kann eine eigene Fitness (siehe Gleichung (32)) berechnet werden. Die Fitness-Werte  $s$  unterschiedlicher Individuen können identisch sein. In der Abbildung 43 ist die Fitness der Individuen in einer früheren Phase der Optimierung dargestellt.



**Abbildung 43: Resultierende Fitness der Individuen bei  $\lambda_{best} = \lambda_{soll}$**

Eine hohe Fitness wird durch einen kleinen Fitnesswert  $s$  dargestellt.

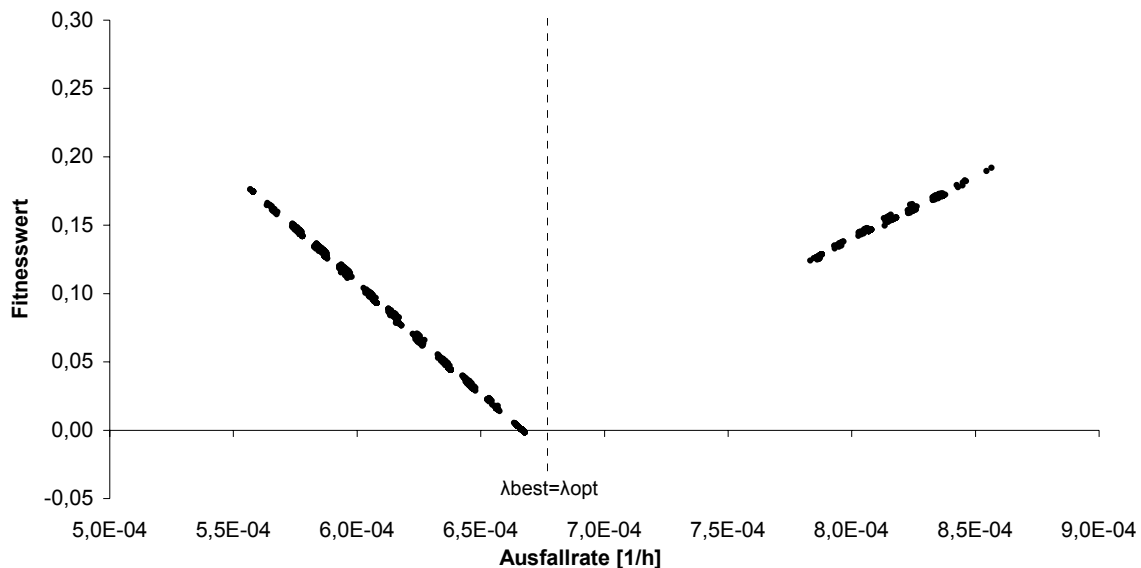
Man erkennt, dass sich auch negative Werte ergeben. Dies ist nach der Fitnessgleichung (Gleichung (32)) der Fall, wenn „ $\lambda_{norm} > W_{norm}$ “ ist.

Die Ausfallrate des besten Individuum  $\lambda_{best}$  der aktuellen Generation entspricht zu diesem Zeitpunkt der vorgegebenen Ausfallrate  $\lambda_{soll}$ . Es ist sinnvoll, diesen Wert auch als Startwert der Optimierung zu nutzen.

Für die Individuen nahe der momentan besten Ausfallrate  $\lambda_{best}$  ergibt sich eine hohe Fitness. Sollte ein Individuum mit einer geringen Ausfallrate und kleiner Wirtschaftlichkeitsmaßzahl gefunden werden, so wird dieses, trotz der offensichtlich guten Werte, mit einer geringen Fitness bewertet. Dies ist auch so gewollt, da zu erwarten ist, dass es in einer späteren Optimierungsphase eine höhere Fitness bekommt. Dieses Individuum könnte aus der aktuellen Generation dennoch durch die Selektion der Besten in die nächste Generation übernommen werden.

In nachfolgenden Generationen sollten bessere Individuen gefunden werden. Die Ausfallrate des jeweils besten Individuums wird dann der Normierungswert für die Ausfallraten-Normierung nach Gleichung (33).

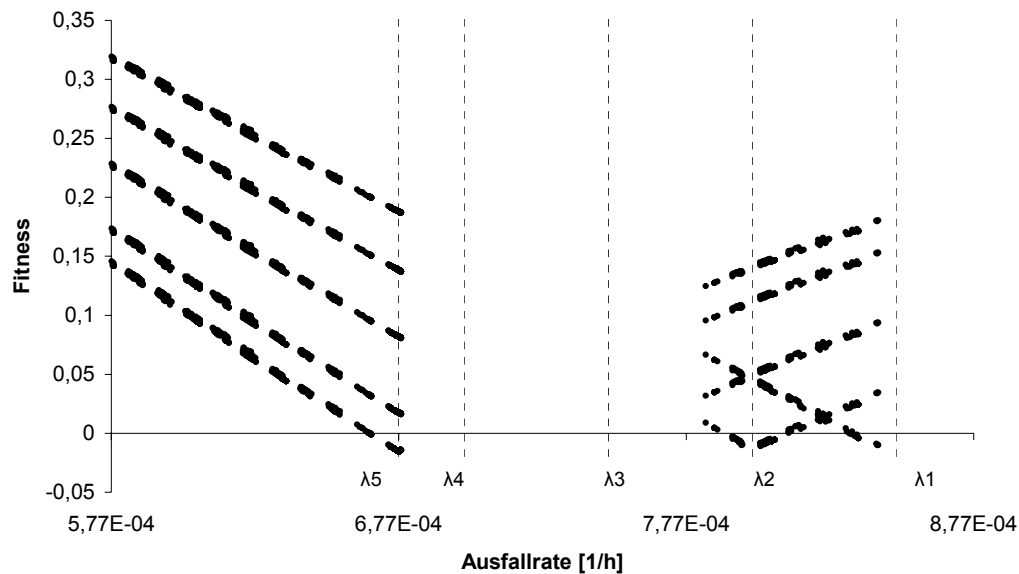
Für die in Abbildung 44 dargestellten Fitnesswerte der EMB wurde bereits  $\lambda_{\text{best}} = \lambda_{\text{opt}}$  vorausgesetzt.



**Abbildung 44: Resultierende Fitness der Individuen bei  $\lambda_{\text{best}} = \lambda_{\text{opt}}$**

Die Fitnesswerte der rechts vom Optimum (gestrichelte Linie) liegenden Individuen sind zwar sehr gering, aber dennoch größer als die sich in der Nähe des Optimums befindlichen. Da dieser Zustand voraussichtlich erst zu einem späteren Punkt der Optimierung erreicht wird, wird vorausgesetzt, dass die „schlechteren“ Individuen bereits Teil der Optimierungsgeneration gewesen sind. Diese Individuen werden dann erst nach und nach in den Generationen ersetzt, da nicht sofort alle neuen Individuen eine höhere Fitness aufweisen.

Die Abbildung 45 zeigt die Veränderung der Fitness der EMB-Individuen während der Optimierung.

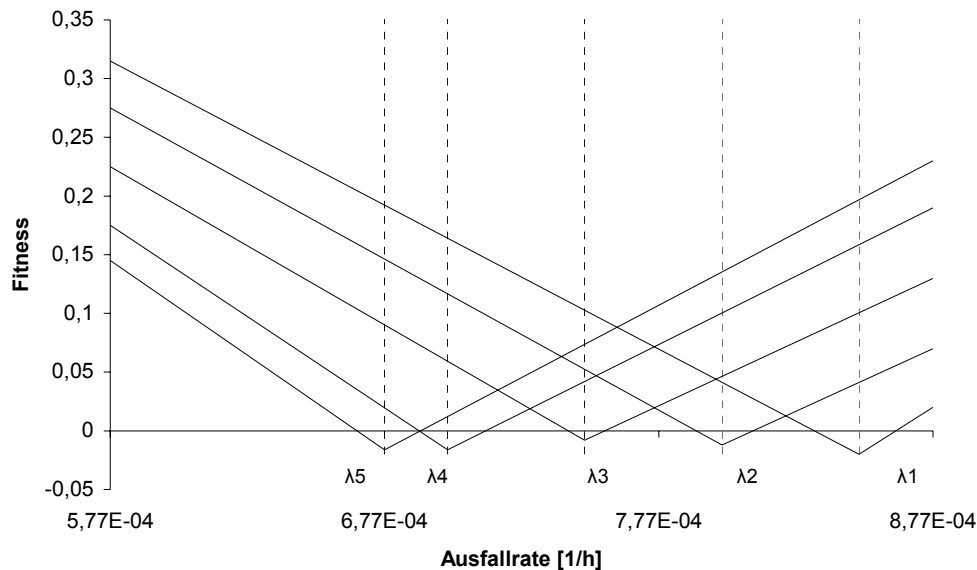


**Abbildung 45: Veränderung der Fitness während der Optimierung**

Die gestrichelten senkrechten Linien stellen jeweils die beste Ausfallrate dar, die den Fitnessberechnungen in verschiedenen Phasen der Optimierung zugrunde lag. Diese fünf Ausfallraten ( $\lambda_1$  bis  $\lambda_5$ ) sind die Berechnungsgrundlagen (Referenzen) der fünf dargestellten Fitness-Verläufe.

Der Referenzwert wird in einer frühen Phase der Optimierung eine vergleichbar hohe Ausfallrate aufweisen, die sich jedoch im Laufe der Optimierung verbessern wird. Dieser Startwert könnte bei der EMB in der Größenordnung der in Abbildung 45 mit  $\lambda_1=8,5 \cdot 10^{-4}/h$  dargestellten Ausfallrate sein. Er wird durch die Ausfallrate des jeweils wirtschaftlichsten (nicht des besten!) Individuums definiert. Es ist wahrscheinlich, dass sich diese Ausfallrate im Laufe der Optimierung verkleinert. Da jedoch die Wirtschaftlichkeitsmaßzahl von der Ausfallrate nicht linear abhängt, eine Verbesserung der Ausfallrate also nicht gleichbedeutend mit einer Verschlechterung der Wirtschaftlichkeitsmaßzahl einhergeht, kann der Referenzwert im Laufe der Optimierung wieder ansteigen.

In der Abbildung 46 sind die Fitness-Verläufe der Individuen aus Abbildung 45 interpoliert und vereinfacht dargestellt.



**Abbildung 46: Vereinfachte Darstellung der Verschiebung der Fitness-Werte während der Optimierung**

Um die Veränderung der Fitness der Individuen darstellen zu können, wurden in Abbildung 46 beispielhaft fünf verschiedene Referenzwerte, die sich während der Optimierung ergaben, dargestellt. Vereinfachend wurde hier eine stetige Verbesserung der Referenzwerte vom Startwert  $\lambda_1=8,5 \cdot 10^{-4}/h$  bis zur Ausfallrate des Optimums  $\lambda_5=6,77 \cdot 10^{-4}/h$  angenommen. Die V-förmigen Verläufe der Fitness-Werte haben ihr Minimum jeweils beim aktuellen Referenzwert. Dieses Minimum ist kleiner als 0, wenn die normierte Wirtschaftlichkeitsmaßzahl kleiner als die normierte Ausfallrate ist. Durch die stetige Verbesserung (Sonderfall für dieses Beispiel) der Referenz-Ausfallrate „wandert“ der Punkt des kleinsten Fitnesswertes  $s$  nach links.

#### 8.2.4 Praxisrelevanz der Fitness

Im Kapitel 4.3.1.2 wurde dargestellt, dass Individuen mit einer im Vergleich zum Referenzwert höheren Ausfallrate bei gleichem Abstand höher bewertet werden als Individuen mit einer niedrigeren. Dies ergibt sich aus dem flacheren Verlauf des Gradienten bei der Berechnung der normierten Ausfallrate (vgl. Abbildung 19) für höhere Ausfallraten. Individuen mit identischen Ausfallraten haben also eine im Vergleich höhere Fitness, wenn ihre Ausfallrate schlechter als der Referenzwert ist.

In diesem frühen Stadium der Optimierung besteht die Generation noch aus vielen unterschiedlichen und „schlechten“ Individuen. Die Ausfallrate hat einen höheren Einfluss auf die Berechnung der Fitness, als in einem späteren Optimierungsstadium. Durch diese Vielfalt können vom Beginn der Optimierung an viele unterschiedliche Optimierungswege verfolgt werden.

Im späteren Optimierungsstadium existieren bereits viele Individuen, deren Ausfallrate nahe dem Optimum liegen. Der Gradient bei der Berechnung der normierten Ausfallrate (vgl. Abbildung 19) ist in dem dann relevanten Bereich auf beiden Seiten ähnlich, abgesehen vom Vorzeichen. Die normierte Wirtschaftlichkeitsmaßzahl hat somit, bei ähnlichen Ausfallraten-Abständen vom



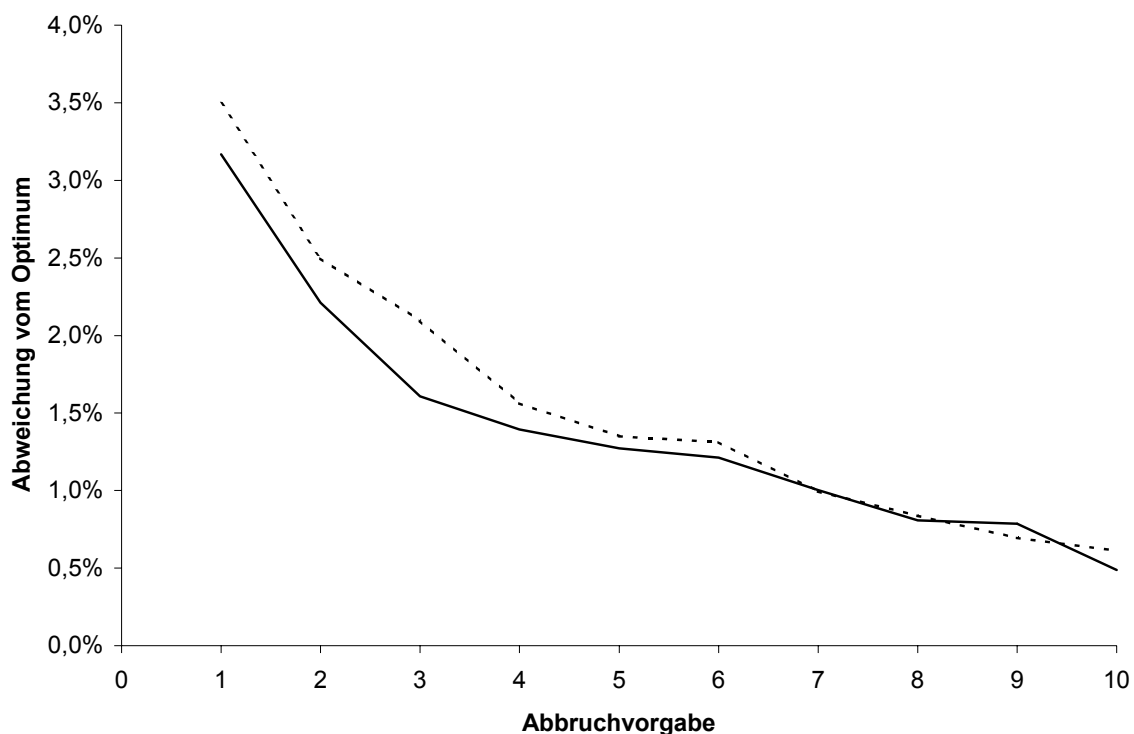
Referenzwert, den wesentlichen Einfluss auf die Berechnung der Fitness. Im späten Optimierungsstadium dient die Fitness also verstärkt der Suche nach dem wirtschaftlichen Optimum.

### 8.3 Darstellung der Effektivität der gezielten Genmanipulation

Nach Beendigung der Optimierung wird die in Kapitel 4.3.6 beschriebene gezielte Genmanipulation durchgeführt. Hierbei werden die Gene der gefundenen Lösung mit denen der Datenbasis verglichen. Werden in der Datenbasis „offensichtlich bessere“ Gene gefunden, werden die „schlechteren“ Gene des betrachteten Individuums durch diese ersetzt. Ein Gen der Datenbasis ist „offensichtlich besser“, wenn es bei gleicher oder besserer Ausfallrate gleiche oder bessere Wirtschaftlichkeitsparameter aufweist. Keiner der Wirtschaftlichkeitsparameter des Gens der Datenbasis darf dann einen schlechteren Wert als das potenziell zu ersetzende des gefundenen Individuums haben.

Für die nachfolgende Betrachtung der gezielten Genmanipulation wurde das adaptive Abbruchkriterium nicht angewendet. Die Abbruchvorgabe jeder Optimierung war konstant und wurde für die nächste Optimierung inkrementiert.

In der Abbildung 47 sind zwei Verläufe dargestellt. Die durchgezogene Kennlinie entspricht der Optimierung mit abschließender gezielter Genmanipulation. Bei der gestrichelten Kennlinie wurde darauf verzichtet.



**Abbildung 47: Optimierungsfehler in Abhängigkeit von der Abbruchvorgabe mit und ohne gezielte Genmanipulation**

Der Sinn der gezielten Genmanipulation besteht darin, bereits in einer frühen Phase der Optimierung gute Individuen zu bekommen. Die Abbildung 47 ist daher auf die Darstellung der Werte für kleine Abbruchvorgaben beschränkt. Die bei kleiner Abbruchvorgabe gefundenen Individuen sind vergleichbar mit denen, die bei der

Optimierung mit großen Abbruchvorgaben in der frühen Optimierungsphase gefunden wurden.

