

# Multidisziplinäre Optimierung im Produktentwicklungsprozess der Automobilindustrie

Fabian Duddeck<sup>1\*</sup>

<sup>1</sup> Queen Mary College, London University, London, UK

## Zusammenfassung

Nachdem die numerische Simulation mit den Finite-Element-Methoden im Produktentwicklungsprozess (PEP) der Automobilindustrie nunmehr etabliert ist, ist in den letzten Jahren das Interesse an Optimierungsverfahren sehr gestiegen. Wurden anfangs nur kleinere, sehr detailbezogene Probleme optimiert, so werden in jüngerer Zeit erste so genannte multidisziplinäre Optimierungen (MDO) realisiert. Hierbei bezieht man sich auf Designverbesserungen, die mittels simultaner Optimierung verschiedener Funktionen der Karosserie erzielt wurden. Meist betraf dies die Lastfälle der passiven Sicherheit und die Fragestellungen des NVH-Bereichs (statische und dynamische Steifigkeiten der Karosserie). Ein ganzheitliches Erfassen der Funktionalität in einer solchen simultanen Optimierung ermöglicht eine sehr effektive, belastungsgerechte Ausnutzung aller Bauteile. In der Vergangenheit war der industrielle Einsatz der MDO aufgrund des hohen Rechenaufwands bei den Crashsimulationen fragwürdig und auf methodische Untersuchungen beschränkt. Hier wird nun gezeigt, dass die MDO durch den Einsatz von effizienten Optimierungsalgorithmen deutlich beschleunigt werden kann. Bei BMW konnte somit erstmals die MDO sinnvoll in den realen Produktionsprozess integriert werden.

**Keywords:** Produktentwicklungsprozess, multidisziplinäre Optimierung, Crash, NVH, evolutionäre Algorithmen, mehrkriterielle Optimierung.

---

\* Kontakt: Dr.-Ing. habil. Fabian Duddeck, Department of Engineering, Queen Mary College, London University, Mile End Road, London E1 4NS, UK, E-Mail: [f.duddeck@qmul.ac.uk](mailto:f.duddeck@qmul.ac.uk)

# 1 Kontext

Steigende Komplexität der Fragestellungen und zunehmende Verflechtung aller Disziplinen zur Reduktion der Entwicklungszeiten prägen das Bild des heutigen Produktentwicklungsprozesses (PEP) in der Automobilindustrie. Oft wird in den einzelnen Disziplinen - wie z.B. passive Sicherheit, Akustik, Strukturmechanik und andere Aspekte des NVH (Noise, Vibration, and Harshness) - numerische Optimierung bereits erfolgreich eingesetzt. Fragen der Identifikation, Sensitivität und Robustheit gehen einher mit wirklichen Optimierungsaufgaben betreffs Topologie, Form oder Material- und Geometrieparametern (Abbildung 1).

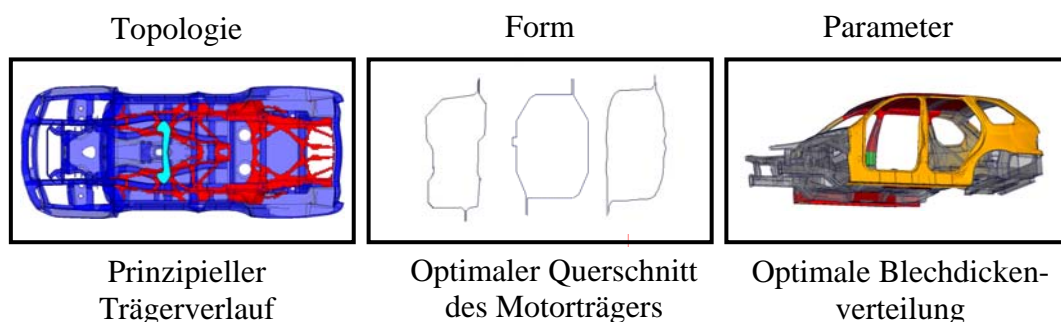


Abbildung 1: Kategorien der Optimierung.

Optimierungsziele sind meist Reduktion der Masse der Karosserie – damit verbunden eine Reduktion des Treibstoffverbrauchs und der CO<sub>2</sub>-Emission – oder die Erhöhung der Funktionalität der Karosserie wie z.B. höhere Eigenfrequenzen und/oder Steifigkeiten in der Strukturmechanik und -statik, geringere Intrusionen in die Fahrgastzelle und geringere Belastungen der Insassen in den Crashlastfällen oder geringere Geräuschbelastungen in der Akustik.

Teils werden mono-kriterielle Optimierungen realisiert, in denen entweder nur eine Zielgröße definiert werden kann oder in denen alle Zielgrößen zu einem einzigen Optimierungskriterium zusammengefasst werden. Teils konnten in jüngerer Zeit auch multi-kriterielle Optimierungen durchgeführt werden, in denen eine Paretofront erzeugt wird, die eine a posteriori Bewertung der einzelnen Zielgrößen ermöglicht. Diese Trade-Off-Kurven bilden z.B. Grundlage für Entscheidungen, ob eine Erhöhung der Ansprüche betreffs einer Zielgröße noch vertretbar ist, obwohl sich andere Zielgrößen verschlechtern; der beste Kompromiss kann so ermittelt werden. Abbildung 2 zeigt ein Beispiel hierfür: die Abhängigkeit der Intrusion in die Fahrgastzelle bei einem Seitencrash von der Masse kann abgelesen werden. Der numerische Aufwand für eine mehrkriterielle Optimierung ist i.d.R. höher als für die mono-kriterielle, steigt bei einem geeigneten Algorithmus jedoch weniger als proportional zur Anzahl der Zielkriterien. Für eine Zwei-Kriterien Optimierung benötigt man weniger als den doppelten Aufwand einer mono-kriteriellen Optimierung.

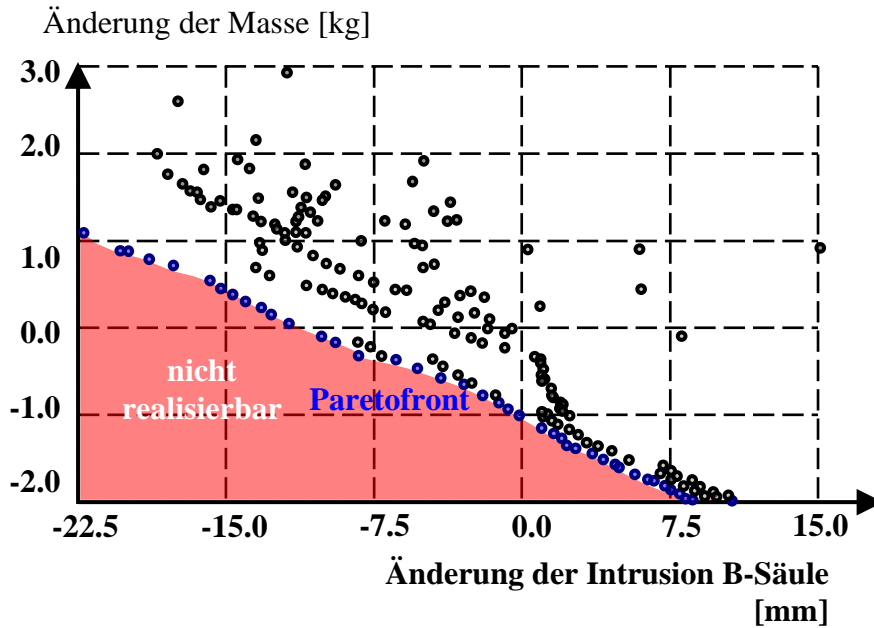


Abbildung 2: Multi-kriterielle Optimierung für den Seitencrash. Die Paretofront zeigt den zusätzlichen Aufwand an Masse zur Erreichung einer bestimmten Reduktion der Intrusion der B-Säule.