Der gestrichelte Verlauf entspricht einer Optimierung ohne, der durchgezogene einer Optimierung mit gezielter Genmanipulation. Es wurde nur der Bereich kleiner Abbruchvorgaben ( $V_{ab}=1\dots 10$ ) betrachtet. Für jede dargestellte Abbruchvorgabe wurde der arithmetische Mittelwert der Wirtschaftlichkeitsmaßzahlen aus 100 Optimierungen ermittelt. Die sich ergebende prozentuale Abweichung von der optimalen Wirtschaftlichkeitsmaßzahl ist letztlich dargestellt.

Man erkennt, dass durch die gezielte Genmanipulation bereits in der frühen Phase der Optimierung im Mittel eine geringere Abweichung der Individuen vom Optimum resultiert. Die Qualität der in der frühen Phase gefundenen Individuen ist also tatsächlich höher. Bei großen Abbruchvorgaben unterscheiden sich die Abweichungen kaum noch, denn bei beiden Varianten (mit und ohne gezielte Genmanipulation) verringert sich mit steigender Abbruchvorgabe der Fehler-Mittelwert.

Die Qualitätsverbesserung der Individuen in der frühen Phase wirkt sich jedoch auf den gesamten Optimierungsaufwand aus und beschleunigt diesen bei jeder Abbruchvorgabe. Ohne gezielte Genmanipulation wird dasselbe Ergebnis gefunden, es müssen jedoch mehr Generationen durchlaufen werden

#### **8.4 Grenzen bei der Beherrschung von Lösungsräumen beliebiger Komplexität**

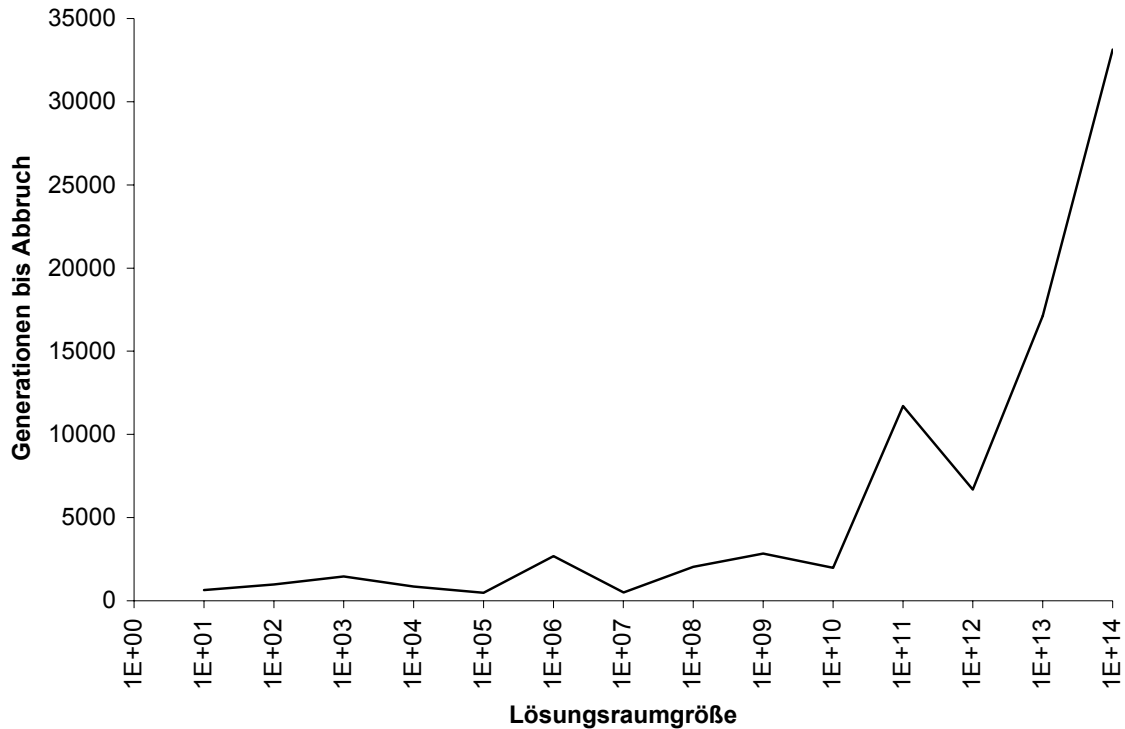
Im Kapitel 4.3 wurde dargestellt, dass die zufallsbasierte Methode der Genetischen Algorithmen eine heuristische Vorgehensweise zur Beherrschung der Komplexität des Lösungsraumes anwendet. Die Stärke des Genetischen Algorithmus liegt in der Beherrschung großer Lösungsräume. Nachfolgend wird diese Theorie durch ein praxisbezogenes Beispiel validiert. Bei Genetischen Algorithmen darf der benötigte Lösungsaufwand nicht im gleichen Maße wie die Komplexität des Lösungsraumes steigen.

Theoretisch ist, aufgrund der angewandten Heuristiken, jede Lösungsraum-Komplexität beherrschbar.

Zur Validierung des realisierten Genetischen Algorithmus wurde eine Testkomponente definiert, durch die eine vereinfachte Systemdefinition und Aufwandsberechnung möglich wird. Diese Testkomponente besteht aus 10 Typen, die sich nur hinsichtlich ihrer Kosten unterscheiden. Sie variieren von 1€ für Typ1 bis zu 10€ für Typ10.

Es wurden verschiedene Testsysteme definiert, die ausschließlich aus einer jeweils unterschiedlichen Anzahl an Testkomponenten bestand. Gemäß Gleichung (7) ergeben sich für ein System mit zwei Testkomponenten  $10^2=100$  mögliche Lösungen, für 3 Testkomponenten 1000, für vier 10000 usw.. Die Lösungsraumgröße wird mit jeder weiteren Komponenten um den Faktor 10 erhöht.

Die Abbildung 48 zeigt die Anzahl der durchlaufenen Generationen bei Systemen mit verschieden großen Lösungsräumen.



**Abbildung 48: Anzahl der durchlaufenen Generationen bei steigender Lösungsraumgröße**

Die Abszisse ist logarithmisch eingeteilt. Man erkennt eine steigende Tendenz der Anzahl der durchlaufenen Generationen. Dieser Anstieg ist jedoch aufgrund des zufallsbasierten Ansatzes nicht monoton.

Aus Abbildung 48 geht weiter hervor, dass der Berechnungsaufwand zwar tendenziell mit einem vergrößerten Lösungsraum steigt, dieser Anstieg jedoch im Verhältnis zur Lösungsraumvergrößerung deutlich geringer ist.

Vergleicht man die Generationenanzahl der Lösungsräume der Größen  $1E+01$  und  $1E+10$  so erkennt man, dass sich der Lösungsraum um den Faktor  $1E+09$  (d. h. um 9 Potenzen) vergrößert hat, die Anzahl der benötigten Generationen hingegen nur um etwa den Faktor 4. Der Aufwand vergrößert sich also deutlich weniger als die Lösungsraumgröße.

Durch den zufallsbasierten Ansatz des Genetischen Algorithmus kann sich der tatsächliche Aufwand, bei Vergrößerung des Lösungsraumes, sogar verringern. Ein Beispiel dafür ist in der Abbildung 48 bei einem Vergleich der Generationenanzahlen der Lösungsräume mit den Größen  $1E+06$  und  $1E+07$  erkennbar. Die Anzahl der durchlaufenen Generationen ist beim größeren Lösungsraum geringer als beim kleineren.

Alle als Beispiel genutzten Systeme wurden vom angewandten Genetischen Algorithmus problemlos beherrscht. Der verwendete Genetische Algorithmus war vollständig, d.h. alle in dieser Arbeit dargestellten Prinzipien inklusive der Genmanipulation und des adaptiven Abbruchkriteriums wurden angewendet.

Die optimale Lösung jedes Testsystems war, aufgrund der Typenstruktur in der Datenbasis, offensichtlich erkennbar. Bei jeder Optimierung wurde immer das

entsprechende Optimum gefunden. Die Untersuchung hätte noch beliebig weitergeführt werden können. Die Komplexität der Beispiel-EMB liegt bei 442368 möglichen Lösungen. Übertragen auf das System aus Abbildung 48 entspricht das etwa einer Generationenanzahl von 2500. Da auch Lösungsräume mit weitaus höherer Komplexität untersucht wurden, kann davon ausgegangen werden, dass die Größe des Lösungsraumes kein begrenzendes Kriterium für den Genetischen Algorithmus darstellt.

### **8.5 Vergleichsmöglichkeit mit ähnlichen Varianten**

Eine Voraussetzung der praktischen Anwendbarkeit der Optimierungsmethode ist ihre Stabilität. Eine kleine Änderung in der Datenbasis sollte nicht das gesamte bisherige System verändern. Vergrößert sich beispielsweise die Ausfallrate eines gewählten Typs, so sollten während des Entwicklungsprozesses möglichst wenige der bisher gewählten Typen verworfen werden.

Ist dies nicht der Fall, so müsste ein Teil des Entwicklungsaufwandes, der in die verworfenen Komponententypen investiert wurde, nochmals aufgebracht werden. In diesem Fall hat man die Alternative, eventuell eine ähnlichen Variante mit geringerer Entwicklungsunsicherheit, weiter zu verfolgen. Bei dieser könnte es sich dann, wenn das Ergebnis der Optimierung mit den ausschließlich scharfen Parameterwerten zugrunde gelegt wird, um eine suboptimale Variante handeln. Die Optimierungsmethode sollte daher die Möglichkeit bieten, auch Varianten nahe des Optimums zu betrachten.

Alle drei untersuchten Optimierungsmethoden erfüllen diese Anforderung mit unterschiedlichem Aufwand:

- Der Graphen-Algorithmus berechnet immer den gesamten Lösungsraum. Der Aufwand für die Betrachtung der „nächsten Nachbarn“, also der dem Optimum ähnlichen Varianten, ist gering. Im Algorithmus müsste statt der besten Variante eine beliebige Anzahl der nächstbesten Varianten gespeichert werden. Der Durchlauf zum Finden dieser Varianten erfolgt nur einmal nach Ende der Berechnung des Lösungsraumes.
- Die Neuro-Fuzzy-Methode vergleicht immer die augenblicklich beste Variante mit der aktuellen Variante nach der Veränderung der Fuzzy-Mengen. Mit ebenfalls geringem Mehraufwand ist hier das Speichern und der Vergleich der gewünschten Anzahl nächstbesten Varianten möglich.
- Bei der Methode des Genetischen Algorithmus werden prinzipiell mehr als nur die augenblicklich beste Variante gespeichert. Zum Finden der gewünschten Anzahl dieser nächstbesten Varianten müssen also die gewünschte Anzahl nächstbesten Individuen in die neue Generation übernommen werden. Im Allgemeinen entsteht somit bei dieser Methode wenig oder sogar kein Mehraufwand.

### **8.6 Voraussetzungen der Übertragbarkeit der Methode auf andere Fragestellungen**

Der vorliegenden Fragestellung liegt ein Wirtschaftlichkeits-Zuverlässigkeits-Problem zugrunde und der Lösungsraum enthält diskrete Werte. Ist die Methode auch noch anwendbar, wenn sich die Anzahl und Art der Optimierungsparameter ändert?

Die Anzahl der Freiheitsgrade (Optimierungsparameter) wurde in der vorliegenden Arbeit durch die Einführung der Wirtschaftlichkeitsmaßzahl auf zwei reduziert. Dies

war möglich, da die Wirtschaftlichkeitsparameter eine „Beziehung“ zueinander haben. Die Wirtschaftlichkeit konnte quantifiziert werden.

Wären die Wirtschaftlichkeitsparameter völlig unabhängig voneinander, könnten die Freiheitsgrade nicht reduziert werden.

Diese Überlegung führt dazu, dass nur solche Systeme optimiert werden können, deren Parameter sich entweder exakt voneinander abgrenzen lassen oder die zu einem Ersatzwert zusammengefasst werden können, d. h. die Qualität der Parameter muss quantifizierbar sein. Diese Problematik ist jedoch unabhängig von der gewählten Optimierungsmethode.

Genetische Algorithmen sind Methoden der kombinatorischen Optimierung und demnach auch nur für diskrete Lösungsräume anwendbar. Für Probleme mit kontinuierlichen Parametern wäre der Lösungsraum theoretisch unendlich groß. Hierfür sollten dann Regelalgorithmen eingesetzt werden. In diesem Rahmen wäre es interessant zu untersuchen, ob beispielsweise eine Neuro-Fuzzy-Regelung bessere Ergebnisse als beim vorliegenden Problem liefern würden.

## 9 Methode zur Beurteilung der vagen Daten

Im Kapitel 4 wurden die Vor- und Nachteile von drei Optimierungsmethoden dargestellt. Durch Anwendung einer (beliebigen) dieser drei, liegt nun eine (optimale) Lösungsvariante vor.

Auf diese Variante aufbauend, zeigt das Kapitel 5 eine Methode zur Quantifizierung und Bewertung der Entwicklungsunsicherheit. Diese Methode der vagen Daten ist allgemeingültig auf die Kosten, die Ausfallrate, die Geometrie und das Bauvolumen anwendbar. Diese bewerteten Terme werden nachfolgend vereinfachend und zusammenfassend „Parameter“ genannt.

Im Kapitel 7 wurde durch die Anwendung des Genetischen Algorithmus die optimale Lösung des Beispielsystems gefunden. Sie dient als Grundlage für die im nachfolgenden Kapitel dargestellte Bewertung der vagen Daten des Beispielsystems.

### 9.1 Beeinflussungsmöglichkeit durch die Methode der vagen Daten

Jeder Entwicklungsprozess birgt eine Unsicherheit hinsichtlich des erzielbaren Ergebnisses. Es gilt, die Idee der Entwickler innerhalb eines vorgegebenen Zeitraumes in ein möglichst zuverlässiges und leistungsfähiges, aber auch kostengünstiges System umzusetzen.

Für jeden verwendeten Typen eines Systems können während jeder Phase der Entwicklung dessen Parameter mit einer entsprechenden Ungenauigkeit angegeben werden. Hieraus resultiert für das Gesamtsystem eine sich während der Entwicklung verändernde Unsicherheit. Diese sollte sich während des Entwicklungsprozesses, von der ersten Idee bis hin zum Serieneinsatz, sukzessive verringern. Zu jedem Zeitpunkt, in jeder Phase, ist die Durchführung von vagen Abschätzungen möglich. Sie erlauben die Überwachung der Verbesserungsfortschritte des Systems.

Auf dem Weg von der Idee bis zum fertigen System gibt es verschiedenste Probleme und Aufgaben, die meist mit all ihren Tücken nur begrenzt vorhersehbar sind. Diese Unsicherheit wurde bisher durch die Erfahrung der entsprechenden Entwickler begrenzt.

Durch die nachstehend vorgestellte Methode zur Beurteilung der vagen Daten wird dem Entwickler eine Möglichkeit gegeben, die Unsicherheit aller Entwicklungsdaten zu quantifizieren. Er kann dadurch gezielt in den Entwicklungsprozess eingreifen und die notwendigen Prioritäten für die Entwicklung setzen. Die wesentlichen Unsicherheitsfaktoren des Systems können erkannt und gezielt verbessert werden.

Drei grundlegende Vorgehensweisen sind denkbar:

1. *In einer Entwicklungsphase* (Konzept-, Prototypen-, Vorserien- und Serienphase) lassen sich Vergleiche *zwischen den unscharfen Parametern* (Ausfallrate, Kosten, Gewicht, Geometrie) ziehen. Die Breite des Unschärfe-Intervalls kann dabei ein Indiz für den Handlungsbedarf und die Priorität zur Verringerung der Unsicherheit sein.
2. Ebenso lassen sich *innerhalb einer Entwicklungsphase* die Unsicherheiten der *Komponenten miteinander vergleichen*. Die Komponente mit der größten Unsicherheit könnte bevorzugt weiterentwickelt werden. In einer Phase dürfen durchaus mehrere Analyse-Entwicklungs-Zyklen durchgeführt werden, um zum Abschluss der jeweiligen Phase das gewünschte Ergebnis vorweisen zu können.

Es ist also denkbar, erst nach dem Erreichen einer vorgegebenen Entwicklungssicherheit in die nächste Phase überzugehen.

3. Im gesamten Entwicklungsprozess lassen sich die Veränderungen der vagen Daten über die einzelnen Phasen hinweg erkennen, quantifizieren und bewerten. Der Entwicklungsfortschritt lässt sich somit gut darstellen.

Ein weiterer, nicht zu unterschätzender Vorteil der Methode ist, dass der Entwickler eine Argumentationshilfe bekommt.

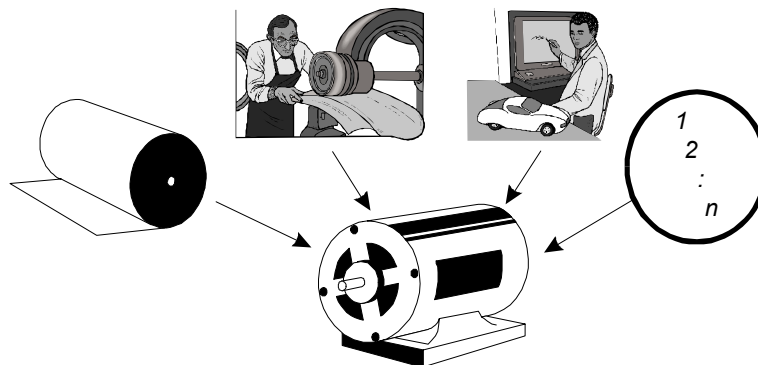
Über Projekte und deren Fortschritt muss ständig berichtet werden. Sowohl vor dem Start als auch im Verlauf von Projekten muss Auskunft über beispielsweise die Einhaltung der anvisierten Kosten gegeben werden. Die dann gemachten Aussagen spiegeln manchmal eine persönliche Abschätzung des Aussagenden über den weiteren Verlauf des Projektes wider.

Durch die hier dargestellte Methode der Auswertung vager Daten können nun die getätigten Aussagen mit Zahlen untermauert werden.

## 9.2 Einflussfaktoren auf die vagen Daten im Entwicklungsprozess

Nachfolgend wird anhand der Kosten eines zu verwendenden Aktors die sich ändernde Unsicherheit während eines Entwicklungsprozesses verdeutlicht.

In einer frühen Phase des Entwicklungsprozesses (*Konzeptphase*) können die Kosten eines Aktors meist nur geschätzt werden. Sie resultieren aus den Kosten für das *Material*, die *Herstellung* und die *Entwicklung*. Diese Kosten werden später auf alle zu produzierenden Aktoren umgelegt (Abbildung 49).



**Abbildung 49: Material, Arbeit, Entwicklung und Teileanzahl als bestimmende Faktoren für die Kosten**

Die Ungenauigkeit bei den *Materialkosten* resultiert überwiegend aus der Ungewissheit, welches Material eingesetzt wird. Ist dies spezifiziert, können die Kosten bereits mit einer begrenzten Ungenauigkeit angegeben werden.

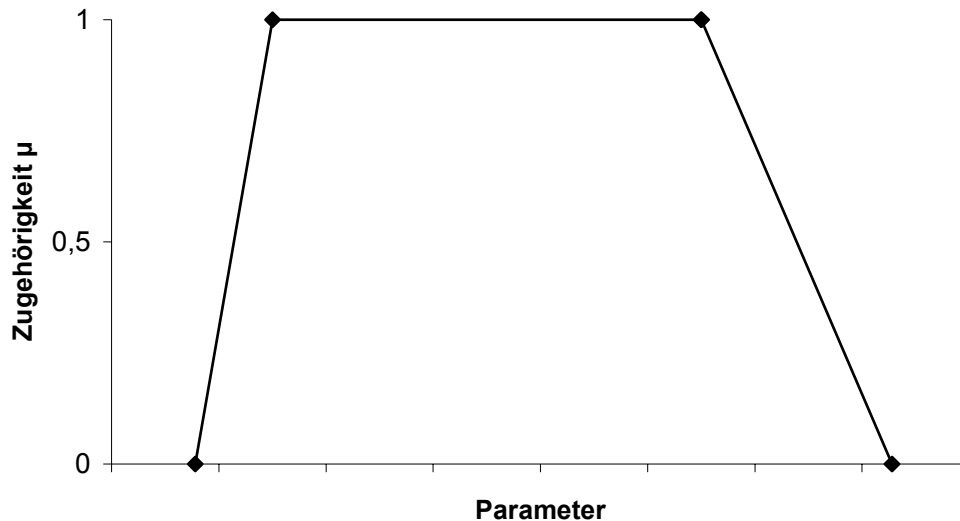
Die Kosten für die *Produktion* sind von vielen Faktoren abhängig. Es müssen ausreichende Produktionskapazitäten und -technologien zur Verfügung stehen. Auch der Produktionsort beeinflusst die Produktionskosten.

Die Kosten für die *Entwicklung* können zunächst nur geschätzt werden. Zum Beginn einer Entwicklung lässt sich meist nicht genau sagen, welche Probleme während der Entwicklung die Entwicklungskosten beeinflussen.

Auch die *Gesamtmenge* der zu produzierenden Aktoren kann oft nur sehr ungenau angegeben werden. Diese ist in der Automobilindustrie beispielsweise vom

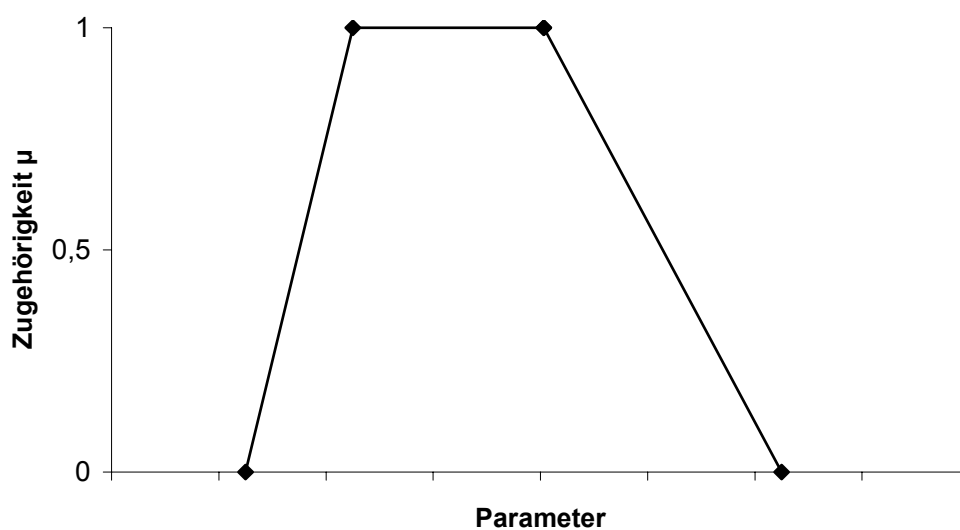
Zielfahrzeug und der Auslegung des Gesamtsystems, aber auch von der Kundenakzeptanz und der Marktlage abhängig.

Aufgrund dieser Unsicherheitsfaktoren würde man in der frühen Entwicklungsphase für neu zu entwickelnde Komponenten meist die Trapezform für die Darstellung der Parameter wählen (Abbildung 50).



**Abbildung 50:** In der frühen Entwicklungsphase wird meist die Trapezform zur Darstellung der Unsicherheit gewählt

In der nächsten Entwicklungsphase (*Prototypenphase*) sollte sich die Unsicherheit dieser Faktoren verringern. Erste Entwürfe wurden mittlerweile umgesetzt, so dass bereits die Menge und die Art der benötigten Materialien konkretisiert wurde. Auch der Aufwand für die Entwicklung kann genauer abgeschätzt werden. Das Unsicherheitsintervall wird insgesamt schmäler (Abbildung 51).

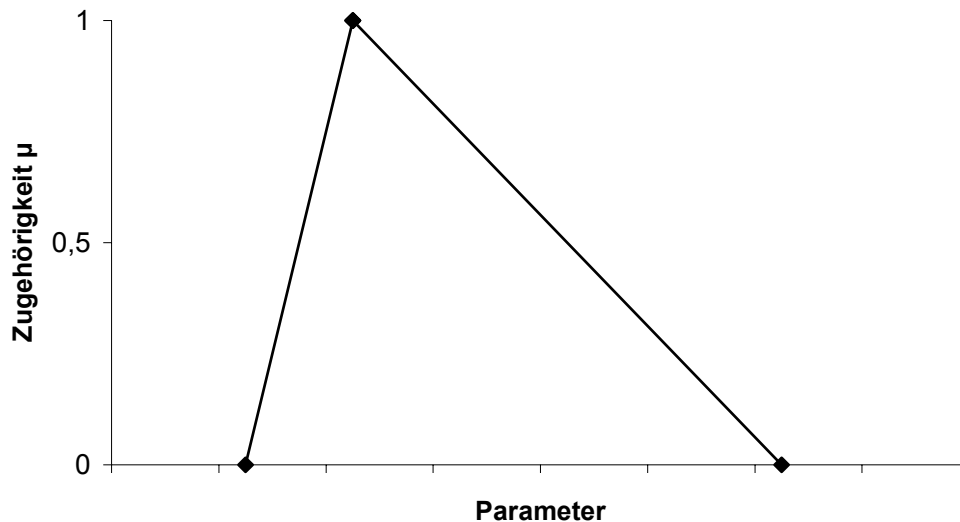


**Abbildung 51:** In der Prototypenphase wird das Unsicherheitsintervall schmäler

Das System wird dann mit dem Ziel der Serienreife in der *Vorserienphase* weiterentwickelt. Das Zielfahrzeug wird spezifiziert und auch das Leistungsvermögen



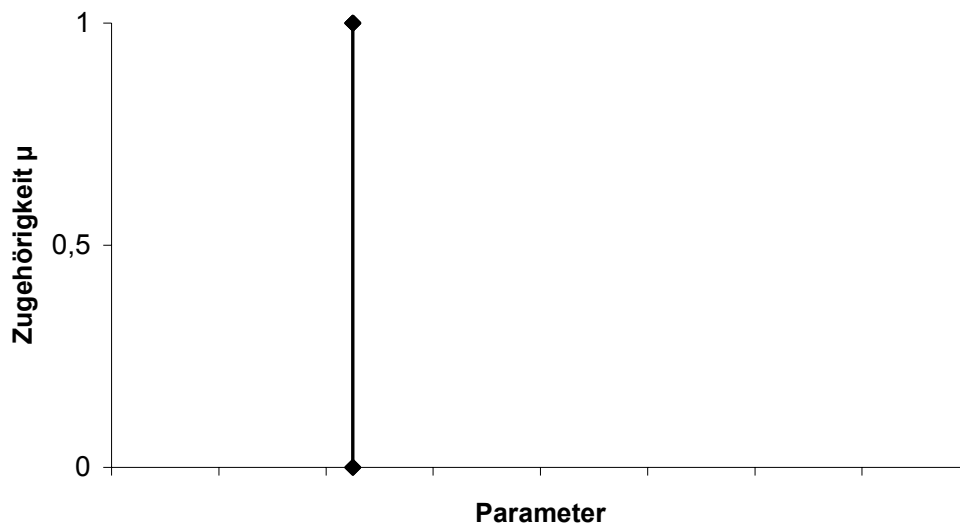
des Gesamtsystems, in das der Aktor eingebaut werden soll, kann genauer angegeben werden. Es sind Rückschlüsse auf die Kundenakzeptanz möglich. Zu diesem Zeitpunkt steht das Gesamtsystem kurz vor der Serieneinführung, so dass auch die Marktlage exakt(er) analysiert werden kann. Das Unsicherheitsintervall hat sich weiter verkleinert, so dass die Angabe durch eine Dreiecksform möglich ist (Abbildung 52).



**Abbildung 52: In der Vorserienphase wird die Unsicherheit weiter präzisiert**

Nach der Vorserienphase erfolgt der *Serieneinsatz*. Zu diesem Zeitpunkt ist der Aktor fertig entwickelt, die Produktionsressourcen und -standorte sind ausgewählt und die Kosten des Gesamtsystems festgelegt. Beispielsweise die Kosten für den Aktor können (spätestens) jetzt mit einer sehr geringen Unsicherheit, oftmals sogar durch einen scharfen Wert angegeben werden.

Die Abbildung 53 zeigt ein Unsicherheitsintervall, das mit fortschreitender Entwicklung präzisiert wurde und in der Serienphase auf einen scharfen Wert reduziert ist.



**Abbildung 53: Mit steigender Dauer der Serienphase können verschiedenen Parameter (z. B. Kosten) sehr genau angegeben werden**

Dieser Wert sollte, sofern die vorangegangene Abschätzung innerhalb eines ausreichend großen Unsicherheitsintervalls erfolgte, nicht außerhalb eines der vorangegangenen Unsicherheitsintervalle liegen. Nun zeigt sich, wie gut die Abschätzungen der Entwicklungsunsicherheit in den bisher durchlaufenen Phasen gewesen sind.

Es ist möglich, dass eventuell auch in der Serienphase eine Unsicherheit verbleibt. Diese könnte für das Kostenbeispiel aus den nur begrenzt kalkulierbaren Kosten für Gewährleistung und/oder Rückrufaktionen resultieren.

Das dargestellte Beispiel ist typisch für die Neuentwicklung einer Komponente in den verschiedenen Meilensteinen der Produktentwicklung. Die Vorgehensweise bei den anderen betrachteten Parametern ist ähnlich, muss jedoch gegebenenfalls angepasst werden. Für die Ausfallrate beispielsweise werden die exakten Daten erst nach der Serieneinführung durch die sogenannten „Felderfahrten“, also durch Rückmeldungen aus den Fahrzeugen im Kundeneinsatz, bekannt.

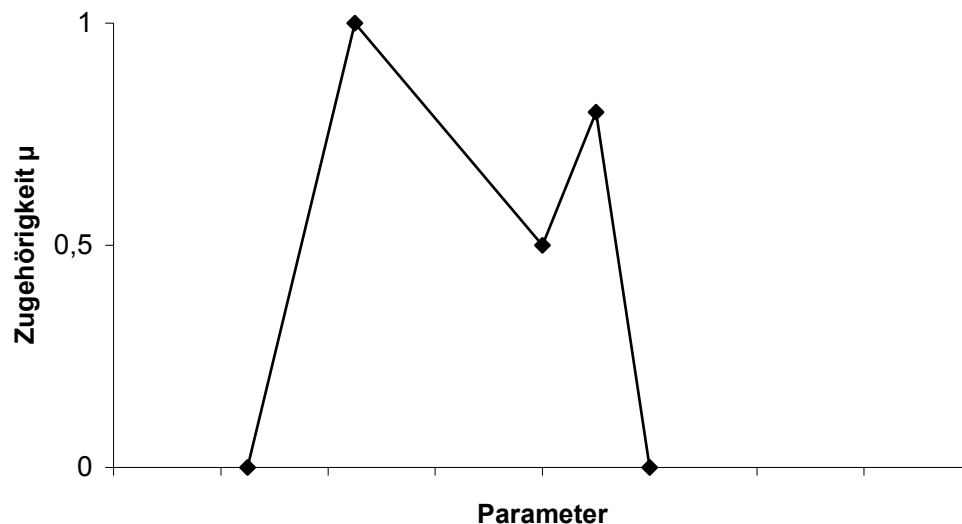
Für neue Systeme werden oft viele Standardkomponenten, die bereits in anderen Applikationen in Serie eingesetzt werden, verwendet. Über diese existieren somit „Felderfahrten“. Ihre Parameter sind von Beginn an ohne, oder nur mit einer sehr geringen Unsicherheit behaftet. Dies trifft jedoch nur zu, wenn die Randbedingungen wie der Einbauort, die Belastungen (mechanisch, elektrisch) oder auch die Anzahl der zu verbauenden Komponenten (zur Kostenermittlung) identisch mit der existierenden Serienapplikation sind. Insbesondere für die Kosten gilt, dass bei der Verwendung von Gleichteilen in verschiedenen Fahrzeugen, diese nochmals gesenkt werden könnten.

In jeder Entwicklungsphase müssen jeweils alle drei in dieser Arbeit verwendeten Verläufe (Punkt, Dreieck, Trapez) miteinander kombiniert und zusammengefasst werden. Die Erfassung der Entwicklungsunsicherheit aller Parameter und die Beherrschung des daraus resultierende Entwicklungsrisiko, entscheidet ganz

erheblich über die Entwicklungsdauer und die einzusetzenden Ressourcen und daher auch über die resultierenden Kosten.

Die erfolgreiche und wirtschaftliche Umsetzung eines Projektes hängt somit stark von der korrekten Auswertung der unscharfen Daten ab!

Es sind noch weitere Verläufe der Darstellung der Entwicklungsunsicherheit denkbar. Für die vorliegende Aufgabenstellung sind jedoch die genannten drei ausreichend und gut anwendbar. Eine weitere Möglichkeit ist ein, wie in Abbildung 54 dargestellter allgemeiner Verlauf (vgl. allgemeines Vieleck [BrS87]), denkbar.



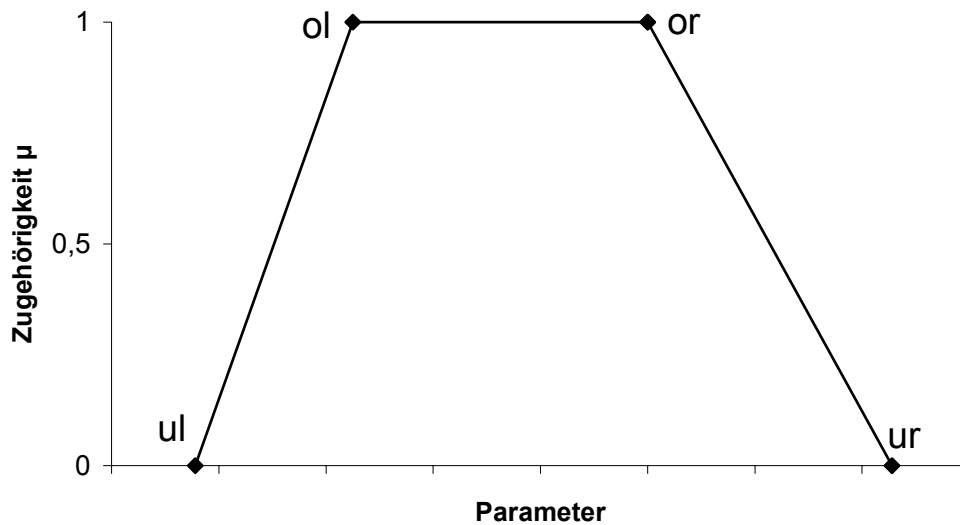
**Abbildung 54: Allgemeines Vieleck als alternative Darstellungsmöglichkeit unscharfer Parameter**

Durch dieses könnten Entwicklungskriterien in beliebiger Weise berücksichtigt werden. Auch die Anwendung von Verteilungsfunktionen (z. B. Normalverteilung, Poisson-Verteilung) ist denkbar. Bei diesen existieren jedoch, im Gegensatz zu den in dieser Arbeit verwendeten Verläufen, keine „Eckpunkte“. Die Zusammenfassung verschiedener Verteilungsfunktionen wäre daher nur über die Auswertung entsprechender Integrale möglich.