Die Optimierung in den einzelnen Disziplinen ist in der Praxis oft nicht ausreichend effektiv. Wird z.B. eine Karosserie bezüglich der NVH-Steifigkeiten optimiert kann dies zu einer Verletzung der Crashkriterien führen. Es sind dann mehrere Iterationsschleifen erforderlich, um zu einem ganzheitlich optimierten Design zu gelangen. Werden alle relevanten Disziplinen (Crash, NVH etc.) simultan optimiert, um diese aufwändigen Iterationen zwischen den Disziplinen zu vermeiden, so spricht man von multi-disziplinärer Optimierung (MDO). Abbildungen 3 und 4 verdeutlichen den Wegfall der Iterationsschleifen.

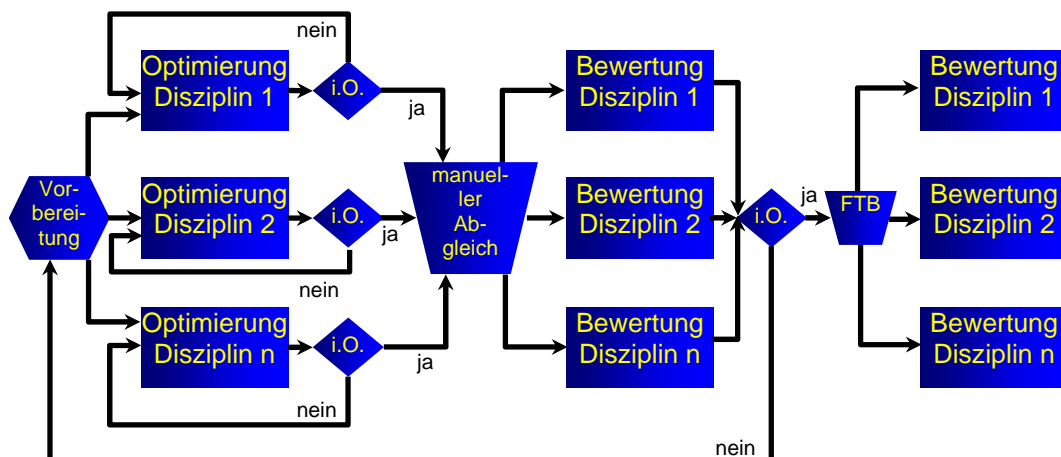


Abbildung 3: Optimierungsschema für eine sequentielle, nicht simultane Optimierung. Hier sind in jeder Disziplin mehrere Iterationsschleifen notwendig.

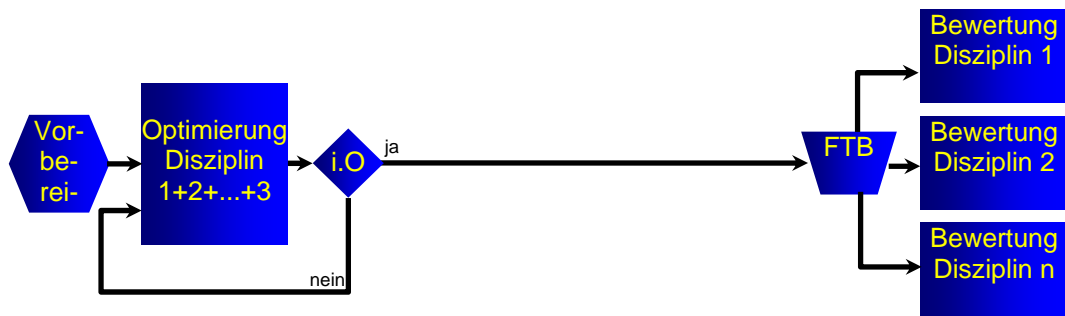


Abbildung 4: Optimierungsschema für eine simultane, multi-disziplinäre Optimierung (MDO). Alle Disziplinen sind in ein Optimierungsverfahren integriert.

Die MDO erfordert jedoch eine sorgfältige Vorbereitung aller numerischen Modelle in allen Disziplinen, Parameter und Modellstände müssen einander angeglichen werden. Teils müssen auch vorhandene Strukturen des PEP speziell für die MDO erweitert und zugänglich gemacht werden. Der immer noch hohe numerische Aufwand lohnt sich nur dann, wenn auch wirklich alle relevanten Randbedingungen bekannt sind und in der Optimierung berücksichtigt werden können. Hierzu zählen Fertigungsrestriktionen, Einschränkungen aus doch nicht berücksichtigten Lastfällen (z.B. Betriebsfestigkeit, Akustik etc.). Die sorgfältige Definition des Optimierungsproblems mit Unterstützung der einzelnen Fachexperten ist unabdingbar.

Erste MDO-Verfahren in der Automobilindustrie wurden meist mithilfe eines Monte-Carlo-Schema realisiert. Vorbereitungszeiten von mindestens einer Woche, Rechenzeiten im Bereich von vier Wochen und Analysezeiten von ebenfalls etwa einer Woche waren die Regel. Im Hinblick auf den engen Zeitrahmen, der vom PEP vorgegeben wird, sind derartige Ansätze nicht produktiv einsetzbar. Eine Verkürzung der Optimierung war unbedingt erforderlich.

Hierzu wurden in der Berechnungsabteilung von BMW Studien durchgeführt. Einerseits sollte der Algorithmus gefunden werden, der robust in allen Fällen eingesetzt werden kann, dennoch ausreichend effektiv ist und zu schnelleren Designverbesserungen führt als die traditionell eingesetzten Methoden. Die wesentlichen Aspekte sind am Ende dieses Abschnitts zusammengestellt. Zusätzlich wünschenswert wäre eine integrierte Robustheits- und Sensitivitätsbewertung. Nochmals betont sei, dass für den mono-kriteriellen Fall streng genommen nicht nach dem Optimum gesucht wird, sondern nach der mit einem gewissen, festgelegten Aufwand maximal erreichbaren Designverbesserung. Auch für die multi-kriterielle Optimierung ist eine genaue Ermittlung der Paretofront in dem aufgezeigten Kontext nicht durchführbar, die Praxis erfordert dies auch nicht, es genügen Abschätzungen.