### **9.3 Quantifizierung der vagen Daten**

Die Unsicherheit des Gesamtsystems wird durch die Zusammenfassung der vagen Daten aller verwendeten Komponententypen zu einem Gesamtergebnis quantifiziert. Hierzu werden alle Parameter aller Typen einzeln beurteilt.

Die größte Daten-Unschärfe wird durch die Trapezform repräsentiert. Sie lässt sich durch vier beschreibende Punkte (Abbildung 55) angeben.



**Abbildung 55: Trapezform mit ihren beschreibenden Punkten**

Diese sind nachfolgend definiert als:

*unten\_links = ul, oben\_links = ol, oben\_rechts = or und unten\_rechts = ur.*

Werden die unscharfen Parameter zweier Typen jeweils durch eine Trapezform dargestellt, so können diese durch punktweise Addition der beschreibenden Eckpunkte zusammengefasst werden.

Die Dreiecksform und die Punktform sind Unterformen der Trapezform, d.h. sie können von dieser abgeleitet werden.

Zur Zusammenfassung der vagen Daten zweier Typen von denen jeder in drei verschiedenen Formen dargestellt werden kann, existieren bekanntlich

$$3^2 = 9 \quad (55)$$

verschiedene Kombinationsmöglichkeiten. Aus diesen resultiert die Ergebnisform.

Die Tabelle 7 stellt die Kombinationsmöglichkeiten dar. Hierin sind die drei verschiedenen Typen Punktform, Dreiecksform und Trapezform durch die Kurzzeichen P, D und T dargestellt (vgl. Abbildung 24).

<b>1</b>	P+P = P	<b>4</b>	D+P = D	<b>7</b>	T+P = T
<b>2</b>	P+D = D	<b>5</b>	D+D = D	<b>8</b>	T+D = T
<b>3</b>	P+T = T	<b>6</b>	D+T = T	<b>9</b>	T+T = T

**Tabelle 7: Kombinationsmöglichkeiten der verschiedenen „Formen“ der vagen Daten**

Die Ergebnisform der Addition zweier Typen unterschiedlicher Genauigkeit ist danach identisch mit der Art der ungenaueren Form (vgl. Kombination 4 und 6). Um

die Formen zusammenfassen zu können, ist für alle drei betrachteten Formen die einheitliche Angabe aller vier beschreibenden Punkte notwendig.

Für die beschreibenden Punkte der Trapezform gilt allgemein

$$ul \leq ol \leq or \leq ur . \quad (56)$$

Für die beiden anderen Formen (Dreieck und Punkt), die Sonderformen der Trapezform sind, lassen sich diese vier beschreibenden Punkte folgendermaßen angeben:

Dreiecksform

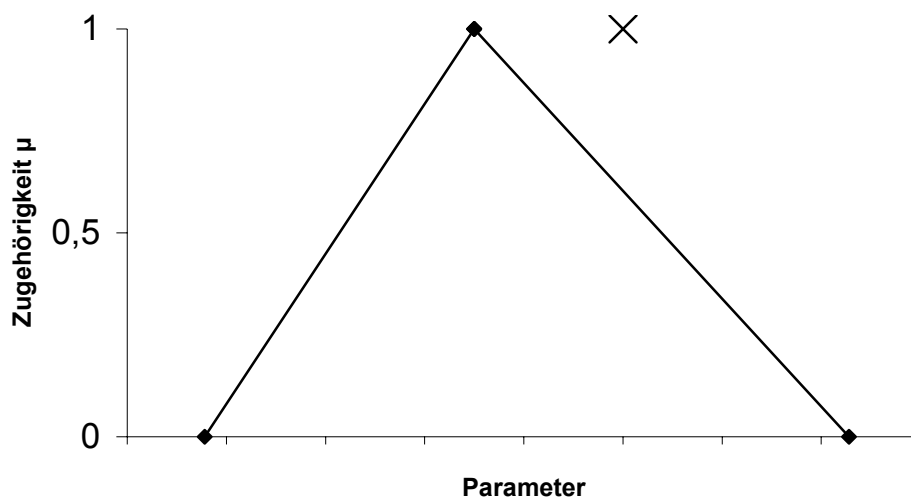
$$ul \leq ol = or \leq ur , \quad (57)$$

Punktform

$$ul = ol = or = ur . \quad (58)$$

Bei der vorliegenden Problemstellung entspricht der Wert der Punktform dem scharf definierten Wert. Dieser wurde auch beim Optimierungsvorgang verwendet.

Die Dreiecksform muss explizit definiert werden. Die Punkte  $ol$  und  $or$  sind nicht automatisch gleichzusetzen mit dem scharfen Wert, können also von diesem abweichen. Dies ist beispielhaft in der Abbildung 56 dargestellt.



**Abbildung 56: Dreiecksform, abweichend vom scharfen Wert**

Der scharfe Wert ist hier durch das Kreuz dargestellt. Er muss während der Entwicklung nicht der wahrscheinlichste Wert sein. In diesem Fall können die unscharfen Parameter eine Dreiecks- oder sogar eine Trapezform haben. Der wahrscheinlichste Wert (das wahrscheinlichste Intervall) muss den scharfen Wert nicht überdecken.

Zur Vereinfachung der Berechnung werden alle zu berücksichtigenden Daten der Komponenten anhand der vier dargestellten Unschärfepunkte unter Berücksichtigung der Gleichungen (56) und (57) definiert.

Die vagen Daten des resultierenden Systems  $\bar{V}$  berechnen sich letztlich durch die Summe der vagen Daten der Einzelkomponenten  $V_i$

$$\bar{V} = \sum_{i=1}^n \bar{V}_i = \sum_{i=1}^n \begin{bmatrix} ul_i \\ ol_i \\ or_i \\ ur_i \end{bmatrix} \quad (59)$$

wobei  $i$  der Index des jeweils zu verwendenden Komponententyps ist und  $n$  die Anzahl der Komponenten.

#### **9.4 Anwendung der Unsicherheitsabschätzung auf das Beispiel der EMB**

Bei der betrachteten EMB werden einige, bereits in anderen Applikationen erprobte Komponententypen verwendet. Diese Standardkomponenten können bereits entwickelte Stecker, Kabel, Lampen oder andere Komponenten sein. Für sie können alle Parameter als scharfe Werte angegeben werden. Einige Sensoren, wie der Gierratensensor oder auch der Drehratensensor, sind ebenfalls bereits in Serienfahrzeugen vorhanden. Andere Komponenten müssen jedoch neu für die Applikation entwickelt werden.

Beim hier betrachteten EMB-System müssen für folgende Komponenten neue Typen entwickelt werden:

1. Die vorhandenen Batterietechnologien (z. B. Blei-Säure-Batterie) sind nicht geeignet für die Anforderungen an die Dynamik und Zyklenfestigkeit in einem EMB-System.
2. Es gibt noch keine serienreifen elektrischen Bremsaktoren für Kfz.
3. Die Verteilung und Überwachung der Energie erfolgt in der VBox, die ebenfalls neu entwickelt werden muss.
4. Im Bremspedal wird zukünftig der Fahrer-Bremswunsch sensiert und in digitale Signale umgewandelt. Solche Sensoren existieren zwar bereits, müssen jedoch in das Bremspedal integriert und auf die Automobil-Anforderungen abgestimmt werden.
5. Die Parkbremse wird heute in hydraulisch gebremsten Fahrzeugen durch die Betätigung von Bowdenzügen realisiert. Dies ist zukünftig nicht mehr erforderlich. Ein einfacher (Tast-) Schalter könnte den Wunsch des Fahrers nach der Parkbremsbetätigung erkennen und weiterleiten. Er muss dementsprechend neu entwickelt werden.
6. Ein wesentliches Merkmal der EMB ist die Verwendung und Wandlung von elektrischer Energie in Bremsenergie. Daher muss auch deutlich mehr elektrische Energie im Fahrzeug verfügbar sein, als in gegenwärtigen Fahrzeugen. Diese Energie muss von einem Generator vollständig bereitgestellt werden. Weiterhin muss der Generator bei einer EMB höhere Dynamik-Anforderungen erfüllen. Es muss also ein neues Generator-Konzept entwickelt und umgesetzt werden.
7. Lichtmodule zur Steuerung des Bremslichtes gibt es bereits in Serie. Ein solches Modul muss jedoch an die sichere und echtzeitfähige Kommunikation, wie sie für X-by-Wire-Systeme gefordert wird, angepasst werden.

8. Die Steuergeräte für die Aktoren enthalten die Algorithmen zur Ansteuerung und müssen vollständig neu entwickelt werden.

### **9.5 Darstellung der Ergebnisse in den vier Phasen**

Wie zuvor dargestellt, verändern sich die vagen Daten der Komponententypen in den verschiedenen Entwicklungsphasen. Der Nutzen der Betrachtung der vagen Daten während der Entwicklungsphasen liegt in der Möglichkeit, den Projektfortschritt kontrollieren zu können. Es lässt sich in jeder Entwicklungsphase erkennen, welche Komponente die größte Entwicklungsunsicherheit in seinen Parametern enthält. Diese Komponente kann gezielt weiterentwickelt werden.

Der Entwickler bekommt durch die Berücksichtigung vager Daten eine Möglichkeit, die Entwicklungsunsicherheit quantifizieren zu können. Anhand der Begleitung des Entwicklungsprozesses lässt sich in jeder Phase der Entwicklung begründen, weshalb die Entwicklung in eine bestimmte Richtung erfolgt und warum bestimmte Technologien verwendet werden.

Der Entwicklungsprozess gliedert sich in der Automobilindustrie in folgende Phasen:

Phase 1: Konzeptphase,

Phase 2: Prototypenphase,

Phase 3: Vorserienphase,

Phase 4: Serieneinsatz.

Die vagen Daten lassen sich durch verschiedene Methoden in jeder Phase unterschiedlich genau ermitteln.

Die Ausfallrate beispielsweise lässt sich durch verschiedene Methoden zur Zuverlässigkeitsanalyse abschätzen. Näheres zur Bestimmung und Vorhersage der Ausfallraten, zu den Gründen für die Ausfälle und den Arten von Ausfällen findet man unter anderem in [Bir91], [Mey94], [Pau98], [PeM85], [PeM86].

Im Anhang 13.6 bis 13.9 sind die sich verändernden vagen Daten der verwendeten Komponententypen in den verschiedenen Phasen dargestellt. Vereinfachend soll während des gesamten Entwicklungsprozesses angenommen werden, dass sich die Typenauswahl nicht ändert.

Es werden ausschließlich die vagen Daten der ausgewählten Typen in den vier Entwicklungsphasen geändert. Dieser Veränderungsprozess wird nachfolgend beispielhaft an den Kosten des am Bremsaktors dargestellt.

#### **9.5.1 Konzeptphase**

Erste Ansätze und Ideen zur Realisierung eines Systems werden in der Konzeptphase zusammengetragen und verfolgt. Weiterhin werden Vergleiche zu ähnlichen bereits bestehenden Applikationen angestellt und verwendbare Komponenten für das neue System ausgesucht. Einige der benötigten Komponenten müssen jedoch neu oder weiterentwickelt werden. Um das Risiko eines neuen Systems hinsichtlich seiner Zuverlässigkeit, seiner Wirtschaftlichkeit oder der Verbaubarkeit abschätzen zu können, kann die Unsicherheit hinsichtlich der Unsicherheits-Parameter (Ausfallrate, Kosten, Gewicht, Volumen) einer jeden Komponente bereits in der ersten Entwicklungsphase, der Konzeptphase, quantifiziert werden. In den nachfolgenden Phasen wird diese Abschätzung

sukzessiv verfeinert. Diese Möglichkeit der Quantifizierung der Unsicherheit ist unabhängig vom Entwicklungsstand jeder Komponente.

Es sei angenommen, dass zu Beginn der Entwicklung der betrachteten EMB noch kein elektrischer Bremsaktor existiert. Er muss neu entwickelt werden. Die Kosten des Aktors in der Konzeptphase resultieren weitestgehend aus Schätzungen.

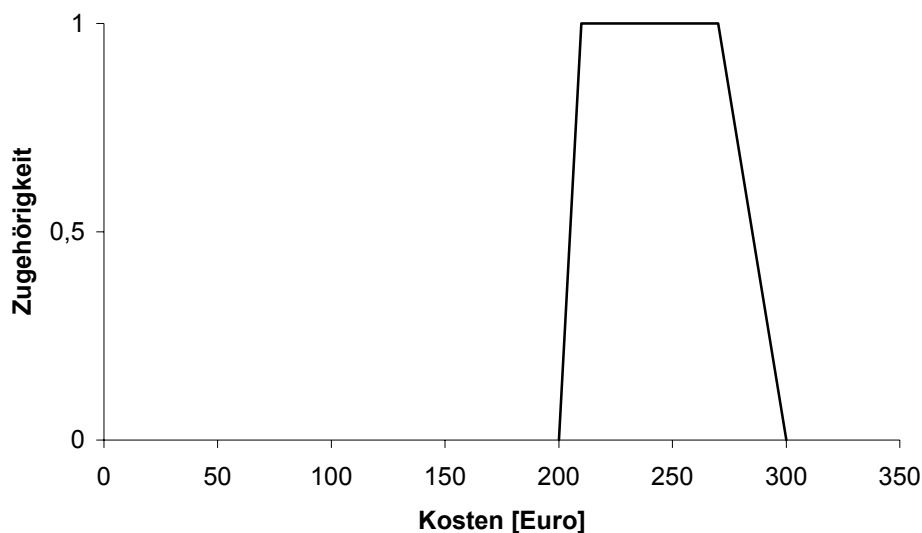
Zu Beginn des Kapitel 9 wurden die kostenbestimmenden Faktoren dargestellt. Sie sind bereits in der Konzeptphase mit einem wahrscheinlichsten Intervall abschätzbar.

Für das Beispiel des Bremsaktors werden die Kosten in der Konzeptphase mit einer Trapezform abgeschätzt. Zusätzlich zum wahrscheinlichsten Intervall existieren zwei Unsicherheitsintervalle verminderter Zugehörigkeit. Die Kostenunsicherheit ergibt sich einerseits aus den hohen Anforderungen an das Package und an die Leistungsfähigkeit des Aktors, andererseits aus der Ungewissheit, welche technischen Probleme sich während der Entwicklung noch ergeben. Die Erfüllung der Anforderungen könnte neben der Anwendung bekannter Technologien die Entwicklung neuer Technologien erfordern.

Die Tabelle 8 und die dazugehörige Abbildung 57 stellen die Kostenabschätzung des Bremsaktors während der Konzeptphase dar.

	links		rechts	
	unten	oben	oben	unten
Kosten [€]	200	210	270	300

**Tabelle 8: Vage Kosten des Bremsaktors nach Abschluss der Konzeptphase**



**Abbildung 57: Vage Kosten des elektrischen Bremsaktors nach Abschluss der Konzeptphase**

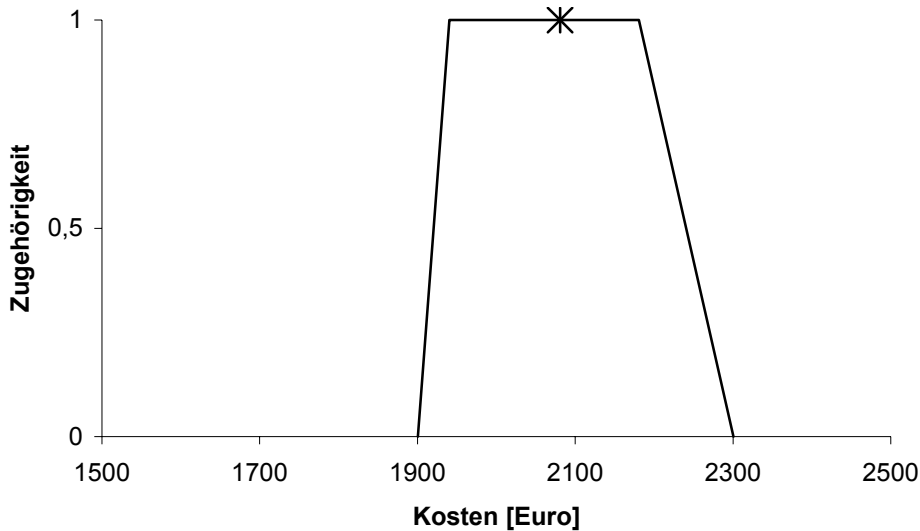
Die vagen Daten aller anderen Komponententypen werden ebenfalls auf diese Art erfasst. Berücksichtigt man nur die vagen Kosten dieser Komponente und nimmt man für alle anderen Komponenten exakte Werte an, so resultieren die in Tabelle 9 dargestellten vagen Kosten.



Es ergibt sich nach Abschluss der Konzeptphase für das Gesamtsystem der in Tabelle 9 und Abbildung 58 dargestellte unscharfe Verlauf.

	links		rechts	
	unten	oben	oben	unten
Kosten [€]	1900,60	1940,60	2180,60	2300,60

**Tabelle 9: Vage Kosten des Gesamtsystems nach Abschluss der Konzeptphase unter ausschließlicher Berücksichtigung der vagen Daten des Bremsaktors**



**Abbildung 58: Vage Kosten des Gesamtsystems nach Abschluss der Konzeptphase unter ausschließlicher Berücksichtigung der vagen Kosten des Bremsaktors**

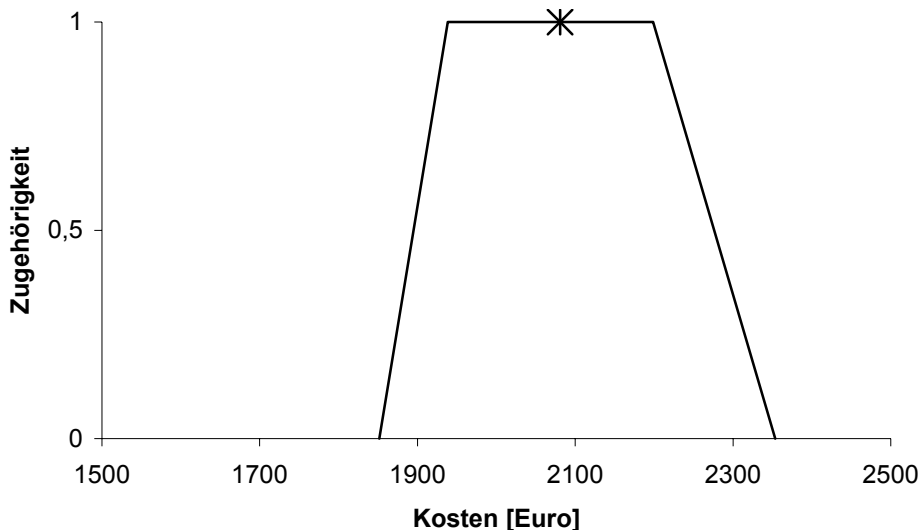
Das Kreuz in Abbildung 58 markiert den Wert, der sich durch die ausschließliche Berücksichtigung der scharfen Werte zur Optimierung ergab. Ein Vergleich mit der Darstellung in Abbildung 57 zeigt, dass der unscharfe Verlauf artgleich geblieben (Trapez blieb Trapez) und in seinem Wertebereich verschoben ist. Da bei der EMB 4 Bremsaktoren benötigt werden, verbreitert sich der vage Bereich um den Faktor 4 im Vergleich zu den Werten eines Bremsaktors. Bei dessen Kosten hat der unscharfe Bereich eine Gesamtbreite von 100 € (200 € bis 300 € je Aktor). Beim Gesamtsystem resultiert daraus eine Kosten- Ungenauigkeit von 400 € (1900,60 € bis 2300,60 €).

Veränderte Formen beim Gesamtsystem resultieren erst dann, wenn auch die vagen Daten der anderen verwendeten Typen berücksichtigt werden.

Für die Kosten des Beispielsystems resultieren in diesem Fall mit Abschluss der Konzeptphase folgende Werte und Verlauf (Tabelle 10, Abbildung 59).

	links		rechts	
	unten	oben	oben	unten
Kosten [€]	1851,6	1938,6	2198,6	2353,6

**Tabelle 10: Vage Kosten des Gesamtsystems nach Abschluss der Konzeptphase unter Berücksichtigung der vagen Daten aller Typen**



**Abbildung 59: Vage Kosten des Gesamtsystems nach Abschluss der Konzeptphase unter Berücksichtigung der vagen Kosten aller Komponententypen**

Für alle Parameter existieren nun trapezförmige Verläufe. Da sich die Unschärfe vergrößert hat (mehr vage Kosten wurden berücksichtigt) ist das Intervall breiter als in Abbildung 57 und Abbildung 58. Die resultierenden Kosten können nun, um bis zu 502 € variieren, wobei die wahrscheinlichsten Kosten in dieser Konzeptphase in einem 260 € „breiten“ Intervall liegen.

Ein Vergleich der Werte in der Abbildung 58 und der Abbildung 59 zeigt, dass in dieser Beispielapplikation der betrachtete Bremsaktor den größten „Unsicherheitsfaktor“ darstellt. Dies liegt daran, dass es sich bei dem Aktor um eine vollständige Neuentwicklung handelt und die Kosten infolgedessen einen großen Einfluss auf die Kostenunsicherheit der Applikation haben. Dieser mit großer Unsicherheit belegte Bremsaktor ist viermal in der Applikation vorhanden.

### 9.5.2 Prototypenphase

Nachdem das Gesamtkonzept definiert ist, werden Prototypen der neu zu entwickelnden und zu verändernden Komponenten erstellt. In der Prototypenphase sollten die Anforderungen an die Komponenten mit dem voraussichtlich Machbaren abgeglichen sein und Ideen zur Realisierung existieren.

Die erarbeiteten Konzepte werden durch die Prototypen der Komponenten und das daraus resultierende Gesamtsystem validiert. Für den Bremsaktor geschieht dies insbesondere hinsichtlich der Dynamikanforderungen, der Bauform oder auch der

elektromechanischen Energiewandlung. Die Anforderungen an Prototypen sind in der Regel jedoch noch nicht so hoch, wie die an Serienkomponenten.

Neben den Materialien und der Produktion, ist auch der Entwicklungsaufwand bei der Abschätzung der Kosten zu berücksichtigen.

Aus den erstellten Konzepten und den Anforderungen lässt sich der verbleibende Entwicklungsaufwand abschätzen und anhand der erstellten Konzepte sollte absehbar sein wie, aber auch in welcher Zeit, etwas realisiert werden kann.

Es muss meist ermittelt werden, welche Techniken, Technologien und Prozesse eingesetzt werden müssen und auch verfügbar sind, damit alle Komponenten später in Serie produziert und als Gesamtsystem eingebaut werden können. Eventuell müssen in oder nach der Prototypenphase die Techniken, Technologien oder Prozesse neu entwickelt oder angepasst werden. Dieser Aufwand ist nur ungenau abschätzbar.

Der Aufwand für unvorhergesehene Probleme kann auch in diesem Stadium nur pauschal berücksichtigt werden. Eventuell helfen Erfahrungen aus vorangegangenen Projekten.

Die Anzahl der beschäftigten Entwickler und der zeitliche Aufwand lässt sich direkt, mit nur geringer Unsicherheit, in finanzielle Mittel umrechnen.

Die Materialien werden im wesentlichen in der Prototypenphase definiert. Die daraus folgende Kostenunsicherheit ist begrenzt.

Einerseits verringert sich also die Unsicherheit, da die Komponenten immer genauer hinsichtlich ihrer Kosten abgeschätzt werden können, andererseits lässt sich der Entwicklungsaufwand nur relativ unpräzise abschätzen. Die Unsicherheit könnte so wieder steigen.

Nach Abschluss der Prototypenphase ergibt sich für die Kosten des Bremsaktors, wie auch während der Konzeptphase, ein Trapezverlauf (Tabelle 11).

			links		rechts	
			unten	oben	oben	unten
Bremsaktor	Kosten	[€]	195	210	275	295

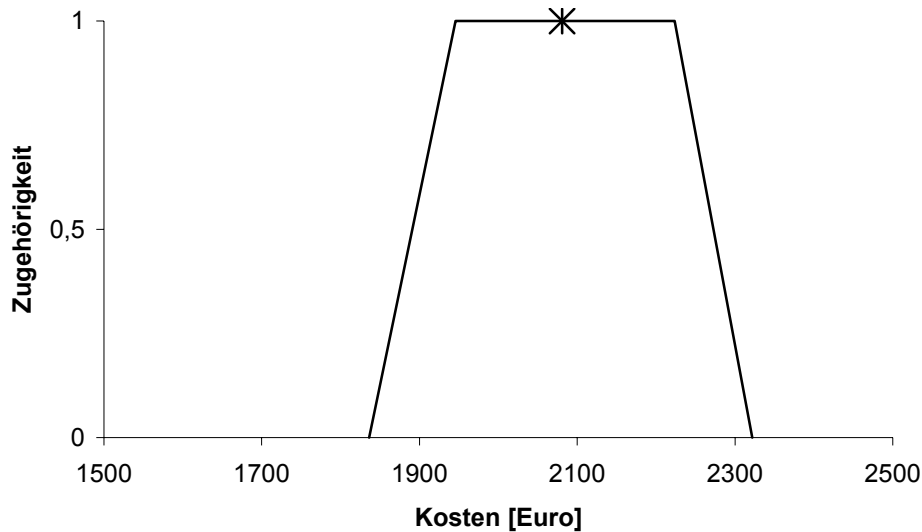
**Tabelle 11: Vage Kosten des Bremsaktors nach Abschluss der Prototypenphase**

Dies wirkt sich unter Berücksichtigung der vagen Daten der anderen Komponenten wie folgt auf das Gesamtsystem aus (Tabelle 12, Abbildung 60).

			links		rechts	
			unten	oben	oben	unten
Bremsaktor	Kosten	[€]	1836,1	1945,6	2223,6	2321,6

**Tabelle 12: Vage Kosten des Gesamtsystems nach Abschluss der Prototypenphase unter Berücksichtigung der vagen Daten aller Typen**

Das resultierende Unsicherheitsintervall ist im Vergleich zur Konzeptphase nach Abschluss der Prototypenphase etwas schmaler.



**Abbildung 60: Vage Daten des Gesamtsystems nach Abschluss der Prototypenphase unter Berücksichtigung der vagen Daten aller Komponententypen**

Es lässt sich die erwartete und beabsichtigte Präzisierungstendenz erkennen (vgl. Abbildung 59 mit Abbildung 60). Das Kosten-Intervall des Gesamtsystems wird schmaler, ist aber noch immer trapezförmig.

### 9.5.3 Vorserienphase

Nach Abschluss der Prototypenphase sollte das Gesamtsystem mindestens hinsichtlich seiner Primärfunktionen funktionsfähig sein. In der anschließenden Vorserienphase soll es zur Serienreife weiterentwickelt werden. Hier werden alle Sekundärfunktionen integriert. Ein Abgleich mit dem geplanten Serienstand des Zielfahrzeugs ist erforderlich und fahrzeugspezifische Anpassungen werden vorgenommen. Die Bauteileform und das Package werden optimiert.

Bei der EMB würden in der Vorserienphase beispielsweise Diebstahlschutz-Algorithmen, aber auch Komfort-Funktionen, wie die Anfahrhilfe am Berg, in das System integriert werden. Auch im Vorserien-Stadium könnten sich die Vorgaben für die Applikation noch ändern.

In den Bremsaktor werden spätestens in dieser Phase die Kommunikation und die Energieversorgung integriert. Er soll anschließend alle an ihn gestellten Anforderungen, beispielsweise auch die des Einbauortes [Vol00], erfüllen.

Während der Vorserienphase werden die Gesamtkosten des Systems ermittelt. Die Entwicklungskosten stehen fest, die Systemkosten werden mit den Zulieferern verhandelt und auch die Produktionskosten sind kalkulierbar. Aus diesen ziemlich exakten Daten ergeben sich die Minimalkosten des Systems. Hinzu kommen noch Kosten, die sich durch spätere Garantieleistungen für das System ergeben können. Auch eventuelle Rückrufaktionen bei Systemmängeln müssen einkalkuliert werden.

Es lassen sich nun die wahrscheinlichsten Kosten für die Komponenten und somit für das Gesamtsystem angeben. Der Wert mit der höchsten Zugehörigkeit setzt sich aus den tatsächlichen Systemkosten zuzüglich einer Pauschalen für Garantie und Rückruf zusammen. Diese Pauschale entsteht durch Umlegung der möglicherweise entstehenden Gesamtkosten auf das einzelne System.

Die Bereiche verminderter Wahrscheinlichkeit hin zu den höheren bzw. niedrigeren Kosten unterscheiden sich in ihrer Breite. Die Seite mit den niedrigeren Kosten wird schmaler sein, als die zu den höheren. Das System wird nicht deutlich billiger werden, kann aber im Falle gravierender Mängel sehr hohe Zusatzkosten verursachen.

Für den Bremsaktor verlagern sich die wahrscheinlichsten Kosten in den unteren Bereich des zuvor wahrscheinlichsten Intervalls der Prototypenphase. Die Intervalle verminderter Wahrscheinlichkeit behalten jedoch ihre Schnittpunkte mit der Abszisse. Nach Abschluss der Vorserienphase können die Kosten des Bremsaktors daher mit einem dreiecksförmigen Verlauf abgeschätzt werden (Tabelle 13, Tabelle 14 und Abbildung 61).

		links		rechts	
		unten	oben	oben	unten
Bremsaktor	Kosten [€]	195	215	215	295

**Tabelle 13: Vage Daten des Bremsaktors nach Abschluss der Vorserienphase**

Bei Berücksichtigung der vagen Kosten aller Komponenten ergeben sich folgende Werte und Verläufe (Tabelle 14, Abbildung 61).

		links		rechts	
		unten	oben	oben	unten
Bremsaktor	Kosten [€]	1852,1	1950,6	1950,6	2316,6

**Tabelle 14: Vage Kosten des Gesamtsystems nach Abschluss der Vorserienphase unter Berücksichtigung der vagen Daten aller Typen**



**Abbildung 61: Vage Kosten des Gesamtsystems nach Abschluss der Vorserienphase unter Berücksichtigung der vagen Daten aller Komponententypen**

Es lässt sich eine Abweichung der wahrscheinlichsten Kosten vom Ergebnis der Optimierung (Kreuz in Abbildung 61) erkennen.

### 9.5.4 Serieneinsatz

In den vorangegangenen Phasen wurden die unscharfen Werte nach Abschluss der jeweiligen Phase bewertet. Dies ist in der Serienphase nicht möglich. Dies würde bedeuten, dass die Parameter erst quantifiziert werden, wenn das System nicht mehr am Markt ist.

Es stellt sich die Frage, wie die nach der Vorserienphase existierenden vagen Daten noch exakter angegeben werden können. Welche Voraussetzungen müssen erfüllt sein?

Für die Bewertung der Kosten können folgende Überlegungen herangezogen werden:

Mit zunehmendem Alter des Systems verringert sich die Wahrscheinlichkeit, dass dieses schwerwiegende, unerkannte Fehler enthält, die einen (teuren) Rückruf des Systems erforderten. Die unkalkulierbaren Kosten streben dann gegen ein Mindestmaß. Idealerweise sollte dieses Minimum gleich Null sein.

In der frühen Phase nach der Einführung des Systems kann das Kosten-Risiko auch aus der Frage der Kunden-Akzeptanz des Systems bestehen. Das dargestellte Beispiel der EMB ist ein sicherheitsrelevantes System. Dieses muss, um vom Kunden auch als sicheres System akzeptiert zu werden, besser als das vergleichbare System der hydraulischen Bremse sein. Sollte es in einer frühen Phase - nach der Systemeinführung - im Feld zu einem Unfall kommen, der auf mangelnde Systemsicherheit zurückzuführen ist, können die Kosten für die (Wieder-) Herstellung des Images die Kosten für die Behebung des eigentlichen Fehlers um ein Vielfaches übersteigen. Mit steigendem Alter des Systems sollte dieses jedoch mit dem beabsichtigten Image (z. B. „...das System ist sicher und besser als das vorherige...“) im Markt akzeptiert und eventuelle Systemfehler erkannt sein. Das Risiko unvorhersehbarer Kosten sinkt also mit steigendem Alter des Systems.

Die Kosten zur Wiederherstellung des Images werden auf alle Systeme umgelegt. Mit der steigenden Anzahl der betrachteten Systeme sinkt die Unsicherheit für jedes einzelne System.

Für den Bremsaktor wurde im Anschluss an die Phase der Frühausfälle zur Darstellung der Kosten eine Dreiecksform mit geringer Unsicherheit (Tabelle 15) gewählt.

			links		rechts	
			unten	oben	oben	unten
Bremsaktor	Kosten	[€]	205	215	215	223

**Tabelle 15: Vage Kosten des Bremsaktors im Anschluss an die Phase der Frühausfälle in der Serienphase**

Im weiteren Serieneinsatz werden diese Werte ständig an die aktuellen Daten angepasst und gegebenenfalls korrigiert. Diese Korrektur geschieht für alle Komponenten des Systems, abhängig davon, wie sorgfältig die Fehler analysiert werden.

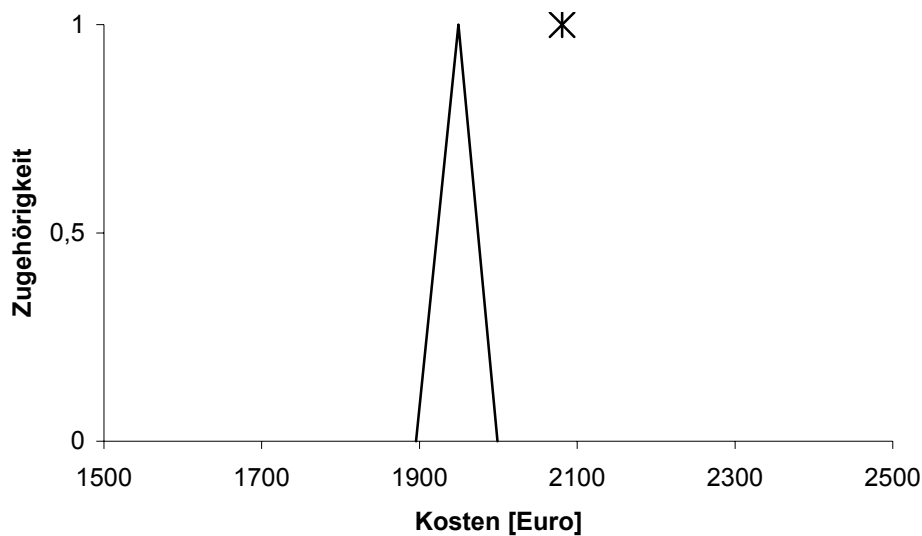
In der Tabelle 16 und der sich hieraus ergebenden Abbildung 62 sind die vagen Kosten aller Komponenten berücksichtigt. Ein Vergleich mit den ursprünglichen Annahmen aus der Konzeptphase (Abbildung 58) zeigt, dass das nun resultierende Intervall der Kosten vollständig in dem der ursprünglichen Annahme liegt und das Intervall der wahrscheinlichsten Kosten zu teuer abgeschätzt war. Die

kostenbestimmende Komponente mit der größten Unsicherheit war der Bremsaktor. Dessen Kosten verringerten sich im Laufe der Entwicklung stetig und zum Ende hin deutlich.