Der Algorithmus sollte

- für alle auftretenden Fälle geeignet sein,
- sich an die Problemstellung automatisch anpassen,
- ausreichende Verbesserungen selbst bei stark unregelmäßigen und verrauschten Antwortflächen liefern,
- mit nicht konvexen Optimierungsproblemen zurechtkommen,
- erfolgreich sein, wenn keine sinnvollen Gradienten ermittelt werden können,
- robust gegenüber Ausfall einzelner Rechnungen sein,
- einen (eventuell veränderten) Neustart nach z.B. Netzwerkproblemen oder Hardware-Problemen ermöglichen,
- sowohl mit Fließkomma, Integer oder diskreten Parametern arbeiten können,
- eine Vielzahl an Optimierungsvariablen zulassen (bis zu 100 oder 200),
- alle Arten von zusätzlichen Randbedingungen erfassen,
- in einem nicht zulässigen Designraum starten können,
- wenig CPU Zeit erfordern, d.h. mit einer geringen Anzahl von Designpunkten auskommen,
- schnell und einfach in Pre- und Postprocessing sein,
- ein Monitoring während der Optimierung ermöglichen,
- nur einfache und leicht verständliche Einstellungen erfordern,
- mono- und multi-kriterielle Optimierung ermöglichen,
- etc.

## 2 Charakteristik der Crash- und NVH-Lastfälle

Die Charakteristik der numerischen Simulation der Crashlastfälle und der NVH-Disziplinen ist grundverschieden. Viele Fragen der NVH werden gewöhnlich mit einem linearen FE-Solver bearbeitet (z.B. NASTRAN). Dort integriert ist eine Optimierung, die auf einem Gradientenverfahren beruht (SOL 200). Erfahrungen haben gezeigt, dass die Statik und die niederfrequenten Probleme der Struktur-dynamik gut lösbar sind mittels dieses Optimierungsverfahrens. Die Verfolgung der Moden während der Optimierung bereitet manchmal Schwierigkeiten, teils identifiziert der Optimierer die falschen Moden, wenn das ursprüngliche Design sich zu stark vom augenblicklichen unterscheidet. In solchen Fällen ist ein Neustart sinnvoll. Kriterium für die Identifikation der Moden ist meist der MAC-Wert (Modal Assurance Criterion), der die Kollinearität zweier Moden aufzeigt.

Optimierungsprobleme mittlerer und höherer Frequenzen sind nicht immer erfolgreich über diese Gradientenmethoden zu lösen. Die Antwortfläche ist nicht mehr konvex, der Gradientenlöser bringt oft entweder keine konvergierende Lösung zustande oder landet in einem Nebenmaximum. Dies bedeutet, dass für Probleme der Akustik nicht deterministische Optimierungsverfahren empfohlen werden können, die in der Lage sind mit den komplexeren Antwortflächen umzugehen.

Für die meisten Crashlastfälle ist dies Problem noch gravierender. Die Antwortflächen sind hoch nichtlinear, können Bifurkationen aufweisen, d.h. sie sind unter Umständen nicht mehr stetig. Die Erfahrung hat jedoch gezeigt, dass die verschiedenen Lastfälle unterschiedlich „gutmütig“ sind. Während Front- und Heckcrash-Lastfälle oft stark nichtlineare Antwortflächen mit Verzweigungen aufweisen, sind Seitencrash-Lastfälle physikalisch einfacher, sie sind im Grunde dominiert von der einfachen Biegung der B-Säule und der umgebenden Bauteile, ohne dass Verzweigungen auftreten. Ähnlich einfacher sind die AZT-Lastfälle (Danner-Test oder Versicherungstest mit geringen Geschwindigkeiten). In den zuletzt genannten Fällen sind die Antwortflächen von geringerer Komplexität, sie lassen sich daher einfacher durch Ersatzmodelle approximieren und dann in einem RSM-Verfahren (Response Surface Method) optimieren. Das in Abbildung 2 gezeigte Optimierungsergebnis eines Seitencrashes wurde über ein RSM-Ansatz erzeugt.

Aufgrund der Parallelität der Crashsimulation (Distributed Memory Parallel DMP) sind die meisten Berechnungen verrauscht, in Extremfällen kann dies zu sehr unterschiedlichen Ergebnissen führen. Abbildung 5 zeigt einen solchen Extremfall, in dem trotz identischem Rechenmodell und identischen Rechnern große Differenzen im Verhalten des Motorträgers auftraten. Durch spezielle Optionen lässt sich das Rauschen in der Crashberechnung reduzieren, jedoch steigt dadurch der Rechenaufwand erheblich.

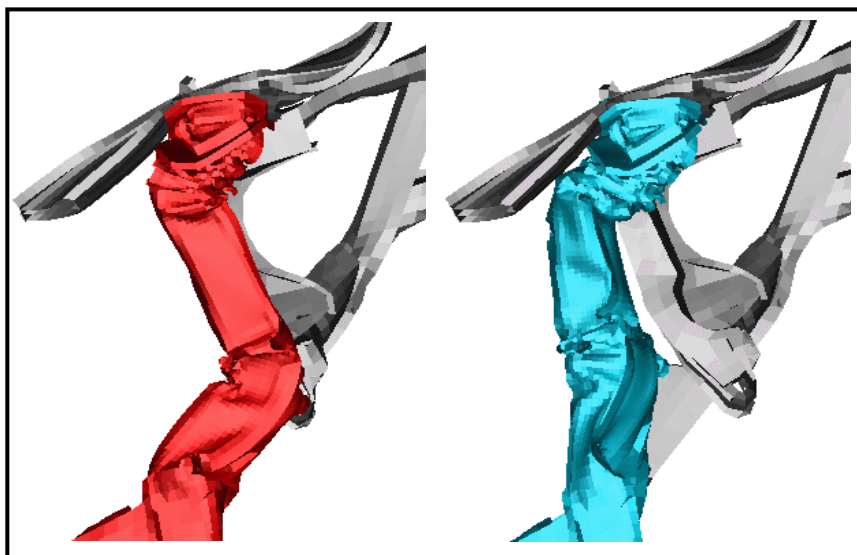


Abbildung 5: Zwei Berechnungsergebnisse einer Crashsimulation identischer Modelle und identischer Rechnerkonfigurationen als Extremfall des numerischen Rauschens der Crashberechnung.