Dies ist nachfolgend dargestellt (Tabelle 16, Abbildung 62).

		links		rechts	
		unten	oben	oben	unten
Bremsaktor	Kosten [€]	205	215	215	223

**Tabelle 16: Vage Daten des Gesamtsystems im Anschluss an die Phase der Frühausfälle in der Serienphase unter Berücksichtigung der vagen Daten aller Typen**



**Abbildung 62: Vage Kosten des Gesamtsystems im Anschluss an die Phase der Frühausfälle in der Serienphase unter Berücksichtigung der vagen Daten aller Komponententypen**

Je länger das System in Serie eingesetzt wird und je höher die Stückzahl ist, umso genauer werden sich später auch die Kosten und die Ausfallrate abschätzen lassen.

## 10 Anwendbarkeit und Erweiterbarkeit der dargestellten Optimierungsmethoden

Bei der in dieser Arbeit durchgeführten Systemoptimierung erfolgte eine Bewertung von Parametern in unterschiedlichen Wertebereichen und Einheiten. Dies wird durch die Einführung der Wirtschaftlichkeitsmaßzahl  $W$  ermöglicht. Es ergibt sich für das vorliegende Problem eine Reduzierung der Freiheitsgrade von 6 auf 2.

Es ist denkbar, diese Wirtschaftlichkeitsmaßzahl mit weiteren Parametern zu ergänzen. Eine Anpassung auf ähnliche Aufgabenstellungen ist möglich.

In der Industrie wird es immer wichtiger, auch die Energiebilanz bei der Erstellung von Typen oder deren Wiederverwendbarkeit (Recycling) zu berücksichtigen. Diese Parameter beeinflussen ebenfalls die Wirtschaftlichkeit eines Systems. Nachdem diese Terme quantifiziert sind, kann die Wirtschaftlichkeitsmaßzahl, ähnlich wie für die in Kapitel 2.1 beschriebenen bisherigen Wirtschaftlichkeitsparameter, erweitert werden.

Die Voraussetzung für die Erweiterung der Wirtschaftlichkeitsmaßzahl bzw. die Reduzierung der Freiheitsgrade ist, dass die Parameter in einem begründbaren Verhältnis stehen. Solche Verhältnisse können in der Betrachtung der Wirtschaftlichkeit, der Umweltverträglichkeit oder ähnlichem bestehen.

Wenn man die Datenbasis und das System nicht verändert, bleibt die Lösungsraumgröße konstant. Sie ist unabhängig von der Anzahl der Freiheitsgrade. Das Finden der optimalen Variante ist dennoch abhängig von der Lösungsraumgröße und der eingesetzten Methode.

Ist die Zusammenfassung der Parameter zur Wirtschaftlichkeitsmaßzahl nicht möglich, so wird eventuell die Online-Beurteilung der Varianten schwierig. Es muss dann eine andere geeignete Beurteilungsmöglichkeit der Parameter gefunden und diese auf die einzusetzende Optimierungsmethode und die Aufgabenstellung angepasst werden.

Insbesondere der Graphenalgorithmus und der Genetische Algorithmus sind auch auf andere ähnliche Aufgabenstellungen anwendbar.

Bei der Neuro-Fuzzy-Methode müsste die Regelbasis geändert werden. Man hätte bei 6 Freiheitsgraden 6 Prämissen, um auf die Konklusion zu schließen. Der Aufwand würde sich deutlich vergrößern und die Regelbasis wäre nicht mehr überschaubar. Der ursprüngliche Vorteil der Neuro-Fuzzy-Methode, nämlich seine übersichtliche und einfach strukturierte Entscheidungslogik, wäre bei der Erhöhung der Freiheitsgrade hinfällig.



## 11 Zusammenfassung

Ein Ziel der vorliegenden Arbeit war es, für das Problem der kombinatorischen Optimierung, insbesondere die beschriebene Wirtschaftlichkeits-Zuverlässigkeits-Optimierung, eine geeignete Optimierungsmethode zu finden und anzupassen. Es wurden daher im Kapitel 4 drei Methoden betrachtet, von denen jede ihrer individuellen Stärken und Schwächen hat.

### 11.1 Optimierungsmethoden

Die Forderung nach einer „guten“ Lösung erfüllt der Graphenalgorithmus am zuverlässigsten. Er liefert immer optimale Ergebnisse, ist jedoch nicht für jede Lösungsraumgröße anwendbar. Diese Komplexitätsgrenze wird bereits bei einfachen realen Systemen schnell erreicht.

Obwohl bei Genetischen Algorithmen der Lösungsraum heuristisch durchsucht wird, ist es durch die Anwendung eines im Rahmen dieser Arbeit gefundenen adaptiven Abbruchkriteriums (Kapitel 8.1) gelungen, mit einer sehr hohen Wahrscheinlichkeit die optimale Lösung zu finden. Trotz der Einführung des adaptiven Abbruchkriteriums verbleibt eine geringe Restfehlerwahrscheinlichkeit, die jedoch vernachlässigbar klein ist. Nachdem das adaptive Abbruchkriterium gefunden und seine Parameter eingestellt waren, wurde bei allen durchgeführten Optimierungen mit dem Genetischen Algorithmus das Optimum gefunden.

Bei der Neuro-Fuzzy-Methode weichen die gefundenen Lösungen leider oft stark vom Optimum ab.

Der Aufwand zur Eingabe des Systems ist für alle Methoden gleich. Alle drei sind unabhängig von der Topologie des zu optimierenden Systems und können die Einbauorte berücksichtigen. Die Optimierungsmethoden erfordern nur die Definition der Komponenten des Systems und die vorgesehenen Einbauorte.

Der Berechnungsaufwand ist beim Genetischen Algorithmus und der Neuro-Fuzzy-Methode etwa identisch. Die Dauer der Optimierung ist abhängig von der Einstellung der Optimierungsparameter. Beide Methoden können sehr komplexe Systeme optimieren.

Der Graphenalgorithmus sollte dann angewendet werden, wenn die Komplexität des Systems eine Berechnung zulässt. Er findet stets die optimale Lösung. Durch die Nicht-Beherrschung von Systemen beliebig hoher Komplexität ist er jedoch nicht allgemeingültig anwendbar. Ist der Aufwand für die vollständige Berechnung durch den Graphenalgorithmus zu groß, so sollte die Methode der Genetischen Algorithmen angewendet werden. Sie findet auch bei diesen „großen“ Systemen die optimale Lösung mit einer sehr hohen Wahrscheinlichkeit.

Die Neuro-Fuzzy-Methode ist, da sie oft suboptimale Lösungen findet, nicht für die in dieser Arbeit betrachtete Art der Optimierung anwendbar.

### 11.2 Bewertung der Entwicklungsunsicherheit

Für alle drei Methoden wird dieselbe Datenbasis verwendet. Ein weiteres Ziel dieser Arbeit war die Quantifizierung und der anschließende Vergleich der unscharfen Entwicklungsparameter der verwendeten Systemtypen.

Im Kapitel 5 ist diese Methode der vagen Daten dargestellt, durch die sich zukünftig unscharfe Entwicklungsparameter im Entwicklungsprozess quantifizieren lassen. Die

Unsicherheit der einzelnen Komponententypen wird auf das resultierende System übertragen.

In jeder Phase des Entwicklungsprozesses lässt sich der aktuelle Systemstatus mit den Vorgaben vergleichen. Weiterhin lässt sich dieser Systemstatus mit dem Status des Systems aus vorangegangenen Phasen vergleichen und so der Entwicklungsfortschritt im wirtschaftlichen und technischen Umfeld angeben. Im Kapitel 9 wird beispielhaft eine Abschätzung mit vagen Daten am Entwicklungsprozess einer EMB gezeigt.

Für alle drei Optimierungsmethoden wurden im Rahmen dieser Arbeit je ein Tool mittels der Programmiersprache C++ programmiert. Außerdem wurde das Tool für die Methode der Genetischen Algorithmen, da diese als am besten anzuwendende Methode identifiziert wurde, um die Möglichkeit der Bewertung der Entwicklungsunsicherheit (vage Daten) erweitert.

### **11.3 Weiterführende Aufgaben**

Eine Absicht dieser Arbeit war im wesentlichen das Finden einer Optimierungsmethode und die Anpassung dieser an das beschriebene Problem. Da in der Praxis die Kriterien eines optimalen Systems jedoch weitgreifender sind, könnten die Methode und das entsprechende Tool zukünftig um weitere Parameter erweitert werden.

Die Ausfallrate von Komponententypen ist tatsächlich nicht konstant, wie vereinfachend in dieser Arbeit angenommen wurde. Die Belastungen der Typen und die Umgebungsbedingungen an den verschiedenen Einbauorten sind wesentliche beeinflussende Faktoren. Eine Möglichkeit zur Berechnung der von der Belastung abhängigen Ausfallrate ist in [Usd86] dargestellt. Diese variable Ausfallrate könnte in weiterführenden Untersuchungen diese Arbeit ergänzen.

Ein weiterer denkbarer Optimierungsaspekt, der zukünftig berücksichtigt werden könnte, ist der Lebenszyklus (Lifecycle) der Komponententypen. Wie viel Energie wird bei der Produktion der betrachteten Typen benötigt? Welche Werkstoffe wurden verwendet? Sind diese oder ganze Teile der Typen wiederverwendbar? Solche und ähnliche Aspekte könnten in zukünftigen Arbeiten quantifiziert und als Erweiterung in die vorliegende Optimierung eingearbeitet werden.

In Kapitel 7.1 wurden die bei der Entwicklung von Systemen angewandten Sicherheitsanalysemethoden angesprochen. Für zukünftige Aufgaben ist es denkbar, die Ergebnisse von Sicherheitsanalysen in die Optimierung einzubringen. Hierdurch würde auch eine Topologieoptimierung möglich. In diese könnte man wiederum die Berechnung der Ausfallrate und die sich daraus ergebende Veränderung der Wirtschaftlichkeit der Typen einbeziehen.

Nachdem die besten Parameter der zu bevorzugenden Methode gefunden sind, sollte die Praxistauglichkeit an verschiedenen Projekten getestet werden.

Bei der Methode der Bewertung der vagen Daten wird die Unsicherheit der Wirtschaftlichkeit und der Zuverlässigkeit bewertet. Während der Entwicklung ist nicht immer sichergestellt, dass mit den vorgesehenen Komponenten, insbesondere mit den neu zu entwickelnden, auch die gewünschten Funktionen erfüllt werden können. Eine Möglichkeit für ergänzende Untersuchungen zu der vorliegenden Arbeit ist daher die Integration eines weiteren, diese Funktions-Unsicherheit berücksichtigenden Parameters.

## 12 Literaturverzeichnis

- [Akl89] Akl, S. G.: The Design and Analysis of Parallel Algorithms. Prentice-Hall International, Englewood Cliffs, New Jersey, 1989.
- [BGU95] Becker, P., Gottschalk, A., Ulbricht, H.: Qualität und Zuverlässigkeit elektronischer Bauelemente und Geräte - bestimmen, voraussagen und sichern. expert-verlag, Renningen-Malmsheim, 1995.
- [Bin01] Binfet-Kull, M.: Entwicklung einer Steer-by-Wire-Architektur nach zuverlässigkeits- und sicherheitstechnischen Vorgaben. Dissertation BUGH Wuppertal, Verlag Mainz, Aachen, 2001.
- [Bir91] Birolini, A.: Qualität und Zuverlässigkeit technischer Systeme. 3. Auflage. Berlin, Heidelberg, New York: Springer Verlag, 1991.
- [BrS87] Bronstein, I. N., Semendjajew, K. A.: Taschenbuch der Mathematik. 23. Auflage, Verlag Harri Deutsch, Thun und Frankfurt/Main, 1987.
- [Dar00] Darwin, C. R.: Über die Entstehung der Arten durch natürliche Zuchtwahl oder die Erhaltung der begünstigten Rassen im Kampf um's Dasein. Köln, Parkland-Verlag, 2000.
- [Die00] Diestel, R.: Graphentheorie. Elektronische Ausgabe 2000, 2. Auflage, Springer-Verlag, Heidelberg, 1996, 2000.
- [Dod91] Department of Defense: Military Handbook- Reliability prediction of electronic equipment (MIL-HDBK-217F). Washington DC 20301, 02.12.1991.
- [Dör01] Döring, M.: Optimierung der Verfügbarkeit und des Aufwands von Kfz-Bordnetzen mit Hilfe der Neuro-Fuzzy-Methode. In Tagungsband „Automatisierungs- und Assistenzsysteme für Transportmittel“, Braunschweig 2001, Fortschritt-Berichte / VDI : Reihe 12, Verkehrstechnik, Fahrzeugtechnik, VDI-Verlag, Düsseldorf, 2001.
- [GRS00] Görz, G., Rollinger C.-R., Schneeberger, J. (Hrsg.): Handbuch der künstlichen Intelligenz. Speziell: Borgelt, C., Timm, H., Kruse, R.: Kapitel 9: Unsicheres und vages Wissens. Oldenburg, München, 2000.
- [Has98] Hasenknopf, T.: Das künstliche Hirn: Neuronale Netze in der Technik 1998. Serie in ELRAD: Magazin für Elektronik und technische Rechneranwendungen, ELRAD, Hannover, 1998.
- [Heu97] Heuer, J.: Neuronale Netze in der Industrie. Deutscher Universitäts-Verlag Wiesbaden, Gabler, Wiesbaden, 1997.
- [KrS96] Kröse, B., Smagt, P. van der: An introduction to neural networks. University of Amsterdam, Amsterdam, 1996.
- [KNK95] Klawonn, F., Nauck, D., Kruse, R.: Generating Rules from Data by Fuzzy and Neuro-Fuzzy Methods. In: *Proc. Fuzzy-Neuro-Systeme'95*, pp. 223-230, Darmstadt, 1995.
- [Lnk01] Link zur Internetseite der Universität in Magdeburg:  
[http://fuzzy.cs.uni-magdeburg.de/welcome\\_d.html](http://fuzzy.cs.uni-magdeburg.de/welcome_d.html)
- [Lüb99] Lübke, A.: Systematischer Bordnetzentwurf - Optimierung der Bordnetzarchitektur mit Hilfe von genetischen Algorithmen. Dissertation TU- Dresden, Verlag Mainz,

Wissenschaftsverlag, Aachen, Wolfsburg, 1999.

- [MeM02]** Meinel, C., Mundhenk, M.: Mathematische Grundlagen der Informatik: mathematisches Denken und Beweisen: eine Einführung. 2., durchgel. Aufl., Stuttgart, Teubner 2002.
- [Mey00]** Meyer, M.: Sicherheitsanalyse für einen Spurwechsel- und Abbiegeassistent. Diplomarbeit an der BUGH Wuppertal, Fachbereich 14 – Sicherheitstechnik, Wuppertal, 2000.
- [Mey94]** Meyna, A.: Zuverlässigkeitsbewertung zukunftsorientierter Technologien. Vieweg Verlag, Wiesbaden, 1994.
- [Mic94]** Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer Verlag, Berlin, 1994.
- [Usd86]** US Department of Defense: Military Handbook – MIL HDBK, reliability of electronic equipment. US Department of Defense, Band 217E (Hauptband), Washington DC, 1986.
- [Mue95]** Mühlenbein, H.: Genetische Algorithmen und Evolutionstheorien. Auf der Suche nach verschollenen Schätzen. GMD-Spiegel, S. 12-19, Heft 2/95.
- [NaK95]** Nauck, D., Kruse, R.: Neuronale Fuzzy-Systeme. In: Georg Dorffner, Knut Möller, Gerhard Paaß, and Stephan Vogel (Eds.), *Konnektionismus und Neuronale Netze. Beiträge zur Herbstschule HeKoNN95*, pp. 1-10, GMD-Forschungszentrum Informationstechnik GmbH, Münster/Westfalen, 1995.
- [NaK99]** Nauck, U., Kruse, R.: Konzeption und Implementierung eines Neuro-Fuzzy-Datenanalysetools in Java. Diplomarbeit an der TU - Braunschweig, Institut für Betriebssysteme und Rechnerverbund, Braunschweig, 03.01.99.
- [Otw93]** Ottmann, T.; Widmayer, P.: Algorithmen und Datenstrukturen. 2. vollständig überarbeitete und erweiterte Auflage. BI-Wissenschaftsverlag, Mannheim, Leipzig, Wien, Zürich, 1993.
- [OKC01]** Olsson, P., Kelling, N., Chaumette, P.: EU-Project BRAKE, Development of a Distributed Brake by Wire System. Artikel VDI-Berichte 1646 „Elektronik im Kraftfahrzeug“, S. 807 ff. VDI-Verlag Düsseldorf, Baden-Baden, 27.-28.09.2001.
- [Pau98]** Pauli, B.: Zuverlässigkeitsprognosen für elektronische Steuergeräte im Kraftfahrzeug: Modellbildung und deren praktische Anwendung. Dissertation an der BUGH Wuppertal, Shaker Verlag, Aachen, 1997.
- [PeM85]** Peters, O. H., Meyna, A.: Handbuch der Sicherheitstechnik. Band 1 der Ausgabe in 2 Bänden. München; Wien: Hanser, 1985.
- [PeM86]** Peters, O. H., Meyna, A.: Handbuch der Sicherheitstechnik. Band 2 der Ausgabe in 2 Bänden. München; Wien: Hanser, 1986.
- [Pie69]** Piehler, J.: Einführung in die lineare Optimierung. Verlag Harry Deutsch, Zürich, Frankfurt/Main, 1969.
- [Roj93]** Rojas, R.: Theorie der Neuronalen Netze- Eine systematische Einführung. Springer-Verlag, Berlin, Heidelberg, 1993.
- [Rüd95]** Rüdte, U.: Technik des Wissenschaftlichen Rechnens. Vorlesungsskript TU-

München, TU - Chemnitz, Überarbeitung: WS 94/95, München, Chemnitz, 1995.