## 3 Evaluierung der Algorithmen für die MDO

### 3.1 Auswahl der Algorithmen

Aufgrund der hohen Nichtlinearität der Crashlastfälle und des auftretenden Rauschens wurden nur nicht deterministische Optimierungsverfahren herangezogen. Ausgewählt wurden folgende Typen:

1. **Kriging-unterstützte Evolution (Kr-EA)**
  - Lokale Approximation (Gauß-Prozesse) der Antwort-Fläche,
  - Optimierung auf der approximierten Fläche.
2. **Simulated Annealing (SA)**
  - Exponentielles Einschränken des Designraumes.
  - Akzeptanz von Verschlechterungen mit abnehmender Wahrscheinlichkeit.
3. **Genetische Algorithmen (GA)**
  - Aus  $N$  Eltern werden  $N$  Kinder erzeugt.
  - Rekombination und Mutation operieren auf der Bit-Ebene.
4. **Evolutionäre Algorithmen (EA)**
  - Mutation und Rekombination
  - Flexible Eltern/Kinder, Real- und Integer-Variablen

Es wurden jeweils mehrere unterschiedliche Parametereinstellungen untersucht und die einzelnen Optimierungen wegen ihrer Stochastizität wiederholt, um eine ausreichende Signifikanz der Vergleiche zu gewährleisten. Beispielsweise wurden bei den EA sowohl elitistische ( $\lambda+\mu$ )- als auch nicht elitistische ( $\lambda, \mu$ )-Strategien, mutative und derandomisierte Schrittweitensteuerung bewertet. Speziell auf integer Variablen oder auf Float-Größen ausgerichtete Algorithmen wurden ebenso analysiert wie Strategien, die auf einem oder mehreren Eltern basieren. Alle Algorithmen wurden von der Firma NuTechSolutions für die Studien zur Verfügung gestellt.

### 3.2 Definition der Benchmarks

Zur Analyse der Optimierungsalgorithmen wurden eine Vielzahl von Problemen behandelt, Abbildung 6 gibt eine Auswahl. Die Beispiele sollten repräsentativ sein, d.h. sowohl Parameter-Optimierungen als auch Formoptimierungen, Crash- als auch NVH-Lastfälle, mono- als auch multi-kriterielle Optimierungen sowie vereinfachte Modelle als auch Praxismodelle umfassen.

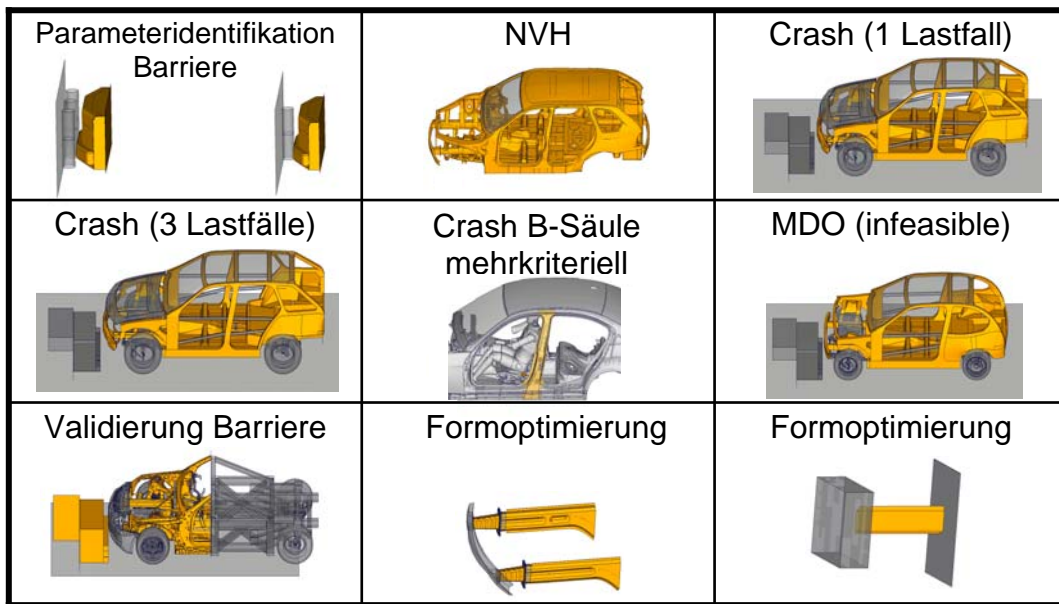
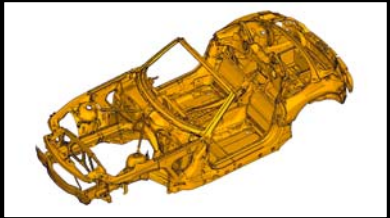


Abbildung 6: Einige der Benchmarks zur Bewertung der Algorithmen.

### 3.3 Ergebnisse der Evaluierung

Zuerst werden die Ergebnisse für das NVH-Beispiel (Abbildung 7) ausführlich dargestellt (Tabellen 1 und 2). In der ersten Tabelle erkennt man, dass genetische Algorithmen, Kriging unterstütztes evolutionäre Optimierung und Simulated Annealing nicht erfolgreich waren.

1. Benchmark: NVH		
FE-Modell	NVH	Crash
Knoten	328.608	-
Schalenelemente	229.510	-
Volumenelemente	56.650	-

Parameter der Optimierung	
# unabhängige Parameter	136
# Lastfälle (NVH)	2
# Zielgrößen	1
# Randbedingungen	11
# Simulationen	2x28x10
# CPU (Crash)	-
# CPU (NVH)	8
Dauer der Optimierung	ca. 28h

Abbildung 7: Parameter des ersten Benchmarks (NVH).



Algorithmus	Mittelwert	Standard-abweichung	Anzahl der Optimierungen	Kommentare
Monte-Carlo Schema	$\Delta \underline{m} = -14,6 \text{ kg}$	$\sigma = 1,43$	n = 8	Referenzmethode
Kriging	$\Delta \underline{m} = -9,3 \text{ kg}$	$\sigma = 1,61$	n = 3	keine signifikante Verbesserung
Simulated Annealing	$\Delta \underline{m} = -10,1 \text{ kg}$	$\sigma = 1,57$	n = 3	keine signifikante Verbesserung
Genetische Algorithmen	$\Delta \underline{m} = -12,2 \text{ kg}$	$\sigma = 2,10$	n = 3	keine signifikante Verbesserung

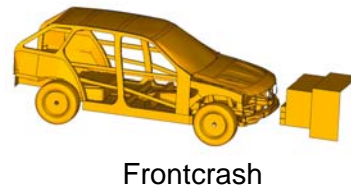
Tabelle 1: Vergleich des Krigingverfahrens, Simulated Annealings und der genetischen Algorithmen (6 Typen) mit dem Referenzverfahren (Monte-Carlo-Suche).