- [TFL83]** Tanaka, H., Fan, L. T., Lai, F. S., Toguchi, K.: Fault Tree Analysis by Fuzzy Probability. IEEE Transactions on reliability, Vol. 32, No. 5, 1983.
- [Vol00]** Volkswagen AG: Elektrische und elektronische Baugruppen in Kraftfahrzeugen. Normierte allgemeine Prüfbedingungen. Technische Lieferbedingung 80101, Wolfsburg, 09/2000.
- [Vor98]** Vornberger, O.: Graphenalgorithmen. Vorlesungsskript Universität Osnabrück, Fachbereich Mathematik/Informatik, [http:// www-lehre. informatik. uni-osnabrueck. de/ ~graph](http://www-lehre.informatik.uni-osnabrueck.de/~graph), Osnabrück, 1998.
- [ZeA95]** Zell, M et Al.: SNNS- Stuttgart Neural Network Simulator, Version 4.1. Universitär verfügbares Tool von der Universität Stuttgart, Institute of Parallel and Distributed High-Performance Systems (IPVR), [http:// www. informatik. uni-stuttgart. de/ ipvs/ bv/ projekte/ snns/ snns.html](http://www.informatik.uni-stuttgart.de/ipvs/bv/projekte/snns/snns.html), Stuttgart, 1995.
- [YaZ87]** Yager, R. R., Zadeh, L. A.: Fuzzy-Sets and applications: selected papers. New-York, Wiley, 1987.
- [Zad65]** Zadeh, L. A.: Fuzzy-Sets. Information and Control, Vol. 8, 338-53, 1965.
- [ZaK92]** Zadeh, L. A., Kacprzyk, J.: Fuzzy logic for the management of uncertainty. Verlag Wiley, New York, 1992.

## 13 Anhang

Die in der vorliegenden Arbeit und diesem Anhang aufgeführten Zahlen sind gegenüber den realen Zahlen, wie man sie eventuell in gängigen Tabellenbüchern [Dod91] findet, bewusst verändert wurden. Von den vorliegenden Werten kann daher kein Rückschluss auf die Wirtschaftlichkeitszahlen oder Ausfallraten der betroffenen Systeme gezogen werden.

### 13.1 Datenbasis

Komponente	Typ	Gewicht [kg]	Länge [mm]	Breite [mm]	Höhe [mm]	Kosten [€]	Ausfallrate [1/h]
Aktorsteuerung	Typ1	1,00	200	210	220	33,2	4,0E-07
Aktorsteuerung	Typ2	1,10	200	220	240	33	2,0E-07
Batterie	Typ1	4,00	200	250	300	250	7,0E-05
Batterie	Typ2	4,20	100	150	200	220	1,0E-04
Batterie	Typ3	4,40	200	250	200	180	3,0E-04
Beschleunigungssensor	Typ1	0,20	80	50	50	5	1,0E-06
Beschleunigungssensor	Typ2	0,25	90	60	30	5,5	3,0E-10
Beschleunigungssensor	Typ3	0,30	70	40	40	5,6	1,0E-10
Bremsaktor	Typ1	3,00	150	200	220	245	3,0E-05
Bremsbackenkraftsensor	Typ1	0,20	80	50	50	8	2,0E-06
Bremsbackenkraftsensor	Typ2	0,25	90	60	30	8,8	2,0E-10
Bremsbackenwegsensor	Typ1	0,20	80	50	50	8	8,0E-07
Bremsbackenwegsensor	Typ2	0,40	60	50	50	8,7	8,0E-10
Bremspedal	Typ1	0,80	120	100	140	80	1,0E-12
Bremsschalter	Typ1	0,50	50	50	50	12	1,0E-07
Bremsschalter	Typ2	0,60	60	50	50	14	1,0E-10
Drehratensensor	Typ1	0,10	30	40	50	5	1,0E-09
Drehratensensor	Typ2	0,15	40	50	60	2	1,0E-05
Generator	Typ1	4,00	250	200	200	75	2,0E-05
Generator	Typ2	4,50	200	280	270	82	2,0E-07
Generator	Typ3	5,00	250	280	270	95	5,0E-08
Generator	Typ4	6,00	300	280	270	97	2,0E-08
Gerätestecker	Typ1	0,02	14	14	14	0,3	7,0E-09
Gierratensensor	Typ1	0,10	50	50	50	7	1,0E-06
Kabel12V	Typ1	0,10	0,01	100	0,01	0,1	7,0E-10
Kabel42V	Typ2	0,10	0,03	100	0,03	0,2	1,0E-10
KabelDaten	Typ3	0,10	0,01	100	0,01	0,1	2,0E-10
KabelSensor	Typ3	0,10	0,01	100	0,01	0,1	2,0E-10
Kombiinstrument	Typ1	5,00	400	230	200	95	4,0E-06
Lampe	V5	0,10	30	30	50	0,5	8,0E-05
Lenkwinkelsensor	Typ1	0,10	80	50	50	5	1,0E-09
Lichtmodul	hinten	2,00	150	200	220	22	4,0E-08
Parkbremse	Typ1	0,80	400	150	100	25	2,0E-06
VBox	Typ1	1,00	100	100	100	50	1,0E-07
Zündung	Typ1	0,50	60	60	60	12	4,0E-05

**13.2 Komponentenliste der EMB**

<b>Nr.</b>	<b>Komponente</b>	<b>Einbauort</b>
1-4	Drehratensensor	Fahrwerk
5-6	Batterie	Wasserkasten
7-10	Bremsaktor	Fahrwerk
11	Bremsbackenkraftsensor	Fahrwerk
12-13	VBox	Wasserkasten
14	Bremspedal	Innenraum, Boden, Querwand
15	Beschleunigungssensor	Innenraum, Boden, Querwand
16	Bremsschalter	Innenraum, Boden, Querwand
17-18	Bremsbackenwegsensor	Fahrwerk
19-20	Parkbremse	Innenraum, Boden, Querwand
21	Generator	Motorraum
22	Zündung	Schaltafel, Mittelkonsole
23	Kombiinstrument	Schaltafel, Mittelkonsole
24-26	Lampe	Heckbereich
27	Lichtmodul	Innenraum, Boden, Querwand
28	Lenkwinkelsensor	Schaltafel, Mittelkonsole
29-32	Aktorsteuerung	Fahrwerk
33-35	Gierratensensor	Innenraum, Boden, Querwand
36	Gerätestecker	Motorraum
37-51	Gerätestecker	Fahrwerk
52-55	Gerätestecker	Wasserkasten
56-58	Gerätestecker	Schaltafel, Mittelkonsole
59-67	Gerätestecker	Innenraum, Boden, Querwand
68-71	Gerätestecker	Heckbereich
72-75	Kabel (12V)	Schaltafel, Mittelkonsole
76-79	Kabel (12V)	Innenraum, Boden, Querwand
80-86	Kabel (12V)	Fahrwerk
87	Kabel (12V)	Heckbereich
88	Kabel (12V)	Schaltafel, Mittelkonsole
89	Kabel (12V)	Motorraum
90	Kabel (12V)	Schaltafel, Mittelkonsole
91	Kabel (12V)	Heckbereich
92	Kabel (12V)	Heckbereich
93	Kabel (12V)	Innenraum, Boden, Querwand
94-97	Kabel (12V)	Motorraum
98	Kabel (Daten)	Fahrwerk
99	Kabel (Daten)	Motorraum
100-101	Kabel (Daten)	Fahrwerk
102-108	Kabel (Sensor)	Fahrwerk
109-114	Kabel (Sensor)	Schaltafel, Mittelkonsole
115-116	Kabel (42V)	Wasserkasten
117-120	Kabel (42V)	Fahrwerk
121-122	Kabel (42V)	Motorraum
123-124	Kabel (42V)	Fahrwerk
125	Kabel (Daten)	Schaltafel, Mittelkonsole
126-127	Kabel (Daten)	Innenraum, Boden, Querwand

Nr.	Komponente	Einbauort
128	Kabel (Daten)	Motorraum
129	Kabel (Daten)	Wasserkasten
130	Kabel (Daten)	Schalttafel, Mittelkonsole
131-132	Kabel (Daten)	Motorraum
133-134	Kabel (Daten)	Innenraum, Boden, Querwand

### 13.3 Werte des optimalen Systems (Gleichgewichtung aller Parameter) bei der Optimierung mit exakten Werten

Ausfallrate [1/h]	Kosten [€]	Gewicht [kg]	Volumen [mm <sup>3</sup> ]	Aufwand
6,8E-04	2.080,60	49,32	133701785	0,984413

### 13.4 Verwendete Typen des optimalen Systems (Gleichgewichtung aller Parameter) bei der Optimierung mit exakten Werten

Nr.	Komponente	Typ	Ausfallrate [1/h]	Kosten [€]	Gewicht [kg]	Volumen [mm <sup>3</sup> ]	Aufwand
1	Drehratensensor	Typ2	1,0E-05	2,00	0,10	200000	0,8
2	Drehratensensor	Typ2	1,0E-05	2,00	0,10	200000	0,8
3	Drehratensensor	Typ2	1,0E-05	2,00	0,10	200000	0,8
4	Drehratensensor	Typ2	1,0E-05	2,00	0,10	200000	0,8
5	Batterie	Typ2	1,0E-04	220,00	4,00	10000000	0,96
6	Batterie	Typ2	1,0E-04	220,00	4,00	10000000	0,96
7-10	Bremsaktor	Typ1	3,0E-05	245,00	3,00	6600000	1
11	Bremsbackenkraftsensor	Typ1	2,0E-06	8,00	0,20	200000	0,969697
12-13	VBox	Typ1	1,0E-07	50,00	1,00	1000000	1
14	Bremspedal	Typ1	1,0E-12	80,00	0,80	1680000	1
15	Beschleunigungssensor	Typ1	1,0E-06	5,00	0,20	200000	0,969697
16	Bremsschalter	Typ1	1,0E-07	12,00	0,50	125000	0,952381
17-18	Bremsbackenwegsensor	Typ1	8,0E-07	8,00	0,20	125000	0,84818
19-20	Parkbremse	Typ1	2,0E-06	25,00	0,80	6000	1
21	Generator	Typ1	2,0E-05	75,00	4,00	18900000	0,924399
22	Zündung	Typ1	4,0E-05	12,00	0,50	216000	1
23	Kombiinstrument	Typ1	4,0E-06	95,00	5,00	18400000	1



Nr.	Komponente	Typ	Ausfallrate [1/h]	Kosten [€]	Gewicht [kg]	Volumen [mm <sup>3</sup> ]	Aufwand
24-26	Lampe	V5	8,0E-05	0,50	0,10	45000	1
27	Lichtmodul	hinten	4,0E-08	22,00	2,00	6600000	1
28	Lenkwinkelsensor	Typ1	1,0E-09	5,00	0,10	200000	1
29-32	Aktorsteuerung	Typ2	2,0E-07	33,00	1,00	9240000	0,997992
33-35	Gierratensensor	Typ1	1,0E-06	7,00	0,10	175000	1
36-71	Gerätestecker	Typ1	7,0E-09	0,30	0,02	2744	1
72-95	Kabel12V	Typ1	7,0E-10	0,10	0,10	0,01	1
96-101	KabelDaten	Typ3	2,0E-10	0,10	0,10	0,01	1
102-114	KabelSensor	Typ3	2,0E-10	0,10	0,10	0,01	1
115-124	Kabel42V	Typ2	1,0E-10	0,20	0,10	0,09	1
125-134	KabelDaten	Typ3	2,0E-10	0,10	0,10	0,01	1

**13.5 Lösungen einer Optimierung mit adaptivem Abbruchkriterium**

Abbruchvorgabe	Wirtschaftlichkeitsmaßzahl	Häufigkeit	Abbruchvorgabe	Wirtschaftlichkeitsmaßzahl	Häufigkeit
10	0,811185	1	100	0,808631	1
10	0,809507	1	110	0,80862	2
10	0,808965	1	110	0,808663	1
10	0,808641	1	120	0,80862	7
10	0,809137	1	120	0,808944	1
20	0,809863	1	130	0,80862	7
30	0,808652	1	130	0,808641	1
40	0,808631	2	140	0,80862	1
40	0,80862	1	140	0,808631	1
40	0,808923	1	150	0,80862	3
50	0,808663	1	150	0,808663	1
60	0,808631	1	160	0,808663	1
70	0,80862	2	170	0,80862	4
70	0,808631	1	170	0,808933	1
80	0,80862	4	180	0,80862	7
80	0,808663	1	180	0,808631	1
90	0,80862	3	190	0,80862	3
90	0,808631	1	190	0,808631	1
100	0,80862	3	200	0,80862	10

### 13.6 Vage Daten der verwendeten Typen in der Konzeptphase

		links		rechts	
		unten	oben	oben	unten
Batterie	Ausfallrate [1/h]	1,00E-05	9,00E-05	4,00E-04	8,00E-04
Typ2	Kosten [€]	210	220	225	230
Wasserkasten	Gewicht [kg]	4,1	4,2	4,2	4,3
	Volumen [mm <sup>3</sup> ]	3000000	3000000	3000000	3000000
Bremsaktor	Ausfallrate [1/h]	3,00E-08	2,00E-05	5,00E-05	8,00E-05
Typ1	Kosten [€]	200	210	270	300
Fahrwerk	Gewicht [kg]	2,9	3	3,1	3,2
	Volumen [mm <sup>3</sup> ]	5000000	6600000	6600000	7600000
VBox	Ausfallrate [1/h]	1,00E-10	1,00E-08	1,00E-07	5,00E-07
Typ1	Kosten [€]	45	50	50	54
Wasserkasten	Gewicht [kg]	1	1	1	1
	Volumen [mm <sup>3</sup> ]	1000000	1000000	1000000	1000000
Bremspedal	Ausfallrate [1/h]	1,00E-13	4,00E-12	4,00E-12	9,00E-12
Typ1	Kosten [€]	77	81	81	84
Innenraum, Boden, Querwand	Gewicht [kg]	0,8	0,8	0,8	0,8
	Volumen [mm <sup>3</sup> ]	1680000	1680000	1680000	1680000
Parkbremse	Ausfallrate [1/h]	1,00E-07	2,00E-06	2,00E-06	6,00E-06
Typ1	Kosten [€]	22	26	26	28
Innenraum, Boden, Querwand	Gewicht [kg]	0,7	0,8	0,8	1
	Volumen [mm <sup>3</sup> ]	6000	6000	6000	6000
Generator	Ausfallrate [1/h]	9,00E-06	2,00E-05	2,00E-05	7,00E-05
Typ1	Kosten [€]	65	70	80	90
Motorraum	Gewicht [kg]	3,5	3,8	4,5	6
	Volumen [mm <sup>3</sup> ]	8000000	9000000	15000000	17000000
Lichtmodul	Ausfallrate [1/h]	2,00E-08	4,00E-08	4,00E-08	6,00E-08
hinten	Kosten [€]	22	22	22	22
Innenraum, Boden, Querwand	Gewicht [kg]	2	2	2	2
	Volumen [mm <sup>3</sup> ]	6600000	6600000	6600000	6600000
Aktorsteuerung	Ausfallrate [1/h]	9,00E-08	4,00E-07	8,00E-07	9,00E-07
Typ1	Kosten [€]	25	30	40	50
Fahrwerk	Gewicht [kg]	0,8	1	1	1,1
	Volumen [mm <sup>3</sup> ]	9240000	9240000	9240000	9240000

### 13.7 Vage Daten der verwendeten Typen in der Prototypenphase

		links		rechts	
		unten	oben	oben	unten
Batterie	Ausfallrate [1/h]	1,50E-05	9,50E-05	4,00E-04	7,80E-04
Typ2	Kosten [€]	212	221	225	228
Wasserkasten	Gewicht [kg]	4,1	4,2	4,2	4,3
	Volumen [mm <sup>3</sup> ]	3000000	3000000	3000000	3000000
Bremsaktor	Ausfallrate [1/h]	2,80E-08	2,05E-05	4,95E-05	6,00E-05
Typ1	Kosten [€]	195	210	275	295
Fahrwerk	Gewicht [kg]	2,95	3,1	3,15	3,18
	Volumen [mm <sup>3</sup> ]	5500000	6400000	6400000	7400000
VBox	Ausfallrate [1/h]	1,00E-10	1,00E-08	1,00E-07	5,00E-07
Typ1	Kosten [€]	45	50	50	54
Wasserkasten	Gewicht [kg]	1	1	1	1
	Volumen [mm <sup>3</sup> ]	1000000	1000000	1000000	1000000
Bremspedal	Ausfallrate [1/h]	1,80E-13	4,00E-12	4,00E-12	8,00E-12
Typ1	Kosten [€]	77,5	81	81	82
Innenraum, Boden, Querwand	Gewicht [kg]	0,8	0,8	0,8	0,8
	Volumen [mm <sup>3</sup> ]	1680000	1680000	1680000	1680000
Parkbremse	Ausfallrate [1/h]	1,50E-07	2,10E-06	2,10E-06	5,50E-06
Typ1	Kosten [€]	22	26	26	26
Innenraum, Boden, Querwand	Gewicht [kg]	0,85	0,85	0,85	0,85
	Volumen [mm <sup>3</sup> ]	6000	6000	6000	6000
Generator	Ausfallrate [1/h]	1,00E-05	2,00E-05	2,00E-05	5,00E-05
Typ1	Kosten [€]	65	75	85	88
Motorraum	Gewicht [kg]	4	4,2	4,5	5,8
	Volumen [mm <sup>3</sup> ]	8500000	9000000	13000000	16000000
Lichtmodul	Ausfallrate [1/h]	2,00E-08	4,00E-08	4,00E-08	6,00E-08
hinten	Kosten [€]	22	22	22	22
Innenraum, Boden, Querwand	Gewicht [kg]	2	2	2	2
	Volumen [mm <sup>3</sup> ]	6600000	6600000	6600000	6600000
Aktorsteuerung	Ausfallrate [1/h]	9,00E-08	4,00E-07	8,00E-07	9,00E-07
Typ1	Kosten [€]	28	32	38	45
Fahrwerk	Gewicht [kg]	0,9	1	1	1,1
	Volumen [mm <sup>3</sup> ]	9240000	9240000	9240000	9240000