Algorithmus	Mittelwert	Standard-abweichung	Anzahl der Optimierungen	Kommentare
Monte-Carlo Schema	$\Delta \underline{m} = -14,6 \text{ kg}$	$\sigma = 1,43$	n = 8	Referenzmethode
Mutative Evolution	$\Delta \underline{m} = -14,0 \text{ kg}$	$\sigma = 1,41$	n = 8	keine signifikante Verbesserung
Derandomized Evolution	$\Delta \underline{m} = -15,5 \text{ kg}$	$\sigma = 1,67$	n = 15	signifikante Verbesserung
Single parent	$\Delta \underline{m} = -15,2 \text{ kg}$	$\sigma = 1,10$	n = 12	geringe Verbesserung
Multiple parents	$\Delta \underline{m} = -15,3 \text{ kg}$	$\sigma = 1,04$	n = 11	geringe Verbesserung
$(\lambda, \mu)$ -Strategie	$\Delta \underline{m} = -15,8 \text{ kg}$	$\sigma = 1,73$	n = 15	signifikante Verbesserung
$(\lambda + \mu)$ -Strategie	$\Delta \underline{m} = -14,4 \text{ kg}$	$\sigma = 1,30$	n = 8	keine signifikante Verbesserung

Tabelle 2: Vergleich der evolutionären Algorithmen (6 Typen) mit dem Referenzverfahren (Monte-Carlo-Suche).

Aus Tabelle 2 kann man entnehmen, dass eine nicht-elitistische Strategie mit einer derandomisierten Adaption das am meisten geeignete Verfahren ist. Das einfache Referenzverfahren schnitt relativ gut ab, was bedingt ist durch die lange Erfahrung mit diesem Ansatz und der günstigen Einstellung der Verfahrensparameter.

2. Benchmark: MDO		
FE-Modell	NVH	Frontcrash
Knoten	83.743	136.902
Schalenelemente	88.586	128.949
Volumenelemente	-	15.207



Parameter der Optimierung	
# unabhängige Parameter	109
# Lastfälle (Crash + NVH)	1+2
# Zielgrößen	1
# Randbedingungen	18
# Simulationen	3x28x10
# CPU (Crash)	800
# CPU (NVH)	8
Dauer der Optimierung	ca. 56 h

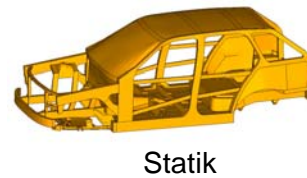


Abbildung 8: Parameter des zweiten Benchmarks (vereinfachte MDO).

Als zweites Benchmark wird ein vereinfachtes MDO-Problem betrachtet (Abbildung 8). Ein Gesamtfahrzeugmodell wurde betreffs der Elementanzahl und der modellierten Detailgenauigkeit reduziert, so dass die Elementanzahl anstatt der üblichen 1 bis 1,5 Millionen nur etwa 150.000 beträgt. Als Lastfälle wurden der Frontcrash, Statik und Dynamik betrachtet. Wie im ersten Beispiel wurde die Anzahl der möglichen Simulationen je Lastfall auf 280 beschränkt. Eine Crashberechnung auf 8 CPU dauerte etwa drei Stunden, die NVH-Berechnungen nur etwa 10 Minuten. Die Ergebnisse waren vergleichbar zu denen des ersten Benchmarks, Tabelle 3 gibt gegenüber der Referenzmethode die Verbesserung an, die mit dem besten Algorithmus im Mittel erreicht wurde. Dieser Lastfall ist aus der Sicht der Optimierung deutlich anspruchsvoller, die starke Nichtlinearität des Frontcrashes lässt die Differenz zwischen dem einfachen Monte-Carlo-Schema und dem lernenden, sich anpassenden Evolutionsalgorithmus markanter werden.

Algorithmus	Mittelwert	Standardabweichung	Anzahl der Optimierungen	Kommentare
Monte-Carlo Schema	$\Delta m = -6,6 \text{ kg}$	$\sigma = 1,45$	n = 15	Referenzmethode
Derandomized Evolution	$\Delta m = -8,9 \text{ kg}$	$\sigma = 2,11$	n = 15	signifikante Verbesserung

Tabelle 3: Vergleich der gewählten EA-Strategie mit dem Monte-Carlo-Schema für den MDO Benchmark.

### 3.4 Bewertung der Benchmarkergebnisse

Die Benchmarkstudien ergaben, dass deutlich bessere Ergebnisse für die verschiedensten Optimierungsprobleme erzielt werden können, wenn der Algorithmus sich automatisch an das Problem anpasst. Hierbei wurde auch ersichtlich, dass es nicht genügt, einen beliebigen evolutionären oder genetischen Algorithmus heranzuziehen. Wichtig sind die folgenden Punkte.

- Das Ausgangsmodell sollte dann in der ersten Generation berücksichtigt werden, wenn es schon im „normalen“ Designprozess per Hand voroptimiert wurde. Dies ist in den meisten Fällen der MDO sinnvoll. Es wird damit akzeptiert, dass nicht der gesamte Designraum abgesucht wird und dass somit irgendwo eine größere Verbesserung möglich ist. Die Erfahrungen zeigen jedoch, dass dies Vorgehen sinnvoll ist, da ein genaueres Untersuchen des Designraums i.d.R. auf Grund des hohen numerischen Aufwands von vorneherein nicht erfolgen kann. Die MDO wird eher in den späteren Phasen des PEP eingesetzt. Es ist daher meist zu erwarten, dass das verbesserte Design „in der Nähe“ des ursprünglichen liegt.
- Der Algorithmus sollte sich während der Optimierung an das Problem anpassen, d.h. die Schrittweiten der Veränderungen der Designparameter dürfen nicht konstant sein, wie dies der Fall ist für die Monte-Carlo-Suche. Ferner sollte diese Anpassung für jeden Parameter durchgeführt werden und nicht global vorgenommen werden.
- Rekombination erscheint gegenüber Mutation als der weniger ausschlaggebende Operator. Ein gutes Design betreffs Crashlastfällen ist meist bedingt durch eine gute Abstimmung der Bauteile zueinander, durch ein gutes Zusammenspiel dieser während der Belastung. Tauscht man einzelne über einen Rekombinationsoperator aus, so zerstört man dieses Zusammenwirken. Sinnvoll wäre es, eine Rekombination von geeigneten Bauteilgruppen zu definieren; in der Praxis ist dies jedoch meist daran gescheitert, dass diese Gruppen erst identifiziert werden müssen.
- Eine nicht-elitistische EA-Strategie war in den durchgeführten Beispielen von Vorteil, da relativ große Parametervariationen zielführend waren. Es sollte verhindert werden, dass der Algorithmus in Nebenextrema stecken bleibt.
- In den Studien hat sich die von NuTech vorgegebene Strategie bewährt, die ein Verhältnis von etwa 7:1 von Offsprings zu Eltern vorschlägt. In den untersuchten Benchmarks wurden insgesamt 280 Simulationen je Disziplin zugelassen. Getestet wurden dann entweder 10 Generationen mit 28 Individuen oder 28 Generationen mit jeweils 10 Varianten. Die Unterschiede zwischen diesen beiden Varianten waren nicht signifikant. Eine Generationsgröße kleiner als etwa 10 erwies sich als nicht sinnvoll.
- Für multi-kriterielle Optimierungen gelten andere Zusammenhänge. Für eine ausreichende Diversivität der Paretofront sollte eine ausreichende Generationsgröße gewählt werden (je nach Problem etwa 30-50). Ferner ist es sinnvoller, eine elitistische Strategie zu wählen, um nicht zu viel Information über die Paretofront während der Optimierung zu verlieren.

Auch das Verhältnis zwischen Nachkommen und Eltern sollte geringer als im mono-kriteriellen Fall sein. Um abschließend urteilen zu können, sollten noch weitere Studien durchgeführt werden. Bislang haben sich die Algorithmen von NuTech auch sehr gut für diese mehrkriteriellen Probleme bewährt.

- Die bislang diskutierten Ergebnisse folgten aus den durchgeführten Parameter-Optimierungen, einige Studien, vgl. [Volz 2005] haben jedoch gezeigt, dass sie auf Probleme der Formoptimierung übertragbar sind. Abbildung 9 zeigt ein Beispiel aus dem Promotionsvorhaben von Karlheinz Volz. Hier wurde der Vorderbau (in der Abbildung gelb hinterlegt) betreffs Frontcrash optimiert. Optimierungsvariablen waren echte Formparameter. Die Optimierung erfolgte hier mit den modifizierten evolutionären Algorithmen der Firma Dynardo.

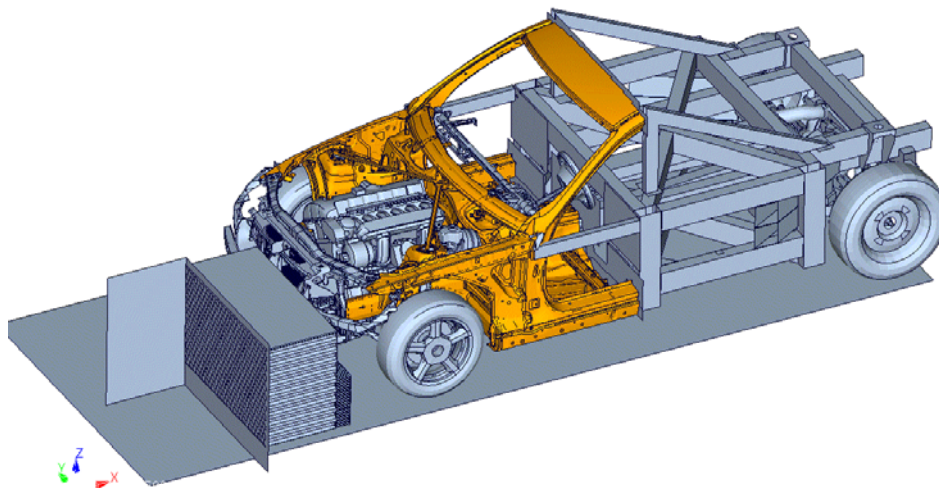


Abbildung 9: Formoptimierung des Vorderbaus, siehe [Volz 2005].

## 4 Validierung mittels vollständiger MDO

Abschließend wurden die Erkenntnisse an Hand eines Praxismodells validiert, Abbildung 10 gibt eine Übersicht über die Lastfälle und der Abbildung 11 kann man die betreffende Charakteristik entnehmen. Das Beispiel war in einer den Evaluierungen vorhergehenden Studie mittels eines Monte-Carlo-Schemas (STORM) bereits optimiert worden. Ausgangssituation war – typisch für MDO-Fragestellungen – ein Modell, dass bereits im normalen Entwicklungsprozess weitgehend betreffs aller Disziplinen intensiv bearbeitet worden war, so dass das prinzipielle Strukturkonzept alle Funktionen im wesentlichen erfüllte und die Optimierung vor allem die Masse reduzieren sollte. Die MDO wird hier zur Feinabstimmung zwischen den Disziplinen verwendet. Sensitivitätsstudien betreffs der NVH-Disziplinen konnten über den Gradientenlöser von NASTRAN realisiert werden, betreffs der Crashdisziplinen wäre dies mit zu großem Aufwand verbunden. Studien hatten gezeigt, dass mindestens hundert Simulationen notwendig

sind, um zumindest die wichtigsten zehn Bauteile zu identifizieren. Das Optimierungsproblem hatte 136 unabhängige Parameter (Blechdicken) und 47 Randbedingungen und das FE-Modell bestand aus etwa 1,1 Millionen Finiten Elementen. Auf einem HP-Itanium Cluster betrug die Rechenzeit auf 8 CPU für eine einzelne Crashsimulation zwischen 12 und 24 Stunden, NVH-Rechnungen dauerten auf einer CPU etwa eine Stunde.

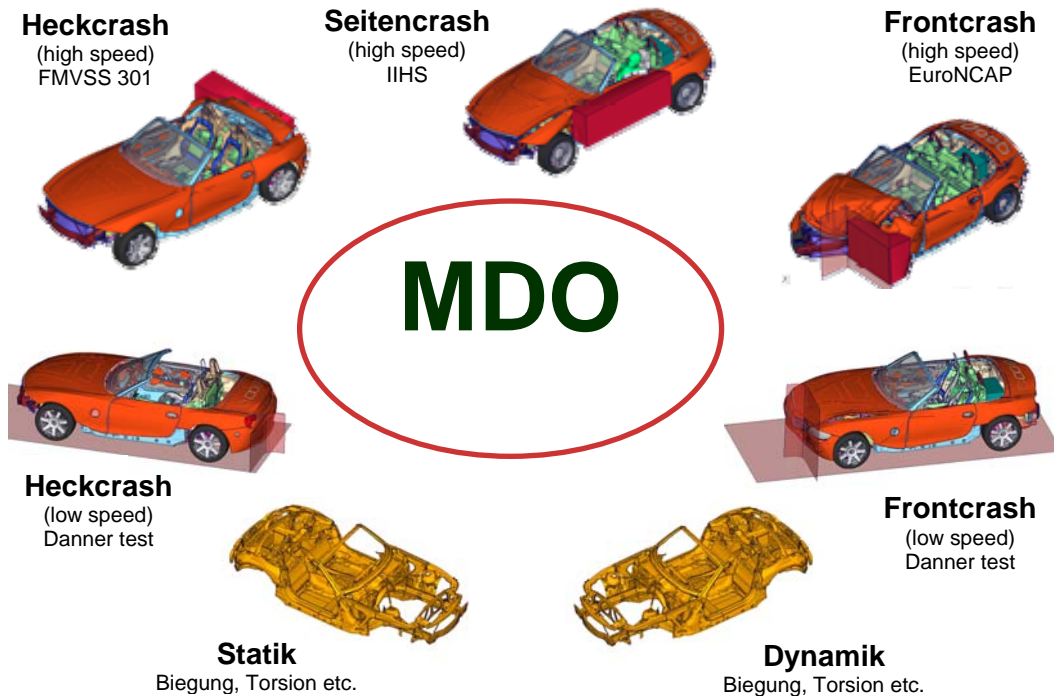
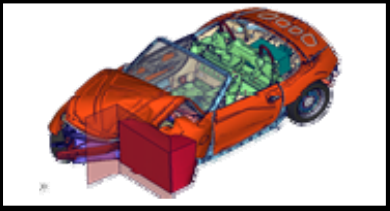


Abbildung 10: MDO-Validierungsbeispiel: betrachtet wurden fünf Crashlastfälle und die beiden linearen FE-Berechnungen für Statik und Dynamik.