### 13.8 Vage Daten der verwendeten Typen in der Vorserienphase

		links		rechts	
		unten	oben	oben	unten
Batterie	Ausfallrate [1/h]	4,00E-05	6,00E-05	6,00E-05	1,00E-04
Typ2	Kosten [€]	212	215	215	228
Wasserkasten	Gewicht [kg]	4,2	4,2	4,2	4,2
	Volumen [mm <sup>3</sup> ]	3000000	3000000	3000000	3000000
Bremsaktor	Ausfallrate [1/h]	2,00E-06	2,00E-05	2,00E-05	6,00E-05
Typ1	Kosten [€]	195	215	215	295
Fahrwerk	Gewicht [kg]	3	3	3	3
	Volumen [mm <sup>3</sup> ]	6400000	6400000	6400000	6400000
VBox	Ausfallrate [1/h]	7,00E-08	1,00E-07	1,00E-07	3,00E-07
Typ1	Kosten [€]	48	50	50	53
Wasserkasten	Gewicht [kg]	1	1	1	1
	Volumen [mm <sup>3</sup> ]	1000000	1000000	1000000	1000000
Bremspedal	Ausfallrate [1/h]	4,00E-13	4,00E-12	4,00E-12	7,00E-12
Typ1	Kosten [€]	78	80	80	81
Innenraum, Boden, Querwand	Gewicht [kg]	0,8	0,8	0,8	0,8
	Volumen [mm <sup>3</sup> ]	1680000	1680000	1680000	1680000
Parkbremse	Ausfallrate [1/h]	2,50E-07	2,10E-06	2,10E-06	5,00E-06
Typ1	Kosten [€]	22	22,5	22,5	25
Innenraum, Boden, Querwand	Gewicht [kg]	0,85	0,85	0,85	0,85
	Volumen [mm <sup>3</sup> ]	6000	6000	6000	6000
Generator	Ausfallrate [1/h]	1,00E-05	4,00E-05	4,00E-05	5,00E-05
Typ1	Kosten [€]	75	80	80	88
Motorraum	Gewicht [kg]	4,2	4,2	4,2	4,2
	Volumen [mm <sup>3</sup> ]	10000000	10000000	10000000	10000000
Lichtmodul	Ausfallrate [1/h]	3,00E-08	4,80E-08	4,80E-08	5,00E-08
hinten	Kosten [€]	22	22	22	22
Innenraum, Boden, Querwand	Gewicht [kg]	2	2	2	2
	Volumen [mm <sup>3</sup> ]	6600000	6600000	6600000	6600000
Aktorsteuerung	Ausfallrate [1/h]	1,00E-07	4,00E-07	4,00E-07	6,00E-07
Typ1	Kosten [€]	28	33,2	33,2	40
Fahrwerk	Gewicht [kg]	1	1	1	1
	Volumen [mm <sup>3</sup> ]	9240000	9240000	9240000	9240000

**13.9 Vage Daten der verwendeten Typen im Serieneinsatz**

		links		rechts	
		unten	oben	oben	unten
Batterie	Ausfallrate [1/h]	4,00E-05	6,00E-05	6,00E-05	8,00E-05
Typ2	Kosten [€]	212	215	215	218
Wasserkasten	Gewicht [kg]	4,2	4,2	4,2	4,2
	Volumen [mm <sup>3</sup> ]	3000000	3000000	3000000	3000000
Bremsaktor	Ausfallrate [1/h]	8,00E-06	1,50E-05	1,50E-05	3,00E-05
Typ1	Kosten [€]	205	215	215	223
Fahrwerk	Gewicht [kg]	3	3	3	3
	Volumen [mm <sup>3</sup> ]	6400000	6400000	6400000	6400000
VBox	Ausfallrate [1/h]	9,00E-08	1,20E-07	1,20E-07	2,00E-07
Typ1	Kosten [€]	48	50	50	51
Wasserkasten	Gewicht [kg]	1	1	1	1
	Volumen [mm <sup>3</sup> ]	1000000	1000000	1000000	1000000
Bremspedal	Ausfallrate [1/h]	1,00E-12	4,00E-12	4,00E-12	6,00E-12
Typ1	Kosten [€]	79	80	80	81
Innenraum, Boden, Querwand	Gewicht [kg]	0,8	0,8	0,8	0,8
	Volumen [mm <sup>3</sup> ]	1680000	1680000	1680000	1680000
Parkbremse	Ausfallrate [1/h]	8,00E-07	2,10E-06	2,10E-06	4,00E-06
Typ1	Kosten [€]	22	22,5	22,5	24
Innenraum, Boden, Querwand	Gewicht [kg]	0,85	0,85	0,85	0,85
	Volumen [mm <sup>3</sup> ]	6000	6000	6000	6000
Generator	Ausfallrate [1/h]	3,00E-05	4,00E-05	4,00E-05	5,00E-05
Typ1	Kosten [€]	77	79	79	84
Motorraum	Gewicht [kg]	4,2	4,2	4,2	4,2
	Volumen [mm <sup>3</sup> ]	10000000	10000000	10000000	10000000
Lichtmodul	Ausfallrate [1/h]	3,00E-08	4,80E-08	4,80E-08	5,00E-08
hinten	Kosten [€]	22	22	22	22
Innenraum, Boden, Querwand	Gewicht [kg]	2	2	2	2
	Volumen [mm <sup>3</sup> ]	6600000	6600000	6600000	6600000
Aktorsteuerung	Ausfallrate [1/h]	2,00E-07	3,00E-07	3,00E-07	5,00E-07
Typ1	Kosten [€]	30	33,1	33,1	37
Fahrwerk	Gewicht [kg]	1	1	1	1
	Volumen [mm <sup>3</sup> ]	9240000	9240000	9240000	9240000

### 13.10 Screenshots

Nachfolgend sind einige Screenshots der Ein- und Ausgabefenster des realisierten Optimierungstools des Genetische Algorithmus dargestellt.

Zunächst können die Komponenten mit jeweils beliebig vielen Typen definiert werden (Abbildung 63).

The screenshot shows a dialog box titled "Neue Komponente hinzufügen, ändern oder löschen". It contains the following fields and controls:

- Name der Komponente:** A dropdown menu.
- Typ der Komponente:** A dropdown menu.
- Ausfallrate dieses Typs:** A section containing:
  - $y \times 10^{-x}$  with input fields for  $y = 0$  and  $x = 0$ , and a unit label  $[1/t]$ .
  - Kosten:** An input field with the value  $0$  and a unit label  $[€]$ .
- Abmessungen:** A section containing:
  - Länge:** An input field with the value  $0$  and a unit label  $[mm]$ .
  - Höhe:** An input field with the value  $0$  and a unit label  $[mm]$ .
  - Breite:** An input field with the value  $0$  and a unit label  $[mm]$ .
  - Gewicht:** An input field with the value  $0$  and a unit label  $[Kg]$ .
- Buttons:** "Übernehmen", "Löschen", "Datenunsicherheit", "OK", and "Abbrechen".

Abbildung 63: Eingabefenster für die Komponentendatenbank

Von diesem Fenster aus, lässt sich zum Eingabefenster für die Datenunsicherheit verzweigen (Abbildung 64).

The screenshot shows a dialog box titled "Dialog" with a close button (X) in the top right corner. The dialog is organized into several sections:

- Component Selection:** Two dropdown menus labeled "Komponente" and "Typ".
- Location Selection:** A dropdown menu labeled "Einbauort".
- Parameter Inputs:** Four input fields for "Ausfallrate", "Kosten", "Gewicht", and "Volumen", each containing the value "0".
- Data Uncertainty Selection:** A section titled "Die Datenunsicherheit liegt bei" containing seven radio buttons:
  - Ausfallrate
  - Kosten
  - Gewicht
  - Volumen
  - scharfer Wert
  - leichte Datenunsicherheit
  - große Datenunsicherheit
- Unit and Interval Definition:** A section titled "Einheit:" containing several input fields:
  - A central input field with "0".
  - Two input fields labeled "Intervall gleicher Wahrscheinlichkeit" with "0" in each.
  - Two input fields labeled "Intervall verminderter Wahrscheinlichkeit" with "0" in each.
  - Two additional input fields at the bottom with "0" in each.
- Action Buttons:**
  - "Übernehmen" (Accept)
  - "Übernehmen und Schließen" (Accept and Close)
  - "Schließen" (Close)
- Footnote:** "Wenn Wert == 0, dann nicht definiert" (If value == 0, then not defined).

**Abbildung 64: Eingabefenster zur Definition der Vagen Daten**

Die Datenunsicherheit kann individuell für jeden relevanten Typ angegeben werden. Sind zu einem Typ keine vagen Daten angegeben, so werden die entsprechenden in Abbildung 63 definierten Parameter als scharfe Werte angenommen.



An den Einbauorten existieren nach [Vol00] Anforderungen. In einem Eingabefenster werden die Einbauorte definiert und die entsprechenden Anforderungen und Resistenzen zugeordnet (Abbildung 65).

Definition der Orte nach VW TL 80101

Einbauort  Lfd.Nr.:

Für diesen Ort gilt ...

Schwingungsbelastungsschärfegrad  
 1  2  3  sinus

Betriebstemperatur  
Tiefstwert (°C)   
Höchstwert (°C)

Lagertemperatur  
Tiefstwert (°C)   
Höchstwert (°C)

Umweltbeständigkeit  
 innen  
 außen

Beständigkeit gegen Reagenzien  Auswahl umkehren

- a) Dieselmotorkraftstoff
- b) FAM-Prüfkraftstoff
- c) Batteriesäure
- d) Bremsflüssigkeit
- e) Kühlmittelzusatz
- f) Konservierungsmittel
- g) Entkonservierungsmittel
- h) Motorenöl
- i) Kaltreiniger
- j) Spiritus
- k) Getriebeöl
- l) ATF
- m) Innenreiniger
- n) M15-Prüfkraftstoff
- o) Dieselmotorkraftstoff (FAME)
- p) Zentralhydrauliköl
- q) Scheibenreinigungsflüssigkeit
- r) Coffein- und zuckerhaltige Getränke
- s) Glasreiniger
- t) Fleckentferner

Staub- und Spritzwasserdichtheit nach DIN 40 050-9  
 IP 5K 4K  IP 5K 4K mit Zusatzprüfung  IP 5K 0  IP 5K 3

Zulässige Geometrie  
maximale Länge (cm)  maximale Breite (cm)  maximale Höhe (cm)

Abbildung 65: Eingabefeld für die an den Einbauorten existierenden Anforderungen nach [Vol00]

Über ein weiteres Fenster (Abbildung 66) kann das System definiert werden, wobei den entsprechenden Komponenten deren Einbauorte zugewiesen werden. Auch die Gewichtung der Wirtschaftlichkeitsmaßzahl wird in dieser Ebene angegeben.

Knotennr.	Modulnr.	Komponente (Typ, Ort)
1	10	Drehratensensor (Typ1, Fahrwerk)
2	10	Drehratensensor (Typ1, Fahrwerk)
3	10	Drehratensensor (Typ1, Fahrwerk)
4	10	Drehratensensor (Typ1, Fahrwerk)
5	10	Batterie (Typ1, Wasserkasten)
6	10	Batterie (Typ1, Wasserkasten)
7	10	Bremsaktor (Typ1, Fahrwerk)
8	10	Bremsaktor (Typ1, Fahrwerk)
9	10	Bremsaktor (Typ1, Fahrwerk)
10	10	Bremsaktor (Typ1, Fahrwerk)
11	10	Bremsbackenkraftsensor (Typ1, Fahrwerk)
12	10	Bremspedal (Typ1, Innenraum,Boden,Querwand)
13	10	Beschleunigungssensor (Typ1, Innenraum,Boden,Querwand)
14	10	Bremsschalter (Typ1, Innenraum,Boden,Querwand)
15	10	Bremsbackenwegsensor (Typ1, Fahrwerk)
16	10	Bremsbackenwegsensor (Typ1, Fahrwerk)
17	10	Parkbremse (Typ1, Innenraum,Boden,Querwand)
18	10	Parkbremse (Typ1, Innenraum,Boden,Querwand)
19	10	Generator (Typ1, Motorraum)
20	10	Zündung (Typ1, Schalttafel,Mittelkonsole)
21	10	Kombiinstrument (Typ1, Schalttafel,Mittelkonsole)
22	10	Lampe (V5, Heckbereich)
23	10	Lampe (V5, Heckbereich)
24	10	Lampe (V5, Heckbereich)
25	10	Lichtmodul (hinten, Innenraum,Boden,Querwand)
26	10	Lenkwinkelsensor (Typ1, Schalttafel,Mittelkonsole)
27	10	Aktorsteuerung (Typ1, Fahrwerk)
28	10	Aktorsteuerung (Typ1, Fahrwerk)
29	10	Aktorsteuerung (Typ1, Fahrwerk)
30	10	Aktorsteuerung (Typ1, Fahrwerk)
31	10	Gierratensensor (Typ1, Innenraum,Boden,Querwand)
32	10	Gierratensensor (Typ1, Innenraum,Boden,Querwand)
33	10	Gierratensensor (Typ1, Innenraum,Boden,Querwand)
34	10	Gerätestecker (Typ1, Motorraum)

Abbildung 66: Eingabefenster für die Definition des Systems

Nachdem alle Komponenten, Einbauorte und das System definiert sind, kann die optimale Lösung ermittelt werden.

Die maximal zulässige Ausfallrate (hier:  $\lambda=8 \cdot 10^{-4}$  1/t), die Zeitbasis für die Ausfallrate (hier:  $t=1$  h) und die Abbruchvorgabe (hier:  $V_{ab}=100$ ) müssen angegeben werden (Abbildung 67).

Soll-Werte

Ausfallrate in [1/t]  $y =$  8  
 $y \cdot \exp(-x)$   $x =$  4

Berechne Überleben t [h] 1

Leerlaufepochen bis Abbruch? 100

Einbauorte prüfen

Übernehmen

Abbildung 67: Vorgaben für die Optimierung, einzugeben im Fenster für die Berechnung

Anschließend wird die Optimierung durchgeführt. Als Ergebnis wird zunächst das billigste System dargestellt (Abbildung 68).

Anzahl der durchlaufenen Generationen  
0

Mit den billigsten Komponenten ergibt sich:

Ausfallrate in [1/t]	min. Kosten in [€]:

geplant ist

Knoten		
Nr.	Komponente	Typ

Abbildung 68: Angabe der billigsten Lösung, dargestellt im Fenster für die Berechnung

Da anschließend das optimale System dargestellt wird (Abbildung 69), lässt sich durch einen Vergleich der beiden Varianten erkennen, wo die Optimierung Veränderungen durchgeführt hat.

Optimierung					
Verwendet wurden					
Ausf.	Komponente	Typ	Preis	Gewicht	Aufw.

**Abbildung 69: Darstellung der Komponenten des optimalen Systems, dargestellt im Fenster für die Berechnung**

Für dieses optimale System werden auch die Wirtschaftlichkeitsdaten und die Ausfallrate angegeben.

Es ergeben sich...		
... Systemkosten von	<input type="text"/>	[€]
... ein Gesamtgewicht von	<input type="text"/>	[kg]
... ein Gesamtvolumen von	<input type="text"/>	[ccm]
... ein Aufwandswert von	<input type="text"/>	[1]
... eine Ausfallrate	<input type="text"/>	[1/t]

**Abbildung 70: Parameter der optimalen Lösung, dargestellt im Fenster für die Berechnung**

Letztlich werden auch die vagen Daten des optimalen Systems (Abbildung 71) dargestellt.

Ausfallrate [1/t]		
Intervall gleicher Wahrscheinlichkeit		
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Intervall verminderter Wahrscheinlichkeit		Intervall verminderter Wahrscheinlichkeit
<input type="text" value="0"/>		<input type="text" value="0"/>

Kosten [€]		
Intervall gleicher Wahrscheinlichkeit		
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Intervall verminderter Wahrscheinlichkeit		Intervall verminderter Wahrscheinlichkeit
<input type="text" value="0"/>		<input type="text" value="0"/>

Gewicht [Kg]		
Intervall gleicher Wahrscheinlichkeit		
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Intervall verminderter Wahrscheinlichkeit		Intervall verminderter Wahrscheinlichkeit
<input type="text" value="0"/>		<input type="text" value="0"/>

Volumen [ccm]		
Intervall gleicher Wahrscheinlichkeit		
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Intervall verminderter Wahrscheinlichkeit		Intervall verminderter Wahrscheinlichkeit
<input type="text" value="0"/>		<input type="text" value="0"/>

Abbildung 71: Vage Daten der optimalen Lösung, dargestellt im Fenster für die Berechnung