Zu betonen sei hier nochmals, dass Optimierungen dieses Umfanges sorgfältig vorbereitet werden müssen. Alle Parameter müssen überprüft werden, die zugehörigen Designfreiräume mit allen Disziplinen diskutiert werden. Selbst diese MDO mit ihren 7 Disziplinen umfasst bei weitem nicht alle zu untersuchenden Fragestellungen. Restriktionen aus nicht integrierten Aspekten müssen sich in den zulässigen Designräumen der Variablen widerspiegeln. Die erfassten Disziplinen müssen aufeinander abgestimmt werden, teils mussten und müssen die vorhandenen PEP erweitert werden, so dass die notwendigen Informationen schnell zugänglich sind.

Der Rechenaufwand ist auch bei verbesserten Algorithmen noch erheblich, es muss unbedingt gewährleistet werden, dass ein Restart möglich ist. Erfahrungen haben ferner gezeigt, dass oft unerwartete Änderungen im komplexen Rechnerumfeld einer CAE-Umgebung in der Automobilindustrie zu einem Abbruch der Optimierung führen. In Zusammenarbeit mit NuTech wurde ein möglichst redundantes System entwickelt, das die meisten Probleme dieser Art verhindert.

MDO – Beispiel		
FE-Modell	NVH	Crash
Knoten	328.608	1.191.597
Schalenelemente	229.510	1.087.424
Volumenelemente	56.650	33.314

Parameter der Optimierung	
# unabhängiger Parameter	136
# Lastfälle (Crash, NVH)	5+2
# Zielgrößen	1
# Randbedingungen	47
# Simulationen	7x28x10
# CPU (Crash)	ca. 1.000
# CPU (NVH)	8
Dauer der Optimierung	ca. 14d

Abbildung 11: Charakteristik des MDO-Validierungsbeispiels.

In Abbildung 12 wird der Optimierungsfortschritt des Referenzverfahrens (Monte-Carlo-Schema) und des gewählten Evolutionsschemas abgebildet. Da wegen der hohen Rechenzeit in der ursprünglich durchgeführten Monte-Carlo-Optimierung nur 180 Simulationen je Lastfall durchgeführt wurden, wird hier als Vergleich der Stand der EA dargestellt, der sich ebenfalls nach 180 Simulationen ergeben hatte. Deutlich erkennbar ist die höhere Effizienz des neuen Verfahrens.

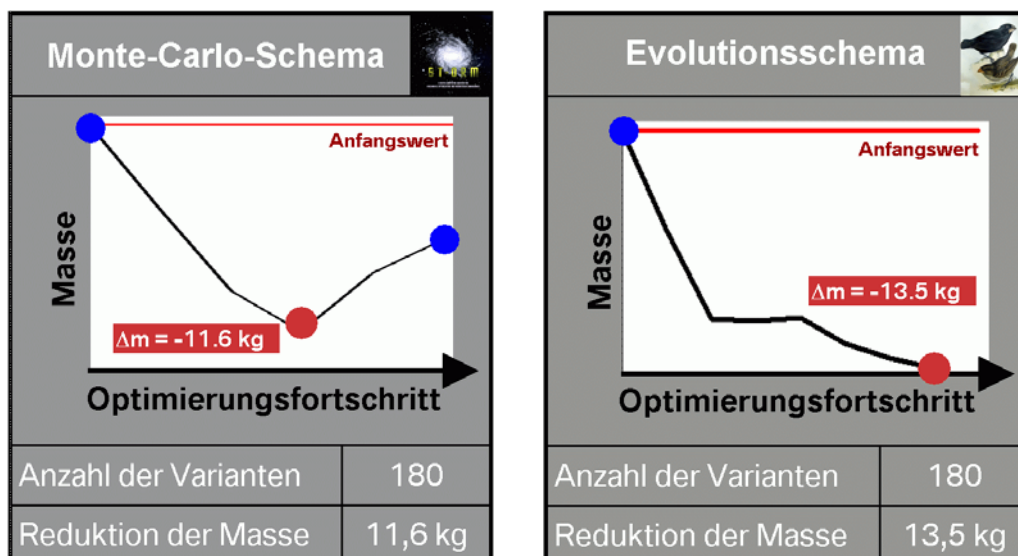


Abbildung 12: Optimierungsfortschritt der Validierungs-MDO für die Monte-Carlo-Optimierung (links) und für die evolutionäre Optimierung (rechts).

Analysiert man tiefer gehend die Gründe für die bessere Performance, so erkennt man, dass das adaptive Lernen, d.h. die Anpassung der Mutationsschrittweiten ausschlaggebend ist, vgl. Abbildung 13.

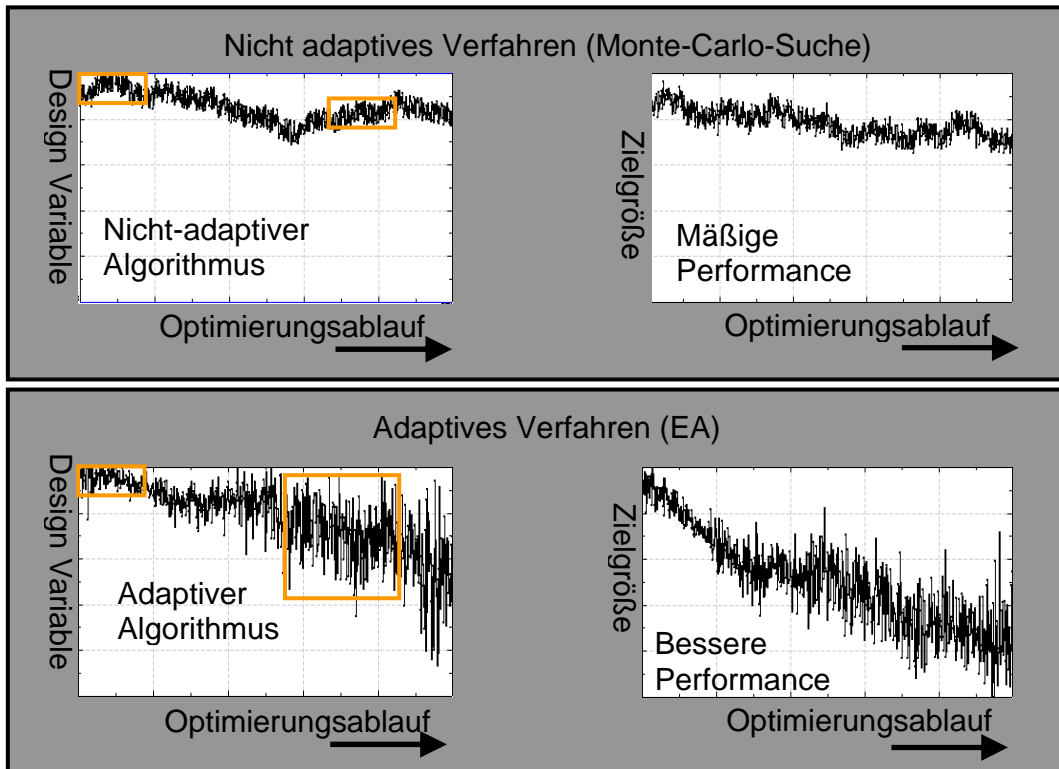


Abbildung 13: Fehlende Adaptivität des Monte-Carlo-Verfahrens (oben) und erfolgreiche Anpassung des adaptiven evolutionären Algorithmus (unten).

## 5 Integration in den Produktentwicklungsprozess

Die Ermittlung eines möglichst geeigneten Algorithmus' ist grundlegend für die Integration der Optimierung in den PEP. Zusätzlich erforderlich ist eine Anpassung der vorhandenen Entwicklungsstrukturen, um eine korrekte, zeit-gerechte und effiziente Bereitstellung der Daten zu Beginn der Optimierung zu ermöglichen. Schnelles und effektives Pre- und Postprocessing, das größtenteils automatisiert sein sollte, sowie ein gutes Monitoring sind unabdingbar.

Aus der Sicht des Autors, ist der organisatorische Teil noch ausschlaggebender für den Erfolg der Optimierung als die Frage nach dem besten Algorithmus. Die hier vorgestellte Evaluierung war Teil der erfolgreichen Integration der MDO in den Entwicklungsprozess von BMW. Optimierungszeiten von insgesamt etwa vier bis fünf Wochen konnten auf etwa zwei Wochen reduziert werden. Erste prozessrelevante MDOs konnten realisiert werden.

## 6 Zusammenfassung

Die Integration der Multidisziplinäre Optimierung in den Produktentwicklungsprozess erfordert einerseits den best möglichen Algorithmus, andererseits eine Umstellung und Erweiterung des Prozesses selbst. Umfangreiche Studien im Forschungs- und Innovationszentrum von BMW konnten zeigen, dass evolutionäre Algorithmen am besten geeignet sind. Wesentlich ist hierbei dass diese Algorithmen sich ausreichend gut an die Probleme anpassen. Dies war nicht bei allen getesteten evolutionäre Algorithmen der Fall. Eine derandomisierte Strategie mit mehreren Eltern und einer nicht elitistischen Selektion erwies sich als optimal.

Zukünftig wird an der weiteren Verkürzung des Zeitfensters gearbeitet. Die Potenziale dafür sind in der intelligenten Ressourcenausnutzung und im Feintuning des Algorithmus zu suchen. Hybride Verfahren, d.h. die Mischung stochastischer Optimierungsmethoden mit RSM-Verfahren sollten getestet werden, um zu einer echten multidisziplinären Strategie zu gelangen. Weitere Disziplinen könnten hinzu genommen werden, Fragen der Akustik und aus dem CFD-Bereich erscheinen sinnvoll.

## 7 Danksagung

Der Autor dankt der Firma BMW für die außergewöhnlich guten Arbeitsmöglichkeiten, die erst diese Studien ermöglicht haben. Persönlich gilt mein Dank allen Kollegen und Kolleginnen, ohne deren Unterstützung keine dieser Optimierung erfolgreich gewesen wäre. Ferner möchte ich mich bei den Mitarbeitern der Firma NuTech, besonders Herrn Prof. Bäck und Herrn Willmes für die Algorithmen und die Beratung bedanken. Der Firma Dynardo gilt mein Dank für die Bereitstellung ihrer Software für zusätzliche Studien, speziell für die Formoptimierungen.

## Literatur

- BÄCK, T.: *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.
- BEYER, H.-G.: *The Theory of Evolution Strategies*. Natural Computing Series, Heidelberg, Springer Verlag, 2001.
- BLUMHARDT, R.: *Numerische Optimierung des Crashverhaltens von Fahrzeugstrukturen und –komponenten*. Technische Universität München, Dissertation, Shaker-Verlag, 2002.
- BÜCHE, D.: *Multi-Objective Evolutionary Optimization of Gas Turbine Components*. Dissertation, ETH Zürich, 2003.



- COELLO, C. A.; VAN VELDHUIZEN, D.A. UND LAMONT, G. B.: *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Acad. Publ., 2002.
- COLLETTE, Y. und Siarry P.: *Optimisation multiobjective*. Editions Eyrolles, 2002.
- DEB, K.: *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons Ltd, 2001.
- DUDDECK, F. UND VOLZ, K.: *Evaluation of Optimization Algorithms for Crash and NVH*. In: BATHE, K. J. (Hrsg.): *Computational Fluid and Solid Mechanics*. Cambridge, USA, 2005.
- FORSBERG, J.: *Simulation based Crashworthiness Design – Accuracy Aspects of Structural Optimization using Response Surfaces*. Linköpings universitet, Dissertation, 2002.
- HEISERER, D.: *Fast Reanalysis for Large Multicriteria Structural Optimization*. Technische Universität München, Dissertation, Shaker-Verlag, 2005.
- LESCHETICKY, J.; DUDDECK, F.; GIRONA, S. UND WILLMES, L.: *Effizientere Produktentwicklung durch den Einsatz multidisziplinärer Optimierungs-verfahren im Entwurfsprozess von Karosserien*. In: *Numerical Analysis and Simulation, VDI-Conference, Würzburg, 2004*.
- MYERS, R. H. UND MONTGOMERY, D. C.: *Response Surface Methodology*. John Wiley & Sons Ltd, 2002.
- REDHE, M.: *On Vehicle Crashworthiness Design Using Structural Optimization*. Linköpings universitet, Dissertation, 2004.
- STANDER, N.; ROUX, W.; GIGER, M., REDHE, M.; FEDOROVA, N.; HAARHOFF, J.: *Crashworthiness optimization in LS-OPT: case studies in metamodeling and random search techniques*. In: *Proc of the 4th European LS-DYNA Users Conference, Ulm, Germany, 2003*.
- VOLZ, K. UND DUDDECK, F.: *Crash optimization of car bodies in the concept stage of vehicle development*. In: BATHE, K. J. (Hrsg.): *Computational Fluid and Solid Mechanics*. Cambridge, USA, 2005.